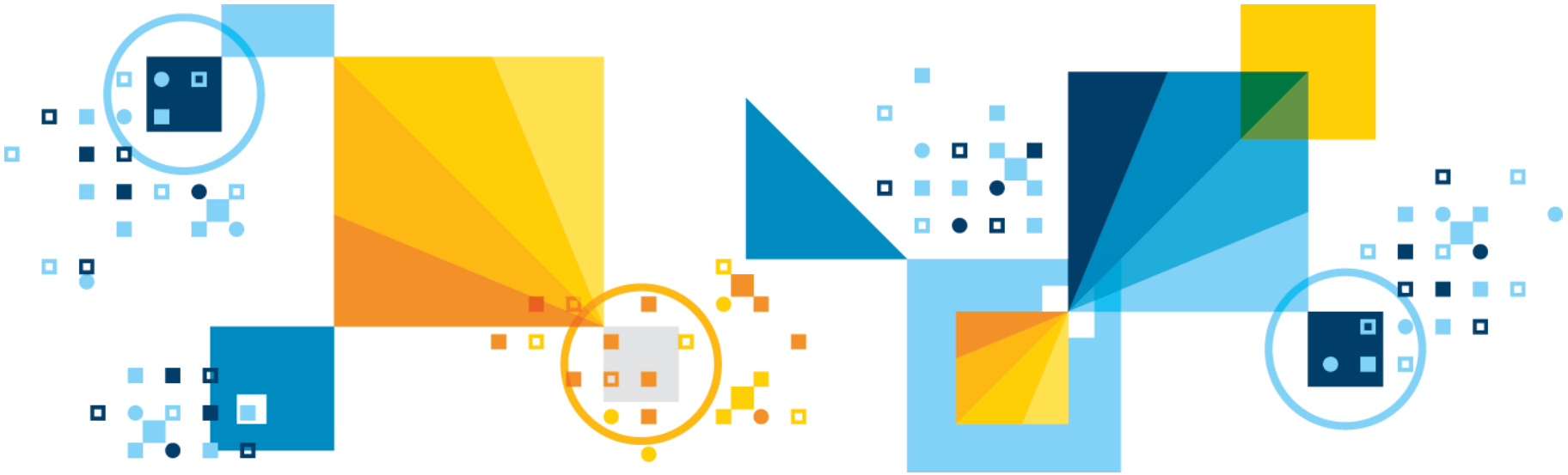
Scott Pickett – WW Informix Technical Sales

May 8, 2018

Informix Roadmap 2018



Disclaimers

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

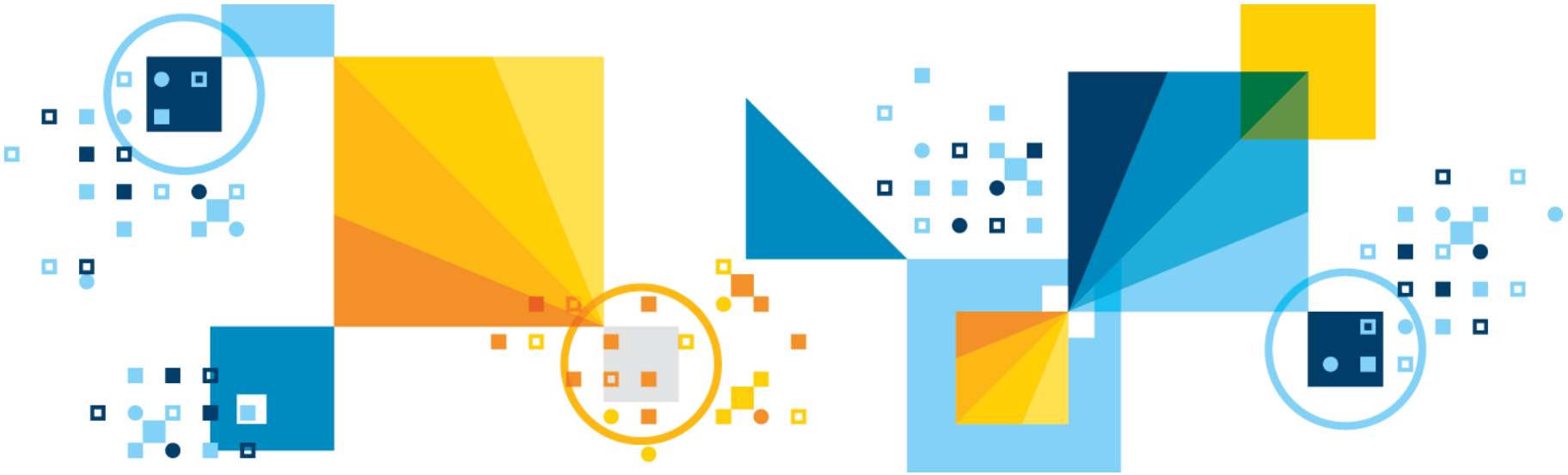
The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Scott Pickett – WW Informix Technical Sales

May 8, 2018

General Data Protection Rules (GDPR) and Informix



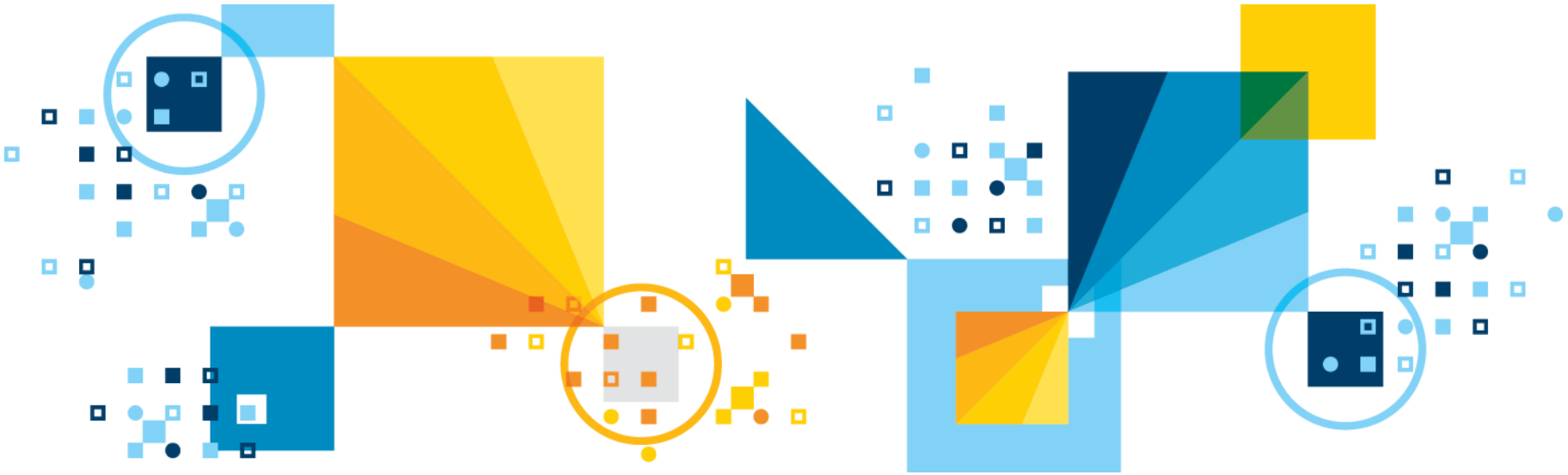
Agenda

- Selected GDPR Articles and Recitives
- Informix Onprem Features & GDPR
- Informix on Cloud & GDPR
- Appendix A – Complete Unedited GDPR Articles – Speaker notes
- Appendix B – Complete Unedited GDPR Recitives – Speaker notes
- Appendix C – GSKit Setup
- Appendix D – GDPR in a Little More Depth

Scott Pickett – WW Informix Technical Sales

May 8, 2018

Selected GDPR Articles and Recitives



Forward

- This presentation looks at some of the actual laws and regulations for GDPR in Europe as adopted by the European Union.
- The actual text of the rules and regulations are used.
- As applied to databases only, the presentation points out how Informix can be beneficial to its customers in meeting the GDPR goals and aims and how we can assist client through our product features, meet the strict goals required of GDPR.
- Implementation of some of these features is discussed in detail.
- There is no attempt to expand this presentation beyond the scope of the Informix database features.

GDPR

- **The European Union (EU) General Data Protection Regulation (GDPR) intends to strengthen and unify data protection for all individuals within the EU**
  - It also addresses the export of personal data outside the EU
    - Therefore, it also applies to businesses whom do not necessarily have offices within the EU but do business with the citizens of the EU.
    - This part is not so obvious but is there and will be enforced

- **Primary objectives of the GDPR:**
  - Return control to citizens and residents over their personal data
  - Simplify the regulatory environment for international business by unifying the current disparate regulations of the same personal data within the EU
    - Replaces the previous data protection directive from 1995

- **GDPR was adopted and is legal as of 27 April 2016 by the EU.**

- **It applies and will be enforced as of 25 May 2018**

Summary – Why does this apply to me?

- **"The proposed new EU data protection regime extends the scope of the EU data protection law to all foreign companies processing data of EU residents, even data stored and processed outside of the EU."**

- **"It provides for a harmonization of the data protection regulations throughout the EU, thereby making it easier for non-European companies to comply with these regulations.."**

- **"However, this comes at the cost of a strict data protection compliance regime with severe maximum fines to be the highest of 4% of global turnover or 20 million Euros."**

- **The law has teeth:**
  - Regulatory boards are setup in each EU member state to hear complaints and enforce regulations.
  - If a business operates in multiple EU states, one EU state Board will lead the investigation

Responsibility and Accountability (1)

- **"Notice requirements expanded over previous rules. Must include:**
  - Retention time for personal data
  - Contact info for the data controller and data protection officer"

- **"Automated individual decision-making, including profiling, is made contestable (Article 22):**
  - EU Citizens now have the right to question and fight decisions affecting them … made on a purely algorithmic basis"

- **"Compliance with the GDPR is proven by the data controller implementing measures meeting the principles of data protection by design and data protection by default:**
  - Privacy by Design and by Default (Article 25) require data protection measures are designed into the development of business processes for products and services
    - Such measures include pseudonymising personal data, by the controller, as soon as possible (Recital 78)

Responsibility and Accountability (2)

- **It is the responsibility and liability of the data controller to implement effective measures and be able to demonstrate the compliance of processing activities even if the processing is carried out by a data processor on behalf of the controller. (Article 47).**

- **"Data Protection Impact Assessments (Article 35) have to be conducted when specific risks occur to the rights and freedoms of data subjects:**
  - Risk assessment and mitigation is required and prior approval of the Data Protection Authorities (DPA) is required for high risks"

Responsibility and Accountability (3)

- **“Data Protection Officers (DPO) (Articles 37–39) are to ensure compliance within organizations**
  - They have to be appointed
    - For all public authorities, except for courts acting in their judicial capacity
    - If the core activities of the controller or the processor consist of
      - Processing **operations** which, by virtue of their nature, their scope and/or their purposes, require regular and systematic monitoring of data subjects on a large scale
      - Processing on a large **scale** of special categories of data pursuant to Article 9 and personal data relating to criminal convictions and offences referred to in Article 10”

Consent

- **“Valid consent must be explicit for data collected and the purposes data is used for (Article 7; defined in Article 4)**
- **Consent for children must be given by the child’s parent or custodian, and verifiable (Article 8)**
- **Data controllers must be able to prove "consent" (opt-in) and consent may be withdrawn”**

Data Protection Officer (DPO)

- **“For public authorities (except judiciary) or private sector where processing is carried out by a controller whose core activities consist of processing operations that require regular and systematic monitoring of the data subjects:**
  - A person with expert knowledge of data protection law and practices should assist the controller or processor to monitor internal compliance with this Regulation.
  - The DPO is similar but not the same as a Compliance Officer as they are also expected to be proficient at managing IT processes, data security (including dealing with cyber-attacks) and other critical business continuity issues around the holding and processing of personal and sensitive data. “
    - Calls for persons with “skill set … beyond understanding legal compliance with data protection laws and regulations.”
    - “The DPO will need to create their own support team and will also be responsible for their own continuing professional development as they need to be independent of the organization that employs them, effectively as a "mini-regulator“ “.

GDPR Pseudonymisation (1)

- **"A process that transforms personal data in such a way that the resulting data cannot be attributed to a specific data subject without the use of additional information.**
  - **An example of pseudonymisation is encryption**, which renders the original data unintelligible and the process cannot be reversed without access to the correct decryption key.
  - The GDPR requires that this additional information (such as the decryption key) be kept separately from the pseudonymised data.
  - Pseudonymisation is recommended to reduce the risks to the concerned data subjects and also help controllers and processors to meet their data-protection obligations (Recital 78)."

- **"If the personal data is pseudonymised with adequate internal policies and measures by the data controller, then it is considered to be effectively anonymized, and not subject to controls and penalties of the GDPR."**

Encryption Specifically Spelled Out as Risk Mitigation

- **Recitive 83 -** *"(83) In order to maintain security and to prevent processing in infringement of this Regulation, the controller or processor should evaluate the risks inherent in the processing and implement measures to mitigate those risks,* ***such as encryption****. Those measures should ensure an appropriate level of security, including confidentiality, taking into account the state of the art and the costs of implementation in relation to the risks and the nature of the personal data to be protected. In assessing data security risk, consideration should be given to the risks that are presented by personal data processing, such as accidental or unlawful destruction, loss, alteration, unauthorised disclosure of, or access to, personal data transmitted, stored or otherwise processed which may in particular lead to physical, material or non-material damage."*

GDPR Pseudonymisation (2)

- **“The policies and measures that meet the principles of data protection by design and data protection by default should be considered adequate for this purpose.**

- **Example measures would include:**
  - Pseudonymizing the data as soon as possible (Recital 78),
  - Encrypting the data locally:
    - Keeping the decryption keys separately from the encrypted data.

- **The regulation does not concern the processing of information that is deemed anonymous, including for statistical or research purposes.”**

Data Breaches and GDPR

- **"Data Controller will be under a legal obligation to notify the Supervisory Authority without undue delay:**
  - The reporting of a data breach is not subject to any *de minimis* standard and must be reported to the Supervisory Authority within 72 hours of the data breach (Article 33).
  - Individuals have to be notified if adverse impact is determined (Article 34).
  - In addition, the data processor will have to notify the controller without undue delay after becoming aware of a personal data breach (Article 33).

- **However, the data processor or controller does not have to notify the data subjects if anonymized data is breached:**
  - Specifically, the **notice to data subjects is not required** if the data controller has implemented pseudonymisation techniques like encryption along with adequate technical and organizational protection measures to the personal data affected by the data breach (Article 34)."

Sanctions

- **The following sanctions can be imposed:**
  - A warning in writing in cases of first and non-intentional non-compliance
  - Regular periodic data protection audits
  - A fine up to 10,000,000 EUR or up to 2% of the annual worldwide turnover of the preceding financial year in case of an enterprise, whichever is greater (Article 83, Paragraph 4)
  - A fine up to 20,000,000 EUR or up to 4% of the annual worldwide turnover of the preceding financial year in case of an enterprise, whichever is greater (Article 83, Paragraph 5 & 6)

Right to Erasure

- **"A *right to be forgotten* was replaced by a more limited *right to erasure* in the version of the GDPR adopted by the European Parliament in March 2014"**

- **Article 17 provides that "the data subject has the right to request erasure of personal data related to them on any one of a number of grounds"**
  - Within one month of request, to be achieved
  - Serial columns, sequences ….. anyone ……
    - Does your SQL check for NULL ?

- **"This right includes "non-compliance with Article 6.1 (lawfulness) that includes a case (f) where the legitimate interests of the controller is overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data"**
- **So individual rights here can override corporate rights.**

Data Portability & GDPR

- **“A person shall be able to transfer their personal data from one electronic processing system and into another, without being prevented from doing so by the data controller. In addition, the data must be provided by the controller in a structured and commonly used Open Standard electronic format.”**
  - Just how many data formats are there out there today ?????

- **“The right to data portability is provided by Article 20 of the GDPR. Legal experts see in the final version of this measure a "new right" created that "reaches beyond the scope of data portability between two controllers as stipulated in Article 20".**

Data Protection by Design and by Default – GDPR Article 25

- **Requires that data protection is designed into the development of business processes for products and services:**
  - Privacy settings must be set at a high level by default
  - Technical and procedural measures should be taken care by the controller in order to make sure that the processing, throughout the whole processing lifecycle, complies with the regulation
  - Controllers should also implement mechanisms to ensure that personal data is only processed when necessary for each specific purpose

Data Protection by Design and by Default – GDPR Article 25

- **The European Union Agency for Network and Information Security (ENISA) elaborates on what needs to be done to achieve privacy and data protection by default:**
  - Encryption and decryption operations must be carried out locally, not by remote service, because both keys and data must remain in the power of the data owner if any privacy is to be achieved.
    - Outsourced data storage on remote clouds is practical and relatively safe, as long as only the data owner, **not** the cloud service, holds the decryption keys.
    - So even with data in the Cloud, hold the keys locally and not in the hands of your cloud provider.

Records of Processing Activities

- **“Records of processing activities must be maintained, that include purposes of the processing, categories involved and envisaged time limits.**

- **These records must be made available to the supervisory authority on request. (Article 30).”**

Lawful Basis for Processing

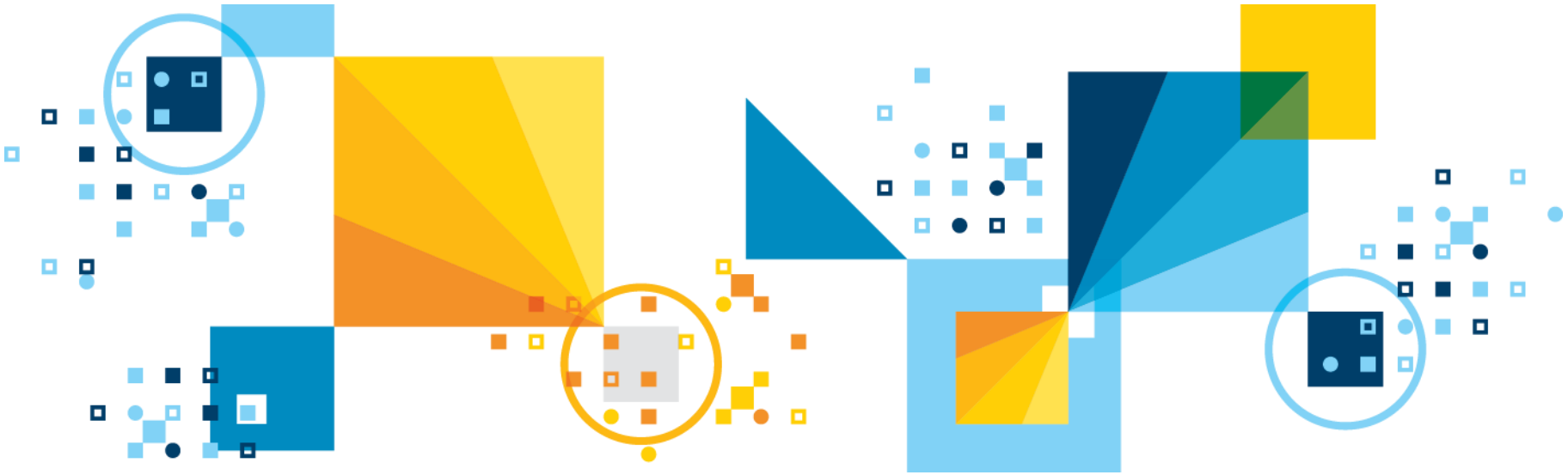
- **Data may not be processed unless there is at least one lawful basis to do so:**
  - The data subject has given consent to the processing of personal data for one or more specific purposes
  - Processing is necessary for the performance of a contract to which the data subject is party or to take steps at the request of the data subject prior to entering into a contract
  - Processing is necessary for compliance with a legal obligation to which the controller is subject
  - Processing is necessary to protect the vital interests of the data subject or of another natural person
  - Processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller
  - Processing is necessary for the purposes of the legitimate interests pursued by the controller or by a third party unless such interests are overridden by the interests or fundamental rights and freedoms of the data subject, which require protection of personal data, in particular if the data subject is a child

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Informix OnPrem Editions and GDPR



GDPR and Informix

- **GDPR is a whole laundry list of things that an organization needs to do to protect the data privacy and rights on individuals; primarily, GDPR falls to applications to do a lot of the work involved in making the goals and aims of GDPR a reality.**

- **Informix as a data repository, nevertheless has features within it to help with the security, access, authentication, auditing, location and data disposal, the use of which must obviously fit within a end-user tailored solution.**

- **From the Articles and Regulation of the GDPR, one thing is clear, encryption is considered the way to pseudonymise data and relieve companies from the enforcement and penalties of the GDPR, as well as a legal entry way to taking into consideration of both individual and corporate privacy rights and concerns.**
  - Informix has a whole lot of that ……

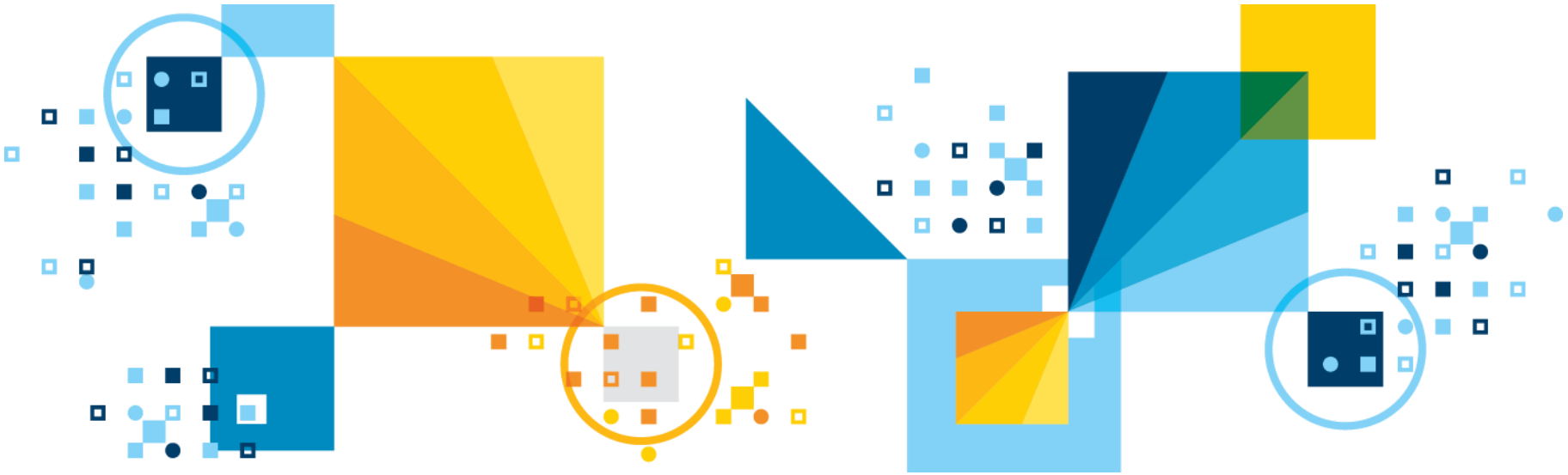
Two areas

- Security, Encryption, Authentication, Permissions
- Data Sunset Capabilities

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Informix Security, Encryption, Authentication, Permissions, Auditing



Security, Encryption, Authentication, Permissions (1)

- **Encrypt all data at rest**
- **Encrypt all backups via BACKUP/RESTORE FILTER**
  - Multiple differing encryption methods supported.
- **Encrypt Communications (ENCCSM) between client/server and non-HDR/ER server/server**
- **Encrypt all communications between servers and client to server**
- **Encrypt column defined passwords**
- **Encrypted data columns of personal sensitive information, application controlled.**
- **Grant and revoke access, user defined roles & permissions (dba, connect, resource, insert, update, delete , select, index)**

Security, Encryption, Authentication, Permissions (2)

- Audit DML, DDL via Informix or Guardium Auditing
- Secure role separations for DBSA, DBA, Database System Security Officer (DBSSO) and Audit Analysis Officer (AAO)
- Secure 24 x 7 Informix install binary directory from malware, etc.
- SQL based language execution code to remove/delete/modify data upon user request via secured and authorized users.
- Default O/S based user authentication, logins, passwords, groups
- Pluggable Authentication Module (PAM) application attached security via: the client connection password, correct input to a challenge-response prompt (for example, a RADIUS authentication server), or a combination of both.

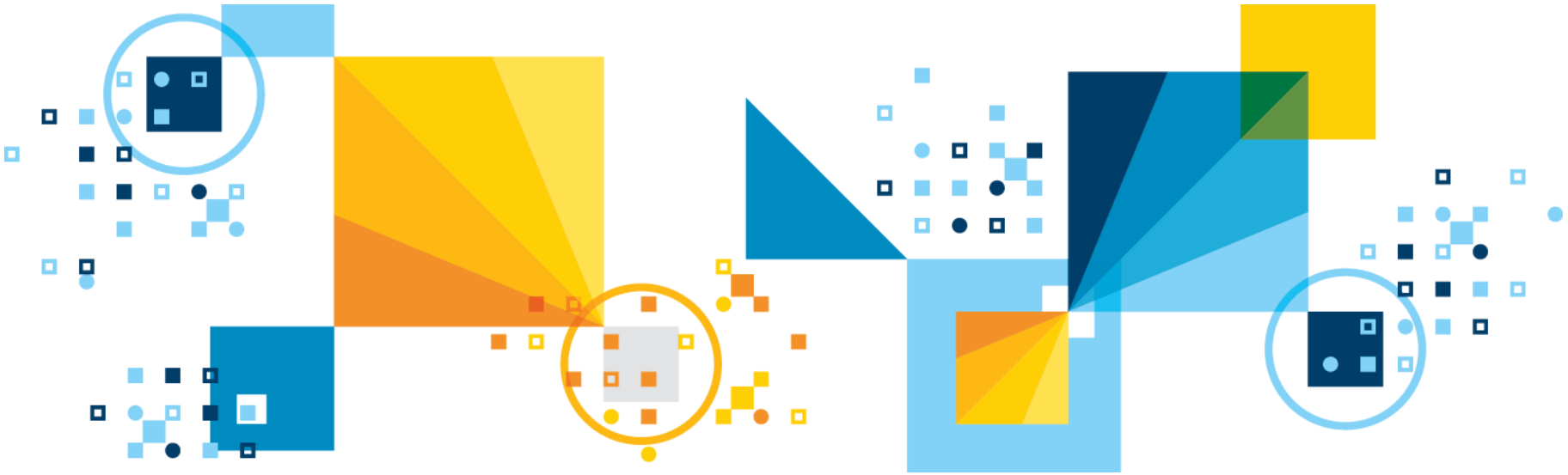
Security, Encryption, Authentication, Permissions (3)

- REST authentication with HTTPS support
- Mapped Users support for non-O/S based authentication,
- Label Based Access Control (LBAC)
- Possible to lock/unlock tables/databases exclusively online, awaiting final disposition
- **Applications permissions can be programmatically assigned and security for the same as well.**
  - Internal application identifier label assignment and database table lookup checks for each user.

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Encryption at Rest



Encrypt Storage Spaces or a Whole Instance

- **Encrypting storage spaces is an effective way to protect sensitive information that is stored on media**
  - Data in encrypted storage spaces is unintelligible without the encryption key.
    - Customer is responsible for managing the keys.

- **Enable storage space encryption by setting the new DISK_ENCRYPTION configuration parameter.**
  - Subsequently, storage spaces created are default automatically encrypted.

- **Create an unencrypted storage space with onspaces -c or SQL Admin API commands.**

- **Encrypt or decrypt storage spaces during a restore with the ON-Bar or ontape utilities.**
- **Check if storage spaces are encrypted with the onstat -d and oncheck -pr commands.**

DISK_ENCRYPTION configuration parameter

- **Controls the encryption of storage spaces**
  - Not set by default
  - Not dynamic.
- **Once enabled, any storage spaces created are encrypted by default.**
  - Previously created storage spaces will not be encrypted.
- **Set the encryption file names, cipher to use, the configuration parameter to enable storage space encryption.**
- **When storage space encryption is enabled, you can restore a storage space as encrypted or unencrypted, regardless of whether the space was encrypted at the time of the back up.**
- **Backup data and Restore data are encrypted/decrypted via the BACKUP_FILTER and RESTORE_FILTER parameter.**

DISK_ENCRYPTION configuration parameter

```
>>-DISK_ENCRYPTION--keystore--=--keystore_name----------------->

>--+----------------------+----------------------------->
   '-,--cipher--=--+-aes128-+-'
                   +-aes192-+
                    '-aes256-'

>--+-----------------------------------+----------------><
   '-,--rollfwd_create_dbs--=--+-encrypt-+-'
                               '-decrypt-'
```

DISK_ENCRYPTION configuration parameter

- **keystore**
  - The **keystore** specifies the name of the **keystore** and **stash file** names.
  - The files are created in the **INFORMIXDIR/etc** directory:

- **keystore.p12**
  - The keystore file that contains the security certificates.
- **keystore.sth**
  - The stash file that contains the encryption password.

- **Manual back up via operating system backup is presently required for the keystore and password stash files.**
  - Files are not backed up when **ON-Bar** or **ontape** backs up.

DISK_ENCRYPTION configuration parameter

- **cipher**
  - The encryption cipher:
    - **aes128** - **Default.** Advanced Encryption Standard cipher with 128-bit keys.
    - **aes192** - Advanced Encryption Standard cipher with 192-bit keys.
    - **aes256** - Advanced Encryption Standard cipher with 256-bit keys.

- **rollfwd_create_dbs**
  - Whether to encrypt a storage space created by the rolling forward of the logical log during a restore:
    - **encrypt** - Encrypt the newly created storage space
    - **decrypt** - Do not encrypt the newly created storage space
  - Default, storage spaces that are created by the rolling forward of the logical log have the same encryption state as the original storage space.

onspaces Unencrypted Option

- To create an unencrypted storage space, even if DISK_ENCRYPTION is turned on:

```
onspaces –c –d unencrypted_space –p /usr/storage/unencrypted_dbs1 –o
0 –s 2000000 –k 2 –u


execute function task("create unencrypted dbspace…

execute function task("create unencrypted blobspace…

etc…
```

- In shared memory, if a dbspace is encrypted the flags value for this will be 0x10000000, and onstat –d in the dbspaces portion of the output will reflect encryption status with an 'E' in position 6 for the flags column.

onstat -d

```
IBM Informix Dynamic Server Version 12.10.FC8 --On-Line --Up 00:03:16 --38324 Kbytes
Dbspaces
address  number flags      fchunk nchunks pgsize flags   owner    name
4484f028 1      0x1        1      1       2048    N  BA  informix rootdbs
4484fdd0 2      0x10000001 2      1       2048    N  BAE informix jcdbs
2 active, 2047 maximum

Chunks
Address  chunk/dbs offset   size     free   bpages flags    pathname
4484f268 1        1        0        100000 35118          PO-B-- /work3/JC/rootchunk
44958450 2        2        0        5000   3209           PO-B-- /work3/JC/chunk2
2 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are displayed in
terms of "pgsize" of the DBspace to which they belong.

Expanded chunk capacity mode: always
```

Quick Start (1)

- Set DISK_ENCRYPTION in onconfig file
- DISK_ENCRYPTION keystore=jc_keystore
- oninit –ivy …..

Quick start (2) – Message Log File

...

...

Initializing Dictionary Cache and SPL Routine Cache...succeeded Initializing encryption-at-rest if necessary...succeeded Initializing encryption-at-rest structures (part 1)...succeeded Bringing up ADM VP...succeeded

Creating VP classes...succeeded Forking main_loop thread...succeeded Initializing DR structures...succeeded

Forking 1 'ipcshm' listener threads...succeeded Starting tracing...succeeded

Initializing 1 flushers...succeeded

Clearing encrypted root chunk 1 before initialization... 25% done.

50% done.

75% done.

100% done.

Initializing encryption-at-rest structures (part 2)...succeeded

Initializing log/checkpoint information...succeeded

...

...

Quick Start (3) – Instance Results (default)

- **A new instance with one chunk, encrypted using the default cypher (aes128).**
  - This will be key1 on dbspace1 (rootdbs) for this instance

- **Installation locations, keystore and stash files:**
  - **$INFORMIXDIR/etc/jc_keystore.p12**
  - **$INFORMIXDIR/etc/jc_keystore.sth**
    - When performing a cold restore of an instance to encrypt critical storage spaces these files should be **mv'd** first within the same directory\
      - These files are recreated during the restore.

- **Each space in an instance uses a different encryption key.**
  - Keys 2-2047 are derived from Key 1 at run-time and never stored anywhere on disk.

What's in the Key Store File and Stash Files?

- **The Key Store file ($INFORMIXDIR/etc/<keystore name>.p12) contains a single encryption key, which is used only for ROOTDBS (Dbspace 1).**
  - Key Store file is encrypted.
  - To decrypt the Key Store file, the server needs the Master Key.

- **The Master Key is stored in a stash file**
  - (**$INFORMIXDIR/etc/<keystore name>.sth**)
  - The stash file is encrypted.
  - The server knows how to read it only because IBM **GSKit** knows how to read it.
    - **gskit** is installed with Informix initially
    - See Appendix E for more details

- **Best practice is to store encrypted chunks on a separate disk from $INFORMIXDIR.**

- **Users are expected to back up $INFORMIXDIR with some regularity.**

What's in memory

- Pages in the buffer pool are not encrypted.
- Decryption happens during the read from disk, at a low level in the I/O code.
- Encryption happens at the same low level during a write.
- onstat -g dmp will display decrypted data.
- Shared memory dump files will contain decrypted data, but not encryption keys.

Encryption and Replication

- **Encryption on a secondary is entirely independent of encryption on a primary.**
- **A primary may be encrypted while a secondary is not, and vice-versa.**
- **A different set of spaces may be encrypted in a primary vs. a secondary.**
- **An SDS secondary must use exactly the same encryption keys as used on the primary:**
  - When a shared-disk secondary is first created for an encrypted primary, the primary's keystore file is automatically copied to the secondary's **$INFORMIXDIR/etc** directory
  - File is then encrypted with a master key stored in the stash file

ontape/onbar – Changing Encryption During Restores

- **If storage space encryption is enabled, storage spaces are restored with the same encryption state as during the back up, by default**
  - Can specify to restore storage spaces as encrypted or unencrypted

- **The encryption state of storage spaces on disk does not affect the encryption state of backups**
  - Storage spaces that are encrypted on disk are unencrypted during a backup
    - To encrypt backed up storage spaces, set the **BACKUP_FILTER** configuration parameter to the name of an encryption utility
    - When you restore a storage space that was encrypted on disk before its backup, the storage space is encrypted during the restore, unless you specify to restore the space as unencrypted
    - Similarly, you can restore a storage space that was not encrypted on disk by specifying to encrypt the space during the restore

ontape/onbar – Changing Encryption During Restores

- **The following shows ways you can encrypt and decrypt storage spaces during a physical restore with the ON-Bar or ontape utilities when storage space encryption is enabled:**

Task – Encrypt or Decrypt …..	Method
All existing storage spaces	Full restore with the **-encrypt** or **-decrypt** option. Set/unset **DISK_ENCRYPTION**
Critical storage spaces	Cold restore with the **-encrypt** or **-decrypt** option and specify the spaces with the **-D** option.
Some non-critical storage spaces	Warm restore with the **-encrypt** or **-decrypt** option and specify the spaces with the **-D** option.
All storage spaces for a tenant database	Tenant restore with the **onbar -T** command and include the **-encrypt** or **-decrypt** option.
Storage spaces created by a roll-forward of logical logs	Include **rollfwd_create_dbs=encrypt** or **rollfwd_create_dbs=decrypt** option on the **DISK_ENCRYPTION** parameter value.

ontape/onbar – Changing Encryption During Restores

- **The -encrypt or -decrypt arguments to onbar or ontape apply to the physical restore only:**
  - The server can't use them for the logical restore.

- **Decrypt an entire instance but still enable encryption at rest by setting DISK_ENCRYPTION and perform a cold restore using the -decrypt argument:**

  ontape -r -decrypt

  onbar -r -decrypt

- **During a rollforward, spaces may be re-created.**
  - Assuming Encryption at Rest is enabled, by default they will be created with the same encryption status they were given originally.
- **This default can be overridden by adding rollfwd_create_dbs to the DISK_ENCRYPTION setting, as in:**

  DISK_ENCRYPTION keystore=jc_keystore,rollfwd_create_dbs=encrypt

  DISK_ENCRYPTION keystore=jc_keystore,rollfwd_create_dbs=decrypt

Encryption and Restores

- **During an external restore, storage spaces are restored to the same encryption state as during the backup**
  - Cannot change the encryption state of storage spaces during an external restore

- **When storage space encryption is not enabled, you see the following:**
  - Encrypting storage spaces during a restore with the **-encrypt** option, restore fails
  - Restoring encrypted storage spaces, storage spaces are restored as unencrypted

- **Encrypt all existing storage spaces during a whole-system restore:**

  onbar -r -encrypt -w
- **Encrypt two storage spaces during a physical restore:**

  ontape -p -encrypt -D dbspace1 dbspace2
- **Decrypt all storage spaces that belong to a tenant database:**

  onbar -T tenant1 -decrypt -t "08-08-2016 00:00:00"

How Can I Tell Whether Encryption at Rest Is Enabled?

- **oncheck**
  - **oncheck -pr | head -15**
  - **oncheck –pr | grep rest**
- **select from sysmaster:sysshmhdr**
  - **select value from sysshmhdr where name = "sh_disk_encryption";**
- **Look for "Encryption-at-rest is enabled using cipher" in the message log.**
- **onstat -g dmp**
  - **onstat -g dmp <rhead addr> rhead_t | grep sh_disk_encryption**

Overwriting the Key Store and Stash Files

- **Each instance has its own key store and stash file. Instances cannot share these files, but stored in a directory that may be shared**
  - Instances that share an **$INFORMIXDIR** must use different key store names in their **DISK_ENCRYPTION** settings.

- **We attempt to prevent clobbering of these files by insisting that FULL_DISK_INIT is set before they can be overwritten.**

- **With encryption enabled, the key store and stash files will be overwritten under the following conditions:**
  - **oninit -i**
  - Cold restore
  - Clone creation via **ifxclone**.

Caveats

- **You have to find somewhere to store your keys ……**
  - Future Release
- **Presently, backup data is not encrypted, use BACKUP_FILTER and RESTORE_FILTER with another encryption method established at the O/S level**
  - Future Release
- **Don't forget your keys ……**
  - Fortunately, since backups are not encrypted yet, you can restore the storage spaces unencrypted.
- **Use of ifxclone on an existing instance**

Change the Storage Space Encryption Key

- **master_key reset argument: (SQL administration API)**

- **Use the master_key reset argument with the admin() or task() function to change the master key for storage space encryption**
  - When encryption is first established, a key is automatically generated and stored encrypted in the stash file.
  - User supplied master key of **32** bytes maximum encrypts the keystore for storage space encryption.
    - Users informix or root only may change at any time
    - Stores the new encrypted key in the stash file
    - Accepts no argument, in which case a **random** master key is generated.
      - **Make sure you write it down ……..**

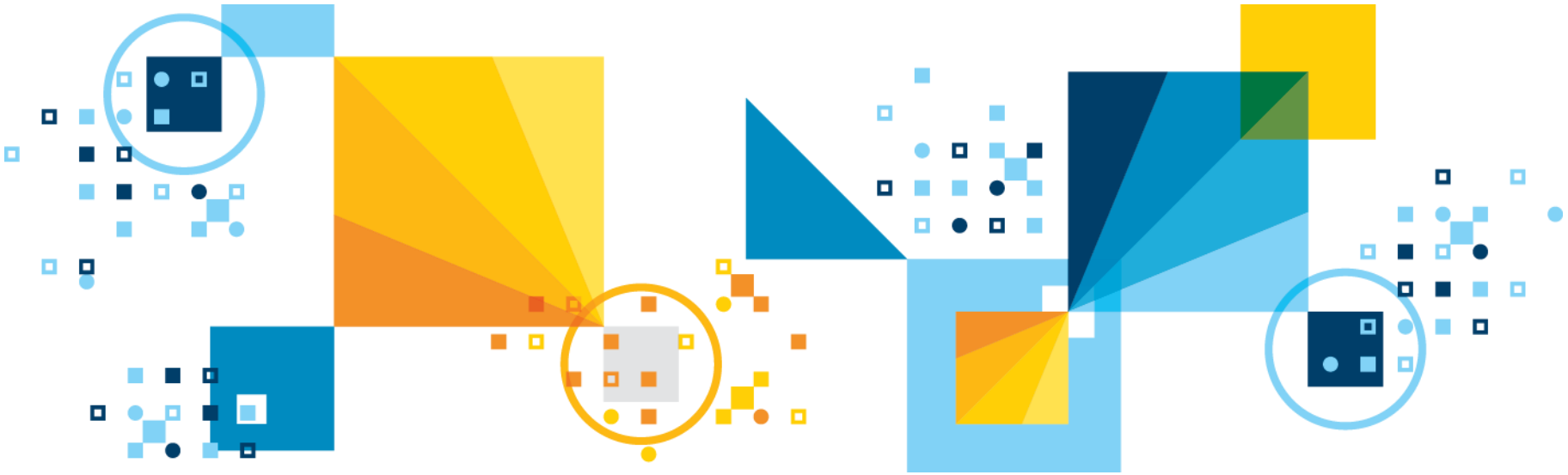
execute function task("master key reset","new master key, hopefully not");

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Encryption of Backups



Encrypted Backups

- **In Informix, presently the backups taken by onbar or ontape are default not encrypted.**
  - This does mean that stored either on disk or on tape, the data on them can be seen.
  - **Ontape** and **onbar** are two different free Informix supplied binaries to perform backups and restores on Informix Databases
    - A backup taken by **ontape** does not work with **onbar**, and vice versa.
    - **Onbar** needs a storage manager, **ontape** does not.
      - Informix supplies a free storage manager, the Primary Storage Manager

- **It is possible to encrypt the backups, and via a user-chosen encryption method, and not necessarily the one used by the Informix database server, which adds an extra level of security.**

- **The example below shows the use of openSSL, which comes free with Linux distributions. Other products could be used as well……**

OpenSSL steps - Simplified

- **Step 1 below generates the key, 2 & 3 are command line filters to encrypt/decrypt data.**

1. **Generate private key (as root, key is stored in the root directory)**
   openssl rand -base64 32 > key.bin

2. **Encryption Filter**
   /usr/bin/openssl enc -aes-256-cbc -salt -pass file:/root/key.bin

3. **Restore filter**
   /usr/bin/openssl enc -d -aes-256-cbc -pass file:/root/key.bin

Informix Configuration for Encryption of Backups

- Informix presently has no facilities to automatically encrypt/decrypt backups as a database server operation.

- Informix has two configuration parameters, BACKUP_FILTER and RESTORE_FILTER, used by both Informix backup and restore utilities, ontape and onbar, that operate on backups and restores in a pass thru manner, applying the setting that is there (which is usually a call to a program to filter and transform the data being backed up or restored).

- Settings 2 and 3 from the previous slide now apply to configuration parameters BACKUP_FILTER and RESTORE_FILTER.
- BACKUP_FILTER

  /usr/bin/openssl enc -aes-256-cbc -salt -pass file:/root/key.bin
- RESTORE_FILTER

  /usr/bin/openssl enc -d -aes-256-cbc -pass file:/root/key.bin

Implementation via ontape shown (1)

- So it looks like this before operations begin

```
Inst_1:
Inst_1: onstat -g cfg BACKUP_FILTER

IBM Informix Dynamic Server Version 12.10.FC8DE -- On-Line -- Up 00:38:10 -- 376676 Kbytes

name                     current value
BACKUP_FILTER            /usr/bin/openssl enc -aes-256-cbc -salt -pass file:/root/key.bin

Inst_1: onstat -g cfg RESTORE_FILTER

IBM Informix Dynamic Server Version 12.10.FC8DE -- On-Line -- Up 00:38:25 -- 376676 Kbytes

name                     current value
RESTORE_FILTER           /usr/bin/openssl enc -d -aes-256-cbc -pass file:/root/key.bin

Inst_1:
```

- Full system backup:

```
Inst_1:
Inst_1: ontape -s -L 0
Using the backup and restore filter /usr/bin/openssl enc -aes-256-cbc -salt -pass file:/root/key.bin.
10 percent done.
100 percent done.
File created: /opt/IBM/informix/backups/inst1_L0

Please label this tape as number 1 in the arc tape sequence.
This tape contains the following logical logs:

 13

Program over.
Inst_1:
```

Implementation via ontape shown (2)

- What does this backup look like, shown here via a cat of the output file (file name from previous slide):
- cat /opt/IBM/informix/backups/inst1_L0

- It’s junk to the eyes. Can I restore an instance with the backup encrypted ? Yes, you can.

Implementation via ontape shown (2)

- **Couple of small steps here:**
  - Server has to be down to do the full system restore
  - The encryption keys and stash file used by Informix encryption at rest must not exist (the existing files can be **mv**'d, before the restore)
- **ontape -r**

```
Inst_1:
Inst_1: ontape -r
Restore will use level 0 archive file /opt/IBM/informix/backups/inst1_L0. Press Return to continue ...

Using the backup and restore filter /usr/bin/openssl enc -d -aes-256-cbc -pass file:/root/key.bin.

Archive Tape Information

Tape type:      Archive Backup Tape
Online version: IBM Informix Dynamic Server Version 12.10.FC8DE
Archive date:   Wed Jun 14 16:25:30 2017
User id:        informix
Terminal id:    /dev/pts/6
Archive level:  0
Tape device:    /opt/IBM/informix/backups/
Tape blocksize (in k): 32
Tape size (in k): system defined for directory
Tape number in series: 1
Backup filter:    /usr/bin/openssl enc -aes-256-cbc -salt -pass file:/root/key.bin

Spaces to restore:1 [rootdbs
2 [data_space_1
3 [data_space_2
4 [log_space
5 [plog_space
6 [slob_space

Archive Information

IBM Informix Dynamic Server Copyright 2001, 2016  IBM Corporation
Initialization Time       05/31/2017 12:30:32
System Page Size          2048
Version                   29
Index Page Logging        OFF
Archive CheckPoint Time   06/14/2017 16:25:29
```

Implementation via ontape shown (2)

- ontape –r (cont'd)

```
Dbspaces
number   flags    fchunk   nchunks  flags    owner                          name
1        10000001 1        1        N  A     informix                       rootdbs
2        10000001 2        1        N  AE    informix                       data_space_1
3        10000001 3        1        N  AE    informix                       data_space_2
4        10000001 4        1        N  AE    informix                       log_space
5        10002001 5        1        N T AE    informix                       work_space
6        10008001 6        1        N S AE    informix                       slob_space
7        11000001 7        1        N P AE    informix                       plog_space


Chunks
chk/dbs offset   size     free     bpages   flags pathname
1   1   0        150000   137713            PO--- /opt/IBM/informix/devices/inst_1/rootspace
2   2   0        256000   246501            PO--- /opt/IBM/informix/devices/inst_1/data1
3   3   0        250000   249939            PO--- /opt/IBM/informix/devices/inst_1/data2
4   4   0        200000   99947             PO--- /opt/IBM/informix/devices/inst_1/logspace
5   5   0        25000    24922             PO--- /opt/IBM/informix/devices/inst_1/tmpspace
6   6   0        25000    1269              POS-- /opt/IBM/informix/devices/inst_1/slob1
7   7   0        37500    0                 PO--E /opt/IBM/informix/devices/inst_1/plogspace

error writing output file
Continue restore? (y/n)y
Do you want to back up the logs? (y/n)n
Using the backup and restore filter /usr/bin/openssl enc -d -aes-256-cbc -pass file:/root/key.bin.
Restore a level 1 archive (y/n) n
Do you want to restore log tapes? (y/n)n
/opt/IBM/informix/12_10/bin/onmode -sy

Program over.
Inst_1: onstat -

IBM Informix Dynamic Server Version 12.10.FC8DE -- Quiescent -- Up 00:00:52 -- 376676 Kbytes

Inst_1: onmode -m
Inst_1: onstat -

IBM Informix Dynamic Server Version 12.10.FC8DE -- On-Line -- Up 00:01:10 -- 376676 Kbytes
```

- So the server came backup. Can we see the data ? The flags columns shows Encrypted dbspaces, which is what we want.

Data via dbaccess

```
NEW:    ESC    = Done editing      CTRL-A = Typeover/Insert     CTRL-R = Redraw
        CTRL-X = Delete character  CTRL-D = Delete rest of line

---------------------- stores@inst_1 ---------- Press CTRL-W for Help --------

select * from my_table
```

```
File  Edit  View  Search  Terminal  Help
SQL:  New  Run  Modify  Use-editor  Output  Choose  Save  Info  Drop  Exit
Run the current SQL statements.

---------------------- stores@inst_1 ---------- Press CTRL-W for Help --------


col1  1
col2  2017-06-08 08:39:12
col3  some data

col1  2
col2  2017-06-08 08:39:12
col3  some more data

col1  3
col2  2017-06-08 08:39:12
col3  some data that I didn't have to type in

col1  4
col2  2017-06-08 08:39:12
col3  still more data

col1  5
col2  2017-06-08 08:39:12
col3  are we done yet?

col1  6
col2  2017-06-16 12:36:20
col3  test of some data
```

Behind the Scenes, Encryption and Cold Restores (1)

- **Remembering for a second that to encrypt an entire instance you need to do a cold restore ……**
  - And, as a prerequisite to the cold restore, to be safe, you should mv the keystore and stash files in $INFORMIXDIR/etc just in case of emergency …….

- **Let's say you have a standalone server that is encrypted. Its keystore file contains Encryption Key EK1. You take an Level-0 archive, and then restore that archive, keeping DISK_ENCRYPTION set in your onconfig file.**

- **The restored instance will be encrypted, but the keystore file will have been recreated during the cold restore, so that now it contains Encryption Key EK2. Any time you do a cold restore with encryption enabled, we blow away and recreate the keystore file, and generate a new encryption key to store in there.**

Behind the Scenes, Encryption and Cold Restores (2)

- The same principle applies to an HDR secondary, which is simply an instance you're creating using a cold restore (this assumes you're not using ifxclone).

- A brand new keystore file will be created for the secondary during that cold restore, which will contain a newly-generated encryption key that has nothing to do with any other instance's key.

- All encrypted instances are entirely independent of one another, in terms of encryption keys, with the obvious exception being a shared-disk (SDS) primary and a shared-disk (SDS) secondary. The key contained in the SDS's keystore file is identical to the primary's. The DBA doesn't have to do anything to that -- IDS takes care of it.

Behind the Scenes, Encryption and Cold Restores (3)

- The encryption feature will encrypt every base page in an encrypted space and does not care at all what the page looks like. The page can contain absolute garbage from top to bottom--we'll still happily encrypt it.

- Whether the pages originally contained some compressed data or uncompressed data or in the future contains some compressed data or uncompressed data is neither here nor there to the encryption feature. They're entirely unaware of each other and therefore compatible."

Backups and Logical Log Encryption

- By default the backups taken, of both the logical logs and the instance dbspaces, are not encrypted.
- You can encrypt them yourself, and quite easily.
- In Linux, for example, most distributions come with openssl.
- This is a different encryption method than that employed by Informix, which relies on GSKit, installed automatically with Informix.
- This feature relies on the BACKUP_FILTER and RESTORE_FILTER parameters being populated in the ONCONFIG file.

Steps to Encrypt via openssl

- **Create a key, on the command line, assuming openssl is present here on the machine:**

  openssl rand -base64 32 > key.bin

- **vi $INFORMIXDIR/etc/$ONCONFIG**

  – Set **BACKUP_FILTER**

    /usr/bin/openssl enc -aes-256-cbc -salt –pass file:/root/key.bin

  – Set **RESTORE_FILTER**

    /usr/bin/openssl enc -d -aes-256-cbc -pass file:/root/key.bin

- **Save the changes and close the file.**

- **Run backups**

Backups Encrypted – Operation / Garbage to See

```
Inst_1:
Inst_1: ontape -s -L 0
Using the backup and restore filter /usr/bin/openssl enc -aes-256-cbc -salt -pass file:/root/key.bin.
10 percent done.
100 percent done.
File created: /opt/IBM/informix/backups/inst1_L0

Please label this tape as number 1 in the arc tape sequence.
This tape contains the following logical logs:

 13

Program over.
Inst_1:
```

Cat the backup file (do not use more, session control loss can result) :

Steps

- **The system needs a cold restore to demonstrate a full encrypted restore to decrypted as you cannot warm restore the rootdbs as it is a critical storage space.**
  - Since the entire backup is encrypted, the restore will apply a decrypted backup to the entire instance, all dbspaces included.
  - Since the instance is still encrypted, we must also move the instance stash and password files.

onmode –c and Enter

onmode –ky and Enter

cd $INFORMIXDIR/etc and Enter

mv sfp_keystore.p12 sfp_keystore.p12.old and Enter

mv sfp_keystore.sth sfp_keystore.sth.old and Enter

ontape –r and Enter

onstat – and Enter

onmode –m and Enter

onstat -d and Enter

onstat –d – Encrypted Still

```
Inst_1:
Inst_1: onstat -d

IBM Informix Dynamic Server Version 12.10.FC8DE -- On-Line -- Up 00:29:04 -- 376676 Kbytes

Dbspaces
address          number   flags      fchunk   nchunks  pgsize   flags    owner    name
44e23028         1        0x10000001 1        1        2048     N  BAE   informix rootdbs
44f53790         2        0x10000001 2        1        2048     N  BAE   informix data_space_1
44f539d0         3        0x10000001 3        1        2048     N  BAE   informix data_space_2
44f53c10         4        0x10000001 4        1        2048     N  BAE   informix log_space
45e45028         5        0x10002001 5        1        2048     N TBAE   informix work_space
45e45268         6        0x10008001 6        1        2048     N SBAE   informix slob_space
45e454a8         7        0x11000001 7        1        2048     N PBAE   informix plog_space
 7 active, 2047 maximum

Chunks
address          chunk/dbs     offset     size       free       bpages     flags pathname
44e23268         1      1      0          150000     137713                PO-B-- /opt/IBM/informix/devices/inst_1/rootspace
45e46028         2      2      0          256000     246501                PO-B-- /opt/IBM/informix/devices/inst_1/data1
45e47028         3      3      0          250000     249939                PO-B-- /opt/IBM/informix/devices/inst_1/data2
45e48028         4      4      0          200000     99947                 PO-B-- /opt/IBM/informix/devices/inst_1/logspace
45e49028         5      5      0          25000      24947                 PO-B-- /opt/IBM/informix/devices/inst_1/tmpspace
45e4a028         6      6      0          25000      23128      23241      POSB-- /opt/IBM/informix/devices/inst_1/slob1
                                 Metadata 1706       1269       1706
45e4b028         7      7      0          37500      0                     PO-BE- /opt/IBM/informix/devices/inst_1/plogspace
 7 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are
      displayed in terms of "pgsize" of the DBspace to which they belong.


Expanded chunk capacity mode: always

Inst_1:
```

Restore (1)

```
Inst_1:
Inst_1: ontape -r
Restore will use level 0 archive file /opt/IBM/informix/backups/inst1_L0. Press Return to continue ...

Using the backup and restore filter /usr/bin/openssl enc -d -aes-256-cbc -pass file:/root/key.bin.

Archive Tape Information

Tape type:      Archive Backup Tape
Online version: IBM Informix Dynamic Server Version 12.10.FC8DE
Archive date:   Wed Jun 14 16:25:30 2017
User id:        informix
Terminal id:    /dev/pts/6
Archive level:  0
Tape device:    /opt/IBM/informix/backups/
Tape blocksize (in k): 32
Tape size (in k): system defined for directory
Tape number in series: 1
Backup filter:    /usr/bin/openssl enc -aes-256-cbc -salt -pass file:/root/key.bin

Spaces to restore:1 [rootdbs
2 [data_space_1
3 [data_space_2
4 [log_space
5 [plog_space
6 [slob_space

Archive Information

IBM Informix Dynamic Server Copyright 2001, 2016  IBM Corporation
Initialization Time       05/31/2017 12:30:32
System Page Size          2048
Version                   29
Index Page Logging        OFF
Archive CheckPoint Time   06/14/2017 16:25:29
```

Restore (2)

```
Dbspaces
number   flags    fchunk   nchunks  flags    owner                          name
1        10000001 1        1        N   AE   informix                       rootdbs
2        10000001 2        1        N   AE   informix                       data_space_1
3        10000001 3        1        N   AE   informix                       data_space_2
4        10000001 4        1        N   AE   informix                       log_space
5        10002001 5        1        N T AE   informix                       work_space
6        10008001 6        1        N S AE   informix                       slob_space
7        11000001 7        1        N P AE   informix                       plog_space


Chunks
chk/dbs offset   size     free     bpages   flags pathname
1   1   0        150000   137713            PO--- /opt/IBM/informix/devices/inst_1/rootspace
2   2   0        256000   246501            PO--- /opt/IBM/informix/devices/inst_1/data1
3   3   0        250000   249939            PO--- /opt/IBM/informix/devices/inst_1/data2
4   4   0        200000   99947             PO--- /opt/IBM/informix/devices/inst_1/logspace
5   5   0        25000    24922             PO--- /opt/IBM/informix/devices/inst_1/tmpspace
6   6   0        25000    1269              POS-- /opt/IBM/informix/devices/inst_1/slob1
7   7   0        37500    0                 PO--E /opt/IBM/informix/devices/inst_1/plogspace

error writing output file
Continue restore? (y/n)y
Do you want to back up the logs? (y/n)n
Using the backup and restore filter /usr/bin/openssl enc -d -aes-256-cbc -pass file:/root/key.bin.
Restore a level 1 archive (y/n) n
Do you want to restore log tapes? (y/n)n
/opt/IBM/informix/12_10/bin/onmode -sy

Program over.
Inst_1: onstat -

IBM Informix Dynamic Server Version 12.10.FC8DE -- Quiescent -- Up 00:00:52 -- 376676 Kbytes

Inst_1: onmode -m
Inst_1: onstat -

IBM Informix Dynamic Server Version 12.10.FC8DE -- On-Line -- Up 00:01:10 -- 376676 Kbytes
```

Restore (3)

```
Inst_1: onstat -d

IBM Informix Dynamic Server Version 12.10.FC8DE -- On-Line -- Up 00:03:14 -- 376676 Kbytes

Dbspaces
address          number   flags      fchunk   nchunks  pgsize   flags    owner    name
44e23028         1        0x10000001 1        1        2048     N  BAE   informix rootdbs
44f53790         2        0x10000001 2        1        2048     N  BAE   informix data_space_1
44f539d0         3        0x10000001 3        1        2048     N  BAE   informix data_space_2
44f53c10         4        0x10000001 4        1        2048     N  BAE   informix log_space
45e45028         5        0x10002001 5        1        2048     N TBAE   informix work_space
45e45268         6        0x10008001 6        1        2048     N SBAE   informix slob_space
45e454a8         7        0x11000001 7        1        2048     N PBAE   informix plog_space
 7 active, 2047 maximum

Chunks
address          chunk/dbs     offset     size       free       bpages     flags pathname
44e23268         1      1      0          150000     137713               PO-B-- /opt/IBM/informix/devices/inst_1/rootspace
45e46028         2      2      0          256000     246501               PO-B-- /opt/IBM/informix/devices/inst_1/data1
45e47028         3      3      0          250000     249939               PO-B-- /opt/IBM/informix/devices/inst_1/data2
45e48028         4      4      0          200000     99947                PO-B-- /opt/IBM/informix/devices/inst_1/logspace
45e49028         5      5      0          25000      24947                PO-B-- /opt/IBM/informix/devices/inst_1/tmpspace
45e4a028         6      6      0          25000      23128      23241     POSB-- /opt/IBM/informix/devices/inst_1/slob1
                                 Metadata 1706       1269       1706
45e4b028         7      7      0          37500      0                    PO-BE- /opt/IBM/informix/devices/inst_1/plogspace
 7 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are
      displayed in terms of "pgsize" of the DBspace to which they belong.


Expanded chunk capacity mode: always
```

Restore (4)

```
NEW:    ESC    = Done editing      CTRL-A = Typeover/Insert     CTRL-R = Redraw
        CTRL-X = Delete character  CTRL-D = Delete rest of line

---------------------- stores@inst_1 ---------- Press CTRL-W for Help --------

select * from my_table
```

```
File  Edit  View  Search  Terminal  Help
SQL:  New  Run  Modify  Use-editor  Output  Choose  Save  Info  Drop  Exit
Run the current SQL statements.

---------------------- stores@inst_1 ---------- Press CTRL-W for Help --------


col1  1
col2  2017-06-08 08:39:12
col3  some data

col1  2
col2  2017-06-08 08:39:12
col3  some more data

col1  3
col2  2017-06-08 08:39:12
col3  some data that I didn't have to type in

col1  4
col2  2017-06-08 08:39:12
col3  still more data

col1  5
col2  2017-06-08 08:39:12
col3  are we done yet?

col1  6
col2  2017-06-16 12:36:20
col3  test of some data
```

Logical Log Encryption (1)

- When backed up, presently the backups of logical logs are default not encrypted.

- The same BACKUP_FILTER and RESTORE_FILTER commands with openssl can be employed to encrypt/decrypt the backups of the logical logs.

- Steps (assumes the existence of logical logs needing backup):

- ontape –a # May generate an error

```
Inst_1:
Inst_1: ontape -a
The LTAPESIZE configuration parameter cannot be set to 0 when the BACKUP_FILTER
configuration parameter is set; change the value of LTAPESIZE.

Program over.
Inst_1: onstat -g cfg LTAPESIZE

IBM Informix Dynamic Server Version 12.10.FC8DE -- On-Line -- Up 02:06:26 -- 376676 Kbytes

name                     current value
LTAPESIZE                0

Inst_1:
```

Logical Log Encryption (2)

- **Error generated results from BACKUP_FILTER value being changed.**
- **If the BACKUP_FILTER parameter is set in the ONCONFIG file, the LTAPESIZE cannot be set to 0. Otherwise the ontape utility returns an error when backing up logical logs to a directory on disk.**
  - **onbar** ignores this parameter
- **The error message is:**
- **"The LTAPESIZE configuration parameter cannot be set to 0 when the BACKUP_FILTER configuration parameter is set; change the value of LTAPESIZE. Program over."**
- **A workaround is to set the LTAPESIZE configuration parameter to a very high value. Log files are not much higher than the LOGSIZE configuration parameter. Use the value in the LOGSIZE as the upper limit for this database.**

Logical Log Encryption (3

- So set the LTAPEDEV parameter slightly higher than the value of LOGSIZE, say around 10% higher or so to account for the encryption.

- ontape –a

```
Inst_1: ontape -a

Performing automatic backup of logical logs.
Using the backup and restore filter /usr/bin/openssl enc -aes-256-cbc -salt -pass file:/root/key.bin.

File created: /opt/IBM/informix/backups/inst1_Log0000000013
Do you want to back up the current logical log? (y/n) y
Using the backup and restore filter /usr/bin/openssl enc -aes-256-cbc -salt -pass file:/root/key.bin.
File created: /opt/IBM/informix/backups/inst1_Log0000000014

Program over.
Inst_1:
```

- So it is applying the encryption to the logical logs backup as well.

- Next do the backup:

Logical Log Encryption (4)

- ontape –s –L 0

```
Inst_1: ontape -a

Performing automatic backup of logical logs.
Using the backup and restore filter /usr/bin/openssl enc -aes-256-cbc -salt -pass file:/root/key.bin.

File created: /opt/IBM/informix/backups/inst1_Log0000000013
Do you want to back up the current logical log? (y/n) y
Using the backup and restore filter /usr/bin/openssl enc -aes-256-cbc -salt -pass file:/root/key.bin.
File created: /opt/IBM/informix/backups/inst1_Log0000000014

Program over.
Inst_1:
```

- onmode –c and Enter
- onmode –ky and Enter
- cd $INFORMIXDIR/etc and Enter
- mv filename.p12 filename.p12.old and Enter
- mv filename.sth filename.sth.old and Enter

Logical Log Encryption (5)

- **ontape –r** and Enter

```
Inst_1:
Inst_1: ontape -r
Restore will use level 0 archive file /opt/IBM/informix/backups/inst1_L0. Press Return to continue ...

Using the backup and restore filter /usr/bin/openssl enc -d -aes-256-cbc -pass file:/root/key.bin.

Archive Tape Information

Tape type:      Archive Backup Tape
Online version: IBM Informix Dynamic Server Version 12.10.FC8DE
Archive date:   Fri Jun 16 13:07:43 2017
User id:        informix
Terminal id:    /dev/pts/2
Archive level:  0
Tape device:    /opt/IBM/informix/backups/
Tape blocksize (in k): 32
Tape size (in k): system defined for directory
Tape number in series: 1
Backup filter:  /usr/bin/openssl enc -aes-256-cbc -salt -pass file:/root/key.bin

Spaces to restore:1 [rootdbs                                                                                    ]
2 [data_space_1                                                                                                 ]
3 [data_space_2                                                                                                 ]
4 [log_space                                                                                                    ]
5 [plog_space                                                                                                   ]
6 [slob_space                                                                                                   ]

Archive Information

IBM Informix Dynamic Server Copyright 2001, 2016  IBM Corporation
Initialization Time       05/31/2017 12:30:32
System Page Size          2048
Version                   29
Index Page Logging        OFF
Archive CheckPoint Time   06/16/2017 13:07:43
```

Logical Log Encryption (6)

- ontape –r (cont'd)

- Answer y to continue restore, n to back up logs, n to restore a level-1 archive, y to restore logical log tapes.
- onstat –
- onmode –m
- onstat -

Logical Log Encryption (7)

```
Dbspaces
number   flags    fchunk   nchunks  flags    owner                          name
1        10000001 1        1        N   AE   informix                       rootdbs
2        10000001 2        1        N   AE   informix                       data_space_1
3        10000001 3        1        N   AE   informix                       data_space_2
4        10000001 4        1        N   AE   informix                       log_space
5        10002001 5        1        N T AE   informix                       work_space
6        10008001 6        1        N S AE   informix                       slob_space
7        11000001 7        1        N P AE   informix                       plog_space


Chunks
chk/dbs offset   size     free     bpages   flags pathname
1   1   0        150000   137709            PO--- /opt/IBM/informix/devices/inst_1/rootspace
2   2   0        256000   246501            PO--- /opt/IBM/informix/devices/inst_1/data1
3   3   0        250000   249939            PO--- /opt/IBM/informix/devices/inst_1/data2
4   4   0        200000   99947             PO--- /opt/IBM/informix/devices/inst_1/logspace
5   5   0        25000    24922             PO--- /opt/IBM/informix/devices/inst_1/tmpspace
6   6   0        25000    1269              POS-- /opt/IBM/informix/devices/inst_1/slob1
7   7   0        37500    0                 PO--E /opt/IBM/informix/devices/inst_1/plogspace

error writing output file
Continue restore? (y/n)y
Do you want to back up the logs? (y/n)n
Using the backup and restore filter /usr/bin/openssl enc -d -aes-256-cbc -pass file:/root/key.bin.
Restore a level 1 archive (y/n) n
Do you want to restore log tapes? (y/n)y

Roll forward should start with log number 15
Restore will use log backup file /opt/IBM/informix/backups/inst1_Log0000000015. Press Return to continue ...

Using the backup and restore filter /usr/bin/openssl enc -d -aes-256-cbc -pass file:/root/key.bin.
Rollforward log file /opt/IBM/informix/backups/inst1_Log0000000015 ...

Program over.
Inst_1: onstat -

IBM Informix Dynamic Server Version 12.10.FC8DE -- Quiescent -- Up 00:04:43 -- 376676 Kbytes

Inst_1: onmode -m
Inst_1: onstat -

IBM Informix Dynamic Server Version 12.10.FC8DE -- On-Line -- Up 00:04:52 -- 376676 Kbytes
```

Logical Log Encryption (8)

- **hexdump -C /opt/ibm/informix/backups/inst1_Log0000000015 | more**

```
0000e570  07 5e 4f e1 80 6e 01 40  6a e9 a1 66 54 2e f5 13  |.^O..n.@j..fT...|
0000e580  72 95 e8 64 02 26 1d 81  25 95 6e f1 aa f5 a0 08  |r..d.&..%.n.....|
0000e590  c9 06 db 93 4d 8b 7e 46  55 69 07 e3 5d 0c 6f 3e  |....M.~FUi..].o>|
0000e5a0  33 b0 3e 53 e5 b0 2f 1c  bd 32 8b cf 75 c7 71 f5  |3.>S../..2..u.q.|
0000e5b0  c7 96 88 13 78 62 3a 61  2d 2c f7 ef 2d 7a 33 4f  |....xb:a-,..-z3O|
0000e5c0  dc a7 c0 57 91 16 ff 8d  99 78 35 93 64 2a 9d 2b  |...W.....x5.d*.+|
0000e5d0  ef e1 dd 46 05 1c 86 5e  7d f1 27 2b 43 a0 d2 3b  |...F...^}.'+C..;|
0000e5e0  4a 07 93 45 54 cf de 13  10 d8 5a 2d d5 22 e1 ad  |J..ET.....Z-."..|
0000e5f0  92 c8 b8 de a1 cf 30 ab  b1 ec 49 d4 24 5e 96 45  |......0...I.$^.E|
0000e600  84 26 dc e3 50 7c 4f f8  a4 31 c8 f8 ea 6e 9e fa  |.&..P|O..1...n..|
0000e610  91 f3 ec 94 8a 6c 66 b8  7f 34 b4 2e 3f 0c a6 61  |.....lf..4..?..a|
0000e620  f3 53 c8 5a e8 96 72 3f  d6 df a0 b1 dc 99 bd f5  |.S.Z..r?........|
0000e630  84 55 dd a7 31 fd 3c df  28 28 16 52 93 94 d0 32  |.U..1.<.((.R...2|
0000e640  7d 54 b1 13 89 8f 74 b6  6c 9d 42 30 57 0c 14 ae  |}T....t.l.B0W...|
0000e650  c1 07 c8 25 b5 32 92 f4  80 c7 86 8e 1a 97 2a 4d  |...%.2........*M|
0000e660  ff 4a a1 ff 91 a0 fa 55  e6 dd d4 dc 09 3e 2d e0  |.J.....U.....>-.|
0000e670  90 3a 2e ae cd 37 4e da  0e e5 d3 eb a4 03 92 ab  |.:...7N.........|
0000e680  76 c7 44 d3 8e 08 d5 14  82 48 54 4a e8 a9 da 7c  |v.D......HTJ...||
0000e690  a7 16 ac c4 f9 d5 90 b4  ab 5d 8b 62 8a a8 d8 5c  |.........].b...\|
0000e6a0  f2 00 81 d4 e1 1d 01 eb  0b 99 9a 87 7f 89 7e 52  |..............~R|
0000e6b0  66 e2 20 57 8d 03 0e e8  7a 20 f0 78 ef 63 3b fe  |f. W....z .x.c;.|
0000e6c0  3d be 47 0d 97 5a 01 7f  f2 5e a9 33 61 82 75 74  |=.G..Z...^.3a.ut|
0000e6d0  58 78 50 32 5d 2e f0 c0  69 d5 11 e9 00 cb 9c 9a  |XxP2]...i.......|
0000e6e0  f0 81 da 61 a7 57 f6 66  99 c3 ec f6 ef 66 99 30  |...a.W.f.....f.0|
0000e6f0  73 8f 2c 80 6d 88 d5 fe  76 1d 72 a1 61 6e b8 c8  |s.,.m...v.r.an..|
0000e700  e0 24 ab fb 40 6f d9 3c  25 60 99 b6 37 91 0f dc  |.$..@o.<%`..7...|
0000e710  af d9 44 63 76 cf 7f cf  cd fd b5 7b d7 d8 e2 9e  |..Dcv......{....|
0000e720  51 ac c0 d1 cc 0f c3 a1  c3 af 30 b7 92 e6 ba 01  |Q.........0.....|
0000e730  29 ba 31 1d 24 7a 8f 76  15 cb bf c2 ba bc 5e d0  |).1.$z.v......^.|
0000e740  86 6e f1 3f ac a6 d4 60  7a 67 92 32 fb b6 5f b9  |.n.?...`zg.2.._.|
0000e750  c5 63 5e 90 b6 1a 61 6f  88 4d 07 7f 00 d4 a4 f0  |.c^...ao.M......|
0000e760  12 c4 45 16 b4 fb 84 ce  f5 cf 45 f1 e1 ae fd 60  |..E.......E....`|
0000e770  fb 78 ee b6 c0 e5 6d 57  a2 d3 0d be 60 b4 7f 94  |.x....mW....`...|
0000e780  89 d1 74 cb 6b 2a d8 03  d1 4c 6f 13 1d 9f cd 00  |..t.k*...Lo.....|
0000e790  88 d3 3a 83 e0 f7 3c 6f  e2 02 3a cd 47 5f ba 7c  |..:...<o..:.G_.||
0000e7a0  00 be 93 02 31 9c bc b2  5e 41 6b 2d d5 b4 24 fc  |....1...^Ak-..$.|
0000e7b0  b0 74 d6 9e c7 db 75 27  c2 46 55 1e 5d f4 0b 64  |.t....u'.FU.]..d|
0000e7c0  a0 21 b9 4a 4b a7 1a c7  a2 de 57 cc e8 23 a9 5e  |.!.JK.....W..#.^|
0000e7d0  ee 25 c4 f7 0b 0b fb 39  ae 37 20 f9 30 82 80 a9  |.%.....9.7 .0...|
0000e7e0  51 31 39 bf ec ff 30 cb  af 19 ac 6d 4b 1a 79 a0  |Q19...0....mK.y.|
0000e7f0  18 f0 b2 32 53 31 c3 01  25 98 b0 c2 ab 19 6d af  |...2S1..%.....m.|
0000e800  1d 9d 70 30 bd 79 d2 8d  c0 42 2f a5 67 26 67 ef  |..p0.y...B/.g&g.|
```

Top 12 pages or so are visible in each logical log file, they contain header info only and not data. This is from the middle or so of the file.

We're up – Is the Instance Encrypted Still ?

```
Inst_1: onstat -d

IBM Informix Dynamic Server Version 12.10.FC8DE -- On-Line -- Up 00:03:14 -- 376676 Kbytes

Dbspaces
address          number   flags      fchunk   nchunks  pgsize   flags    owner    name
44e23028         1        0x10000001 1        1        2048     N  BAE   informix rootdbs
44f53790         2        0x10000001 2        1        2048     N  BAE   informix data_space_1
44f539d0         3        0x10000001 3        1        2048     N  BAE   informix data_space_2
44f53c10         4        0x10000001 4        1        2048     N  BAE   informix log_space
45e45028         5        0x10002001 5        1        2048     N TBAE   informix work_space
45e45268         6        0x10008001 6        1        2048     N SBAE   informix slob_space
45e454a8         7        0x11000001 7        1        2048     N PBAE   informix plog_space
 7 active, 2047 maximum

Chunks
address          chunk/dbs     offset     size       free       bpages     flags pathname
44e23268         1      1      0          150000     137713               PO-B-- /opt/IBM/informix/devices/inst_1/rootspace
45e46028         2      2      0          256000     246501               PO-B-- /opt/IBM/informix/devices/inst_1/data1
45e47028         3      3      0          250000     249939               PO-B-- /opt/IBM/informix/devices/inst_1/data2
45e48028         4      4      0          200000     99947                PO-B-- /opt/IBM/informix/devices/inst_1/logspace
45e49028         5      5      0          25000      24947                PO-B-- /opt/IBM/informix/devices/inst_1/tmpspace
45e4a028         6      6      0          25000      23128      23241     POSB-- /opt/IBM/informix/devices/inst_1/slob1
                                  Metadata 1706       1269       1706
45e4b028         7      7      0          37500      0                    PO-BE- /opt/IBM/informix/devices/inst_1/plogspace
 7 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are
      displayed in terms of "pgsize" of the DBspace to which they belong.


Expanded chunk capacity mode: always
```

We’re up and encrypted, Can I see data?

```
NEW:    ESC    = Done editing      CTRL-A = Typeover/Insert     CTRL-R = Redraw
        CTRL-X = Delete character  CTRL-D = Delete rest of line

---------------------- stores@inst_1 ---------- Press CTRL-W for Help --------

select * from my_table
```

```
File  Edit  View  Search  Terminal  Help
SQL:  New  Run  Modify  Use-editor  Output  Choose  Save  Info  Drop  Exit
Run the current SQL statements.

---------------------- stores@inst_1 ---------- Press CTRL-W for Help --------


col1  1
col2  2017-06-08 08:39:12
col3  some data

col1  2
col2  2017-06-08 08:39:12
col3  some more data

col1  3
col2  2017-06-08 08:39:12
col3  some data that I didn't have to type in

col1  4
col2  2017-06-08 08:39:12
col3  still more data

col1  5
col2  2017-06-08 08:39:12
col3  are we done yet?

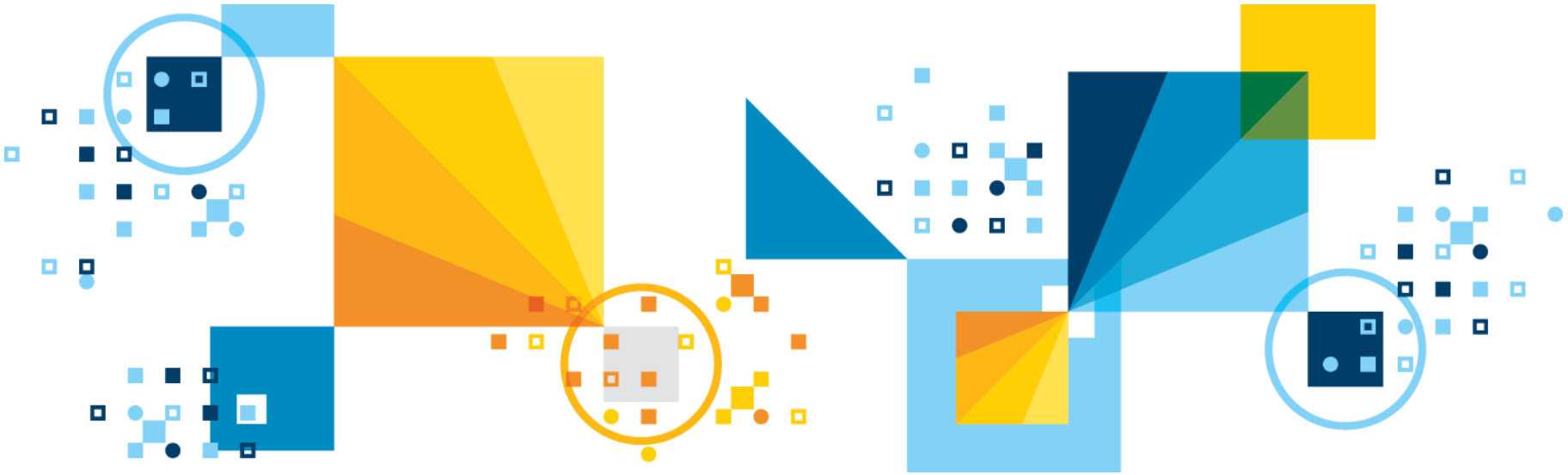
col1  6
col2  2017-06-16 12:36:20
col3  test of some data
```

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Client to Server, Server to Server Encrypted Communications



Server/Server and Client/Server Encryption via SSL & TLS

- **Secure Sockets Layer (SSL) and its successor, Transport Layer Security (TLS), are communication protocols that use encryption to provide privacy and integrity for data communication through a reliable end-to-end secure connection between two network points.**
- **Usage with:**
  - IBM Data Server Driver for JDBC and SQLJ connections with Informix
  - IBM Informix ESQL/C connections with Informix
  - IBM Informix ODBC Driver connections with Informix
  - DB-Access connections
  - Enterprise Replication connections
  - High-availability data replication (HDR) connections between an HDR primary server and one or more secondary servers of any type (HDR secondary, SD secondary, or RS secondary)
  - Distributed transaction connections, which span multiple database servers
  - The dbexport, dbimport, dbschema, and dbload utility connections
  - Connection Manager connections between servers in a cluster

Advantages of SSL over Informix CSM

- **SSL provides these advantages over the Informix Communication Support Modules (CSMs):**
  - SSL is a more widely used alternative to the IBM Informix CSMs.
  - Can use SSL for encrypted communication with both DRDA and SQLI clients.
  - Can use the CSMs only for connections with SQLI clients; you cannot use them for connections with DRDA clients.

- **Can configure the Encrypt and Simple Password Communications Support Modules (ENCCSM and SPWDCSM) with SSL connections:**
  - These CSMs provide encryption functionality, so configuring the ENCCSM or SPWDCSM with SSL involves additional effort with no extra benefit.

- **Can configure Pluggable Authentication Module (PAM) and the Generic Security Services Communications Support Module (GSSCSM), which uses the Kerberos 5 security protocol for single sign-on (SSO) with SSL connections.**

Digital certificates

- **SSL uses digital certificates, which are electronic ID cards issued by a trusted party, to exchange keys for encryption and server authentication.**
  - Trusted entity issuing a digital certificate is known as a Certificate Authority (CA).
  - The CA issues a digital certificate for only a limited time.
  - When the expiration date passes, you must acquire another digital certificate.

- **With SSL, the data that moves between a client and server is encrypted by a symmetric key (secret or private key) algorithm.**

- **An asymmetric key (public key) algorithm is used for the exchange of the secret keys in the symmetric algorithm.**

Digital certificates

- **When a client attempts to connect to a secure server, an SSL handshake occurs. The handshake involves the following events:**
  - The server sends its digital certificate to the client.
  - The client verifies the validity of the server digital certificate. For this to occur, the client must possess the digital certificate of the CA that issued the server digital certificate.

- **If the handshake succeeds, these events occur:**
  - The client generates a random symmetric key and sends it to the server, in an encrypted form, by using the asymmetric key in the server digital certificate.
  - The server retrieves the symmetric key by decrypting it.

- **Because the server and client now know and can use the symmetric key, the server and client encrypt data for the session duration.**

Keystores

- **A keystore is a protected database that stores SSL keys and digital certificates. Both the client and server must have the keystore that stores the digital certificates used in SSL communication.**

- **The keystore stores its digital certificate and the root CA certificate of all other servers that Informix is connecting to.**

- **The server keystore must be located in the $INFORMIXDIR/ssl directory. The name of the keystore file must be *server_name*.kdb, where *server_name* is the value specified in the DBSERVERNAME configuration parameter.**
  - Each Informix instance must have its own keystore.

Server Keystore

- **Each certificate in the keystore has a unique label. When you set up Informix to use SSL, specify the name of the label of the digital certificate in the SSL_KEYSTORE_LABEL configuration parameter in the onconfig file.**

- **If you do not specify a label name in the SSL_KEYSTORE_LABEL configuration parameter, Informix uses the default certificate in the keystore for SSL communication.**
  - Only one certificate in a keystore is the default certificate.

- **The keystore is protected by a password that Informix must know so that it can retrieve the digital certificate for SSL communications. Informix stores its keystore password in an encrypted form in a stash (.sth) file in the $INFORMIXDIR/ssl directory.**
  - The name of the keystore stash file must be ***server_name*.sth**.

Server Keystore

- The password for the keystore is mandatory, because this password protects the private key for the server.

- The permissions on the $INFORMIXDIR/ssl/*server_name*.kdb and $INFORMIXDIR/ssl/*server_name*.sth files must be 600, with informix set as both the owner and the group, even though Informix does not enforce these permissions.

Client Keystore

- **The keystore on the Informix client stores the root CA certificates of all servers to which the client is connecting**
  - A password for the keystore is optional on the client.

- **For Informix SQLI clients (ESQL/C, ODBC, DB-Access, and the dbexport, dbimport, dbschema, and dbload utilities), the location of the keystore and its stash file is not fixed. Instead, the conssl.cfg file in the $INFORMIXDIR/etc directory specifies the keystore and the stash file for Informix clients**

- **The following table shows the client configuration parameters that are in the conssl.cfg file: (over)**

Client Keystore Configuration

Parameter	Description
SSL_KEYSTORE_FILE	This is the fully qualified file name of the keystore that stores the root CA certificates of all of the servers to which the client connects.
SSL_KEYSTORE_STH	This is the fully qualified file name of the stash file containing the encrypted keystore password.

- **If a conssl.cfg file does not exist or the SSL_KEYSTORE_FILE and SSL_KEYSTORE_STH configuration parameters are not set, the client uses $INFORMIXDIR/etc/client.kdb and $INFORMIXDIR/etc/client.sth as the default keystore and keystore stash file names for the client.**

Client sqlhosts File Configuration

- **Update connection information in the sqlhosts file by using the onsocssl protocol for SSL SQLI client connections.**

- **sqlhosts file configured for these client connections:**

Server Name	Protocol	Host Name	Service Name
sf_on	onsoctcp	sanfrancisco	sf_serv
oak_on	onsocssl	oakland	oak_serv

- **Using a text editor, create a conssl.cfg file in the $INFORMIXDIR/etc directory. The file must contain the following information:**
  - **SSL_KEYSTORE_FILE** information that specifies the fully qualified file name of the keystore that stores the root CA certificates of all of the servers to which the client connects
  - **SSL_KEYSTORE_STH** information that specifies the fully qualified file name of the stash file containing the encrypted keystore password.

conssl.cfg

- The format of the conssl.cfg file is:

```
Parameter    Value    # Comment
```

- For example, the conssl.cfg file might contain this information:

```
SSL_KEYSTORE_FILE   /work/keystores/ssl_client.kdb   # Keystore file
SSL_KEYSTORE_STH    /work/keystores/ssl_client.sth   # Keystore stash file
```

GSKCapiCmd

- **Use the GSKCapiCmd command-line interface, which is a part of the GSKit Java runtime environment, to set up a keystore and its password stash file and digital certificate.**

- **When you create the password, be sure that:**
  - You use the command associated with the installed version of GSKit (for example, **gsk7capicmd** or **gsk8capicmd**).
  - The name and location of the keystore and its stash file are as specified in the **conssl.cfg** file.
  - Permissions on the keystore and its stash file are set to 666, even though the permissions are not enforced.

- **If the certificate created for server is self-signed, you must extract the certificate from the server and use FTP to move the extracted certificate to the client, for the client keystore to use. If you use the provided default certificates, you must create the client keystore.**

Keystore Creation

- **For example:**
  - If the certificate is self-signed or is a default CA certificate, run the following commands on the client to create the keystore and add your certificate:

```
GSK_COMMAND -keydb -create -db client.kdb -pw PASSWORD -type cms -stash
```

Certificate Operations

- **If the certificate created for the server is self-signed, additionally:**
  - Log on to the remote server and extract the certificate from the server keystore:

    GSK_COMMAND -cert -extract -db $INFORMIXSERVER.kdb -format ascii -label

    SSL_KEYSTORE_LABEL -pw PASSWORD -target SSL_KEYSTORE_LABEL.cert

  - Use FTP to move the extracted certificate to your client.
  - Add the certificate to the client keystore:

    GSK_COMMAND -cert -add -db client.kdb -pw PASSWORD -label SSL_KEYSTORE_LABEL -file SSL_KEYSTORE_LABEL.cert -format ascii

- **Add the digital certificate of the Certificate Authority that issued the server digital certificate to the keystore.**

SSL Server Configuration

- **Configure Informix for SSL connections by adding connection information to the sqlhosts file, setting SSL configuration parameters, and configuring the keystore and the digital certificates it stores.**

- **To configure the Informix instance for SSL connections:**
  - Update connection information in the **sqlhosts** file to include information about SSL connections. Use the:
    - **onsocssl** protocol for **ESQL/C, ODBC, DB-Access, dbexport** utility, **dbimport** utility, **dbschema** utility, or **dbload** utility connections
    - **drsocssl** protocol for DRDA connections

- **The following table shows an example of an sqlhosts file configured for both SSL and non-SSL connections: (over)**

SQLHOSTS Configuration

Server Name	Protocol	Host Name	Service Name
sf_on	onsoctcp	sanfrancisco	sf_serv
oak_on	onsocssl	oakland	oak_serv
sac_on	drsocssl	sacramento	sac_serv

Server name value is same as **DBSERVERNAME** in the **onconfig** file.
Host name is the same as in **/etc/hosts** or the result of the **hostname** command
Service name (or number) is the same as that in **/etc/services**.

ONCONFIG Configuration

- **Specify the name of the label of the server digital certificate in the SSL_KEYSTORE_LABEL configuration parameter.**
  - The label can contain up to 512 bytes.
  - If a label name is not specified, Informix uses the default keystore certificate.
  - For example:
    - specify: **SSL_KEYSTORE_LABEL sf_ssl**

- **Configure poll threads for SSL connections by using the NETTYPE configuration parameter, if not configured, Informix starts one poll thread.**
  - For the protocol, specify **socssl.**
    - The protocol format is iiippp, where iii=[ipc|soc|tli] and ppp=[shm|str|tcp|imc|ssl].
  - For example, specify: **NETTYPE socssl,3,50,NET**

ONCONFIG Configuration

- **Configure Encrypt Virtual Processors (VPs) for SSL encryption and decryption operations, by using the VPCLASS parameter.**
  - If Encrypt VPs are not configured, Informix starts one Encrypt VP the first time an SSL operation occurs.

- **You can also use the onmode -p command to add or drop Encrypt VPs when the database server is in online mode.**
  - For large systems, configure multiple Encrypt VPs.

Keystore Setup

- **Set up a keystore and its password stash file and digital certificate by using the iKeyman utility, GSKCmd command-line interface, or GSKCapiCmd command-line interface. To use the iKeyman utility and GSKCmd tool, a supported Java runtime environment must be installed. The GSKCapiCmd tool is a part of the GSKit and does not require Java.**
- **When you create the password, be sure to:**
  - Select the option to stash the password to a file.
- **Name the keystore as *servername*.kdb, where *servername* is value of the DBSERVERNAME configuration parameter.**
- **Create the keystore and its stash file in $INFORMIXDIR/ssl directory.**
- **Set the permissions on the $INFORMIXDIR/ssl/*server_name*.kdb and $INFORMIXDIR/ssl/*server_name*.sth files to 600, with informix set as both the owner and the group.**

Keystore Setup – GSKit commands

gsk8capicmd -keydb -create -db sf_server.kdb -pw sf_password -type cms -stash

gsk8capicmd -cert -create -db sf_server.kdb -pw sf_password -label my_ssl_label -size 1024 -default_cert yes

- **Important:**
  - If the DBSA configures the database server to use a different version of GSKit, the version-specific **gsk8capicmd** command must be replaced with command from the different GSKit version:
    - For example, **gsk7capicmd**.

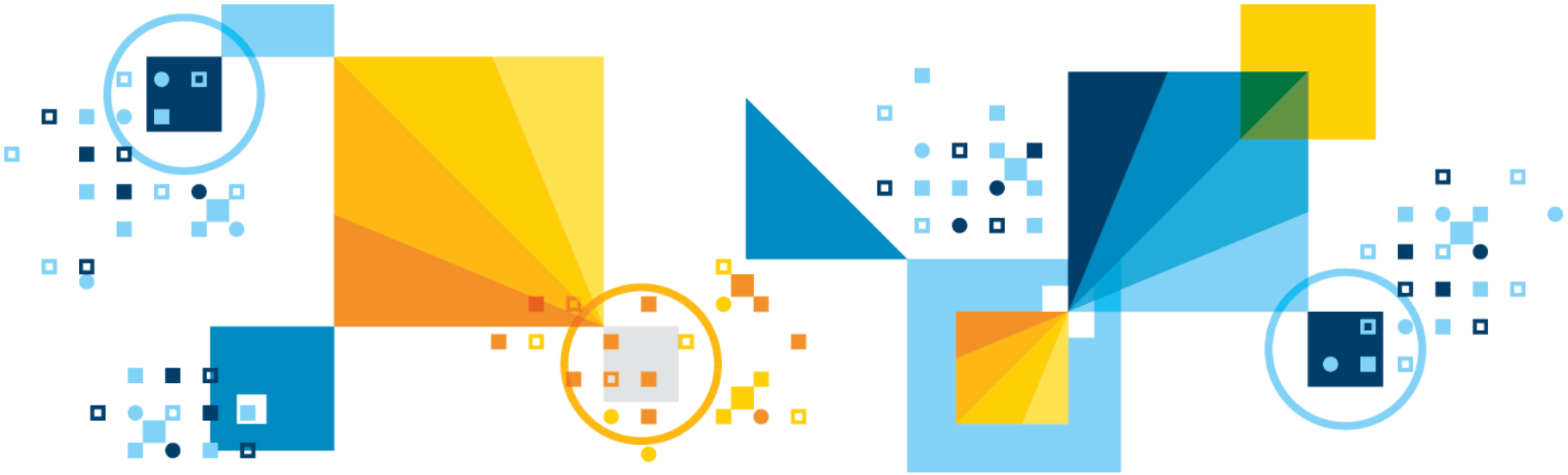
- **Appendix C has a complete discussion about setting up GSKit if not previously setup and implemented.**

Questions

Scott Pickett – WW Informix Technical Sales

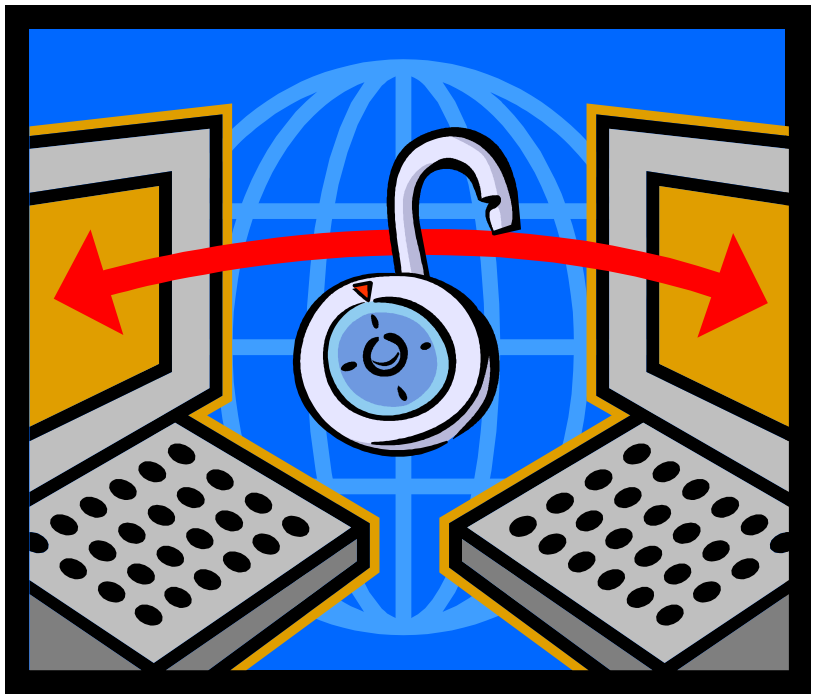
June 27, 2017

HDR & ER Server to Server Encrypted Communications



HDR & ER Encryption Overview

- **This feature provides encryption of data traffic between HDR servers and also ER servers in a secure way.**
- **The implementation uses the following facilities:**
  - IBM Crypto for C as base:
    - FIPS 140-2 certified technology
  - Same ciphers.
- **SSL alternative**



HDR & ER Encryption Description

- **HDR & ER encryption is configured in $ONCONFIG.**
- **Encryption parameters are read only at instance start:**

Instance(s) must be restarted to recognize any changes.

- **ENCRYPT_HDR turns HDR encryption on or off:**
  - Default value: **0** (zero) or off
  - Other value: **1** (one) or on

- **$MSGPATH displays the current value of ENCRYPT_HDR.**

HDR & ER Encryption $ONCONFIG Setup

- **Parameters need to be the same on all servers****

Parameter	Description
ENCRYPT_CIPHERS	**Which ciphers are to be used, can be all, all but or a comma delimited list of specific listing of ciphers and their mode.** **Supported ciphers include:** DES (64 bit), Triple DES, Extended DES, AES (128 &192 bit), Blow Fish (64, 128 & 192 bit). **Modes:** Electronic Code Book, Cipher Block Chaining, Cipher Feedback or Output Feedback.
ENCRYPT_MACFILE	**A list of one or more pathnames to MAC key files. Can also use the built-in files which utilize MAC keys shipped with IDS.**

- ** **ENCRYPT_MACFILE can vary, is physical server dependent, keys they contain must be identical though**.

HDR & ER Encryption $ONCONFIG Setup (cont'd)

Parameter	Description
ENCRYPT_MAC	**Controls the level of message authentication coding that occurs. Possible values:** **Off.** **low** – uses XOR folding **medium** – uses the SHA1 bitwise rotation for messages > 20 bytes and XOR for < 20 bytes. **high** – uses SHA1 for all messages.
ENCRYPT_SWITCH	**A two value, comma separated parameter:** First value (**cipher_switch_time**) -- cipher renegotiation interval Second value (**key_switch_time**) -- secret key renegotiation interval **Both values are measured in minutes.**

HDR & ER Encryption Configuration Setup

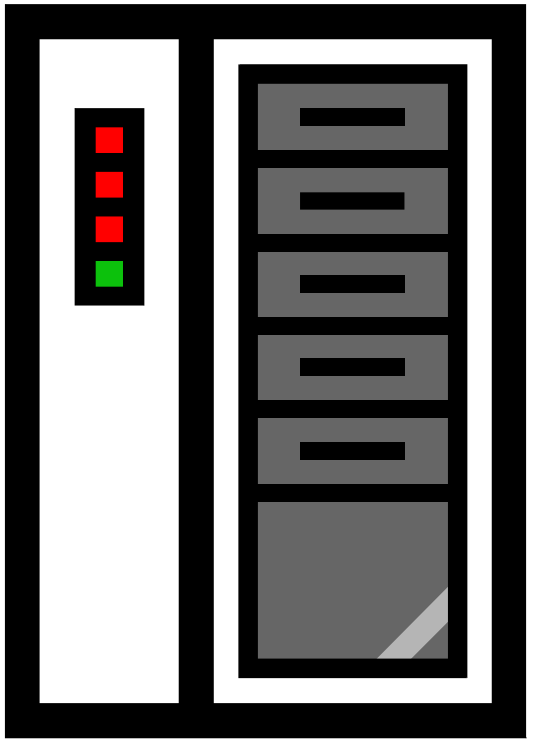
- **No special $INFORMIXSQLHOSTS setup required for HDR & ER encryption:**
  - Uses standard network configuration entries.

- **If applications are using encryption CSM, you will need to define a separate:**
  - **DBSERVERALIAS** and
  - **$INFORMIXSQLHOSTS** and
  - **/etc/services** entry for that connection.

Example ONCONFIG for HDR Encryption

On Primary

ENCRYPT_HDR	1
ENCRYPT_CIPHERS	all
ENCRYPT_MAC	high
ENCRYPT_MACFILE	/usr/local/bin/mac1.dat
ENCRYPT_SWITCH	60,60



On Secondary

ENCRYPT_HDR	1
ENCRYPT_CIPHERS	all
ENCRYPT_MAC	high
ENCRYPT_MACFILE	/usr/local/bin/mac2.dat
ENCRYPT_SWITCH	60,60

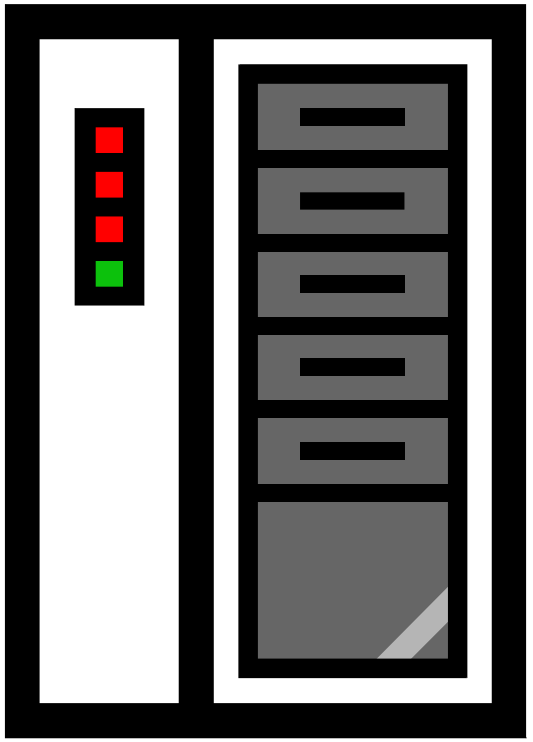
Notes

- **When working in conjunction with each other, that is, live and active at the same time and visible to each other, HDR and Enterprise Replication can share the same ENCRYPT_CIPHER, ENCRYPT_MAC, ENCRYPT_MACFILE and ENCRYPT_SWITCH configuration parameters**
  - The **ENCRYPT_HDR** setting is unique to the HDR
  - The **ENCRYPT_CDR** setting is used for ER
    - Allowed values are:
      - **0** = Default. Do not encrypt.
      - **1** = Encrypt when possible.
        - Encryption is used for Enterprise Replication transactions only when the database server being connected to also supports encryption.
      - **2** = Always encrypt.
        - Only connections to encrypted database servers are allowed.

Example ONCONFIG for ER Encryption

On Primary

ENCRYPT_CDR	1
ENCRYPT_CIPHERS	all
ENCRYPT_MAC	high
ENCRYPT_MACFILE	/usr/local/bin/mac1.dat
ENCRYPT_SWITCH	60,60



On Secondary

ENCRYPT_CDR	1
ENCRYPT_CIPHERS	all
ENCRYPT_MAC	high
ENCRYPT_MACFILE	/usr/local/bin/mac2.dat
ENCRYPT_SWITCH	60,60

Notes

- **For updatable secondary servers in a high-availability cluster environment, encryption from the updatable secondary server to primary server requires Server Multiplexer Group (SMX) encryption.**
  - SMX is a communications interface supporting encrypted multiplexed network connections between database server instances in high availability environments via a reliable, secure, high-performance communication mechanism.

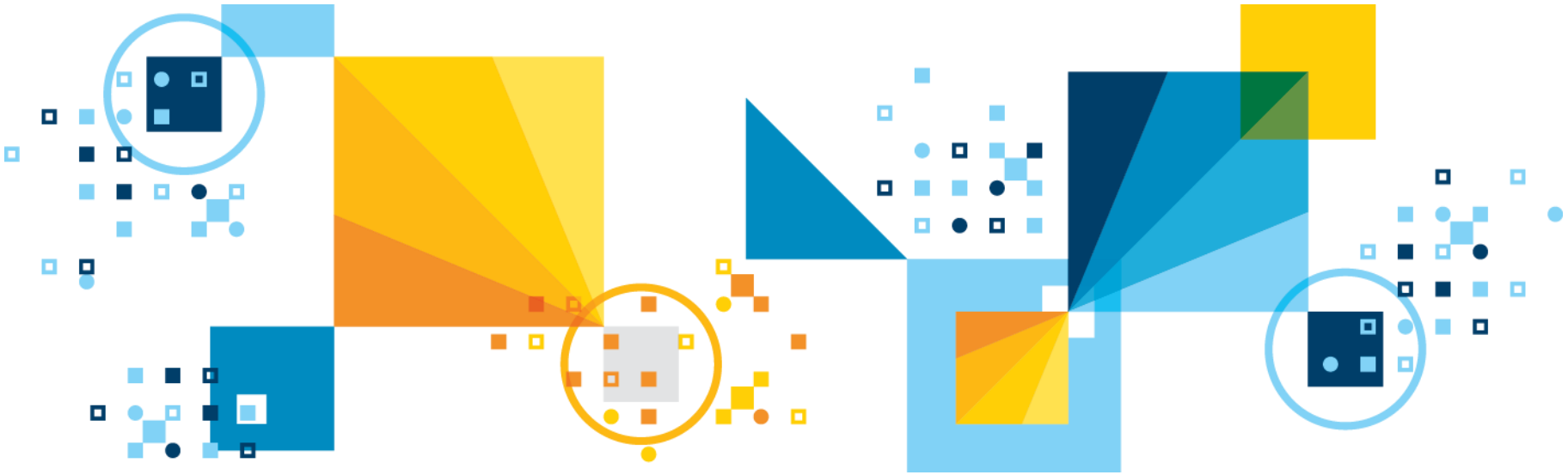
- **To encrypt data sent from an updatable secondary server to the primary server, set the ENCRYPT_SMX configuration parameter on the secondary server.**
  - Valid values for **ENCRYPT_SMX**
    - **0**
      - Disables encryption between servers
    - **1**
      - Encryption is used for SMX transactions only when the database server being connected to also supports encryption.
    - **2**
      - Only connections to encrypted database servers are allowed.

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Encrypted Communications ENCCSM



Data Transmission Encryption End to End

- **The communication support modules (CSM's) are used to encrypt data transmissions, including distributed queries, over the network.**
- **The encryption CSM (ENCCSM) provides network transmission encryption.**
- **This option provides complete data encryption with a standard cryptography library, with many configurable options:**
  - A message authentication code (MAC) is transmitted as part of the encrypted data transmission to ensure data integrity.
  - A MAC is an encrypted message digest.

Data Transmission Encryption End to End

- **CSMs have the following restrictions:**
  - Cannot use an encryption CSM and a simple password CSM simultaneously
    - If you are using the simple password CSM (**SPWDCSM**), and decide to encrypt your network data, you must remove the entries for the **SPWDCSM** in your **concsm.cfg** and **sqlhosts** files.
  - Cannot use either simple password CSM or encryption CSM over a multiplexed connection.

- **Enterprise Replication and high-availability clusters (High-Availability Data Replication, remote stand-alone secondary servers, and shared disk secondary servers) support encryption, but cannot use a connection configured with a CSM.**

- **Encrypted connections and unencrypted connections cannot be combined on the same port.**

Data Transmission Encryption End to End – SSL Alternative

- **Secure Sockets Layer (SSL) communications, which encrypt data in end-to-end, secure TCP/IP and Distributed Relational Database Architecture (DRDA) connections between two points over a network, are an alternative to the Informix specific encryption CSM's.**

Enabling Encryption with Communication Support Modules

- **Modify the concsm.cfg file to use encryption with communication support modules.**
  - Verify that the module can use a port that is not shared with an unencrypted connection before you enable network encryption.

- **To enable network encryption:**
  - Add a line to the **concsm.cfg** file.
    - The **concsm.cfg** file must contain an entry for each communications support module (of the same kind) that you are using.
  - Add an entry to the **options** column of the **sqlhosts** file information.

concsm.cfg file

- An entry is a single line and is limited to 1024 bytes.
- File is default located in the etc directory of $INFORMIXDIR. If you want to store the file somewhere else, you can override the default location by setting the INFORMIXCONCSMCFG environment variable to the full path name of the new location.

- Entries in the concsm.cfg file must conform to the following restrictions:
- The following characters are not allowed to be part of library path names:
  - = (equal sign)
  - " (double quotation mark)
  - , (comma)
- White spaces cannot be used unless the white spaces are part of a path name.

Encryption Ciphers and Modes

- Specify which ciphers and mode to use during encryption.
- The cipher and mode used is randomly selected among the ciphers that are common between the two servers; all servers and client computers that participate in encrypted communication should have common ciphers and modes.
- Encryption is more secure if you include more ciphers and modes that the database server can switch between.
- The Data Encryption Standard (DES) is a cryptographic algorithm designed to encrypt and decrypt data by using 8-byte blocks and a 64-bit key.

Encryption Ciphers and Modes

- **The Triple DES (DES3) is a variation of DES in which three 64-bit keys are used for a 192-bit key. DES3 works by first encrypting the plain text by using the first 64-bits of the key. Then the cipher text is decrypted by using the next part of the key. In the final step, the resulting cipher text is re-encrypted by using the last part of the key**

- **Advanced Encryption Standard (AES) is a replacement algorithm used by the United States government**

- **Two encryption modes are:**
  - *Block Mode*, a method of encryption in which the message is broken into blocks and the encryption occurs on each block as a unit
    - Since each block is at least 8 bytes large, block mode provides the ability for 64-bit arithmetic in the encryption algorithm.
  - *Stream Mode*, a method of encryption in which each individual byte is encrypted. It is generally considered to be a weak form of encryption

Encryption Ciphers and Modes

- **Blowfish is a block cipher that operates on 64-bit (8-byte) blocks of data. It uses a variable size key, but typically, 128-bit (16-byte) keys are considered to be good for strong encryption. Blowfish can be used in the same modes as DES:**
  - Do not specify individual ciphers.
  - For security reasons, all ciphers must be allowed.
  - If a cipher is discovered to have a weakness, you can exclude it.

- **Use the allbut option to list ciphers and modes to exclude. Enclose the allbut list in angled brackets (<>). The list can include unique, abbreviated entries.**
  - For example, **bf** can represent **bf1**, **bf2**, and **bf3**.
    - However, if the abbreviation is the name of an actual cipher, then only that cipher is eliminated.
    - Therefore, **des** eliminates only the DES cipher, but **de** eliminates **des**, **ede**, and **desx**.

Cipher Support

- **The following des, ede, and desx ciphers are supported:**

Cipher	Explanation	Blowfish Cipher	Explanation
des	DES (64-bit key)	bf1	Blowfish (64-bit key)
ede	Triple DES	bf2	Blowfish (128-bit key)
desx	Extended DES (128-bit key)	bf3	Blowfish (192-bit key)

- **The following aes ciphers are supported:**

Cipher	Explanation
aes	AES (128-bit key)
aes128	AES (128-bit key)
aes192	AES (192-bit key)
aes256	AES (256-bit key)

Cipher Modes Support

- **The following modes are supported**

Mode	Explanation
ecb	Electronic Code Book
cbc	Cipher Block Chaining
cfb	Cipher Feedback
ofb	Output Feedback

MAC Key Files

- **MAC key files contain encryption keys used to encrypt messages.**
- **The database servers and client computers that participate in encryption normally require the same MAC key file.**

- **The default MAC key file is the built-in file that provides limited message verification (some validation of the received message and determination that it has come from the IBM Informix client or server):**
  - A site-generated MAC key file performs the strongest verification.
  - Generate key files with the **GenMacKey** utility.
- **Each MAC key file is prioritized and negotiated at connect time. The prioritization for the MAC key files is based on their creation time by the GenMacKey utility. The built-in key file has the lowest priority.**

- **If no MAC key files is present, the built-in MAC key is used by default. However, by using a MAC key file, the default built-in MAC key is disabled.**

Generate a New MAC Key File

- **Can improve the reliability of message verification using encryption.**
- **To generate a new MAC key file:**
  - Run the following command from the command line:

    GenMacKey –o *filename*
  - The ***filename*** is the path and file name of the new MAC key file.

- **Update the central server's configuration to include the location of the new MAC key file in one of the following ways:**
  - **Using encryption tags:**
    - Edit the relevant line in the **concsm.cfg** file to add a path and file name to the mac tag.
  - **Using encryption parameters:**
    - Edit the encryption parameters file to alter the value of the **ENCCSM_MACFILES** parameter.
- **If necessary, remove old MAC key file entries from the configuration.**
- **Distribute the new MAC key file among all appropriate computers.**

MAC Levels

- **MAC levels determine the type of MAC key generation.**
- **The supported generation levels are:**
  - **high.**
    - Uses **SHA1 MAC** generation on all messages.
  - **medium.**
    - Uses **SHA1 MAC** generation for all messages greater than **20** bytes long and **XOR** folding on smaller messages.
  - **low.**
    - Uses **XOR** folding on all messages.
  - **off.**
    - Does not use **MAC** generation.

- **The level is prioritized to the highest value.**
- **The off entry must only be used between servers when it is guaranteed that there is a secure network connection.**

MAC Levels

- All servers and client computers that transmit encrypted communication must have at least one MAC level setting in common.
- For example, if one database server has a level of high and medium enabled and the other database server has only low enabled, then the connection attempt fails.
- If a database server has high and medium settings and the other database server has only the medium setting, the MAC generation levels support a connection.

Examples of Using Encryption Tags

- **The following configuration string states to use all available ciphers except for any of the Blowfish ciphers, and to not use any cipher in ECB mode:**
  - **ENCCSM("$INFORMIXDIR/lib/csm/iencs11a.so", "cipher[allbut:<ecb,bf>]")**

- **The following configuration string states:**
  - Use the **DES/CBC**-mode, **EDE/OFB**-mode, and **DESX/CBC**-mode ciphers for this connection.
  - Use either **SHA1 MAC** generation or **XOR** folding on all messages.
  - Use **mac1.dat, mac2.dat**, or the **builtin MAC** key file for encrypting messages.
  - Switch the cipher being used every **120** minutes and renegotiate the secret key every **15** minutes.
  - **ENCCSM("/$INFORMIXDIR/lib/csm/iencs11a.so", "cipher[des:cbc,ede:ofb,desx:cbc], mac[levels:<high,low>,files:</usr/local/bin/mac1.dat, /usr/local/bin/mac2.dat,builtin>], switch[cipher:120,key:15]")**

Switch Frequency

- The switch frequency defines when ciphers and or secret keys are renegotiated.
- The default time that this renegotiation occurs is once an hour. By using switch options, you can set the time in minutes when the renegotiation occurs.
- The longer that the secret key and encryption cipher remain in use, the more likely that the encryption rules might be broken by an attacker. To avoid this, cryptologists recommend periodically changing the secret key and cipher on long-term connections.

Network Data Encryption Syntax

- **Specify network encryption libraries and options in the concsm.cfg file.**

- **Specify the following types of encryption options:**
  - **DES** and **AES** ciphers to use during encryption
  - Modes to use during encryption
  - Message authentication code (**MAC**) key files
  - **MAC** levels
  - Switch frequency for ciphers and keys

Specifying Network Encryption Options in concsm.cfg

- **Modify encryption communication support module (CSM) options by specifying libraries and encryption tags.**

- **Informix provides the following shared libraries for use as CSM's. The paths and fixed file names are:**
  - Unix and Linux
    - **$INFORMIXDIR/lib/client/csm/iencs11a.so**
  - Windows
    - **%INFORMIXDIR%\bin\client\iencs11a.dll**
- **The shared libraries also have version-specific names that can be used in place of the fixed names. If you use the version-specific name, and the server is updated, you must update the concsm.cfg file.**
- **Specifying encryption options directly in the concsm.cfg file is usually more difficult than specifying libraries and tags in an encryption parameters file because of syntax specifications. A sample file concsm.example is available in $INFORMIXDIR/etc (UNIX and Linux) and %INFORMIXDIR%\etc (Windows).**

$INFORMIXDIR/etc/concsm.example – Ships With Product

```
#**************************************************************************
# Specifying the ENCCSM module with CSM option in SQLHOSTS FILE
# dbservername  nettype hostname service csm=(ENCCSM)
# NB: You must replace <$INFORMIXDIR> with the actual value of $INFORMIXDIR.
# FR: Allow $INFORMIXDIR in the macro values.
# General encryption parametersENCCSM("<$INFORMIXDIR>/lib/csm/iencs11a.so", "cipher[all],
mac[levels:<high>,files:<builtin>], switch[cipher:1440,key:60]")
# Specific encryption parameters (aes:cbc only) ENCCSM("<$INFORMIXDIR>/lib/csm/iencs11a.so",
"cipher[aes:cbc], mac[levels:<high>,files:<builtin>], switch[cipher:1440,key:60]")
# Simple Password Communication SPWDCSM("<$INFORMIXDIR>/lib/csm/libixspw.so", "", "")
# Simple Password Communication (client and server libraries)
SPWDCSM("client=<$INFORMIXDIR>/lib/client/csm/libixspw.so,
server=<$INFORMIXDIR>/lib/csm/libixspw.so", "", "")
# Encryption parameters specified in a file named encrypt.txt ENCCSM("/usr/informix/lib/cms/iencs11a.so",
"config=<$INFORMIXDIR>/etc/encrypt.txt")
# The following example shows an encryption parameter file:
#ENCCSM_CIPHERS       all
#ENCCSM_SWITCH        1440,60
#ENCCSM_MAC           medium
#ENCCSM_MACFILE       <$INFORMIXDIR>/etc/MacKey.dat
```

Example of a Encryption Parameter File

- Specifies values for encryption parameters.

- The following example shows an encryption parameter file:

```
ENCCSM_CIPHERS          all
ENCCSM_SWITCH           120,60
ENCCSM_MAC              medium
ENCCSM_MACFILES         /usr/informix/etc/MacKey.dat
```

- The following example illustrates a line in the concsm.cfg file to specify encryption with a parameter file named encrypt.txt:

```
ENCCSM("usr/informix/lib/cms/iencs11a.so",
config=/usr/lib/encrypt.txt")
```

ENCCSM_MACFILES parameter

- **Specifies the MAC key files to use.**

- **Default Value**
  - **builtin**
- **Units**
  - Path names, up to **1536** bytes in length
- **Range of Values**
  - One or more full path and file names separated by commas, and the optional **builtin** keyword.

- **For Example:**

  ENCCSM_MACFILES /usr/local/bin/mac1.dat,/usr/local/bin/mac2.dat,builtin

ENCCSM_MAC parameter

- **Specifies the MAC level to use.**
- **Default Value**
  - **medium**
- **Range Of Values**
  - One or more of the following options, separated by commas:
    - **off**
      - Does not use MAC generation.
    - **low**
      - Uses XOR folding on all messages.
    - **medium**
      - Uses SHA1 MAC generation for all messages greater than 20 bytes long and XOR folding on smaller messages.
    - **high**
      - Uses SHA1 MAC generation on all messages.

- **For example:**
  - **ENCCSM_MAC medium,high**

ENCCSM_CIPHERS parameter

- **Specifies the ciphers and modes to use during encryption.**

  ENCCSM_CIPHERS
  - **all**|**allbut**:<list of ciphers and modes>|**cipher:mode**{,cipher:mode ...}
- **all:**
  - Specifies to include all available ciphers and modes, except ECB mode.
  - For example:
    - **ENCCSM_CIPHERS all**
- **allbut:<list of ciphers and modes>:**
  - Default with **ecb**. Specifies to include all ciphers and modes except the ones in the list. Separate ciphers or modes with a comma.
  - For example:
    - **ENCCSM_CIPHERS allbut**:<cbc,bf>
- **cipher:mode:**
  - Specifies the ciphers and modes. Separate cipher-mode pairs with a comma.
  - For example:
    - **ENCCSM_CIPHERS des3:cbc,des3:ofb**

ENCCSM_SWITCH parameter

- **Defines the number of minutes between cipher and key renegotiation.**

- **Syntax ENCCSM_SWITCH cipher_switch_time,key_switch_time**

- **cipher_switch_time**
  - Specifies the minutes between cipher renegotiation
- **key_switch_time**
  - Specifies the minutes between secret key renegotiation
- **Default Value**
  - 60,60
- **Units**
  - minutes
- **Range Of Values**
  - positive integers

concsm.cfg syntax (1 of 5)

```
concsm.cfg entry Syntax

>>-name--(--“--+-client--=--client_library--,--server--=--server_library-+--“--,-->
               '-csm_library-----------------------------------------------'

>--“--+---------------------------------+--“--)-----------------><
      +-config--=--parameter_file------+
      | .-,--------------------------. |
      | V  (1)                       | |
      '-------+-| Cipher options |-+-+-'
              +-| MAC options |----+
              '-| Switch options |-'

Cipher options

              .-all-------------------------.
              |               .-,------.    |
              |               V        |    |
|--cipher--[--+-allbut--:--<----cipher-+-->-+--]----------------|
              | .-,---------------.          |
              | V                 |          |
              '---cipher--:--mode-+----------'
```

concsm.cfg syntax (2 of 5)

```
MAC options

                            .-,----------.
                       (1)  V .-medium-. |
|--mac--[--levels--:--<--------+-high---+-+-->------------------>
                               +-low----+
                               '-off----'

                .-builtin-----------------------.
                | .-,---------.                 |
                | V           |                 |
>--files--:--<--+---file_name-+--,--+---------+-+-->--]---------|
                                    '-builtin-'

Switch options

              .-,-----------------------------------------.
              V                                           |
|--switch--[----+-------------------------------------+-+--]--|
                |  (1)                                |
                +-------cipher--:--negotiation_interval-+
                '-key--:--negotiation_interval----------'
```

concsm.cfg Syntax (3 of 5)

Option	Description
all	Include all available ciphers and all available modes, except ECB mode.
allbut	Include all ciphers except the ones listed.
builtin	The default MAC key file provided by IBM Informix. The builtin file provides limited message verification that received messages have come from the IBM Informix client or server).
cipher	Include the specified cipher.
client_library	The path and name of the shared library that is the CSM on the client computer.
csm_library	The path and name of the shared library that is the CSM if the CSM is shared by both the database server and the client computers.
files	The comma-separated list of the full path names of MAC key files.
key	Message authentication code (MAC) keys used for message encryption.
key_file	The path and file name of the MAC key files.

concsm.cfg Syntax (4 of 5)

Option	Description
levels	Comma-separated list of MAC generation levels that the connection supports. **high** Use SHA1 MAC generation on all messages. **medium** Use SHA1 MAC generation for all messages greater than 20 bytes long and XOR folding on smaller messages. **low** Use XOR folding on all messages. **off** Do not use MAC generation.
mode	Use the specified cipher mode. **ecb** Electronic Code Book **cbc** Cipher Block Chaining **cfb** Cipher Feedback **ofb** Output Feedback

concsm.cfg Syntax (5 of 5)

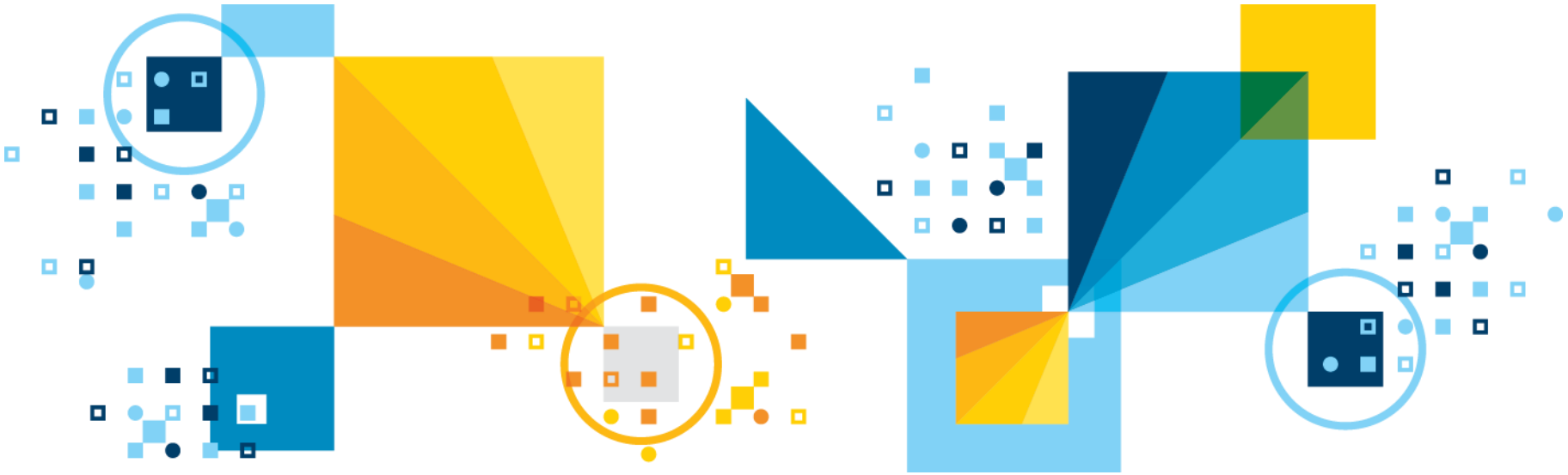
name	The name that you assign to the CSM.
negotiation_interval	The minutes between renegotiations.
parameter_file	The path and file name of the file in which the encryption parameters are defined.ImportantIf the file does not exist at the specified path, then default parameter values are used. No error is returned.
server_library	The full path and name of the shared library that is the CSM on the database server.

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Encrypted Columns & Cells



Agenda

- **Caveat**
- **Encrypted Column**

Caveat

- **If you decide to do encrypted columns, be aware that this is not compatible with encryption at rest of an entire instance.**

- **The encrypted columns feature is carried inside the app and in memory.**
  - Do not index an encrypted column
    - Plain text is stored on disk in the index.

- **The Encryption at Rest feature is carried out on-disk.**

Column-level Encryption

- Use to store sensitive data in an encrypted format.
- After encrypting sensitive data, such as credit card numbers, only users who can provide a secret password can decrypt the data.
- Use the built-in ENCRYPT_AES() and ENCRYPT_TDES() encryption functions to encrypt data in columns containing the following character data types or smart large object data types:
  - CHAR
  - NCHAR
  - VARCHAR
  - NVARCHAR
  - LVARCHAR
  - BLOB
  - CLOB

Column-level Encryption

- **You can also use the SET ENCRYPTION PASSWORD statement to set an encryption password for a session:**
  - Only users providing a secret password can view, copy, or modify encrypted data

- **The built-in ENCRYPT_AES(), ENCRYPT_TDES(), DECRYPT_CHAR(), and DECRYPT_BINARY() encryption and decryption functions can use the session-level password if the password is not explicitly specified in the encryption or decryption function:**
  - If **SET ENCRYPTION PASSWORD** is used, there is no requirement to provide the same password in every encryption or decryption function.

Column-level Encryption Explained

- **After Informix prepares a statement that contains a password (and, optionally, a hint), Informix keeps the password and hint in shared memory in an encrypted format**
- **Informix only decrypts a copy of the password or hint when any statement related to encryption is being executed**
- **Informix uses a randomly generated session key to encrypt the password and hint in memory. This means that if the server fails with an AF (assertion failure) error, or if the shared memory is paged out of main memory, it is hard to find plain text passwords in the core dump**
- **Informix never writes a password to disk**
  - It records hints with the encrypted data, lightly encrypted, not readily understood

Password Encryption Types for Column Data

- **Column-level encryption:**
  - All values in a specific column of a database table are encrypted with the same password (word or phrase), the same encryption algorithm and cipher mode
  - For column-level encryption, you can store the hint outside the encrypted column, rather than repeating it in every row
    - If encryption functions are not used, users can enter unencrypted data into columns that are meant to contain encrypted data
    - To ensure that data entered into a field is always encrypted, use views and **INSTEAD OF** triggers

- **Cell-level encryption (also called *row-column* or *set-column level encryption*):**
  - Within a column of encrypted data, many different passwords, encryption algorithms, or modes are used
  - This type of encryption might be necessary to protect personal data

Passwords and Hints

- **Passwords and hints declared with SET ENCRYPTION PASSWORD are not stored as plain text in any of the system catalog tables.**

- **To prevent other users from accessing the plain text of encrypted data or of a password, you must avoid actions that might compromise the secrecy of a password:**
  - Unless your database is accessible only by a secure network, you must enable the Encryption Communication Support Module (**ENCCSM**) to protect data transmission between the database server and any client system.
  - Do not index encrypted columns and do not create a functional index on a decrypted column:
    - This would store plain-text data in the database, defeating the purpose of encryption.
  - Do not store passwords in a trigger or in a user-defined routine (**UDR**) that exposes the password to the public:
    - Use the session password before you activate the trigger, invoke the **UDR**, or pass any password as a parameter to a **UDR**.

Passwords and Hints

- **When you set a password, the database server transfers it and any hint to a 128-bit key used to encrypt the password and hint**
- **Passwords and hints are not stored as clear text**
- **The key is a time-based random value per instance**
  - The database server starts the key when the server starts and the key is deleted when the database server shuts down

- **Although it is possible to store both encrypted and unencrypted data in a single column, your application must determine which rows contain encrypted data and which rows contain unencrypted data. In addition, the application must provide for using the correct code to handle the difference, because the built-in decryption functions fail if they are applied to unencrypted data:**
  - Simplest way to avoid this error is for all rows to use encryption in a column where any row is encrypted.

Caveats

- **A query for encrypted data must specify an unencrypted column on which to select the rows**
- **An encrypted value uses more storage space in a column than the corresponding plain text value:**
  - All of the information required to decrypt the value, except the encryption key, is stored with the value
  - Therefore, embedding zero bytes in the encrypted result is not recommended

Encrypt Virtual Processor Configuration

- The database server includes an Encrypt Virtual Processor.
- If the VPCLASS parameter encrypt option is not defined in the onconfig file, the database server starts one Encrypt VP the first time that any encryption or decryption functions defined for column-level encryption are called.
- Define multiple Encrypt VPs if necessary to decrease the time required to start the database server.
- When the database server is in online mode, you can use the onmode -p command to add or drop Encrypt VPs.
- For example, to add four more Encrypt VPs, use:
  - onmode -p 4 encrypt
- To drop three Encrypt VPs, use:
  - onmode -p -3 encrypt

Encrypted Columns Sizing

- **The column size must be large enough to store the encrypted data.**
  - The speaker notes show how the size of a Credit Card column is calculated
    - In the speaker notes example, Initialization Vector (IV) is a pseudo-random series of bytes that is used to initiate encryption when using some cipher modes.
    - IV size is the number of random series of bytes; for Informix, this is 8 bytes.

- **If the hint is not stored in the column, the total size in the previous example is 55 bytes.**

- **Alternately, calculate as follows:**
  - **SELECT LENGTH(ENCRYPT_TDES ("1234567890123456", "password", "long….hint")) FROM "informix".systables WHERE tabid = 1;**

- **Without the hint, you can calculate as follows:**
  - **SELECT LENGTH(ENCRYPT_TDES("1234567890123456", "password", "")) FROM "informix".systables WHERE tabid = 1;**

Encrypted Columns Sizing

- **If the column size is smaller than the returned data size from ENCRYPT and DECRYPT functions, the encrypted data is truncated when it is inserted and it is not possible to decrypt the data**
  - Because the header indicates that the length must be longer than the data received

Code to Encrypt a Column - Simple

- **Use the SET ENCRYPTION PASSWORD statement to restrict access to data in a column:**

```
create table emp
(   name char(40),
    salary money,
    ssn lvarchar(67) ); # normally this would be char(11)

set encryption password "one two three 123";
insert into emp values (“Scott", 50000, encrypt_aes ('123-456-7890'));
insert into emp values ("Bob", 65000, encrypt_aes ('213-656-0890'));
select name, salary, decrypt_char(ssn, "one two three 123")
  from emp where name = 'Bob';
```

Querying Data & Encrypted Columns

- Query encrypted data with the DECRYPT function or the SET ENCRYPTION PASSWORD statement.

- The following example shows how to use the decrypt function to query encrypted data:

```
select name, decrypt_char(ssn, "one two three 123") from emp;
```

or

```
set encryption password "one two three 123";
select name, salary, decrypt_char(ssn) from emp where name = 'Bob';
```

What is Column-Level Encryption?

- **Assists in legislative compliance:**
  - HIPAA (Health Insurance Portability and Accountability Act)
  - Sarbanes-Oxley (aka Sarbox or Sox)
  - Basel II
  - Gramm-Leach-Bliley Act (GLBA)
  - California SB 1386 ‘Personal Information: Privacy’
  - EU GDPR (May 23, 2018)

- **Latest cryptographic standards (OpenSSL 1.0.2)**

- **128-bit AES and 112-bit Triple-DES**

Manual Pages – Overview

- Passwords and hints
- ENCRYPT_TDES function
- ENCRYPT_AES function
- DECRYPT_CHAR function
- DECRYPT_BINARY function
- GETHINT function
- SET ENCRYPTION PASSWORD statement
- Data Space Requirements

Manual Pages – Passwords, Hints

- **Passwords are necessary.**
  - Pass-phrase can be from **6** bytes up to **128** bytes.
  - Cryptographic hash is used to make random-seeming key.
  - Random initialization vector also used (and recorded).

- **Hints are optional.**
  - Hints can be up to **32** bytes of text.

- **Passwords (and hints) can be set for a session.**
  - **SET ENCRYPTION PASSWORD** statement.
  - Hence, the password and hint parameters are optional.
  - Explicit values override the session default values.

Manual Pages – ENCRYPT_TDES

- **ENCRYPT_TDES(data [, password [, hint]])**
  - Input data is encrypted.
  - For character input data:
    - Output is base-64 encoded.
  - For binary input data (BLOB):
    - Output is unencoded.

- **Triple-DES encryption**
  - Triple DES uses two 56-bit keys for 112-bits overall.

- ```
  UPDATE SomeTable
      SET EncryptedColumn = ENCRYPT_TDES(?, ?)
      WHERE PK_Column = ?;
  ```

- **Extremely variant function.**
  ```

Manual pages - ENCRYPT_TDES

- The ENCRYPT_TDES function returns a value that is the result of encrypting a character expression, or a BLOB or CLOB value, by applying the TDES (Triple Data Encryption Standard, which is sometimes also called DES3) algorithm to its first argument.
- This algorithm is slower than the AES algorithm that is used by the ENCRYPT_AES function, but is considered somewhat more secure.
- The disk space required as encryption overhead resembles that of ENCRYPT_AES, but is somewhat smaller because of the smaller block size of ENCRYPT_TDES.
- For BLOB or CLOB values, the encrypted object is temporarily stored in the default sbspace that the SBSPACENAME configuration parameter specifies.

Manual pages - ENCRYPT_TDES

- **These differences in performance, tamper-resistance, and in the returned *encrypted_data* size are the practical differences between the ENCRYPT_TDES and ENCRYPT_AES functions, which otherwise follow the same rules, defaults, and restrictions that appear in the description of ENCRYPT_AES on the previous page in regard to the following features:**
  - The required first argument (the plain text *data* value to be encrypted)
  - The explicit or default second argument (the *password* string that must also be an argument to DECRYPT_CHAR or DECRYPT_BINARY to decrypt the returned *encrypted_data* value)
    - This must be specified unless a default *session password* has been set by the SET ENCRYPTION statement
- **The optional third argument (the *hint* value) that might assist users who forget the *password*. If you subsequently cannot remember an explicit or default *hint* that was defined for *password*, you can use the returned value from ENCRYPT_TDES as the argument to GETHINT to retrieve the *hint*.**

Manual pages - ENCRYPT_TDES

- **The following calls ENCRYPT_TDES from the SET clause of an UPDATE statement; here the *session password* is 'PERSEPHONE' and the *hint* string is "pomegranate", with column colU of table tabU the *data* argument. Because the WHERE clause condition of "1=1" is true for all rows of tabU, the effect of this statement is to replace every plain text colU value with encrypted strings returned by the algorithm that ENCRYPT_TDES implements:**

  - **EXEC SQL SET ENCRYPTION PASSWORD 'PERSEPHONE' WITH HINT 'pomegranate';**
  - **EXEC SQL UPDATE tabU SET colU = ENCRYPT_TDES (colU) WHERE 1=1;**
    - Above example assumes that the character data type of colU is of sufficient size to store the new encrypted values without truncation
      (A more cautious example might execute an appropriate ALTER TABLE statement before the UPDATE.)

Manual Pages – ENCRYPT_AES

- **ENCRYPT_AES(data [, password [, hint]])**
  - Input data is encrypted.
  - For character input data:
    - Output is base-64 encoded.
  - For binary input data (BLOB):
    - Output is un-encoded.

- **AES encryption**
  - Advanced Encryption System (aka Rijndael).
    - 128-bit key size.

- ```
  INSERT INTO SomeTable
      VALUES (?, ENCRYPT_AES(?, ?))
  ```

- **Extremely variant function.**
  ```

Manual Pages – ENCRYPT_AES function

- **ENCRYPT_AES returns an encrypted value that it derives by applying the AES (Advanced Encryption Standard) algorithm to its first argument, which must be an unencrypted character expression or a smart large object (that is, a BLOB or CLOB data type)**
  - A character argument max length <= **32640** bytes if an explicit or default *hint* is used, or **32672** bytes if no hint (or a NULL hint) is specified.
  - Theoretical size limits on **BLOB** or **CLOB** arguments are many orders of magnitude larger, but practical limits might be imposed by your hardware, or by time required for encryption and decryption.
- **Encrypted BLOB or CLOB object is temporarily stored in the default sbspace that the SBSPACENAME configuration parameter specifies.**
- **Specify a *password* as its second argument, unless a SET ENCRYPTION statement has specified a *session password*, which the database server uses by default if you omit the second argument.**

Manual Pages – ENCRYPT_AES

- **A *password* must be specified as its second argument, unless a SET ENCRYPTION statement has specified a *session password*; the database server uses it by default if the second argument is omitted:**
  - If a *session password* has been set, any user-specified *password* overrides the *session password* for the function call returned value.
  - The explicit or default *password* will also be required for any subsequent decryption of the returned encrypted value.
  - A valid *password* must have at least **6** bytes but no more than **128**.
- **Optionally, a hint, to help remember the password, can be specified as the third argument. If the SET ENCRYPTION statement specified a default *hint* for this session, and you specify no hint, that default *hint* is stored in an encrypted form within the returned value.**
  - Any *hint* that you specify overrides the default *hint*.
  - A valid *hint* can be no longer than **32** bytes.
  - You can use consecutive quotation marks (") to specify a **NULL** *hint*.
  - If you specify an explicit *hint*, you must also specify an explicit *password*.

Manual Pages – ENCRYPT_AES

- If a hint is not remembered, the returned value from ENCRYPT_AES can be used as the argument to GETHINT to retrieve the *hint*.
- The following example calls ENCRYPT_AES from the VALUES clause of an INSERT statement that stores in tab1 a plain-text string and an *encrypted_data* value that ENCRYPT_AES returns from its 12-byte first argument
  - Here SET ENCRYPTION defines a *session password* and *hint* that are used as default second and third arguments to the ENCRYPT_AES function:
  - EXEC SQL SET ENCRYPTION PASSWORD 'CHARYBDIS' WITH HINT 'messina';
  - EXEC SQL INSERT INTO tab1 VALUES ('abcd', ENCRYPT_AES("111-222-3333"));
- An ENCRYPT_AES call fails with an error if the *password* argument is omitted and no *session password* has been set, or if an explicit *password* argument length is < 6 bytes or > 128 bytes.
- An error can be issued if the encrypted returned value is larger than the data type it is to be stored in.

Manual Pages – DECRYPT_CHAR

- **DECRYPT_CHAR(encrypted_data [, password])**
- **Also known as DECRYPT**
- **Encrypted data in base-64 encoding contains:**
  - Information about encryption method.
  - All other data needed to decrypt it.
  - Except the password!
- **Error if the data is not encrypted.**
-
```
SELECT DECRYPT_CHAR(EncryptedColumn, 'password')
    FROM SomeTable
    WHERE PK_Column = ?
```
- **Invariant function.**

Manual Pages – DECRYPT_CHAR function

- **DECRYPT_CHAR accepts as its first argument an *encrypted_data* character string that can have any character type (CHAR, LVARCHAR, NCHAR, NVARCHAR, or VARCHAR).**

- **A *password* must be specified as its second argument, unless the SET ENCRYPTION statement has specified for this session the same *session password* by which the first argument was encrypted.**

- **The DECRYPT_CHAR function also accepts as its first argument an *encrypted_data* large object of type BLOB or CLOB. A password must be specified as its second argument, unless the SET ENCRYPTION statement has specified as the default for this session the same *password* by which the first argument was encrypted.**
  - If the call to DECRYPT_CHAR is successful, it returns a CLOB large object that contains the plain text version of the *encrypted_data* argument.

Manual Pages – DECRYPT_CHAR

- **If the call to DECRYPT_CHAR with an encrypted string argument is successful, it returns a character string that contains the plain text version of the *encrypted_data* argument.**
- **The following example returns a character string containing a decrypted value from the ssid column of the engineers table for the row whose empno value is 287:**
  - **SELECT DECRYPT_CHAR (ssid) FROM engineers WHERE empno = 287;**
- **If the first argument to DECRYPT_CHAR is not an encrypted value, or if the second argument (or the default *password* specified by SET ENCRYPTION) is not the *password* that was used when the first argument was encrypted, Informix issues an error, and the call to DECRYPT_CHAR fails.**
- **Do not use DECRYPT_CHAR (or any other decryption function) to create a functional index on an encrypted column:**
  - This would store the decrypted values as plain text data in the database, defeating the purpose of encryption.

Manual Pages – DECRYPT_BINARY

- **DECRYPT_BINARY(encrypted_data [, password])**

- **Encrypted data contains:**
  - Information about encryption method.
  - All other data needed to decrypt it.
  - Except the password!

- **Error if the data is not encrypted.**

- ```
  SELECT
      DECRYPT_BINARY(EncryptedByteColumn, ?)
      FROM SomeTable
      WHERE PK_Column = ?
  ```

- **Invariant function.**
  ```

Manual Pages - DECRYPT_BINARY function

- Accepts as its first argument an *encrypted_data* large object of type BLOB or CLOB.
- A *password* must be specified as its second argument, unless the SET ENCRYPTION statement has the same default password for this session by which the first argument was encrypted.
- If the call to DECRYPT_BINARY is successful, it returns a BLOB or CLOB large object that contains the plain text version of the *encrypted_data* argument. The decrypted BLOB or CLOB object is temporarily stored in the default sbspace that the SBSPACENAME configuration parameter setting specifies.
- If the first argument to DECRYPT_BINARY is an encrypted value of a character data type, Informix invokes the DECRYPT_CHAR function and attempts to decrypt the specified value.

Manual Pages - DECRYPT_BINARY function

- If the first argument to DECRYPT_BINARY is not an encrypted value, or if the second argument (or the default *password* specified by SET ENCRYPTION) is not the *password* that was used when the first argument was encrypted, Informix issues an error, and the call to DECRYPT_BINARY fails.

- Do not use DECRYPT_BINARY (or any other decryption function) to create a functional index on an encrypted column as the index stores the decrypted values as plain text data in the database, defeating the purpose of encryption.

Manual Pages – GETHINT

- **GETHINT(encrypted_data)**

- **Returns the hint (if any) from the encrypted data.**
  - Or an empty string (NULL).

- ```
  SELECT GETHINT(enc_cc_number)
      FROM cc_info
      WHERE user_id = ?;
  ```

- **Invariant function.**

- **Anybody can get any hint at any time.**
  ```

Manual Pages – GETHINT function

- **GETHINT returns a character string that a previously executed SET ENCRYPTION PASSWORD statement defined for the *password,* used when *encrypted_data* was encrypted by the ENCRYPT_AES function or by the ENCRYPT_TDES function.**

- **This string typically provides information to the user, known as a *hint,* which helps to specify the *password* needed to return the plain text version of *encrypted_data* with the DECRYPT_CHAR or DECRYPT_BINARY decryption function.**
  - The *hint* string should not be the same as the *password*.
- **In the following example, a query returns the *hint* string into a host variable called myhint:**
  - **EXEC SQL SELECT GETHINT(creditcard) INTO :myhint FROM customer WHERE id = :myid;**
- **An error is returned, rather than a *hint* string, if the *encrypted_data* argument to the GETHINT function is not an encrypted string or an encrypted large object.**

Manual Pages – SET ENCRYPTION PASSWORD

- **SQL statement**
  - `SET ENCRYPTION PASSWORD 'password'`
    `[ WITH HINT 'hint string' ];`

- **Specifies the password that will be used by default.**
  - When no password explicitly provided:
    - To encryption functions.
    - To decryption functions.

- **Optionally specifies the hint that will be used by default:**
  - When no hint provided to encryption functions.

- **Session wide password management:**
  - For easy programming.
  - Support for views, triggers, SPL.

Manual Pages – SET ENCRYPTION PASSWORD statement

- **Define or reset a session password for the encryption and decryption of character, BLOB, or CLOB values.**

- **This is an extension to the ANSI/ISO standard for SQL.**
- **You can use this statement with SQL, ESQL/C.**

Element	Description	Restrictions
hint	String that GETHINT returns from an encrypted argument	(0 byte) ≤ *hint* ≤ (32 bytes). Do not include the *password* in the *hint*.
password	Password (or a multi-word phrase) for data encryption	(6 bytes) ≤ *password* ≤ (120 bytes). Do not specify your login password.

Manual Pages – SET ENCRYPTION PASSWORD statement

- **Declares a password to support data confidentiality through built-in functions that use the Triple-DES or AES algorithms for encryption and decryption.**

- **These functions enable the database to store sensitive data in an encrypted format that prevents anyone who cannot provide the secret password from viewing, copying, or modifying encrypted data.**

- **The password is not stored as plain text in the database and is not accessible to the DBA.**
  - By default, communication between client systems and Informix is in plain text.
  - Unless the database is accessible only by a secure network, the DBA must enable the encryption communication support module (ENCCSM) or SSL to provide data encryption between the database server and any client system.
  - Otherwise, an attacker might read the password and use it to access encrypted data.

Manual Pages – SET ENCRYPTION PASSWORD statement

- If the network is not secure, all of the database servers in a distributed query need ENCCSM enabled, so that the password is not transmitted as plain text.

- Operations on encrypted data tend to be slower than corresponding operations on plain text data, but use of this feature has no effect on unencrypted data.

- The SET ENCRYPTION PASSWORD statements can be prepared, and EXECUTE IMMEDIATE can process a prepared SET ENCRYPTION PASSWORD statement.

Informix Storage Requirements for Column Encryption

- Use ENCRYPT_AES or ENCRYPT_TDES functions to encrypt data.
- Encrypted values of character data types are stored in BASE64 format (also called Radix-64). For character data, this requires significantly more storage than the corresponding unencrypted data.
- Omitting the *hint* can reduce encryption overhead by more than 50 bytes for each encrypted value.
- It is the responsibility of the user to make sufficient storage space available for encrypted values.

Original Data Type	Encrypted Data Type	Base 64 format	Function
CHAR	CHAR	yes	DECRYPT_CHAR
NCHAR	NCHAR	yes	DECRYPT_CHAR
VARCHAR	VARCHAR	yes	DECRYPT_CHAR
NVARCHAR	NVARCHAR	yes	DECRYPT_CHAR
LVARCHAR	LVARCHAR	yes	DECRYPT_CHAR
BLOB	BLOB	no	DECRYPT_BINARY
CLOB	BLOB	no	DECRYPT_CHAR

Informix Storage Requirements for Column Encryption

- You cannot encrypt a column of the IDSSECURITYLABEL data type.
- If the encrypted VARCHAR (or NVARCHAR) value is longer than the 255 byte maximum size for those data types, the encryption function returns a CHAR (or NCHAR) value of sufficient size to store the encrypted value.
- DECRYPT_BINARY and DECRYPT_CHAR both return the same value from encrypted CHAR, NCHAR, VARCHAR, NVARCHAR, or LVARCHAR values.
- No built-in encryption or decryption functions support BYTE or TEXT data types, but you can use BLOB data types to encrypt very large strings.

Informix Storage Requirements for Column Encryption

- If the target column size in which you intend to store encrypted data is smaller than the encrypted data length, truncation occurs when you insert the encrypted data into the column.
- The truncated data cannot subsequently be decrypted, because the data length indicated in the header of the encrypted string does not match what the column stores.
- To avoid truncation, make sure that any column storing encrypted strings has sufficient length.
- Besides the unencrypted data length, the storage required for encrypted data depends on the encoding format, on whether you specify a *hint*, and on the block size of the encryption function.

Manual Pages – Storage Space Requirements - Math

- **Encrypted data will be stored in character columns.**
- **Needs more space than the unencrypted data.**
- **If input data string is N bytes long:**
  - AES = **B64(NGM(N, 16) + H + 8) + 11**
  - Triple-DES = **B64(NGM(N, 8) + H + 8) + 11**
    - **H = 0** with no hint; **H = 40** with hint.
    - **NGM(x,y)** – Next multiple of **y** that is greater than **x**.
      - **NGM(x, y) = ((x + y) ÷ y) × y**
    - **B64(x)** – Base-64 encoding size.
      - **B64(x) = ((x + 2) ÷ 3) × 4**
  - AES can be bigger than Triple-DES, but not by much.
- **Do not normally encrypt 4-byte integer numbers.**

Manual Pages – Storage Space Requirements

Input Size (bytes)	Triple-DES (no hint)	AES (no hint)	Triple-DES (with hint)	AES (with hint)
1..7	35	43	87	99
8..15	43	43	99	99
16..23	55	67	107	119
24..31	67	67	119	119
32..39	75	87	131	139
40..47	87	87	139	139
100	163	171	215	227
200	299	299	355	355
500	695	707	747	759

Page size for storage might matter here ……

Specifying a Session Password and Hint

- The required *password* specification can be quoted strings or other character expression that evaluates to a string whose length is at least 6 bytes but no more than 120 bytes. The optional *hint* can specify a string no longer than 32 bytes.

- The password or *hint* can be a single word or several words. The *hint* should be a word or phrase that helps you to remember the *password*, but does not include the *password*. You can subsequently execute the built-in GETHINT function (with an encrypted value as its argument) to return the plain text of *hint*.

- The following ESQL/C program fragment defines a routine that includes the SET ENCRYPTION PASSWORD statement and executes DML statements:

Specifying a Session Password and Hint - code

```
process_ssn( )
 { EXEC SQL BEGIN DECLARE SECTION;
char password[128];
char myhint[33];
char myid[16], myssn[16];
EXEC SQL END DECLARE SECTION;
 . . .
EXEC SQL SET ENCRYPTION PASSWORD :password WITH HINT :myhint;
...
EXEC SQL INSERT INTO tab1 VALUES (':abcd', ENCRYPT_AES("111-22-3333"))
;
EXEC SQL SELECT pid, DECRYPT(ssn, :password) INTO :myid, :myssn;
...
EXEC SQL SELECT GETHINT(ssn) INTO :myhint, WHERE id = :myid; }
```

SET ENCRYPTION PASSWORD

- **Use with encrypt/decrypt functions to support:**
  - Column-Level Encryption:
    - All values in a given column of a database table are encrypted using the same password, the same encryption algorithm, and the same encryption mode.
      - In this case, you can save disk space by storing the *hint* outside the encrypted column, rather than repeating it in every row.
  - Cell-Level Encryption:
    - Values of a given column in different rows of the same database table are encrypted using different passwords, or different encryption algorithms, or different encryption modes.
      - This technique is sometimes necessary to protect personal data.
      - Cell-level encryption can cause substantial maintenance costs. If you implement this level of encryption, your application is responsible for determining which rows contain encrypted data and for using the correct code to handle the data.
      - The built-in decryption functions fail with error -26005 if they are applied to unencrypted data. The simplest way to avoid this error is to use column-level encryption rather than cell-level encryption.

SET ENCRYPTION PASSWORD

- If encryption functions are not used, users might enter unencrypted data into columns that are meant to contain encrypted data.

- To ensure that data entered into a field is always encrypted, use views and INSTEAD OF triggers.

Password Protection – Column and Cell Level Encryption

- **Passwords and hints that you declare with SET ENCRYPTION PASSWORD are not stored as plain text in any table of the system catalog, which also maintains no record of which columns or tables contain encrypted data.**
- **To prevent other users from accessing the plain text of encrypted data or of a password, however, you must avoid actions that might compromise the secrecy of a password:**
  - Do not create a functional index using a decryption function which would store plain-text index data in the database, defeating the purpose of encryption.
  - On a network that is not secure, always work with encrypted data, or use session encryption, because the SQL communication between client and server sends passwords, hints, and the data to be encrypted as plain text.
  - Do not store passwords in a trigger or in a UDR exposing the password publicly.
  - Do not set the session password prior to creating any view, trigger, procedure, or UDR. Set the session password only when you use the object. Otherwise, the password might be visible in the schema to other users, and queries executed by other users might return unencrypted data.

Password Protection (1) - Simple

```
-- reset session encryption password
set encryption password null;
-- create procedure without password
create procedure p1 ();
      insert into tab2 select (decrypt_char (col1)) from tab1;
end procedure;
-- set session encryption password
set encryption password ("PASSWD2");
-- insert data insert into tab1 values (encrypt_aes ('WXY'));
-- call procedure
```

- **Output from the SET EXPLAIN statement always displays the *password* and *hint* parameters as XXXXX, rather than displaying actual *password* or *hint* values.**

Row Based Encryption (1)

```
CREATE VIEW lm_patient_v AS SELECT * FROM lm_patient;
CREATE TRIGGER lm_patient_v_ins
   INSTEAD OF INSERT ON lm_patient_v REFERENCING NEW AS new
   FOR EACH ROW
(
INSERT INTO lm_patient(idlmpatient, namefampri, namefamfir, namefamsec
lastupdatedt) VALUES
(ENCRYPT_AES(new.idlmpatient),
ENCRYPT_AES(new.namefampri, 'D31st3r' || new.idlmpatient),
ENCRYPT_AES(new.namefamfir, 'D31st3r' || new.idlmpatient),
ENCRYPT_AES(new.namefamsec, 'D31st3r' || new.idlmpatient),
                    new.lastupdatedt)
```

Row Based Encryption (2)

```
CREATE TRIGGER lm_patient_v_upd INSTEAD OF UPDATE ON lm_patient_v
  REFERENCING OLD AS old NEW AS new FOR EACH ROW
(
      UPDATE lm_patient
        SET (idlmpatient, namefampri, namefamfir, namefamsec lastupdatedt)
         = (CASE WHEN new.idlmpatient = old.idlmpatient THEN new.idlmpatient
                    ELSE ENCRYPT_AES(new.idlmpatient) END,
            CASE WHEN new.namefampri  = old.namefampri  THEN new.namefampri
                    ELSE ENCRYPT_AES(new.namefampri, 'D31st3r' || CASE WHEN
new.idlmpatient = old.idlmpatient THEN DECRYPT_CHAR(new.idlmpatient) ELSE
new.idlmpatient END) END,
            CASE WHEN new.namefamfir  = old.namefamfir  THEN new.namefamfir
                     ELSE ENCRYPT_AES(new.namefamfir, 'D31st3r' || CASE WHEN
new.idlmpatient = old.idlmpatient THEN DECRYPT_CHAR(new.idlmpatient) ELSE
new.idlmpatient END) END,
           CASE WHEN new.namefamsec  = old.namefamsec  THEN new.namefamsec
                    ELSE ENCRYPT_AES(new.namefamsec, 'D31st3r' || CASE WHEN
new.idlmpatient = old.idlmpatient THEN DECRYPT_CHAR(new.idlmpatient) ELSE
new.idlmpatient END) END,
            new.lastupdatedt)
       WHERE lm_patient.patid = old.patid
);
```

Row Based Encryption (3)

```
SELECT DECRYPT_CHAR(idlmpatient),
       DECRYPT_CHAR(namefampri, 'D31st3r' || DECRYPT_CHAR(idlmpatient)),
       DECRYPT_CHAR(namefamfir, 'D31st3r‘  || DECRYPT_CHAR(idlmpatient)),
       DECRYPT_CHAR(namefamsec, 'D31st3r' || DECRYPT_CHAR(idlmpatient))
  FROM lm_patient
 WHERE idlmpatient = ENCRYPT_AES('123462‘);
```

How To SET DEFAULT CRYPTO PASSWORD

```
execute procedure dbconfig('30874MYPASS');

create function dbconfig(p_salt varchar(40)) returning varchar(40);
        define m_int varchar(40);
        define m_key varchar(40);
        define m_chk varchar(40);

        LET m_int = '0FDE0F14A352790928';
        LET m_key = p_salt[6,10] || m_int[3,10];
        LET m_chk = LPAD(MOD(TRUNC(((TO_CHAR(TODAY, '%m%d')* 1103515245) + 12345)  / 65536),
32768), 5, '0');

        IF m_chk = p_salt[1, 5] THEN
                SET ENCRYPTION PASSWORD m_key;
        END IF

        RETURN m_chk;
end function;
DELETE FROM sysprocbody WHERE procid =
            (SELECT procid FROM sysprocedures WHERE procname = 'dbconfig');
```

Performance Impact of Encryption

- **Comparing ‘apples to apples’ is hard.**

- **An accurate comparison consists of:**
  - IDS encrypting and decrypting data sent unencrypted by client
    - versus
  - Client encrypting and decrypting data sent encrypted to IDS.

- **Unfortunately, that requires benchmarking an application with cryptography built in.**
  - Can be done, but is fiddly.

- **So, everybody makes an ‘apples to oranges’ comparison:**
  - IDS encrypting and decrypting data
  - vs.
  - IDS not encrypting and decrypting data.

Performance Impact of Encryption

- **The performance impact of encryption is significant.**
  - It depends on direction:
    - Encrypting is slower than decrypting.
  - It does not depend measurably on algorithm:
    - AES performs at the same speed as Triple-DES.
  - It does depend on data size:
    - The relative overhead is less when there is more data to encrypt.

- **Do not use encryption just because it is sexy.**
  - Use it where there is a demonstrable business or legal need.

Performance Impact – Credit Card Number

- **Credit card number plus expiry date**
  - 20 digits + 5 punctuators
  - "**4567-1234-2345-3456□01/99**"
  - Needs 67 characters without hint; 119 with hint.

- **These comparisons should be repeated for your machine!**
  - On an old, small, slow Sun UltraSparc 10:
    - Solaris 8
    - Single CPU at 333 MHz
    - 256 MB
    - Single user
    - No /dev/random or /dev/urandom
    - Running several IDS servers, Apache, etc.

Performance Impact – Credit Card Number

- **Without hints – batch mode processing – 5000 rows.**
  - ```
    INSERT INTO NewTable
        SELECT …, ENCRYPT_TDES(OtherColumn), …
        FROM OtherTable;
    ```

- **Encryption performance:**
  - 424 µs per row (without encryption)
  - 3601 µs per row (with Triple-DES encryption)
  - Ratio: 8.5:1
  - Cost: 3200 µs per call.
  - Even a trivial SPL procedure called in place of encryption levels the playing field a lot.
    ```

Performance Impact – Credit Card Number

- **Without hints – batch mode processing – 5000 rows.**
  - `SELECT …, DECRYPT_CHAR(EncryptedData), …`
    `FROM NewTable;`

- **Decryption performance:**
  - 155 µs per row (without decryption)
  - 1285 µs per row (with Triple-DES decryption)
  - Ratio: 8.3:1
  - Cost: 1100 µs per call.

- **Hence, decryption costs about 1/3 what encryption costs.**
  - Major component of encryption cost:
    - Generating cryptographically random number.
  - Just as well you'll normally do more decryption than encryption.

Modes of Use

- **Column-Level Encryption is an enabling technology**
- **You decide how you are going to use it.**
- **Two main modes of operation:**
  - Web mode:
    - Different keys for each row of data.
  - MIS mode:
    - Same key for each row of data.

Web Application

- **Think of credit card numbers on a web site.**
- **Different keys for each row of data.**
- **Hints are important.**
- **Key management is not a major problem.**
  - You do not store the key (password) for the user.
  - If the user forgets their password, the data is re-enterable.
  - Or you get into more fancy schemes:
    - Encrypt user's passwords with known key.
    - But these systems are generally less secure.
- **SET ENCRYPTION PASSWORD is irrelevant.**

MIS Applications

- **Same key for each row of data**
- **Hints are irrelevant**
- **Key management is critical.**
- **SET ENCRYPTION PASSWORD is critical.**
  - You might be better off coding with a temporary table:
    - Contains one row of data – the password.
    - Join with that table when you need encrypted data.
    - Avoids revealing the password in SQL.

Key Management

- **IDS does not do any key management.**
- **Keys are handled outside the DBMS.**
- **You can store keys in the DBMS if you want to.**
  - Securing them is your problem.
  - Probably encrypted with a single high-security password.

Encryption Without Changing Applications

- **You can do it,**
  - But it is not necessarily a good idea.
- **The application will not work as fast as without encryption,**
  - But it will run about as fast as if you rework it with encryption.
    - So, many people will do it.
- **The application will use more data space.**
  - But you won't be using hints.
- **The easiest approach uses more space than necessary.**
  - And more encryption and decryption operations.
- **It is a bad idea to use encrypted data as keys.**
  - Does the SSN get used as a key?
  - Encrypted keys are very much more difficult to handle.
  - Can you use a hash checksum instead?

Encryption Without Changing Applications: Techniques

- **Changes to database schema:**
  - Rename all tables containing encrypted data.
  - Change data types for columns that must be encrypted.
    - CHAR data type.
    - Expanded size.
  - Create views with the old table names:

```
CREATE VIEW oldname(key, col2, col3)
     AS SELECT key, DECRYPT(col2)::type1,
                    DECRYPT(col3)::type2
               FROM newname;
```

- **The hard part is setting the encryption password!**

Encryption Without Changing Applications: Techniques

- Create INSTEAD OF triggers on views.
  - CREATE TRIGGER ti_on INSTEAD OF INSERT ON oldname
    REFERENCING NEW AS new FOR EACH ROW
    (INSERT INTO newname VALUES
    (new.key, ENCRYPT_AES(new.col2),
    ENCRYPT_AES(new.col3)));
  - CREATE TRIGGER tu_on INSTEAD OF UPDATE ON oldname
    REFERENCING OLD AS old NEW AS new
    FOR EACH ROW
    (UPDATE newname SET (key, col2, col3) =
    (new.key, ENCRYPT_AES(new.col2),
    ENCRYPT_AES(new.col3))
    WHERE key = old.key);
  - CREATE TRIGGER td_on INSTEAD OF DELETE ON oldname
    REFERENCING OLD AS old FOR EACH ROW
    (DELETE FROM newname WHERE key = old.key);

Behind the Scenes

- **The encrypted data contains:**
  - Which encryption algorithm was used.
  - A random initialization vector (IV).
  - The encrypted data.
  - Optionally, the hint.

- **The IV ensures randomization:**
  - If the same data is encrypted with the same key,
  - The encrypted data is different.
    - Assuming the IV is different (and it ‘always’ is).

- **Text data is converted to Base-64 encoding.**
  - Binary data is not Base-64 encoded.
    - More compact.

Behind the Scenes

- Session password in shared memory is encrypted.
- Constant password in shared memory is encrypted.
- "onstat –g sql" will display not constant password.

Behind the Scenes

- **Encryption VPs**
  - Generating (cryptographically) good random numbers can block.
- **Define multiple ENCRYPT VP in ONCONFIG**
  - `VPCLASS encrypt,num=3`
- **Add or drop encryption VPs online**
  - `onmode -p +1 encrypt`
  - `onmode -p -1 encrypt`
- **If you don't define encryption VPs and use encryption**
  - One encryption VP is added automatically.

Gotchas

- **Encrypting BLOB data requires the correct configuration**
  - SBSPACE set in ONCONFIG file
    - One SBSPACE is automatically configured during the instance created during an install of Informix Dynamic Server for the first time.
  - (SYSSBSPACE set in ONCONFIG file, too)
    - One SBSPACE is automatically configured during the instance created during an install of Informix for the first time.

Things to Avoid

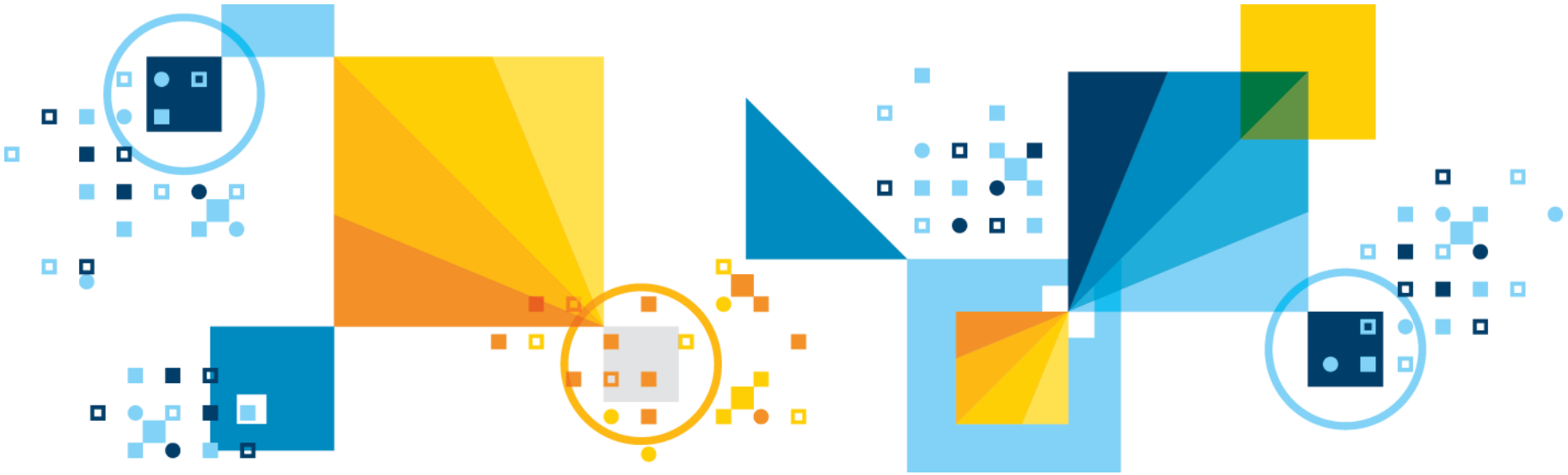
- **Do not index encrypted columns.**
  - You should not be searching for encrypted values.
  - If you do index a column, it is only useful for equality comparisons.
- **Do not create a functional index on the decrypted data.**
  - In a web-style application, you won't have the key.
  - In a MIS-style application, you will be storing the data unencrypted after all.
- **Avoid using encrypted columns as keys for tables.**
- **Do not ask Tech Support to retrieve your passwords.**
  - It can't be done!

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

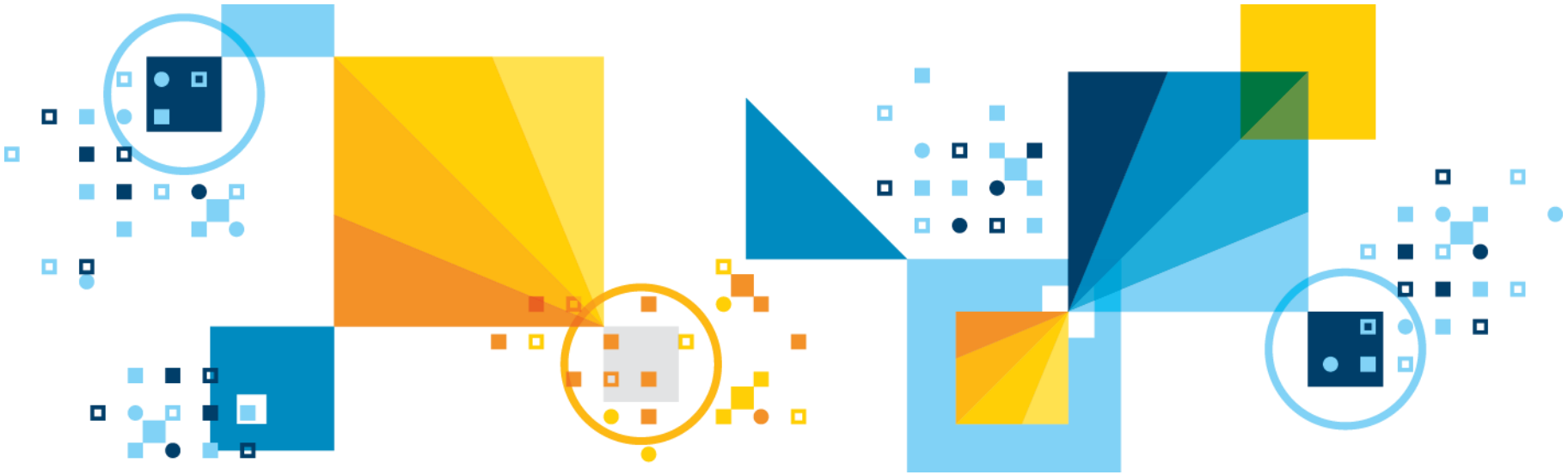
Auditing – Informix and Guardium



Scott Pickett – WW Informix Technical Sales

June 27, 2017

Auditing
Informix Trusted Secure Facility - onaudit



Auditing

- How can you tell who is accessing what?
- How do you detect unauthorized access attempts?
- How do you assess potential security damage?

Auditing

- **Audit**
- **ADTCFG File**
  - **ADTCFG** Parameters
- **onaudit Utility**
  - Audit Configuration
  - Audit Masks
  - Audit Events

Auditing

- **Auditing creates a record of selected activities that users perform.**
- **Audit is based on EVENTS and USERS.**
- **An audit administrator who analyzes the audit trail can use these records:**
  - To detect unusual or suspicious user actions and identify the users who performed those actions
  - To detect unauthorized access attempts
  - To assess potential security damage
  - To provide evidence in investigations, if necessary
  - To provide a passive deterrent against unwanted activities

ADTCFG File

- **Audit configuration file:**
  - **$INFORMIXDIR/aaodir/adtcfg[.nn]**

Parameter	Value		Description
ADTMODE	1	#	Auditing mode
ADTPATH	/auditlog	#	Audit log directory
ADTSIZE	100000	#	Maximum size of audit file
ADTERR	0	#	Error handling modes
ADTROWS	0	#	Row Level Audit Control

- **When Informix server is started or audit configuration is changed, adtcfg.<servnum> file is written.**
  - Restarting the server uses this file.

ADTCFG Parameters

- **ADTMODE** controls the level of auditing:
  - **0**: auditing disabled
  - **1**: auditing on; starts auditing for all sessions
  - **3**: auditing on; audits DBSSO actions
  - **5**: auditing on; audits database server administrator actions
  - **7**: auditing on; audits DBSSO and database server administrator actions

ADTCFG Parameters

- **ADTPATH** specifies the directory in which the database server saves audit files.
  - Make sure that the directory that you specify has appropriate access privileges to prevent unauthorized access to audit records.

- **ADTSIZE** configuration parameter specifies the maximum size of an audit file.
  - When a file reaches the maximum size, the database server saves the current audit file and creates a new one.

- **ADTERR specifies how the database server behaves when it encounters an error while it writes an audit record.**

ADTCFG Parameters

- **ADTROWS configuration parameter to control selective row-level auditing of tables:**
  - **0**: for auditing row-level events on all tables
  - **1**: to allow control of which tables are audited.
    - Row-level events **DLRW**, **INRW**, **RDRW**, and **UPRW** are audited **only** on tables for which the **AUDIT** flag is set.
  - **2**: to turn on selective row-level auditing and also include the primary key in audit records

ADTCFG Parameters

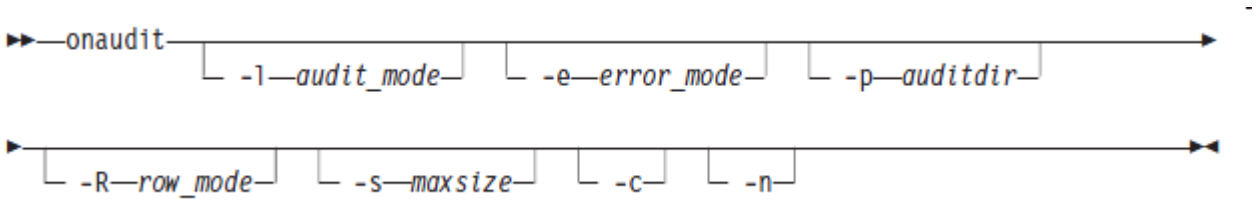
- Designate tables for ADTROWS setting AUDIT flag:
  - `CREATE TABLE existing_syntax WITH AUDIT;`
  - `ALTER TABLE existing_syntax ADD AUDIT;`
  - `ALTER TABLE existing_syntax DROP AUDIT;`

onaudit Utility

- **Manages audit masks and configuration.**
- **Needs to be DBSSO or AAO:**
  - **DBSSO** can perform functions related to audit setup.
  - **AAO** can perform functions related to audit analysis.

Audit Configuration

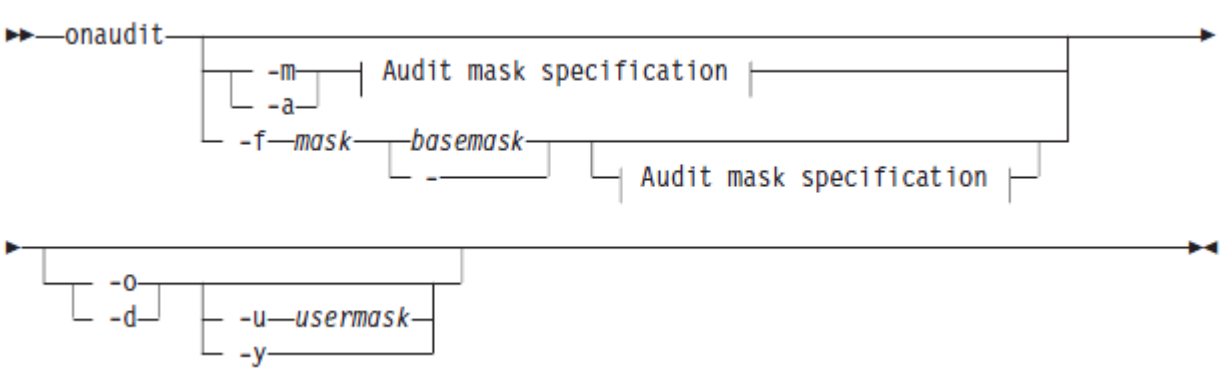
- **Use the onaudit utility to start, stop, and configure auditing.**



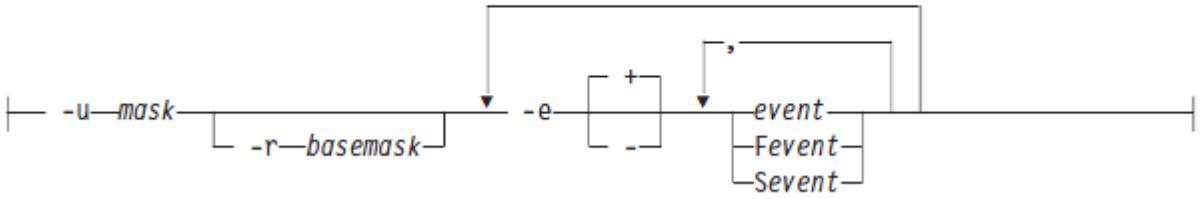
- **onaudit –c**
  - Display current audit configuration.

Audit Masks

- **Use the onaudit utility to add, modify, delete and display audit masks.**



Audit mask specification:



Audit Masks

- **There are 3 defined global masks**
  - _default, _require, _exclude
- **They must be created to have values**
  - By default they are empty
- **The audit mask used for a session is calculated at the beginning of the session**
  - (user or _default) + _require - _exclude
- **onaudit -o**
  - To display existing audit masks.

Audit Events

- **Audit events are represented by a four character mnemonic:**
  - AAOO
    - AA is a letter code for action
    - OO is a letter code for the object
  - e.g. **ACTB** means access table

- **May have prepended ‘S’ or ‘F’ to indicate Success or Failure only.**
  - e.g. **SACTB** or **FACTB**.

Notes

- **Informix onaudit output can be redirected to a database where it can be setup as SQL queryable, albeit at the work of the customer as this is not preconfigured.**

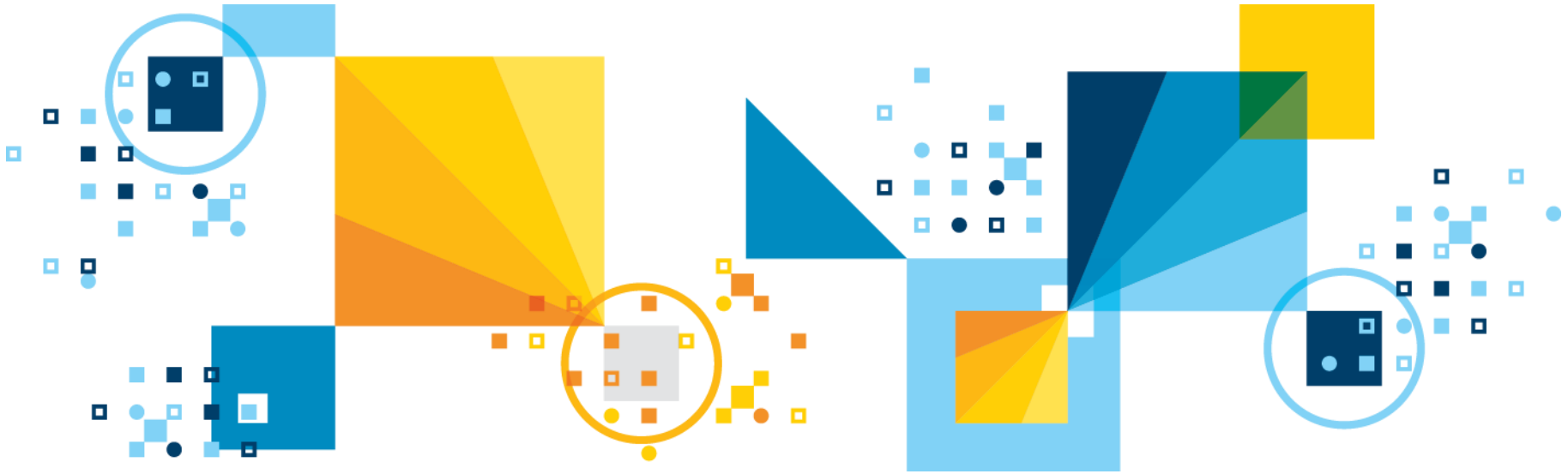
- **Normally, output goes to O/S file output, and the file systems for these should be very large if everything is audited.**
  - Heavy scheduler usage to swap files in and out is often employed, and to compress them as well.

- **Because O/S file output is employed, unless file systems off of the database server are employed, performance can be impacted.**

Questions

Scott Pickett – WW Informix Technical Sales
June 27, 2017

Auditing – Guardium



Guardium – What is it ?

- **Lightweight software or hardware/software based monitoring utility to audit user actions.**
  - Guardium has been ported to Informix for many years now and works quite well.
  - It is actually much more, but we will confine this presentation to databases

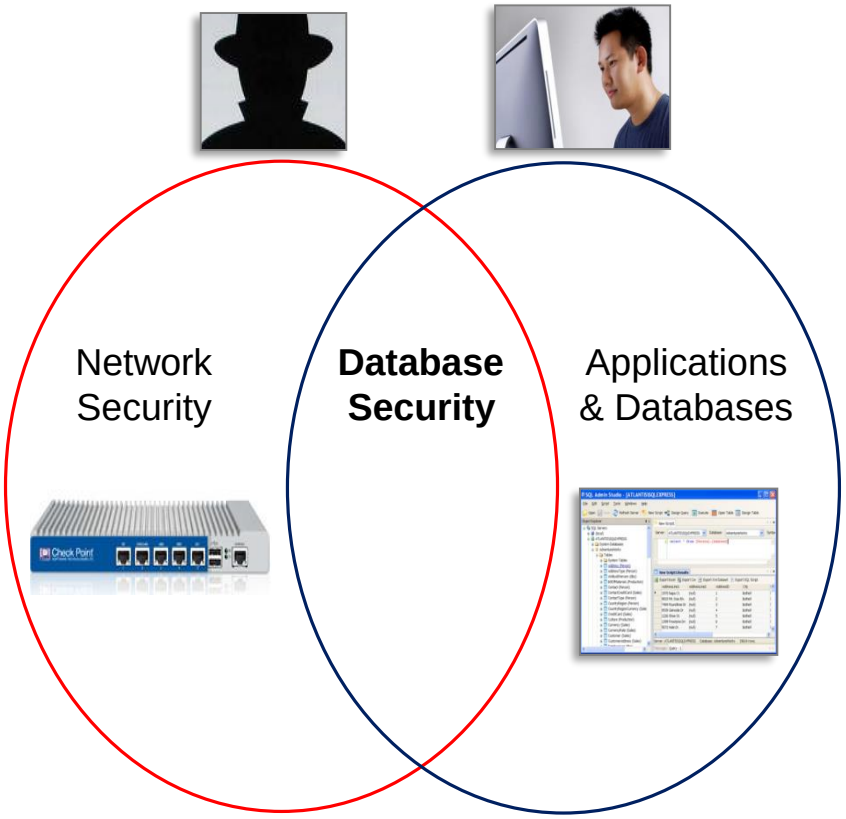
Database Monitoring: 3 Key Business Drivers

- **Internal threats**
  - Identify unauthorized changes (governance)
    - Prevent data leakage
- **External threats**
  - Prevent theft
- **Compliance**
  - Simplify processes
  - Reduce costs

Cultural/Technical Issues

"DBAs spend less than 5% of their time on database security."

Market Overview: Database Security
Noel Yuhanna, Forrester Research, February 2009



"There is more to risk than weak software."

Hackers compromise in 3 ways

1. **Weak Software**

Buffer overflows, OS/application vulnerabilities

2. **Weak Configurations**

Default configurations, weak passwords, failure to harden

3. **Weak People**

Insider threat, social engineering

Josh Corman, **IBM/ISS**

Addressing Key Stakeholders

SECURITY OPERATIONS

- ✓ Real-time policies
- ✓ Secure audit trail
- ✓ Data mining & forensics

COMPLIANCE AUDIT

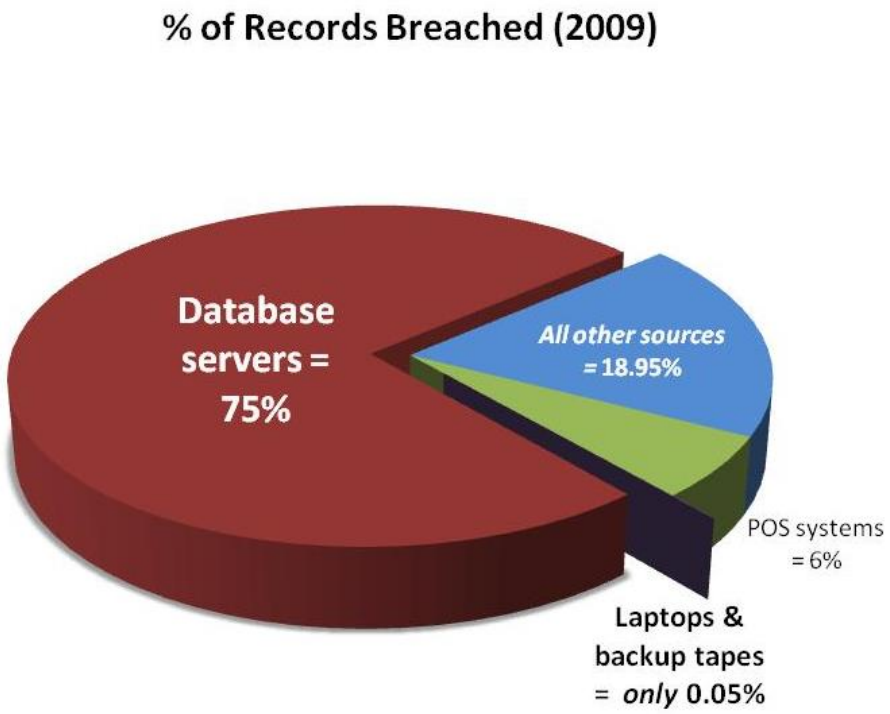
- ✓ Separation of duties
- ✓ Best practices reports
- ✓ Automated controls

APPLICATION & DATABASE

- ✓ Minimal impact
- ✓ Change management
- ✓ Performance optimization

Guardium: 100% Visibility & Unified View

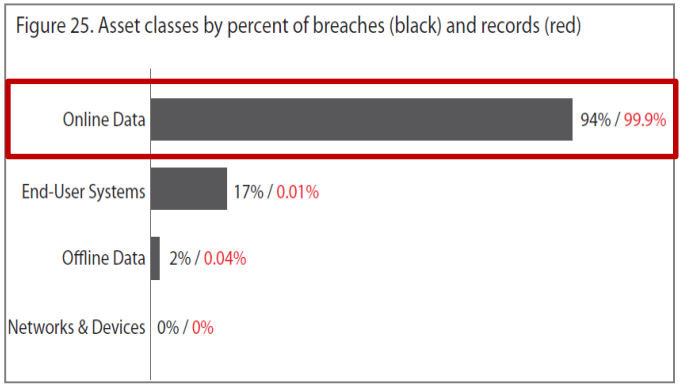
Combined Threats… Database Servers = Vast Majority of Compromised Records



2009 Data Breach Report from Verizon Business RISK Team

http://www.verizonbusiness.com/resources/security/reports/2009_databreach_rp.pdf

Online data = 99.9% of all compromised records



"Although much angst and security funding is given to **offline data, mobile devices,** and **end-user systems,** these assets are simply **not a major point of compromise."**

Example - Other vendors (1)

- **Recently, MongoDB has been exposed by University students testing access to internet sites of private corporations as being vulnerable to simple hacks; 40,000 online databases in France and Germany was the sample for the test:**

  "*The cause is a misconfigured open source database upon which millions of online stores and platforms from all over the world base their services. If the operators blindly stick to the defaults in the installation process and do not consider crucial details, the data is available online, completely unprotected.*"

- **More here:**

  http://www.sciencedaily.com/releases/2015/02/150210083803.htm

- **And here:**

  https://securityintelligence.com/news/insecure-configuration-of-mongodb-other-databases-could-be-leaking-information/

- **This one shows what a hacker could do:**

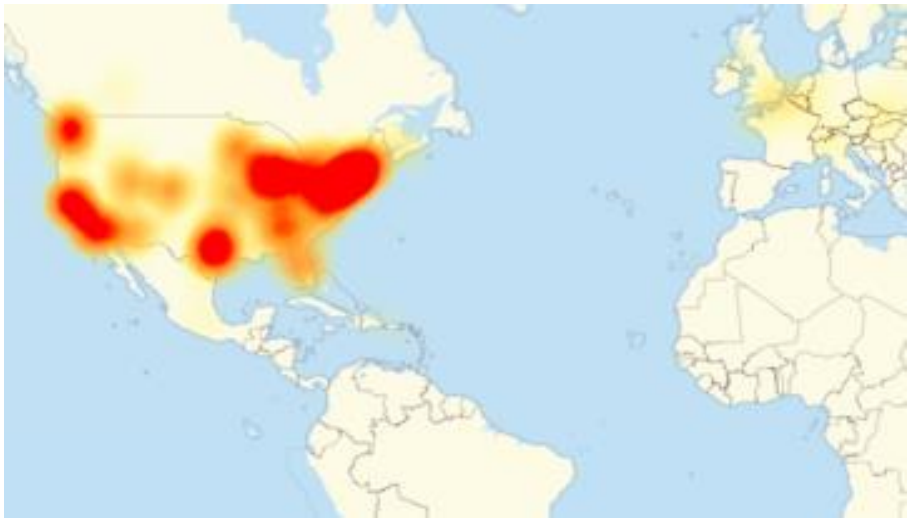
  http://blog.binaryedge.io/2015/08/10/data-technologies-and-security-part-1/

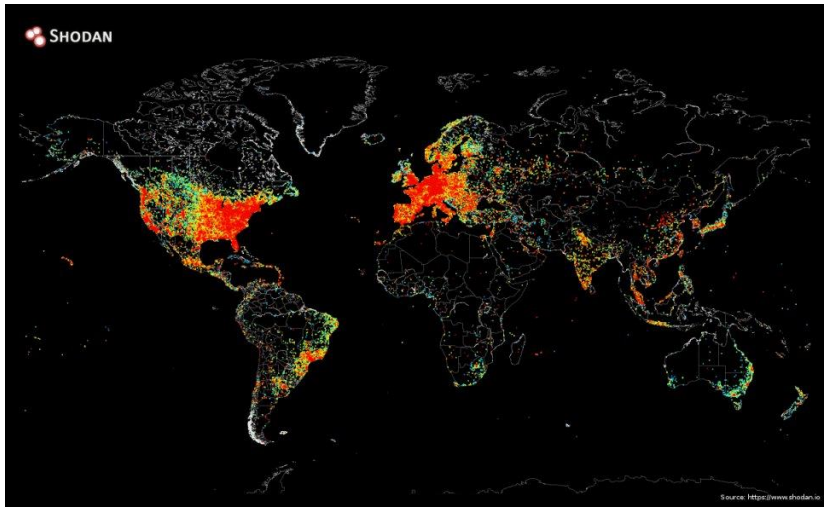
Example - Other vendors (2)

- http://it.slashdot.org/story/15/12/17/0319203/over-650-tb-of-data-up-for-grabs-from-publicly-exposed-mongodb-database

"A scan performed over the past few days by John Matherly, the creator of the Shodan search engine, has found that there are at least 35,000 publicly accessible and insecure MongoDB databases on the Internet, and their number appears to be growing. Combined they expose 684.8 terabytes of data to potential theft. Matherly originally sounded the alarm about this issue back in July, when he found nearly 30,000 unauthenticated MongoDB instances. He decided to revisit the issue after a security researcher named Chris Vickery recently found information exposed in such databases that was associated with 25 million user accounts from various apps and services, including 13 million users of the controversial OS X optimization program MacKeeper, as reported on Slashdot on Wednesday."

Video Cameras, DVR's Take Down Part of the Internet





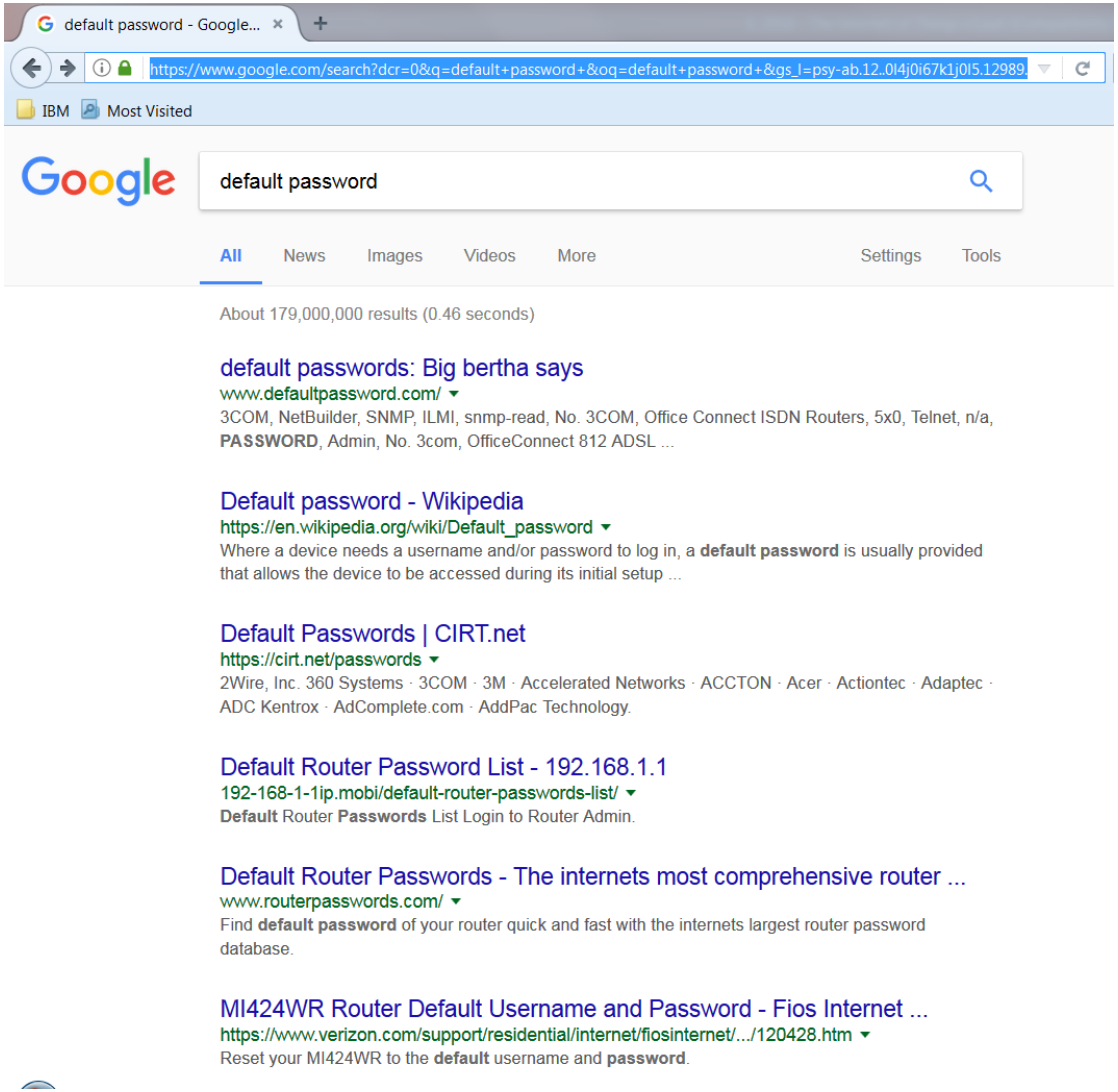
- In Oct. 2016, 100,000 DVR's and video cameras manufactured by one company were infected with a virus that took down one of the Internet's main traffic cop server farms, which in turn led to widespread outages of some heavily traffic'ed web sites on the web.
- The causes were default device usernames and passwords posted to a chat site by the company's technical staff to help customers and no firmware upgrades.

2014 - WW internet Traffic Routing
Today about 70% of the traffic WW starts in or passes thru the US and Europe.

Hackers Financially Incentivized - Ransomware

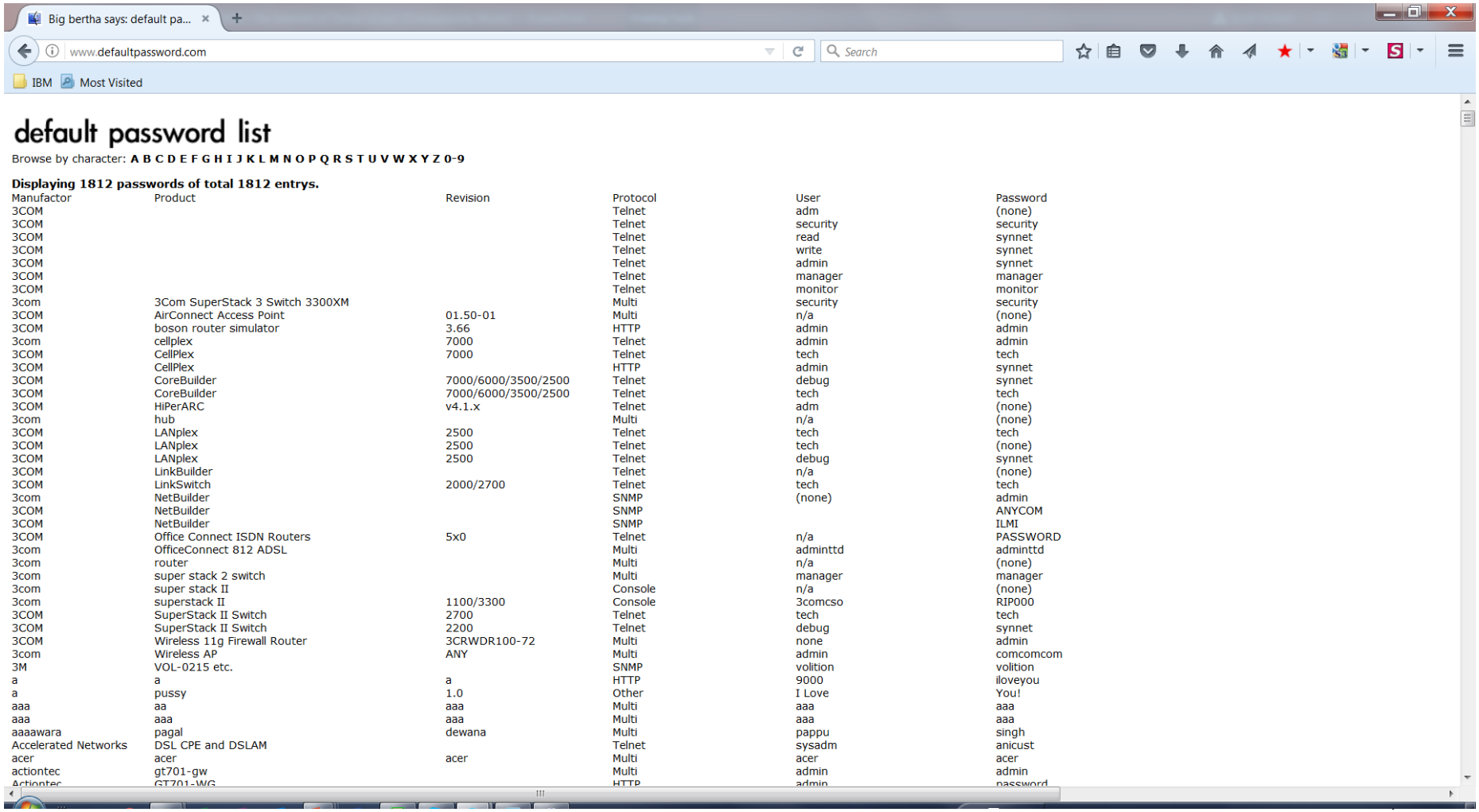
- **Now those whom manufacture viruses have combined all of the latest technology trends into profit making ventures:**
  - Using the Internet for “distribution”
  - Devices of all kinds to embed their viruses within
    - Consequence of everything having a unsecured chip ……
  - Encryption, embedded within the viruses, to render entire machines useless
  - Bitcoin, how they want to be paid, is untraceable, in return they provide the encryption key via a secure channel, which may or may not work ……..
    - Cash is not accepted here ……. Nor Amex, MC or Visa
  - Graphics, with a Visible Countdown Clock, representing the remaining amount of time to pay before there is no hope of getting a key ……
    - T-minus 10 …. and counting …….
  - Consequences of not paying
    - The entire operation of the business or entity failing
      - Police and Sheriff’s Departments
      - Water and Sewer Utilities
      - Hospitals
      - Doctors Offices
      - Subway systems …….

So just what is out there …… First 5 out of 179 Million?



Default Device, Software, Site passwords – 10/10/2017

http://www.defaultpassword.com/



Big bertha says: default pa...

www.defaultpassword.com

Search

IBM Most Visited

default password list

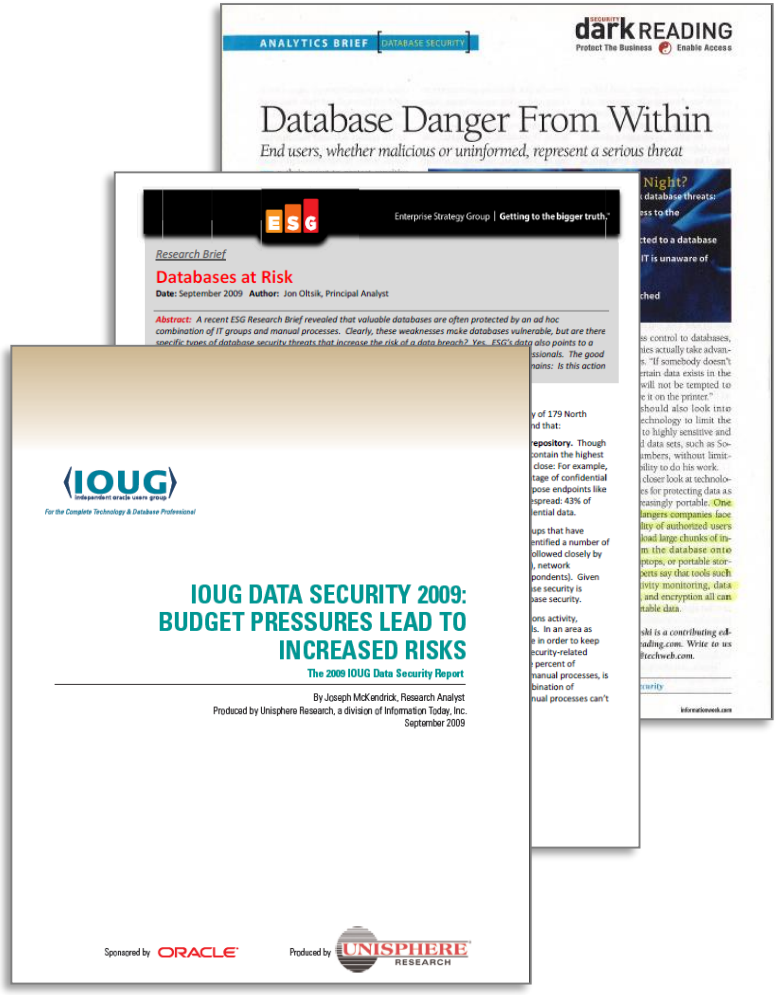
Browse by character: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0-9

Displaying 1812 passwords of total 1812 entrys.

Manufactor	Product	Revision	Protocol	User	Password
3COM			Telnet	adm	(none)
3COM			Telnet	security	security
3COM			Telnet	read	synnet
3COM			Telnet	write	synnet
3COM			Telnet	admin	synnet
3COM			Telnet	manager	manager
3COM			Telnet	monitor	monitor
3com	3Com SuperStack 3 Switch 3300XM		Multi	security	security
3COM	AirConnect Access Point	01.50-01	Multi	n/a	(none)
3COM	boson router simulator	3.66	HTTP	admin	admin
3com	cellplex	7000	Telnet	admin	admin
3COM	CellPlex	7000	Telnet	tech	tech
3COM	CellPlex		HTTP	admin	synnet
3COM	CoreBuilder	7000/6000/3500/2500	Telnet	debug	synnet
3COM	CoreBuilder	7000/6000/3500/2500	Telnet	tech	tech
3COM	HiPerARC	v4.1.x	Telnet	adm	(none)
3com	hub		Multi	n/a	(none)
3COM	LANplex	2500	Telnet	tech	tech
3COM	LANplex	2500	Telnet	tech	(none)
3COM	LANplex	2500	Telnet	debug	synnet
3COM	LinkBuilder		Telnet	n/a	(none)
3COM	LinkSwitch	2000/2700	Telnet	tech	tech
3com	NetBuilder		SNMP	(none)	admin
3COM	NetBuilder		SNMP		ANYCOM
3COM	NetBuilder		SNMP		ILMI
3COM	Office Connect ISDN Routers	5x0	Telnet	n/a	PASSWORD
3com	OfficeConnect 812 ADSL		Multi	adminttd	adminttd
3com	router		Multi	n/a	(none)
3com	super stack 2 switch		Multi	manager	manager
3com	super stack II		Console	n/a	(none)
3com	superstack II	1100/3300	Console	3comcso	RIP000
3COM	SuperStack II Switch	2700	Telnet	tech	tech
3COM	SuperStack II Switch	2200	Telnet	debug	synnet
3COM	Wireless 11g Firewall Router	3CRWDR100-72	Multi	none	admin
3com	Wireless AP	ANY	Multi	admin	comcomcom
3M	VOL-0215 etc.		SNMP	volition	volition
a	a	a	HTTP	9000	iloveyou
a	pussy	1.0	Other	I Love	You!
aaa	aa	aaa	Multi	aaa	aaa
aaa	aaa	aaa	Multi	aaa	aaa
aaaawara	pagal	dewana	Multi	pappu	singh
Accelerated Networks	DSL CPE and DSLAM		Telnet	sysadm	anicust
acer	acer	acer	Multi	acer	acer
actiontec	gt701-gw		Multi	admin	admin
Actiontec	GT701-WG		HTTP	admin	password

Database Danger from Within

- **“Organizations overlook the most imminent threat to their databases: authorized users.” (Dark Reading)**
- **“No one group seems to own database security … This is not a recipe for strong database security” … 63% depend primarily on manual processes.” (ESG)**
- **Most organizations (62%) cannot prevent super users from reading or tampering with sensitive information … most are unable to even detect such incidents … only 1 out of 4 believe their data assets are securely configured (Independent Oracle User Group).**



http://www.darkreading.com/database_security/security/app-security/showArticle.jhtml?articleID=220300753

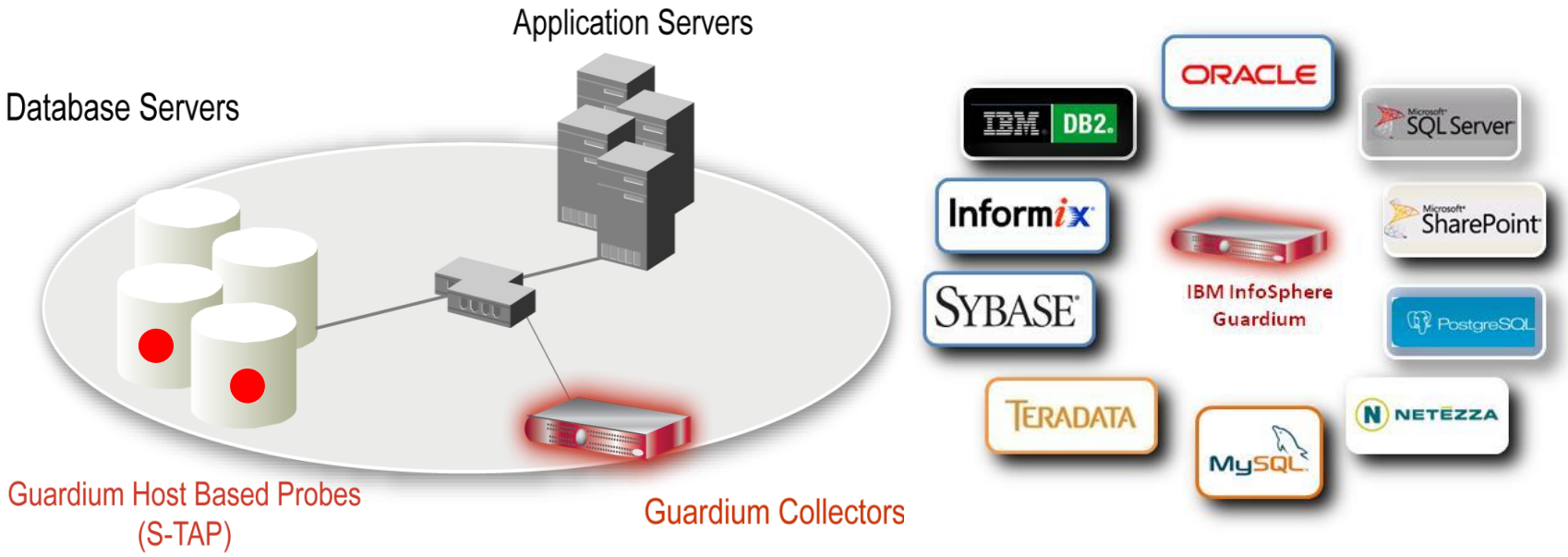
http://www.guardium.com/index.php/landing/866/

The Compliance Mandate

Audit Requirements	COBIT (SOX)	PCI-DSS	ISO 27002	Data Privacy & Protection Laws	NIST SP 800-53 (FISMA)
1. Access to Sensitive Data (Successful/Failed SELECTs)		✓	✓	✓	✓
2. Schema Changes (DDL) (Create/Drop/Alter Tables, etc.)	✓	✓	✓	✓	✓
3. Data Changes (DML) (Insert, Update, Delete)	✓		✓		
4. Security Exceptions (Failed logins, SQL errors, etc.)	✓	✓	✓	✓	✓
5. Accounts, Roles & Permissions (DCL) (GRANT, REVOKE)	✓	✓	✓	✓	✓

DDL = Data Definition Language (aka schema changes)
DML = Data Manipulation Language (data value changes)
DCL = Data Control Language

Real-Time Database Monitoring

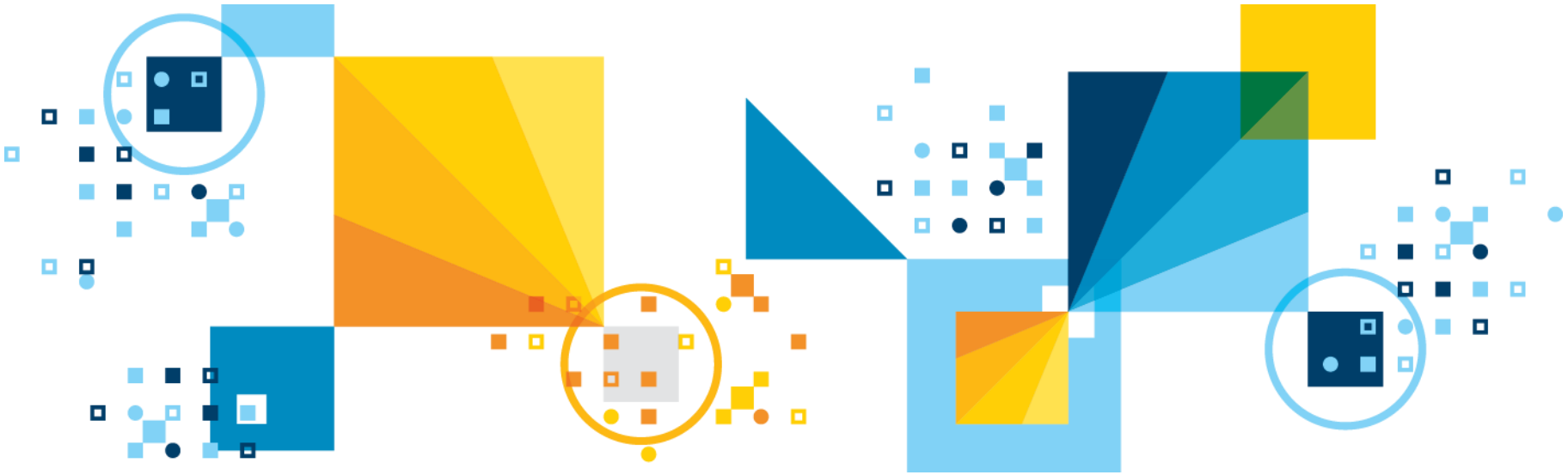


- Non-invasive architecture
  - Outside database
  - Minimal performance impact (3 - 5%)
  - No DBMS or application changes
- Cross-DBMS solution
- 100% visibility including local DBA access
- Enforces separation of duties
- Does not rely on DBMS-resident logs that can easily be erased by attackers, rogue insiders
- Granular, real-time policies & auditing
  - *Who, what, when, how*
- Automated compliance reporting, sign-offs & escalations (SOX, PCI, NIST, etc.)

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Informix and Guardium – ifxguard



ifxguard – What does it do and what is "new"

- **Utility located in $INFORMXIDIR/bin, allows Informix and Guardium to have enhanced capabilities with each other**
  - Utility must be active for the two to talk to each other
- **Loads Guardium shared library binaries to monitor client to server network communication buffers**
  - Can load 3rd party as well, only Guardium presently
- **Dynamic reloading for ifxguard configuration value changes**
- **SQLI and DRDA based clients**
- **Unix/Linux based network protocols supported currently by Informix**
- **Part of Informix installation**
  - Informix server supports security restrictions with the new utility

ifxguard

- **When executed with valid configuration parameters which specifies the shared library path, network protocols monitored, and other attributes.**

- **The Communications Buffer Monitoring Support (CBMS) feature of Guardium allows it to integrate with a database server and analyze the database client-server communication buffers in clear text.**

- **Multiple thread utility application, it can connect to Informix server with IPCSTR, SOCTCP, TLITCP, or SOCSSL protocols.**
  - **IPCSHM** does not natively support threading, works only as a single thread, we will fork a child process for **IPCSHM** instead of thread.
  - Still works though.

ifxguard Operations

- **The functions will be passed the buffer(s) that have been received from the client or are about to be sent to the client.**
- **Allows the CBMS library to examine the communications between Informix and its clients and perform whatever auditing is necessary.**
- **Terminating connections should the buffer contain malicious or unauthorized data is new, but only the DBSA authority can launch this command line utility**
  - Previously, Guardium could only monitor data within Informix
- **This utility will now also support Guardium's version 10 ability to perform data scrubbing (for example, taking a social security or drivers license number and rendering it harmless to human view).**
- **Encrypted data communications via CSM work as well with ifxguard.**

ifxguard Work Flow - Generally

- **Successfully executable by Informix from the Informix server.**

- **The libraries utilized as part of the utility binary install are provided by the Guardium product and attach to Guardium shared memory.**

- **Multiple worker threads are created**
  - Each worker thread connects to Informix server with
    - **IPCSHM ***
    - **IPCSTR ****
    - **SOCTCP** *
    - **SOCSSL** *
    - **DRDA**
  - Worker threads wait for a message from Informix server.
  - Informix server finds one **ifxguard** pipe and passes a client buffer.
  - **ifxguard** worker thread reads a buffer and sends it to Guardium.
  - Worker thread send ack message from Guardium to Informix server.

* Tested

** Tested by Guardium

Purpose, Install Directory

- **Primarily intended for Guardium v10 editions and has tighter integration between Guardium V10 and Informix 12.10.xC6 and higher**

- **Includes full STAP support for Informix**

- **If Guardium requests to kill a session, Informix will onmode –z it**

- **Installation directory:**

```
informix@informixva:/opt/IBM/informix> onstat -

IBM Informix Dynamic Server Version 12.10.FC4W1 -- On-Line -- Up 00:07:28 -- 180956 Kbytes

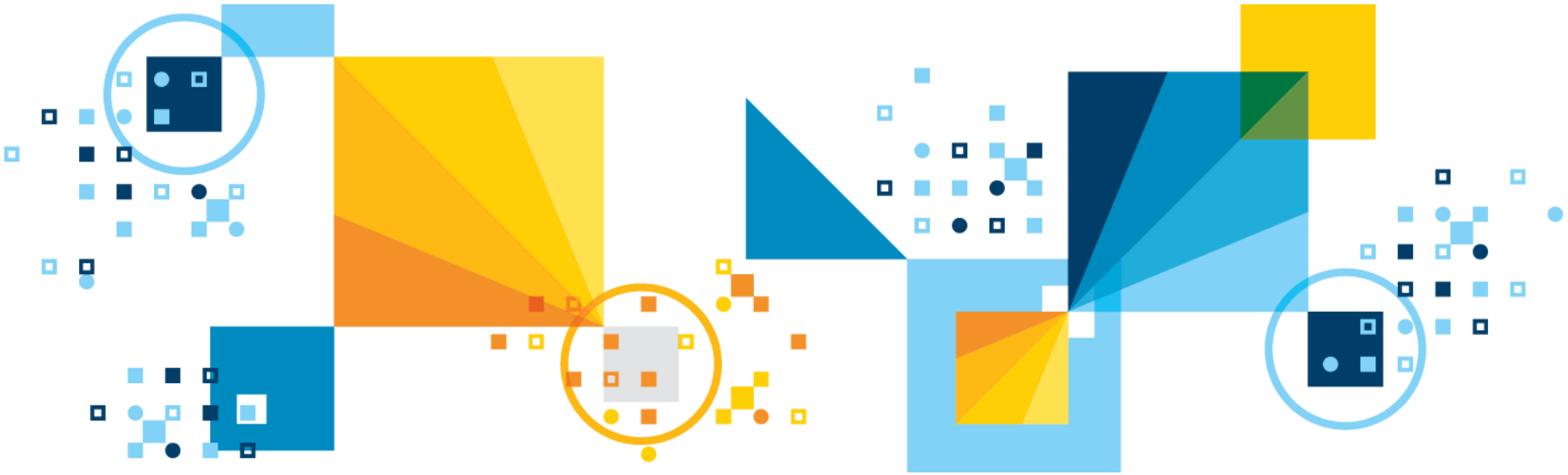
informix@informixva:/opt/IBM/informix> echo $INFORMIXDIR/bin/ifxguard
/opt/IBM/informix/bin/ifxguard
informix@informixva:/opt/IBM/informix>
```

- **Has a configuration file of its own.**
  - Some specific configuration instructions in Appendix A

Questions

Scott Pickett – WW Informix Technical Sales
June 27, 2017

Informix and Guardium - Configuration Libraries



Informix and Guardium

- This is a brief intro to Guardium, focussing on the Informix components involved

- This section does not show how to install Guardium itself; you are referenced here for those instructions:

  http://www.ibm.com/support/knowledgecenter/SSMPHH_10.0.0/com.ibm.guardium.doc.install/install/software_appliance_installation_guide.html?view=embed

- An assumption is made, however, that Guardium has been installed prior to using these instructions operationally

- Next two slides have a little bit of the terminology used in the Guardium documentation, and elsewhere in this presentation, for understanding purposes

Guardium A-TAP vs S-TAP vs K-TAP vs FS-TAP (1)

- **S-TAP - is a lightweight software daemon agent installed on a database server system that monitors database traffic and forwards information about that traffic to a Guardium system**

- **A-TAP - is a lightweight software agent installed on a database server system that monitors communications between internal components of the database server:**
  - Some traffic can only be tapped at the database server application level
  - This may be required because the DBMS uses its own encryption, or because of other internal database implementation details
  - For these cases, the A-TAP (application-level tapping) mechanism monitors communication between internal components of the database server
  - A-TAP uses K-TAP (next slide) as a proxy to pass data to S-TAP, and A-TAP must be configured separately for each database environment

Guardium A-TAP vs S-TAP vs K-TAP vs FS-TAP (2)

- **K-TAP - kernel module that performs interception in the kernel; is the recommended mechanism to collect local and network traffic on a UNIX database server. Observes access to a database server by hooking or tapping the mechanisms used to communicate between the database client and server**

- **FS-TAP - is a lightweight software agent installed on a server that monitors file system traffic and usage and forwards information about that traffic to a Guardium collector system**

Guardium and Informix libraries

- **A special shared library called Informix Exit is part of the Guardium Unix S-TAP installation, loaded at runtime by *ifxguard*. Currently 32 bit and 64 bit .so are available. Static libraries are inclusive as well.**

- **Located under: <*guardium_installation_directory*>/guard_stap**
  - eg **/usr/local/guardium/guard_stap**

    /usr/local/guardium/guard_stap/libguard_informix_exit_32.so
    /usr/local/guardium/guard_stap/libguard_informix_exit_64.so

- **Informix Exit allows Guardium v10 to audit the network protocols of Informix SQL activities:**
  - Includes TCP, Shared Memory and Named Pipe protocols.

- **There is no limit on Informix Exit.**
  - It can support all Guardium features:
    - (S-gate, UID chain, Redaction, query-rewrite, etc).

Informix EXIT with UNIX based S-TAP (Informix 12.1 & above)

- **Informix EXIT supports firewall and UID chain.**
- **Instructions for configuring**
  - Initial Setup - add **db** user to **guardium group**: **/usr/local/guardium/bin/guardctl authorize-user informix**
  - Set up Informix - as user Informix:
    - Copy correct informix exit library from **guard_stap** directory to the informix **lib** directory:
      cp /usr/local/guardium/guard_stap/libguard_informix_exit_64.so ~/lib
  - As user **informix**: bring up **ifxguard** –
    - If **$INFORMIXDIR/etc/ifxguard.$INFORMIXSERVER** file exists and **LIBPATH** is set correctly, then just run **ifxguard**. If not, (e.g. starting **ifxguard** with a 64bit library)
      - **ifxguard -p $HOME/lib/libguard_informix_exit_64.so -l /tmp/logfile.txt**
  - Add **INFX_EXIT IE** in the **guard_tap.ini** file
  - To disable **libguard**, **ifxguard -kill $INFORMIXSERVER**

- **Starting in Informix 12.1, ifxguard is provided and is integrated with Guardium (Informix_Exit) for Informix 12.10 and above protocols**
  - It also provides an ability to support all Guardium features (S-gate, UID chain, Redaction, query-rewrite, etc)

Informix EXIT with UNIX based S-TAP (Informix 12.1 & above)

- When changing the Guardium tap_identifier in its inspection engine, in order for the change to take effect with Informix exit, the database will have to be restarted.
- With ATAP enabled, the database will have to be stopped, ATAP deactivated, reactivated, and finally the database started again.
- To make tap_identifier work for Informix EXIT, make sure db_install_dir is exactly the same with $HOME value in the database.
- The database needs to restart to pick up the new tap_identifier value. For Informix exit,
  - Stop **ifxguard**
  - Restart the database
  - Start **ifxguard**.

Guardium and Informix libraries

- **The Linux platform is a special case where you can use Informix EXIT to replace Informix ATAP to capture shared memory traffic.**
- **You can still capture Informix 12.10 through KTAP by setting db type to Informix in Guardium.**
- **If multiple Informix instances exist in the same database host (eg.. IDS 11.70 and IDS 12.10), you only need either Informix EXIT or Informix KTAP.**
  - Another inspection engine is not needed for Informix KTAP.
- **On installing Informix patches or OS fix packs, it is recommended to stop the ifxguard agent first (using -kill $INFORMIXSERVER).**

Informix EXIT Configuration Instructions

1. **Login as user informix to IDS 12.10 and locate:**
   Its instance name (**echo $INFORMIXSERVER**)
   Installation directory (**echo $INFORMIXDIR**).
2. **Install and start up S-TAP in the db host**

3. **As user root, make sure user informix is in group guardium.**

   You can add user from unix:

   # chgroup users=informix guardium (AIX only)

   Or add user using guardctl:

   # /usr/local/guardium/bin/guardctl authorize-user informix
4. **login as user informix**

   $ id
   uid=501(informix) gid=205(informix) groups=215(guardium)

Informix EXIT Configuration Instructions (as user informix)

5. copy Informix Exit .so file from STAP directory to Informix library path $INFORMIXDIR/lib

$ cp /usr/local/guardium/guard_stap/libguard_informix_exit_64.so $INFORMIXDIR/lib/libguard_informix.so

6. Setup ifxguard
Create a config file under
$INFORMIXDIR/etc/ifxguard.$INFORMIXSERVER .. etc.

```
NAME        ol_informix1210
WORKERS     2
LIBPATH     /home/informix/12.10.FC6/lib/libguard_informix.so
DEBUG       1
LOGFILE     /home/informix/12.10.FC6/etc/ifxguard.msg.txtg.txt
```

Informix EXIT Configuration Instructions (as user informix)

7.

```
$ id
uid=501(informix) gid=205(informix) groups=215(guardium)

$ onstat -

IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line
-- Up 6 days 00:22:25 -- 253104 Kbytes
```

If the ifxguard config file is setup per step 6, start ifxguard this way:

```
$ ifxguard

15:20:17 ifxguard set instance name ol_informix1210
Starting ifxguard ol_informix1210 ...
check log file: /home/informix/12.10.FC6/etc/ifxguard.msg.txt
```

Informix EXIT Configuration Instructions

You should not see any error. In case of error, check file indicated in LOGFILE.

If the ifxguard config file is stored not under $INFORMIXDIR/etc, specify the file's full path with -c option: - for example
$ ifxguard -c /mnt/conf/ifxguard.ol_informix1210

If the ifxguard config file is not set up at all, you can still bring up the agent but must specify the .so library using full-path with -p option and message log file with -l option:

Example

$ ifxguard -p /home/informix/12.10.FC6/lib/libguard_informix.so –l home/informix/12.10.FC6/etc/ifxguard.msg.txt

Informix EXIT Configuration Instructions

8. Make sure ifxguard and S-TAP is up running using ps -ef.

$ ps -ef|grep guard

root 15401210 1 1 15:14:11 - 0:00
/usr/local/guardium/guard_stap/guard_stap
/usr/local/guardium/guard_stap/guard_tap.ini
informix 22609968 1 0 15:20:17 - 0:00 ifxguard

- **The speaker notes section below contains relevant messages which should be similar to what you might experience**

Guardium Configuration Instructions for Informix-EXIT

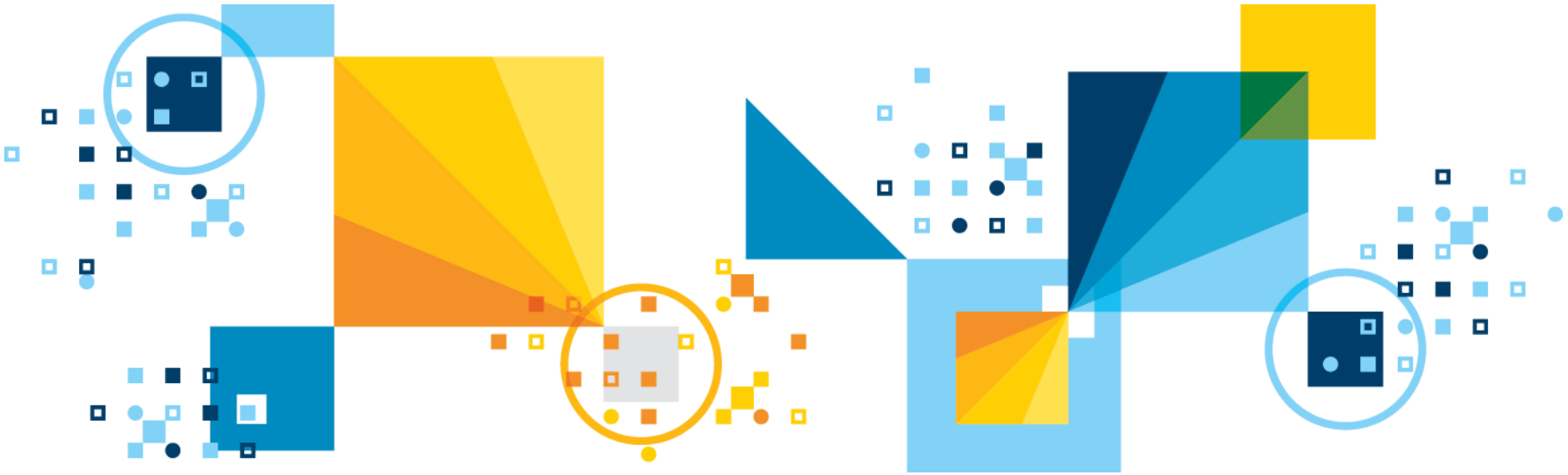
- **9. Setup INFX_EXIT inspection engine per the following example**
  - In Guardium,
    - Go to `GUI`, click `Manage`-> `Activity Monitoring`->`S-TAP Control`:
    - Look for STAP host IP,
  - Click `Modify` to add inspection engine, and the following settings:
    - Protocol: **Informix Exit**
    - DB Install Dir: **/home/informix**
    - Process Name: **/INFORMIXTMP/.inf.sqlexec**
    - Intercept Types: <blank or null>
    - Identifier: <blank or null>
  - Click `Apply`
  - Then click the `Send Command` icon, choose `Restart STAP`.

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Informix and Guardium – Configuration



ifxguard Configuration File (1)

- **Default ifxguard configuration file path is: $INFORMIXDIR/etc/ifxguard.$INFORMIXSERVER**
  - One of these per server connection type, named accordingly for ease of use
- **Contents:**

  NAME < instance-name for a single/multiple instance on one installation >

  WORKERS < number of worker threads to do the job >

  LIBPATH < library file name for buffer monitoring >

  e.g. **$INFORMIXDIR/lib/libguard_informix.so**

  DEBUG < debug message level >

  LOGFILE < message file path >

- **Each WORKER thread connects to Informix server and processes regular client application data communication.**

- **The WORKERS can be configured as equal to or greater than the number of CPUVP's on the server side to avoid heavy lock situations.**

ifxguard Configuration File (2)

- The DEBUG parameter has 3 possible values:
  - 0 none – default
  - 1 light messages – errors only
  - 2 heavy messages

- All DEBUG messages go to the message log file defined by the LOGFILE parameter.

```
informix@informixva:/opt/IBM/informix/etc> cd $INFORMIXDIR/etc
informix@informixva:/opt/IBM/informix/etc> ls -l ifxg*
-rw-r--r-- 1 root root 109 2015-01-20 16:14 ifxguard.cfg
-rw-r--r-- 1 root root  48 2015-01-20 15:59 ifxguard.demo_on
informix@informixva:/opt/IBM/informix/etc> cat ifxguard.cfg
NAME    guard_kx1
LOGFILE /tmp/guard_kx1.log
DEBUG   2
WORKERS 1
LIBPATH /opt/IBM/informix/lib/cbms_test.lin64.so
informix@informixva:/opt/IBM/informix/etc>
```

ifxguard configuration file contents

- The ifxguard configuration file can be reloaded dynamically from the command line when ifxguard configuration parameters change.

ifxguard Configuration File (3)

```
informix@informixva:/opt/IBM/informix/etc> cd $INFORMIXDIR/etc
informix@informixva:/opt/IBM/informix/etc> ls -l ifxg*
-rw-r--r-- 1 root root 109 2015-01-20 16:14 ifxguard.cfg
-rw-r--r-- 1 root root  48 2015-01-20 15:59 ifxguard.demo_on
informix@informixva:/opt/IBM/informix/etc> cat ifxguard.cfg
NAME    guard_kx1
LOGFILE /tmp/guard_kx1.log
DEBUG   2
WORKERS 1
LIBPATH /opt/IBM/informix/lib/cbms_test.lin64.so
informix@informixva:/opt/IBM/informix/etc>
```

- **If user informix does not own the log file, it cannot write to it, however, the utility will still start**
  - **root** can start **ifxguard** as well, not recommended, doesn't work well.
- **LOGFILE, DEBUG, and WORKERS are dynamically reloadable:**
  - **NAME** is not dynamically reloadable …..
- **It is possible to have multiple ifxguard configuration files, named differently and configured the same/differently:**
  - These are dynamically loadable while there is another actively running, which then may or may not be shutdown depending on circumstance.

ifxguard ONCONFIG parameter (1)

- **ifxguard is controlled on the Informix server configuration file**

ifxguard enable=1|0,timeout=<n>[:<action>]

0 disable, reject **ifxguard** connection

1 enable, **default** value, **ifxguard** can connect

timeout **-1** user session waits for **ifxguard**, default value

n:action waits for **n** seconds, and then apply action

- **action:**

ignore	IDS continue
alarm	invoke an alarm
kill	kill **ifxguard** utility
shutdown	shutdown **oninit**

- **This parameter is not initially found in the default instance configuration file, it must be manually added to the configuration file to disable it as it is enabled by default.**

ifxguard ONCONFIG parameter (2)

- This is a dynamic database server configuration file parameter, you may change its value without rebooting. Possible arguments:

  onmode wf/wm:

  onmode –wf IFXGUARD="enable=0"
  onmode –wf IFXGUARD="enable=1,timeout=-1"
  onmode –wm IFXGUARD="enable=1"
  onmode –wm IFXGUARD="enable=0,timeout=-1"

Is It Really Dynamic ? – Memory and Disk test.

```
informix@informixva:/opt/IBM/informix/bin> sh -x tester.sh
+ onmode -wf IFXGUARD=enable=1,timeout=-1
Current value for IFXGUARD (enable=1,timeout=-1) was saved in config file.
+ onstat -g cfg IFXGUARD

IBM Informix Dynamic Server Version 12.10.FC4W1 -- On-Line -- Up 2 days 00:04:41 -- 181504 Kbytes

name                     current value
IFXGUARD                 enable=1,timeout=-1

+ onstat -c
+ grep IFXGUARD
IFXGUARD        enable=1,timeout=-1
+ onmode -wf IFXGUARD=enable=0,timeout=-1
Value of IFXGUARD has been changed to enable=0.
+ onstat -g cfg IFXGUARD

IBM Informix Dynamic Server Version 12.10.FC4W1 -- On-Line -- Up 2 days 00:04:42 -- 181504 Kb

name                     current value
IFXGUARD                 enable=0

+ onstat -c
+ grep IFXGUARD
IFXGUARD        enable=0
+ onmode -wm IFXGUARD=enable=1,timeout=-1
Value of IFXGUARD has been changed to enable=1,timeout=-1.
+ onstat -g cfg IFXGUARD

IBM Informix Dynamic Server Version 12.10.FC4W1 -- On-Line -- Up 2 days 00:04:43 -- 181504 Kb

name                     current value
IFXGUARD                 enable=1,timeout=-1

+ onstat -c
+ grep IFXGUARD
IFXGUARD        enable=0
+ onmode -wm IFXGUARD=enable=0,timeout=-1
Value of IFXGUARD has been changed to enable=0.
+ onstat -g cfg IFXGUARD

IBM Informix Dynamic Server Version 12.10.FC4W1 -- On-Line -- Up 2 days 00:04:44 -- 181504 Kbytes

name                     current value
IFXGUARD                 enable=0

+ onstat -c
+ grep IFXGUARD
IFXGUARD        enable=0
informix@informixva:/opt/IBM/informix/bin>
```

```
informix@informixva:/opt/IBM/informix/bin> cat tester.sh

onmode -wf IFXGUARD="enable=1,timeout=-1"
onstat -g cfg IFXGUARD
onstat -c | grep IFXGUARD
onmode -wf IFXGUARD="enable=0,timeout=-1"
onstat -g cfg IFXGUARD
onstat -c | grep IFXGUARD
onmode -wm IFXGUARD="enable=1,timeout=-1"
onstat -g cfg IFXGUARD
onstat -c | grep IFXGUARD
onmode -wm IFXGUARD="enable=0,timeout=-1"
onstat -g cfg IFXGUARD
onstat -c | grep IFXGUARD

informix@informixva:/opt/IBM/informix/bin>
```

ifxguard Calls – Command Line & Embedded App. (1)

- **ifxguard**
  - Starts **ifxguard** with default configuration file and at default location
    - **$INFORMIXDIR/etc/ifxguard.<$INFORMIXSERVER>**

- **ifxguard -p <exit file libpath> -l <log-file path> [-w <workers>]**
  - Starts **ifxguard** and generate its named configuration file
    - **libpath**
      - Library path for the agent exit files used by Guardium
      - Default is: **$HOME/lib/libguard_informix_exit_64.so**.
    - **path**
      - Fully pathed location of the log file path for the messages generated by Guardium. **<INFORMIXSERVER>_guard**
    - **workers**
      - An integer, delineates the number of worker threads

- **ifxguard -c <configuration filename>**
  - Brings up the **ifxguard** utility with a named configuration file
    - Default
      - **$INFORMIXDIR/etc/ifxguard.$INFORMIXSERVER**

ifxguard Calls – Command Line & Embedded App. (2)

- **ifxguard -r <agent-name>**
  - Reloads the configuration file for **ifxguard**
  - **agent-name**
    - The value of the **NAME** parameter within the appropriate configuration file for the relevant **ifxguard** execution
    - Changes to only the **debug** level and the **log file** are recognized during reloading

- **ifxguard –k <agent-name> – shutdown the ifxguard agent**

- **Examples:**

ifxguard -p $HOME/lib/libguard_informix_exit_64.so -l /tmp/logfile.out -w 8

ifxguard –k ifxguardium1

ifxguard –r ifxguardium1

Ifxguard –c $INFORMIXDIR/etc/ifxguard.apt1

The dbaccess session “view” from ifxguard (1)

Session Data

```
informix@informixva:...t/IBM/informix/etc
File  Edit  View  Terminal  Tabs  Help
DISPLAY: Next  Restart  Exit
Display next page of results.

---------------------- stores@demo_on ----- Press CTRL-W for Help --------

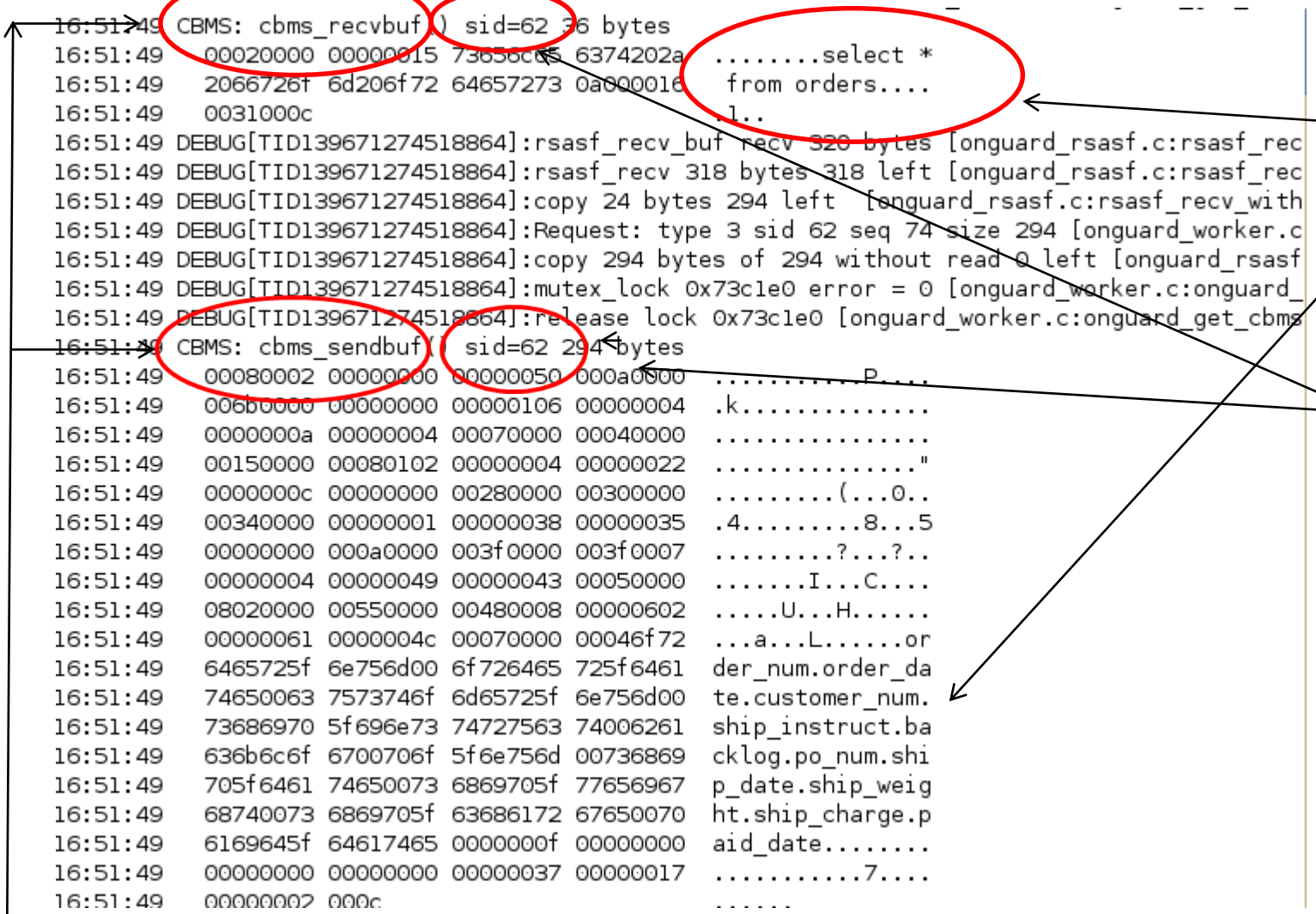

order_num       1015
order_date      06/27/2008
customer_num    110
ship_instruct   closed Mondays
backlog         n
po_num          MA003
ship_date       07/16/2008
ship_weight     20.60
ship_charge     $6.30
paid_date       08/31/2008
```

Ifxguard DEBUG=2 output

```
File  Edit  View  Terminal  Tabs  Help
16:54:27   37303920 20202000 009adcc1 5a000000  709    ..���Z...
16:54:27   c1170000 00009af1 000e0000 00000050  ?.....??.......P
16:54:27   000003fc 00009ad7 0000007b 65787072  ...?..??...{expr
16:54:27   65737320 20202020 20202020 20202020  ess
16:54:27   20202020 20202020 20202020 20202020
16:54:27   20202020 6e573232 38362020 20202000      nW2286     .
16:54:27   009adcc1 0e000000 c1083200 00009b1e  .???....?.2...?.
16:54:27   000e0000 00000050 000003fd 00009ae3  .......P...
16:54:27   0000007c 61736b20 666f7220 456c6169  ...|ask for Elai
16:54:27   6e652020 20202020 20202020 20202020  ne
16:54:27   20202020 20202020 20202020 6e433332              nC32
16:54:27   38382020 20202000 009ae5c1 28000000  88     ..???(...
16:54:27   c10c0000 00009b01 000e0000 00000050  ?.....?........P
16:54:27   000003fe 00009ae4 0000007e 65787072  ...?..??...~expr
16:54:27   65737320 20202020 20202020 20202020  ess
16:54:27   20202020 20202020 20202020 20202020
16:54:27   20202020 6e573939 32352020 20202000      nW9925     .
16:54:27   009aeac1 0f000000 c10d0000 00009b0c  .???....?.....?.
16:54:27   000e0000 00000050 000003ff 00009ae4  .......P......?
16:54:27   0000007f 6e6f2064 656c6976 65726965  ....no deliverie
16:54:27   73206166 74657220 3320702e 6d2e2020  s after 3 p.m.
16:54:27   20202020 20202020 20202020 6e4b4632              nKF2
16:54:27   39363120 20202000 009aeac1 3c000000  961    ..???<...
16:54:27   c1120000 00009b01 000f0000 00000017  ?.....?.........
16:54:27   00000117 00000000 00370000 00170000  .........7......
16:54:27   0002000c                            ....
16:54:34 DEBUG[TID139671296620304]:mutex_lock 0x73b170 error = 0 [onguard_main.c:onguard_re
16:54:34 DEBUG[TID139671296620304]:release lock 0x73b170 [onguard_main.c:onguard_reconfig_u
informix@informixva:/tmp> 16:55:34 DEBUG[TID139671296620304]:mutex_lock 0x73b170 error = 0
[onguard_main.c:onguard_reconfig_lock:666]
16:55:34 DEBUG[TID139671296620304]:release lock 0x73b170 [onguard_main.c:onguard_reconfig_u
nlock:672]
16:56:34 DEBUG[TID139671296620304]:mutex_lock 0x73b170 error = 0 [onguard_main.c:onguard_re
config_lock:666]
16:56:34 DEBUG[TID139671296620304]:release lock 0x73b170 [onguard_main.c:onguard_reconfig_u
nlock:672]
informix@informixva:/tmp> 16:57:35 DEBUG[TID139671296620304]:mutex_lock 0x73b170 error = 0
[onguard_main.c:onguard_reconfig_lock:666]
16:57:35 DEBUG[TID139671296620304]:release lock 0x73b170 [onguard_main.c:onguard_reconfig_u
nlock:672]
```

DEBUG=1 produces only error output.
DEBUG =0 produces only security errors

The dbaccess session “view” from ifxguard (2)



ifxguard Monitoring From the Database Server

- **onstat –g ath & nta report ifxguard activity.**

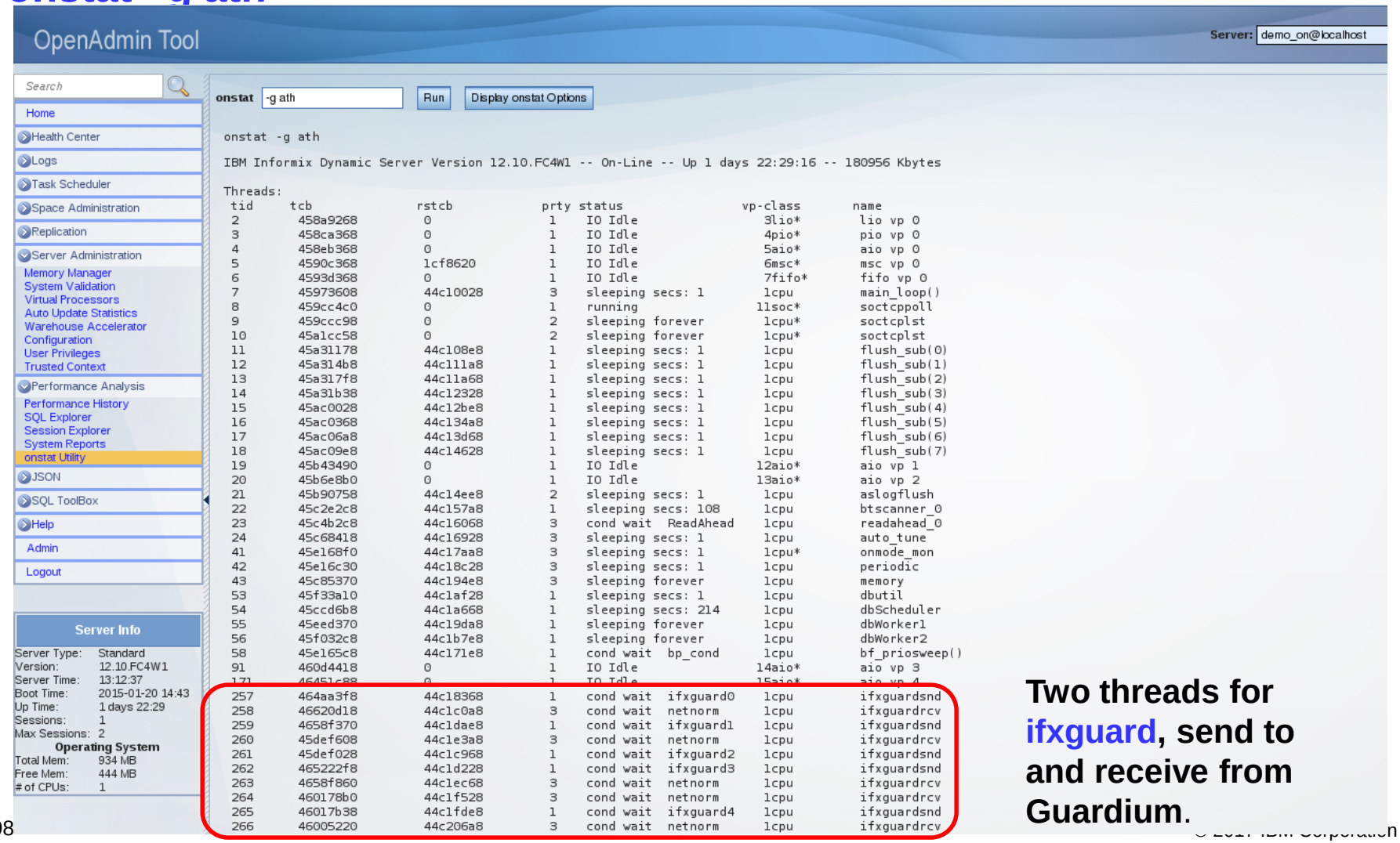
```
informix@informixva:/data/IBM/informix/data/demo_on> onstat -g ath | grep ifxguard
 86       460178b0         44c18368         1    cond wait  ifxguard0    1cpu         ifxguardsnd
 87       46017bf0         44c1c0a8         3    cond wait  netnorm      1cpu         ifxguardrcv
informix@informixva:/data/IBM/informix/data/demo_on>
```

  - Two threads for **ifxguard**:
    - Send to Guardium
    - Receive from Guardium

- **Data from Informix is sent Guardium in a continuous stream, and acknowledgments from Guardium are not waited upon to achieve better performance.**

ifxguard Monitoring From the Database Server

- onstat –g ath

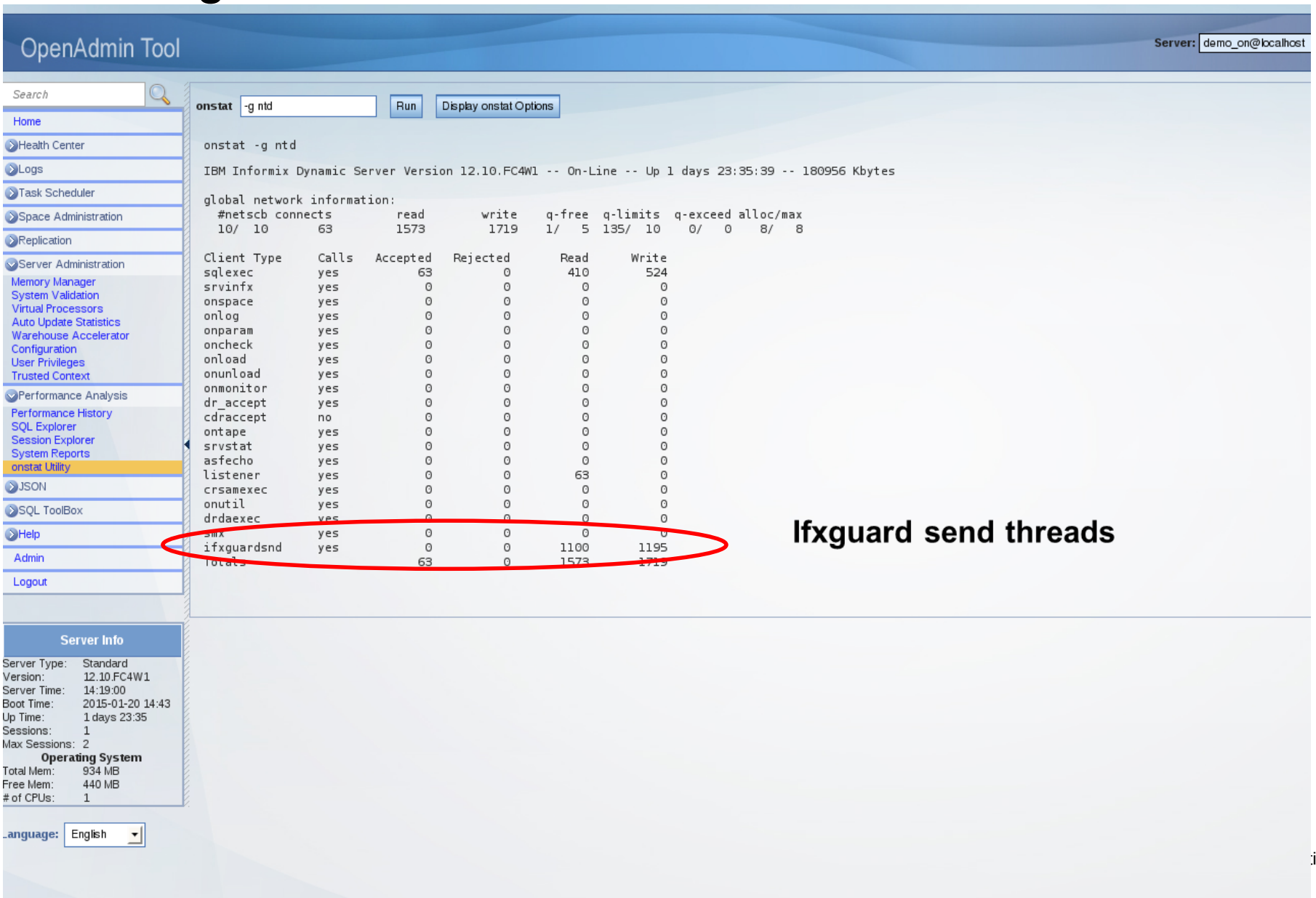


Two threads for ifxguard, send to and receive from Guardium.

onstat –g ath – Showing Encrypt Threads Used for SSL

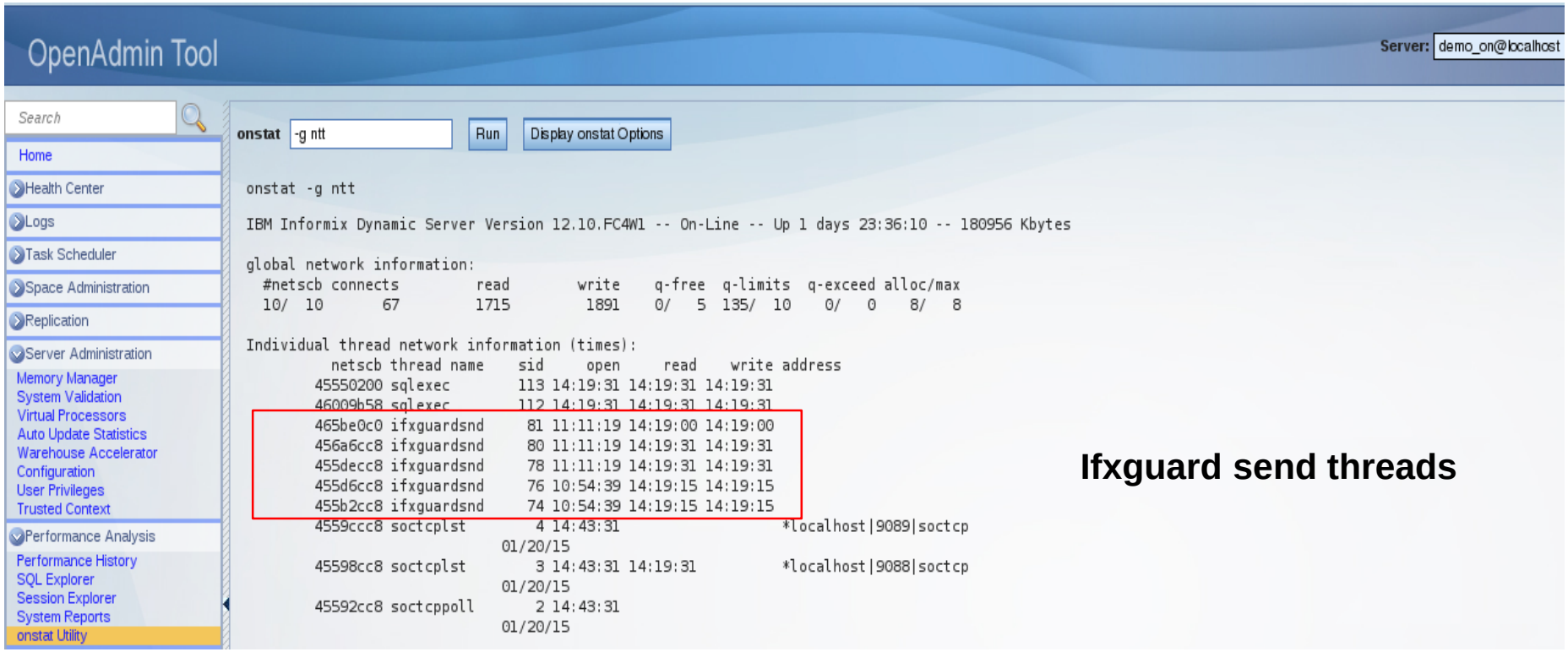
```
informix@informixva:/opt/IBM/informix>   onstat -g ath | grep ifxguard
 64       45dd5808         44c18368         1    cond wait  ifxguard0    1cpu         ifxguardsnd
 65       46078af0         44c1c0a8         1    cond wait  ifxguard1    1cpu         ifxguardsnd
 66       45ebc2c8         44c1c968         3    cond wait  netnorm      1cpu         ifxguardrcv
 67       45ebcae8         44c1d228         1    cond wait  ifxguard2    1cpu         ifxguardsnd
 68       460ee2c8         44c1dae8         3    cond wait  netnorm      1cpu         ifxguardrcv
 69       460ee9e0         44c1e3a8         3    cond wait  netnorm      1cpu         ifxguardrcv
 74       465f8028         44c1fde8         1    cond wait  ifxguard3    1cpu         ifxguardsnd
 75       4661e4c0         44c206a8         3    cond wait  netnorm      17encrypt*   ifxguardrcv
 76       4661eb48         44c20f68         1    cond wait  ifxguard4    1cpu         ifxguardsnd
 77       4667a178         44c21828         1    cond wait  ifxguard5    1cpu         ifxguardsnd
 78       466b6760         44c220e8         3    cond wait  netnorm      17encrypt*   ifxguardrcv
 79       466e0370         44c229a8         3    cond wait  netnorm      17encrypt*   ifxguardrcv
 80       466e06b0         44c23268         1    cond wait  ifxguard6    1cpu         ifxguardsnd
 81       46730d30         44c23b28         3    cond wait  netnorm      17encrypt*   ifxguardrcv
 258      46b919f8         44c243e8         1    cond wait  ifxguard7    1cpu         ifxguardsnd
 259      46a0dc60         44c1f528         1    sleeping secs: 3        1cpu         ifxguardrcv
 260      468c7418         44c24ca8         1    cond wait  ifxguard9    1cpu         ifxguardsnd
 261      45d70b50         44c25568         1    cond wait  ifxguard8    1cpu         ifxguardsnd
 262      469a6220         44c25e28         1    sleeping secs: 1        1cpu         ifxguardrcv
 263      469a68a8         44c266e8         1    sleeping secs: 1        1cpu         ifxguardrcv
informix@informixva:/opt/IBM/informix>
```

onstat –g ntd



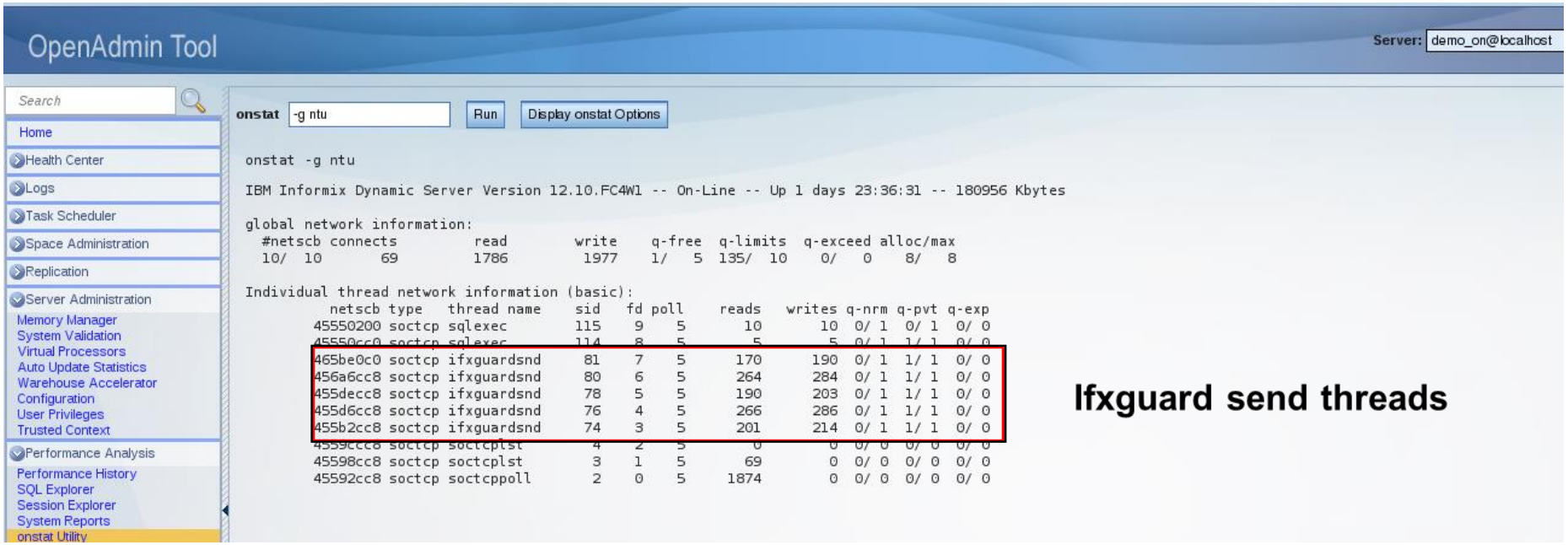
Ifxguard send threads

onstat –g ntt



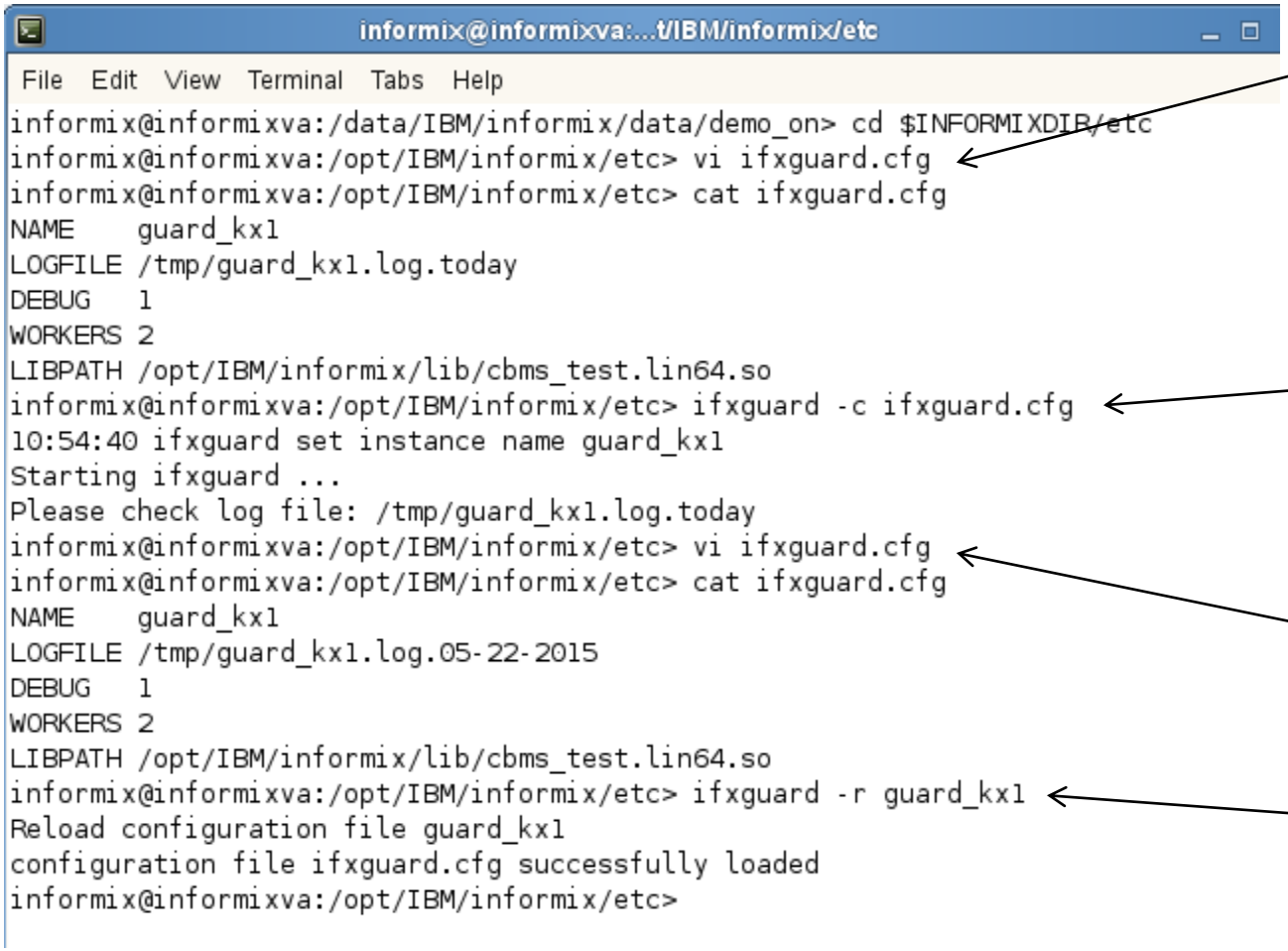
Ifxguard send threads

onstat –g ntu



Ifxguard send threads

Operations – Initialize and Dynamic Reload



- Edit ifxguard file
- Execute ifxguard with the config file
- Re-edit the same configuration file
- Dynamically reload the same configuration file

Operations – Log File Change & Second Initialization (1)

```
informix@informixva:/opt/IBM/informix/etc> cat /tmp/guard_kx1.log.today
Thu Jan 22 10:54:40 2015
10:54:40 INFORMIX-ESQL Version 12.10.FC4W1
10:54:40 Build Number:  N028
10:54:40 Build Host:    lxvm-l172
10:54:40 Build OS:      Linux 2.6.18-128.el5
10:54:40 Build Date:    Tue Sep 9 20:35:45 CDT 2014
10:54:40 GLS Version:   glslib-6.00.FC4
10:54:40
10:54:40 Starting ifxguard ...
10:54:40 DEBUG[TID140080741246736]:Password File /opt/IBM/informix/etc/passwd_file failed error:No such fi
le or directory[2] [onguard_main.c:onguard_pw_init:521]
10:54:40 DEBUG[TID140080741246736]:Ignored Password Manager, trusted host connection [onguard_main.c:ongua
rd_run:2394]
10:54:40 CBMS: cbms_initialized
10:54:40 Connect to server demo_on success process id 7480:360786256
10:54:40 ifxguard started successfully
10:54:40 DEBUG[TID140080741246736]:new daemon pid is 7480 [onguard_main.c:onguard_daemonize:2353]
10:54:40 CBMS: cbms_register() svr=0 sid=76 user=informix pid=0 prog=
10:54:40 CBMS: cbms_recvbuf() sid=76 338 bytes
10:54:40 Connect to server demo_on success process id 7480:358685008
informix@informixva:/opt/IBM/informix/etc> ls -l /tmp/guard_kx1.log.05-22-2015
-rw-r--r-- 1 informix informix 325 2015-01-22 11:00 /tmp/guard_kx1.log.05-22-2015
informix@informixva:/opt/IBM/informix/etc> cat /tmp/guard_kx1.log.05-22-2015
Thu Jan 22 11:00:28 2015
11:00:28 INFORMIX-ESQL Version 12.10.FC4W1
11:00:28 Build Number:  N028
11:00:28 Build Host:    lxvm-l172
11:00:28 Build OS:      Linux 2.6.18-128.el5
11:00:28 Build Date:    Tue Sep 9 20:35:45 CDT 2014
11:00:28 GLS Version:   glslib-6.00.FC4
11:00:28
11:00:28 configuration file ifxguard.cfg successfully loaded
informix@informixva:/opt/IBM/informix/etc> cp ifxguard.cfg ifxguard.demo_on
informix@informixva:/opt/IBM/informix/etc> vi ifxguard.demo_on
informix@informixva:/opt/IBM/informix/etc> ifxguard -c ifxguard.demo_on
11:11:19 ifxguard set instance name guard_kx2
Starting ifxguard ...
Please check log file: /tmp/guard_kx2.log.05-22-2015
informix@informixva:/opt/IBM/informix/etc>
```

- **Display the original log file.**
- **Say I don't like the logfile name.**
- **Change it, create a new config**
- **Initialize a 2nd Configuration, leaving the first up, messages show 2nd instance.**

ifxguard - Shared Memory Operations

```
informix@informixva:~
File  Edit  View  Terminal  Tabs  Help
informix@informixva:~> export INFORMIXSERVER=demo_shm
informix@informixva:~> cat $INFORMIXSQLHOSTS

oltp                onsoctcp    *localhost    1600
webapp              onsoctcp    *localhost    1601
report              onsoctcp    *localhost    1602

dsjdbc              drsoctcp    *localhost    9084

demo_on      onsoctcp *localhost 9088
demo_ondrda  drsoctcp *localhost 9081
demo_shm     onipcshm *localhost demoshm
demo_ssl     onsocssl *localhost 9089
informix@informixva:~> dbaccess stores
```

INFORMIXSERVER=demo_shm

```
SHM
File  Edit  View  Terminal  Tabs  Help
SQL:   New  Run  Modify  Use-editor  Output  Choose  Save  Info  Drop  Exit
Run the current SQL statements.

---------------------- stores@demo_shm -------- Press CTRL-W for Help --------

revoke select, delete, insert, update on orders from public;
grant select, delete, insert, update on orders to igward;
revoke select on orders from igward;
```

```
SHM - igward
File  Edit  View  Terminal  Tabs  Help
SQL:  New  Run  Modify  Use-editor  Output
Modify the current SQL statements using the

---------------------- stores@demo_shm ----

select * from orders

272: No SELECT permission for orders.
```

ifxguard in DEBUG Mode 2 – Connection LOGFILE

```
15:57:03 CBMS: cbms_recvbuf() sid=96 8 bytes
15:57:03   00040000 0007000c                        ........
15:57:03 DEBUG[TID140491292498256]:rsasf_recv_buf recv 54 bytes [onguard_rsasf.c:rsasf_recv_buf:899]
15:57:03 DEBUG[TID140491292498256]:rsasf_recv 52 bytes 52 left [onguard_rsasf.c:rsasf_recv_with_timeout:1837]
15:57:03 DEBUG[TID140491292498256]:copy 24 bytes 28 left  [onguard_rsasf.c:rsasf_recv_with_timeout:1864]
15:57:03 DEBUG[TID140491292498256]:Request: type 3 sid 96 seq 10646 size 28 [onguard_worker.c:onguard_worker:641]
15:57:03 DEBUG[TID140491292498256]:copy 28 bytes of 28 without read 0 left [onguard_rsasf.c:rsasf_recv_with_timeout:1752]
15:57:03 DEBUG[TID140491292498256]:mutex_lock 0x73c840 error = 0 [onguard_worker.c:onguard_get_cbmsctx:91]
15:57:03 DEBUG[TID140491292498256]:release lock 0x73c840 [onguard_worker.c:onguard_get_cbmsctx:101]
15:57:03 CBMS: cbms_sendbuf() sid=96 28 bytes
15:57:03   000f0015 00000000 00000000 00000000  ................
15:57:03   00370000 00010000 0001000c           .7..........
15:57:16 DEBUG[TID140491292498256]:rsasf_recv_buf recv 62 bytes [onguard_rsasf.c:rsasf_recv_buf:899]
15:57:16 DEBUG[TID140491292498256]:rsasf_recv 60 bytes 60 left [onguard_rsasf.c:rsasf_recv_with_timeout:1837]
15:57:16 DEBUG[TID140491292498256]:copy 24 bytes 36 left  [onguard_rsasf.c:rsasf_recv_with_timeout:1864]
15:57:16 DEBUG[TID140491292498256]:Request: type 4 sid 96 seq 10647 size 36 [onguard_worker.c:onguard_worker:641]
15:57:16 DEBUG[TID140491292498256]:copy 36 bytes of 36 without read 0 left [onguard_rsasf.c:rsasf_recv_with_timeout:1752]
15:57:16 DEBUG[TID140491292498256]:mutex_lock 0x73c840 error = 0 [onguard_worker.c:onguard_get_cbmsctx:91]
15:57:16 DEBUG[TID140491292498256]:release lock 0x73c840 [onguard_worker.c:onguard_get_cbmsctx:101]
15:57:16 CBMS: cbms_recvbuf() sid=96 36 bytes
15:57:16   00020000 00000015 73656c65 6374202a  ........select *
15:57:16   2066726f 6d206f72 64657273 0a000016   from orders....
15:57:16   0031000c                             1..
15:57:16 DEBUG[TID140491292498256]:rsasf_recv_buf recv 46 bytes [onguard_rsasf.c:rsasf_recv_buf:899]
15:57:16 DEBUG[TID140491292498256]:rsasf_recv 44 bytes 44 left [onguard_rsasf.c:rsasf_recv_with_timeout:1837]
15:57:16 DEBUG[TID140491292498256]:copy 24 bytes 20 left  [onguard_rsasf.c:rsasf_recv_with_timeout:1864]
15:57:16 DEBUG[TID140491292498256]:Request: type 3 sid 96 seq 10648 size 20 [onguard_worker.c:onguard_worker:641]
15:57:16 DEBUG[TID140491292498256]:copy 20 bytes of 20 without read 0 left [onguard_rsasf.c:rsasf_recv_with_timeout:1752]
15:57:16 DEBUG[TID140491292498256]:mutex_lock 0x73c840 error = 0 [onguard_worker.c:onguard_get_cbmsctx:91]
15:57:16 DEBUG[TID140491292498256]:release lock 0x73c840 [onguard_worker.c:onguard_get_cbmsctx:101]
15:57:16 CBMS: cbms_sendbuf() sid=96 20 bytes
15:57:16   000dfef0 00000000 00150006 6f726465  ..þð........orde
15:57:16   7273000c                             rs..
15:58:02 DEBUG[TID140491338114832]:mutex_lock 0x73b170 error = 0 [onguard_main.c:onguard_reconfig_lock:666]
15:58:02 DEBUG[TID140491338114832]:release lock 0x73b170 [onguard_main.c:onguard_reconfig_unlock:672]
15:59:02 DEBUG[TID140491338114832]:mutex_lock 0x73b170 error = 0 [onguard_main.c:onguard_reconfig_lock:666]
15:59:02 DEBUG[TID140491338114832]:release lock 0x73b170 [onguard_main.c:onguard_reconfig_unlock:672]
16:00:03 DEBUG[TID140491338114832]:mutex_lock 0x73b170 error = 0 [onguard_main.c:onguard_reconfig_lock:666]
16:00:03 DEBUG[TID140491338114832]:release lock 0x73b170 [onguard_main.c:onguard_reconfig_unlock:672]
16:01:04 DEBUG[TID140491338114832]:mutex_lock 0x73b170 error = 0 [onguard_main.c:onguard_reconfig_lock:666]
16:01:04 DEBUG[TID140491338114832]:release lock 0x73b170 [onguard_main.c:onguard_reconfig_unlock:672]
```

Operations – Log File Change & Second Initialization (2)

```
informix@informixva:/opt/IBM/informix/etc> ifxguard -k guard_kx2
Shut down ifxguard guard_kx2
informix@informixva:/opt/IBM/informix/etc> onstat -g ath | grep ifxguard
 257      464aa3f8         44c18368         1    cond wait  ifxguard0    1cpu         ifxguardsnd
 258      46620d18         44c1c0a8         3    cond wait  netnorm      1cpu         ifxguardrcv
 259      4658f370         44c1dae8         1    cond wait  ifxguard1    1cpu         ifxguardsnd
 260      45def608         44c1e3a8         3    cond wait  netnorm      1cpu         ifxguardrcv
informix@informixva:/opt/IBM/informix/etc> ifxguard -k guard_kx1
Shut down ifxguard guard_kx1
informix@informixva:/opt/IBM/informix/etc> onstat -g ath | grep ifxguard
informix@informixva:/opt/IBM/informix/etc>
```

- **Above graphic explained:**

- **Shutdown 2nd ifxguard instance**
  - Validate that its shutdown
- **Shutdown 1st ifxguard instance**
  - Validate that its shutdown

ifxguard and Informix Caveats/Solution (1)

- You must execute ifxguard as user informix.
- A subsequent correct execution of ifxguard as user informix without killing a/all previous incorrectly executed ifxguard session(s) causes the user informix executed ifxguard session to die as well.
- kill -9 works.

ifxguard and Informix Caveats/Solution (2)

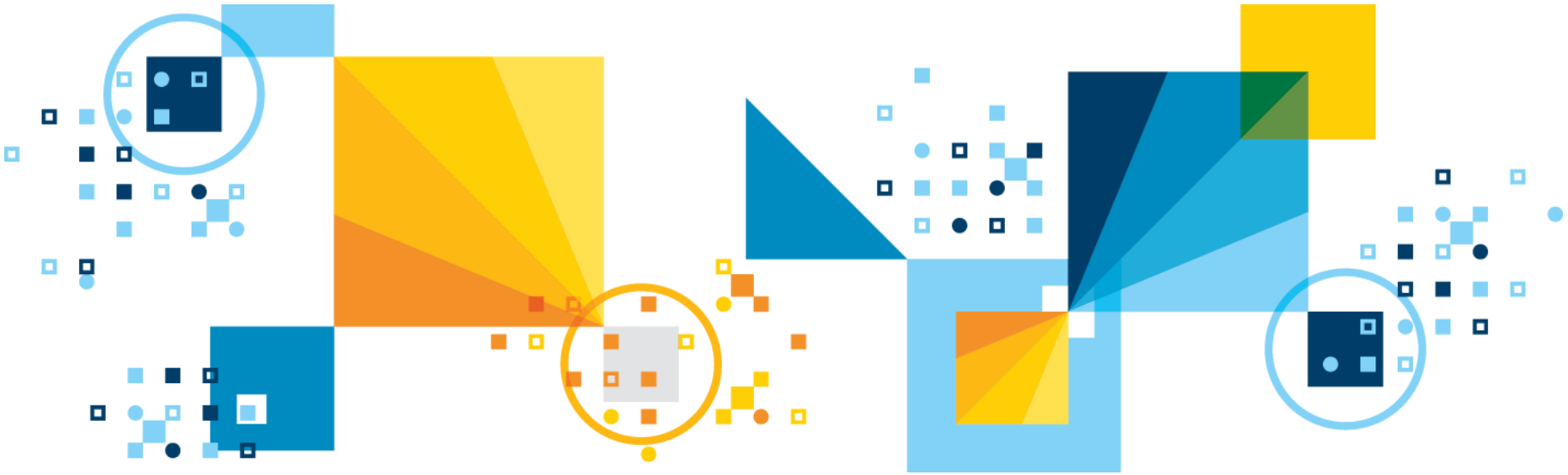
- **When multiple, parallel ifxguards are running different server connection protocols:**
  - **ifxguard –c** ‘seems’ not to work for each additional **ifxguard** executed beyond the first, but….
  - Executing **ifxguard** without an argument ‘finds’ the right **ifxguard** ‘instance’ to connect to the correct **INFORMIXSERVER**
  - **ifxguard –r** and **–k** still require the right **ifxguard** utility name to execute properly.

Questions

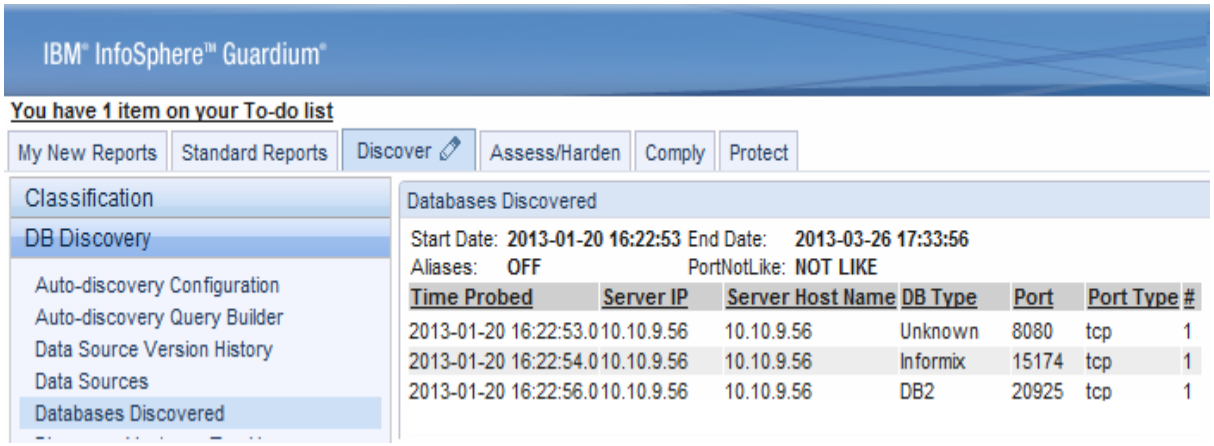
Scott Pickett – WW Informix Technical Sales

June 27, 2017

Guardium and Informix – Some Functionality

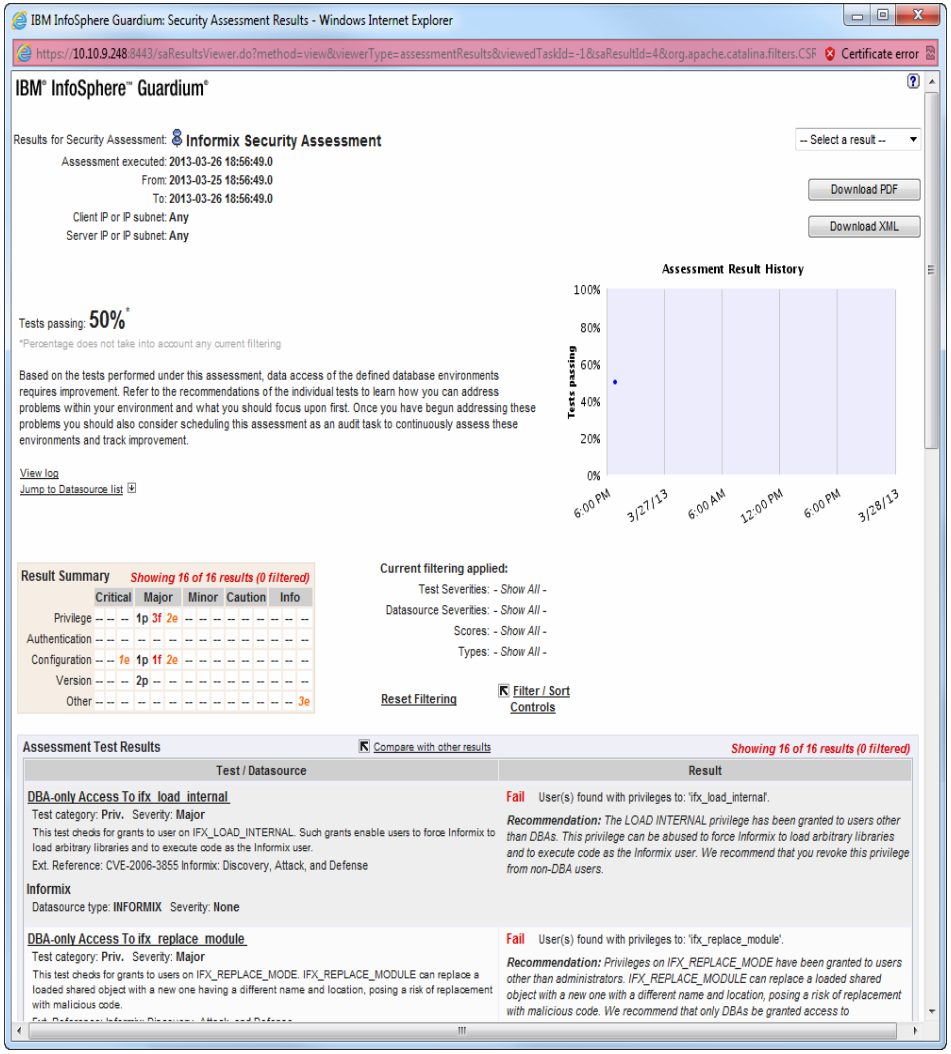


Find Uncataloged Databases and Identify Sensitive Data



- Crawls the network to find uncataloged instances
- Four algorithms to identify sensitive data in databases
- Policy-based responsive actions
  - Alerts
  - Add to group of sensitive objects

Harden Databases



Current Test Results

Prioritized Breakdown

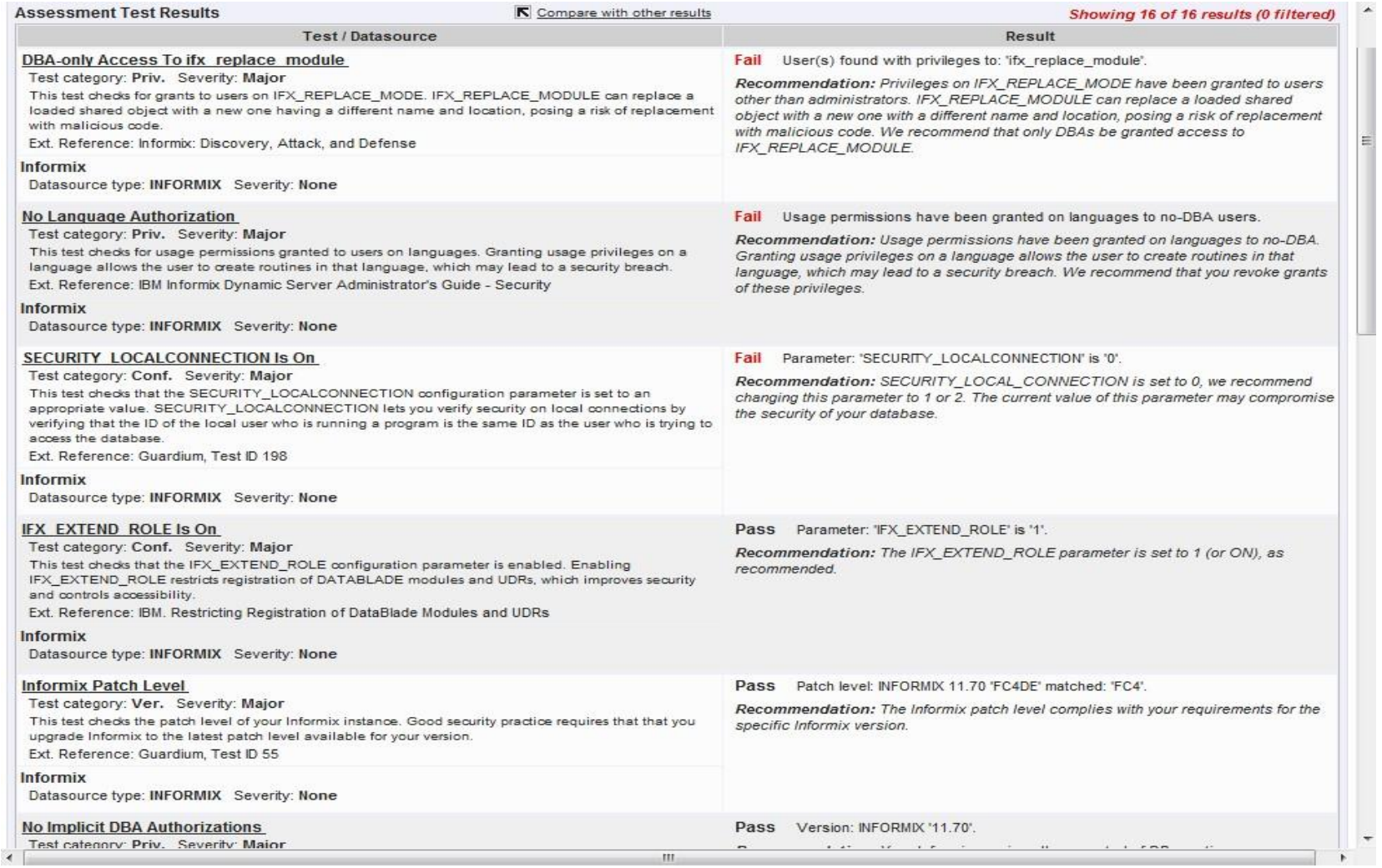
Detailed Test Results

Result History

Filters and Sort Controls

Detailed Remediation Suggestions

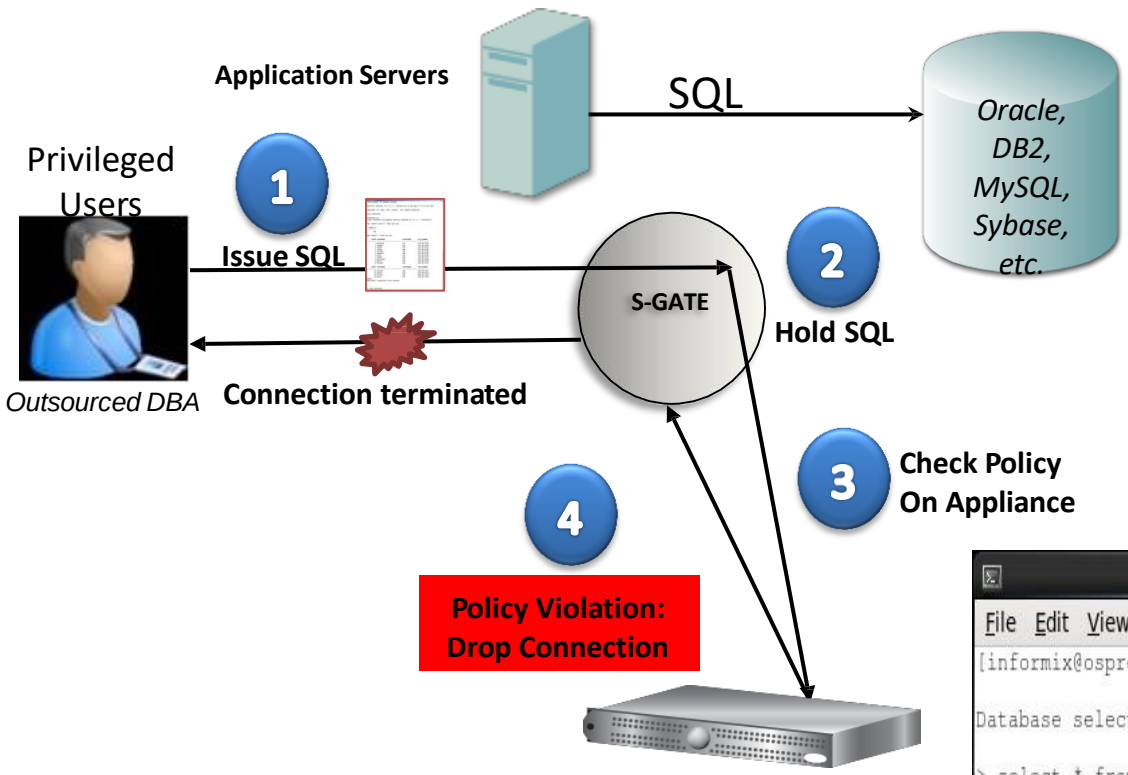
Eliminate Inappropriate Privileges



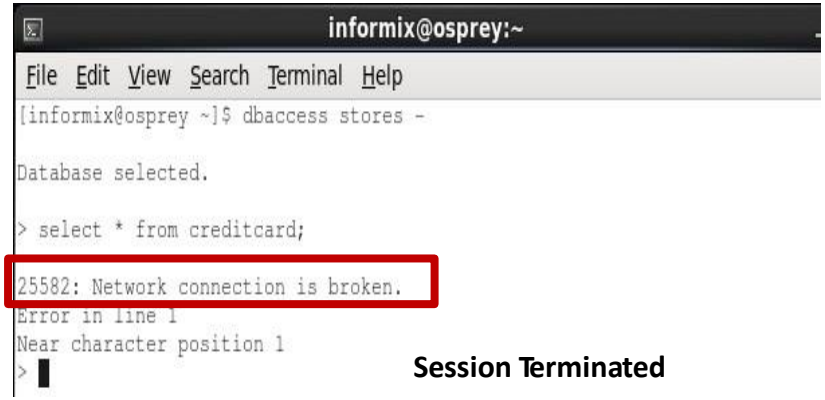
Assessment Test Results — Compare with other results — *Showing 16 of 16 results (0 filtered)*

Test / Datasource	Result
DBA-only Access To ifx_replace_module Test category: **Priv.** Severity: **Major** This test checks for grants to users on IFX_REPLACE_MODE. IFX_REPLACE_MODULE can replace a loaded shared object with a new one having a different name and location, posing a risk of replacement with malicious code. Ext. Reference: Informix: Discovery, Attack, and Defense **Informix** Datasource type: **INFORMIX** Severity: **None**	**Fail** User(s) found with privileges to: 'ifx_replace_module'. ***Recommendation:*** *Privileges on IFX_REPLACE_MODE have been granted to users other than administrators. IFX_REPLACE_MODULE can replace a loaded shared object with a new one with a different name and location, posing a risk of replacement with malicious code. We recommend that only DBAs be granted access to IFX_REPLACE_MODULE.*
No Language Authorization Test category: **Priv.** Severity: **Major** This test checks for usage permissions granted to users on languages. Granting usage privileges on a language allows the user to create routines in that language, which may lead to a security breach. Ext. Reference: IBM Informix Dynamic Server Administrator's Guide - Security **Informix** Datasource type: **INFORMIX** Severity: **None**	**Fail** Usage permissions have been granted on languages to no-DBA users. ***Recommendation:*** *Usage permissions have been granted on languages to no-DBA. Granting usage privileges on a language allows the user to create routines in that language, which may lead to a security breach. We recommend that you revoke grants of these privileges.*
SECURITY_LOCALCONNECTION Is On Test category: **Conf.** Severity: **Major** This test checks that the SECURITY_LOCALCONNECTION configuration parameter is set to an appropriate value. SECURITY_LOCALCONNECTION lets you verify security on local connections by verifying that the ID of the local user who is running a program is the same ID as the user who is trying to access the database. Ext. Reference: Guardium, Test ID 198 **Informix** Datasource type: **INFORMIX** Severity: **None**	**Fail** Parameter: 'SECURITY_LOCALCONNECTION' is '0'. ***Recommendation:*** *SECURITY_LOCAL_CONNECTION is set to 0, we recommend changing this parameter to 1 or 2. The current value of this parameter may compromise the security of your database.*
IFX_EXTEND_ROLE Is On Test category: **Conf.** Severity: **Major** This test checks that the IFX_EXTEND_ROLE configuration parameter is enabled. Enabling IFX_EXTEND_ROLE restricts registration of DATABLADE modules and UDRs, which improves security and controls accessibility. Ext. Reference: IBM. Restricting Registration of DataBlade Modules and UDRs **Informix** Datasource type: **INFORMIX** Severity: **None**	**Pass** Parameter: 'IFX_EXTEND_ROLE' is '1'. ***Recommendation:*** *The IFX_EXTEND_ROLE parameter is set to 1 (or ON), as recommended.*
Informix Patch Level Test category: **Ver.** Severity: **Major** This test checks the patch level of your Informix instance. Good security practice requires that that you upgrade Informix to the latest patch level available for your version. Ext. Reference: Guardium, Test ID 55 **Informix** Datasource type: **INFORMIX** Severity: **None**	**Pass** Patch level: INFORMIX 11.70 'FC4DE' matched: 'FC4'. ***Recommendation:*** *The Informix patch level complies with your requirements for the specific Informix version.*
No Implicit DBA Authorizations Test category: **Priv.** Severity: **Major**	**Pass** Version: INFORMIX '11.70'.

Cross-DBMS, Data-Level Access Control (S-GATE)

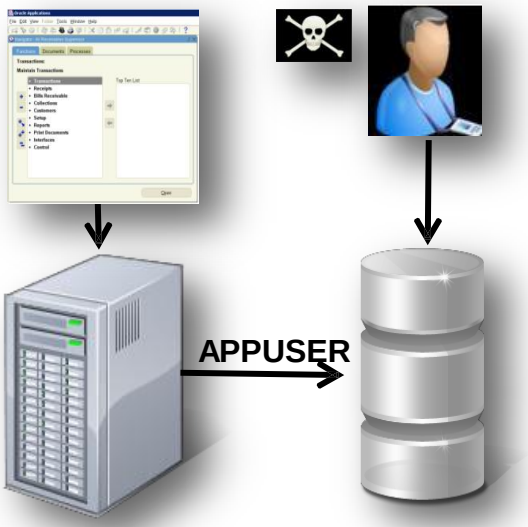


- ✓ Cross-DBMS policies
- ✓ Block privileged user actions
- ✓ No database changes
- ✓ No application changes
- ✓ Without risk of inline appliances that can interfere with application traffic



Session Terminated

A simple policy example: Preventing application bypass



Application Server 10.10.9.244

Database Server 10.10.9.56

Rule #1 Description: non-App Source AppUser Connection

Category: Security | Classification: Breach | Severity: MED

Not ☐ Server IP / and/or Group: Production Servers

Not ☑ Client IP / and/or Group: Authorized Client IPs

Not ☐ Client MAC | Net. Protocol | and/or Group

Not ☐ DB Name

Not ☐ DB User: APPUSER

Field Name

Object: EmployeeTable

Command: Select

Min. Ct.: 0 | Reset Interval (minutes): 0

Continue to next Rule ☐ | Rec. Vals. ☑

Action: ALERT PER MATCH

Notification

☒ Notification Type MAIL Mail User marc_gamache@guardium.com

Sample Alert

From: GuardiumAlert@guardium.com Sent: Wed 4/15/2009 8:00 AM

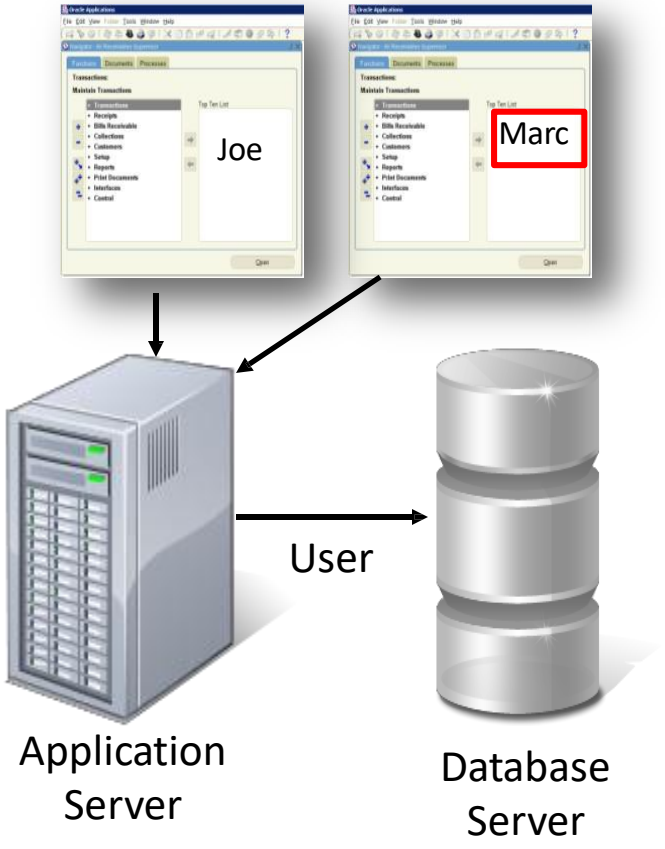
To: Marc Gamache

Cc:

Subject: (c1) SQLGUARD ALERT

Subject: (c1) SQLGUARD ALERT Alert based on rule ID non-App Source AppUser Connection
Category: security Classification: Breach Severity MED
Rule # 20267 [non-App Source AppUser Connection]
Request Info: [Session start: 2009-04-15 06:59:03 Server Type: Informix] Client IP 192.168.20.160 ServerIP: 172.16.2.152 Client PORT: 11787 Server Port: 1521 Net Protocol: TCP DB Protocol: TNS DB Protocol Version: 3.8 DB User: APPUSER
Application User Name
Source Program: JDBC THIN CLIENT Authorization Code: 1 Request Type: SQL_LANG Last Error:
SQL: select * from EmployeeTable

Identifying Fraud at the Application Layer



DB User Name	Application User	Sql
APPUSER	joe	select * from EmployeeRoleView where UserName=?
APPUSER	joe	select * from EmployeeTable
APPUSER	marc	insert into EmployeeTable values (?,?,?,?,?,?,?,?)

- ***Issue***: Application server uses generic service account to access DB
  - ***Doesn't identify who*** initiated transaction (connection pooling)
- ***Solution***: Guardium tracks access to application ***user associated with specific SQL commands***
  - Out-of-the-box support for all major enterprise applications (Oracle EBS, PeopleSoft, SAP, Siebel, Business Objects, Cognos…) and custom applications (WebSphere, WebLogic, ….)
  - Deterministic vs. time-based "best guess"
  - No changes to applications

Identify Inappropriate Use by Authorized Users

Should my customer service rep view 99 records in an hour when the average is 4?

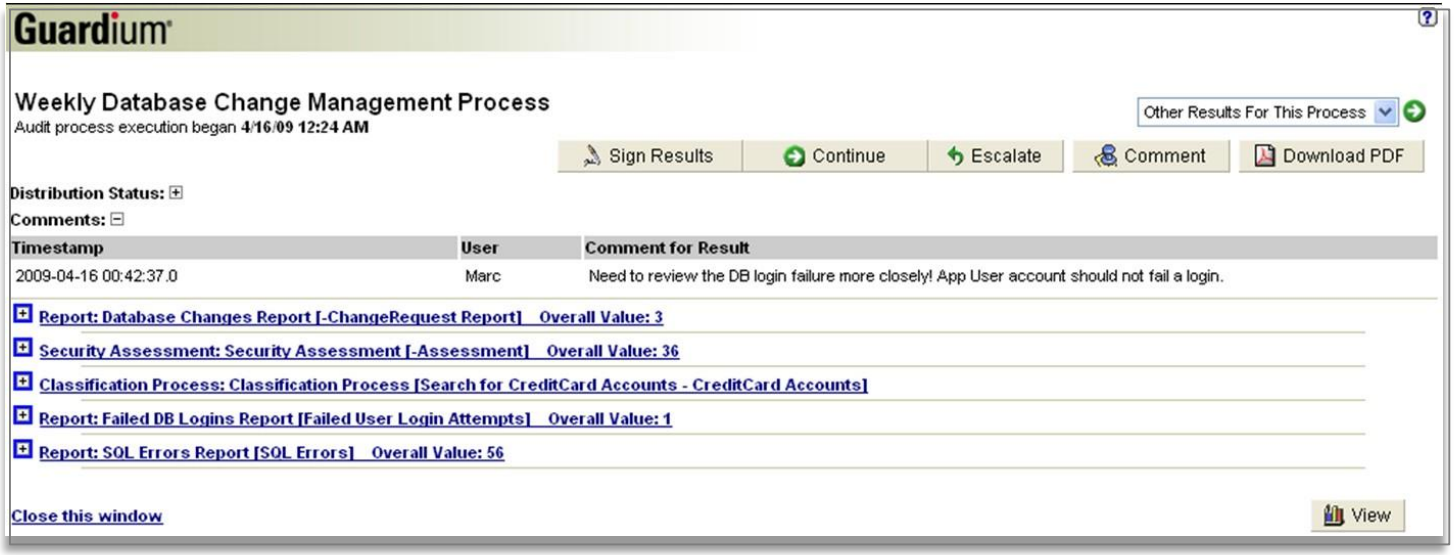
Is this normal?

DB User Name	Sql	Records
STEVE	select * from ar.creditcard where i>? and i<?	4
HARRY	select * from ar.creditcard where i<?	4
JOE	select * from ar.creditcard where i<?	99

What did he see?

HARRY	select * from ar.creditcard where i<?	**********0002, **********0003, **********0004
JOE	select * from ar.creditcard where i<?	**********0001
JOE	select * from ar.creditcard where i<?	**********0002, **********0003, **********0004, **********0005, **********0006, **********0007, **********0008, **********0009, **********0010, **********0011, **********0012, **********0013, **********0014, **********0015, **********0016
JOE	select * from ar.creditcard where i<?	**********0017, **********0018, **********0019, **********0020, **********0021, **********0022, **********0023, **********0024, **********0025, **********0026, **********0027, **********0028, **********0029, **********0030, **********0031
JOE	select * from ar.creditcard where i<?	**********0032, **********0033, **********0034, **********0035, **********0036, **********0037, **********0038, **********0039, **********0040, **********0041, **********0042, **********0043, **********0044, **********0045, **********0046
JOE	select * from ar.creditcard where i<?	**********0047, **********0048, **********0049, **********0050, **********0051, **********0052, **********0053, **********0054, **********0055, **********0056, **********0057, **********0058, **********0059, **********0060, **********0061
JOE	select * from ar.creditcard where i<?	**********0062, **********0063, **********0064, **********0065, **********0066, **********0067, **********0068, **********0069, **********0070, **********0071, **********0072, **********0073, **********0074, **********0075, **********0076
JOE	select * from ar.creditcard where i<?	**********0077, **********0078, **********0079, **********0080, **********0081, **********0082, **********0083, **********0084, **********0085, **********0086, **********0087, **********0088, **********0089, **********0090, **********0091
JOE	select * from ar.creditcard where i<?	**********0092, **********0093, **********0094, **********0095, **********0096, **********0097, **********0098, **********0099

Automating Sign-offs & Escalations for Compliance



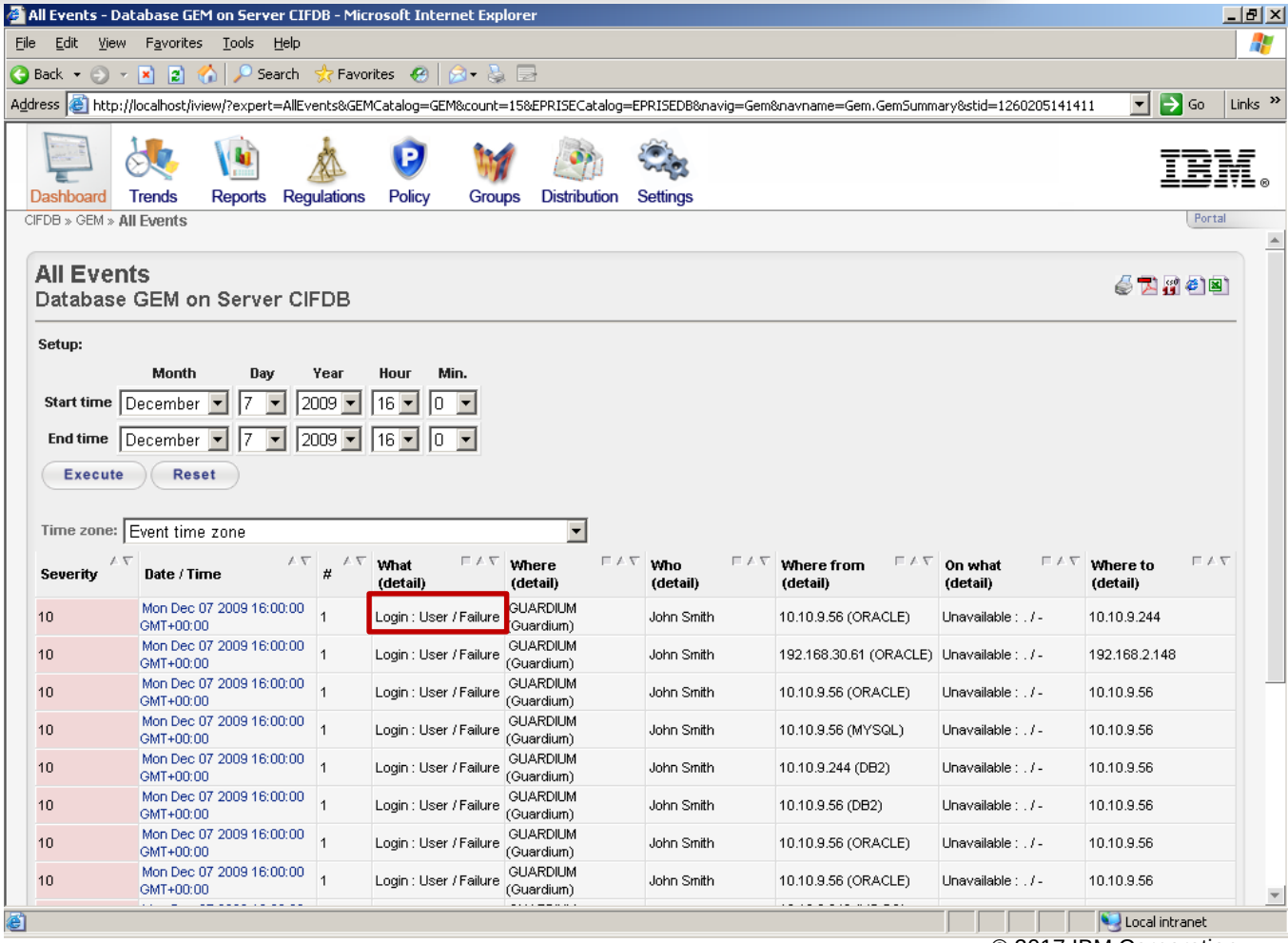
- Automates entire compliance workflow
  - Report distribution to oversight team
  - Electronic sign-offs
  - Escalations, comments & exception handling
- Addresses auditors' requirements to document oversight processes
- Results of audit process stored with audit data in secure audit repository
- Streamlines and simplifies compliance processes

Integrating with IBM TSIEM

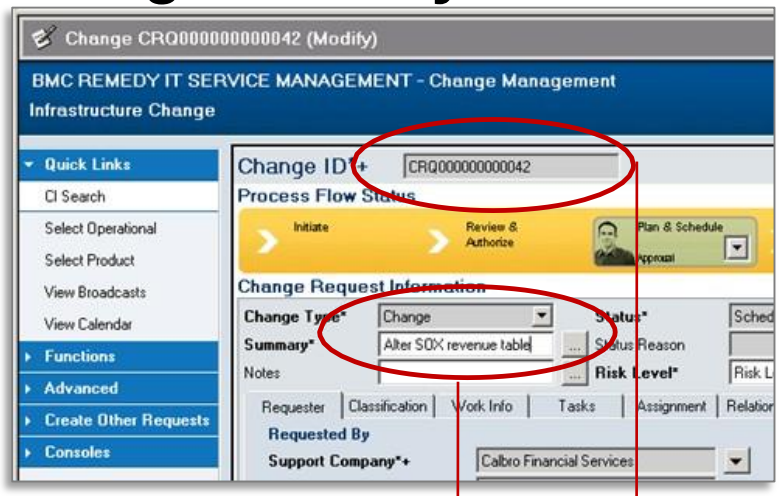
Category Name	Access Rule Description	Client IP	Server IP	DB User Name
security	Login Failures to Production Database Server	10.10.9.56	10.10.9.56	APPUSER

Policy violation in Guardium system

Events in IBM SIEM



Enforcing Change Controls + Integrating with Change Management Systems



Tag DBA actions with ticket IDs

Compare observed changes to approved changes

Identify unauthorized changes (red) or changes with invalid ticket IDs

Start Date: **2009-01-22 15:00:00** End Date: **2009-01-22 16:00:00**

Timestamp	Server Type	risk_level	priority	description	change_id	change_id_entered	Assigned To	DB User Name	Client IP	Server IP	Sql
2009-01-22 15:41:55.0		0	0			crq000000000232	allen	SYSTEM	192.168.8.129	192.168.8.129	Alter table sox_sales_international add total_rev float
2009-01-22 15:08:21.0		0	3	Alter SOX revenue table	CRQ000000000042	crq000000000042	allen	ALLEN	192.168.8.129	192.168.8.129	Alter table sox_sales_east add total_revenue float
2009-01-22 15:08:29.0		0	3	Alter SOX revenue table	CRQ000000000042	crq000000000042	allen	ALLEN	192.168.8.129	192.168.8.129	Alter table sox_sales_central add total_revenue float
2009-01-22 15:08:36.0		0	3	Alter SOX revenue table	CRQ000000000042	crq000000000042	allen	ALLEN	192.168.8.129	192.168.8.129	Alter table sox_sales_west add total_revenue float
2009-01-22 15:08:44.0		0	3	Alter SOX revenue table	CRQ000000000042	crq000000000042	allen	ALLEN	192.168.8.129	192.168.8.129	Alter table sox_sales_international add total_revenue float
2009-01-22 15:12:39.0		0	0					SYSTEM	192.168.8.129	192.168.8.129	alter table allen.sox_sales_east add sum_total float
2009-01-22 15:14:19.0		0	0					SYSTEM	192.168.8.129	192.168.8.129	insert into allen.sox_sales_east (i,customer,zipcode,revenue,total_revenue,sum_total) values(?,?,?,?,?,?

Auditing Database Configuration Changes

Item	Type	Period	Use MD5	Keep Data
INFORMIXSQLHOSTS	Environment Variable	10m	✓	⊘
INFORMIXSERVER	Environment Variable	10m	✓	⊘
HOME	Environment Variable	1h	✓	⊘
$INFORMIX_HOME/bin/.+	File Pattern	1h	✓	⊘
$INFORMIX_HOME/aaodir/.+	File Pattern	1h	✓	⊘
$INFORMIX_HOME/dbssodir/.+	File Pattern	1h	✓	⊘
$INFORMIX_HOME/.login	File	10m	✓	⊘
$INFORMIX_HOME/.bash_profile	File	1h	✓	⊘
$INFORMIX_HOME/.bashrc	File	1h	✓	⊘
$INFORMIX_HOME/.cshrc	File	1h	✓	⊘
$INFORMIX_HOME/.profile	File	1h	✓	⊘
$INFORMIX_HOME/etc/.+	File Pattern	1h	✓	⊘
$INFORMIX_HOME/etc/conv/.+	File Pattern	1h	✓	⊘
$INFORMIX_HOME/etc/sysadmin/.+	File Pattern	1h	✓	⊘

- Tracks changes to files, environment variables, registry settings, scripts, etc.
- 200+ pre-configured templates for all major OS/DBMS configurations
  - Easily customizable via scripts, SQL, etc. (ad hoc tests)
  - Also checks OS permissions for Vulnerability Assessment (VA) tests

Tracking Privileged Users Who "su"

Challenge: How do you track users who 'switch' accounts (perhaps to cover their tracks)?

- Native database logging/auditing & SIEM tools can't capture OS user information
- Other database monitoring solutions only provide OS shell account that was used

User activity

```
Red Hat Enterprise Linux Server release 6.2 (Santiago)
Kernel 2.6.32-220.el6.x86_64 on an x86_64

osprey login: Joe
Password:
Last login: Wed Mar 27 04:52:37 on tty1
[Joe@osprey ~]$ su - informix
Password:
[informix@osprey ~]$ dbaccess stores -

Database selected.

> select * from creditcard;

     cardid cardnumber

          1 4321432143214321

1 row(s) retrieved.
```

What Guardium Shows You

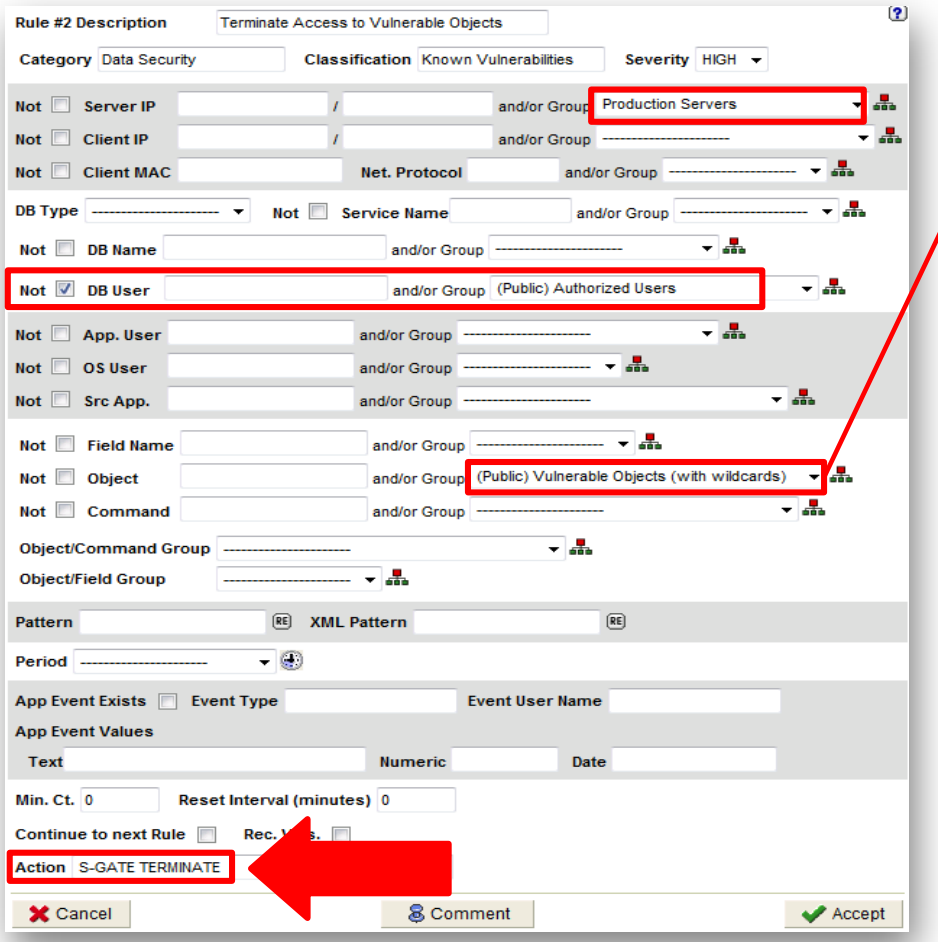
- User Chaining Report

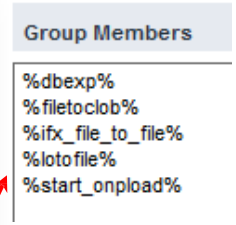
Start Date: 2013-03-27 04:46:59 End Date: 2013-03-27 04:46:59
Aliases: OFF

Timestamp	Server IP	Client IP	DB User Name	Server Port	Database Name	Sql	Uid Chain	Uid Chain Compressed
2013-03-27 18:46:55.0	10.10.9.56	10.10.9.56	INFORMIX	15174	STORES	select * from creditcard	(1,root,/sbin/init)->(2884,root,login -- Joe)->(2922,Joe,-bash)->(2945,Joe,su - informix)->(2949, informix,-bash)->(2978,informix,dbaccess stores -)	Joe

Records 1 to 3 of 3

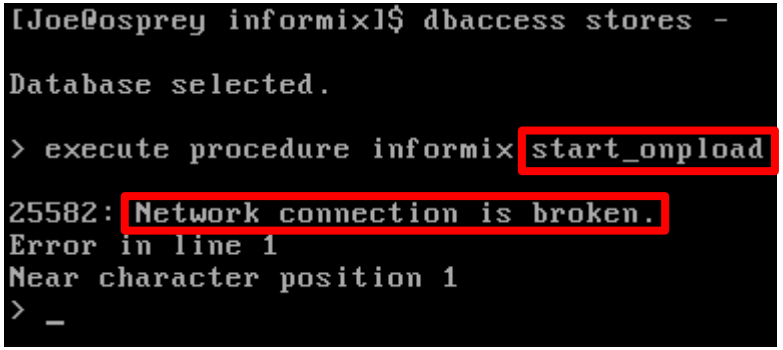
Protecting Against Vulnerabilities With Virtual Patching





Vulnerable procedure that can't be patched right away

What the user sees



InfoSphere Guardium Allows You To Protect Your Most Valuable Information

Continuously monitor access to high-value databases to:

Mitigate external and internal threats

Prevent unauthorized changes to sensitive data or structures

Automate and centralize controls

1. Across PCI DSS, data privacy regulations, HIPAA/HITECH, …
2. Across databases and applications

Simplify processes

Chosen by Leading Organizations Worldwide

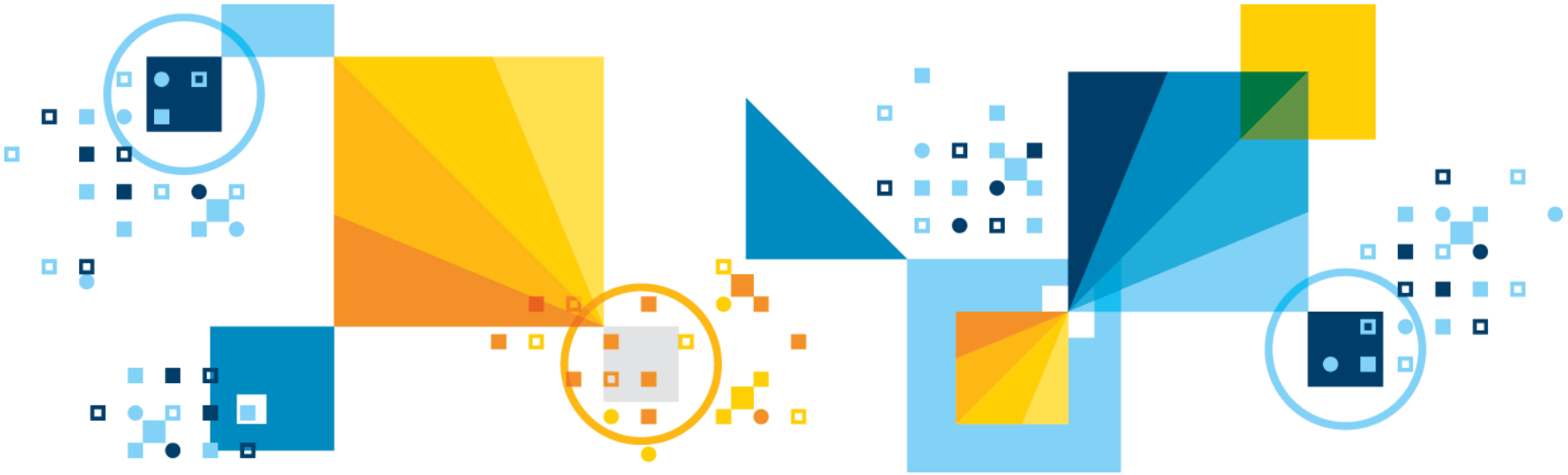
- 8 of the top 10 global banks
- 5 of the top 6 global insurers
- 4 of the top 4 health care providers
- 8 of the top 10 telecoms
- 3 of the top 4 auto makers
- 3 of the world's favorite beverage brands
- 2 of the top 3 global retailers
- Top government agencies
- Top global cardholder brand
- Top energy suppliers
- The most recognized name in PCs
- #1 dedicated security company
- Media & entertainment brands
- International airline brands

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Operating System Based Authentication



User Authentication on Informix

- **Authentication is a way of verifying a user identity or an application**

- **Configure the Informix server authentication mechanisms to meet varying requirements:**
  - Different security methods required for local and remote connections
  - Database access by users without operating system accounts on the servers host computer
  - Non-root installation

- **The simplest, default authentication method operates for a local connection by relying on OS user lookup:**
  - For this type of connection, a user ID and password pair are passed directly to the OS for verification that the user is legitimate
  - Method requires that users have connection privileges granted by the DBSA and have corresponding OS user accounts on the Informix host computer

User Authentication on Informix

- **On UNIX and Linux, the Informix installation can be configured to support other authentication mechanisms that maintain security while reducing the dependency on sysadmin and root-level privileges**

- **Users can develop modules and configure a server to have a self-defined authentication mechanism for local and remote connections.**

- **Authentication-layer mechanism can function so that you are not required to make changes in the application. Informix supports:**
  - Pluggable Authentication Modules (PAM) on Informix systems running on UNIX or Linux.
    - PAM provides an API set for authentication, account, session, and password mgmt.
  - Lightweight Directory Access Protocol (LDAP) Support on Windows
    - Use LDAP Authentication Support if you use an LDAP server to authenticate users.

User Authentication on Informix

- **The Informix client can be a local or a remote user.**
- **For network-based business models, the database server uses the network authentication mechanism provided by the OS, but requires the DBSA to set up trusted-hosts or trusted-user info**
- **Trusted-hosts information is set in the hosts.equiv file or the file specified by the REMOTE_SERVER_CFG configuration parameter.**
  - Trusted-user information is set in each user's **rhosts** file or in the file specified by the **REMOTE_USERS_CFG** configuration parameter.
    - Modify lookup options in the **sqlhosts** file.
- **Users that connect to the database server without login to the host computer OS are internal users.**

Internally Authenticated Users (UNIX, Linux)

- **The DBSA configures the server to authenticate users by checking credentials with a hashed password stored inside the database server.**

- **The DBSA creates internally authenticated users with the CREATE USER statement and sets up a password that is stored in the Informix SYSINTAUTHUSERS catalog table of the SYSUSER database**

- **The DBSA can administer internally authenticated users with the CREATE USER, DROP USER, ALTER USER, and RENAME USER SQL statements**
  - Users change their own password with the **SET USER PASSWORD** statement.

Creating Database Server Users (UNIX, Linux)

- **DBSA privileges can create internally authenticated users or users who do not have accounts on the host system.**
  - To create these types of users, you must map each user to the appropriate user and group privileges, regardless of whether these users have operating system accounts on the database server host computer.

- **After a non-root install, users cannot immediately connect to a DB server with passwords because permission issues prevent OS authentication.**
  - Additionally, users do not yet exist in the internal database.
  - The only way to initially connect to a non-root server is without a password.
    - Because only a DBSA can create users, the database owner must make a connection without a password, and then create users in the database.
    - The DBSA can create a user with or without a password.
    - The method of initial connection creation without a password is provided in this task.

Creating Database Server Users (UNIX, Linux)

- **You must have DBSA privileges.**

- **By default, the owner of a non-root server is a DBSA. When you create or modify user accounts, you can use CREATE USER or ALTER USER statements to grant the DBSA privilege to other users.**

- **For a non-root installation only:**
  - After installation, connect to the database server by using DB-Access.
  - On local clients, you can start DB-Access and establish a connection to the server by using a user name and password.
  - Alternatively, on the command prompt, a user can run the **dbaccess** command and then run other SQL statements to connect without a password, as follows:

```
>dbaccess - -
> database  sysuser;
Database selected.
```

Remote Hosts Connectivity

- **If you want to connect from a remote computer without a password, you must have trusted-host information or trusted-user information specified.**
  - Trusted-host information is in the **hosts.equiv** file or the file specified by the **REMOTE_SERVER_CFG** configuration parameter.
  - Trusted-user information is in each user's **rhosts** file or the file specified by the **REMOTE_USERS_CFG** configuration parameter.

New User Connectivity – Internally Authenticated

- **To enable a new user to successfully connect to the server:**
  - You are not required to specify information in the **USERMAPPING** configuration parameter when you create users.
    - If you want to enable the mapped or internal user to successfully connect to the server, you must set the **USERMAPPING** configuration parameter, as follows:
      - If you do not want mapped users to have administrative privileges, set the **USERMAPPING** parameter to **BASIC.**
      - If you want to make it possible for selected mapped users to have administrative privileges, set the **USERMAPPING** parameter to **ADMIN**.
      - No administrative privileges are given to any users until you provide that access when you run a **CREATE USER** (or **ALTER USER**) statement.
      - You can grant ADMIN privileges to users with surrogate property **AUTHORIZATION**.
      - The valid values are **dbsa**, **dbsso**, **aao** and **bargroup.**

USERMAPPING Configuration Parameter (UNIX, Linux)

- **Determines whether or not the database server accepts connections from mapped users.**

- **Values**
  - **OFF (default)**
    - Only users that are registered in the Informix host computer OS with a login service can connect to the database server.
    - Externally authenticated users without OS accounts on the Informix host computer cannot connect to database server resources.
  - **BASIC**
    - Users can connect to Informix without an OS account.
    - A user without an OS account cannot perform privileged user operations on the database server, even if the user maps to a server administrator user or group ID.
  - **ADMIN**
    - Users can connect to Informix without an OS account.
    - If a user has authenticated with the identity of a privileged user and is mapped to the proper server administrator group ID, the user can perform DBSA, DBSSO, or AAO work on the database server.

USERMAPPING Configuration Parameter (UNIX, Linux) (2)

- **Takes effect**
  - After you edit your **onconfig** file and restart the database server.
  - When you reset the value dynamically in your **onconfig** file by running the **onmode -wf** command.
  - When you reset the value in memory by running the **onmode -wm** command.

- **Usage**
  - Externally authenticated users without operating system (OS) accounts on the Informix host computer can access database server resources when **USERMAPPING** is turned on by setting the parameter with the **BASIC** or **ADMIN** value.
  - The setting of **BASIC** or **ADMIN** also determines whether or not mapped users can be granted administrative privileges.

USERMAPPING Configuration Parameter (UNIX, Linux)

- Changing the USERMAPPING configuration parameter from OFF to ADMIN or BASIC is not the only step in setting up Informix for mapped users.

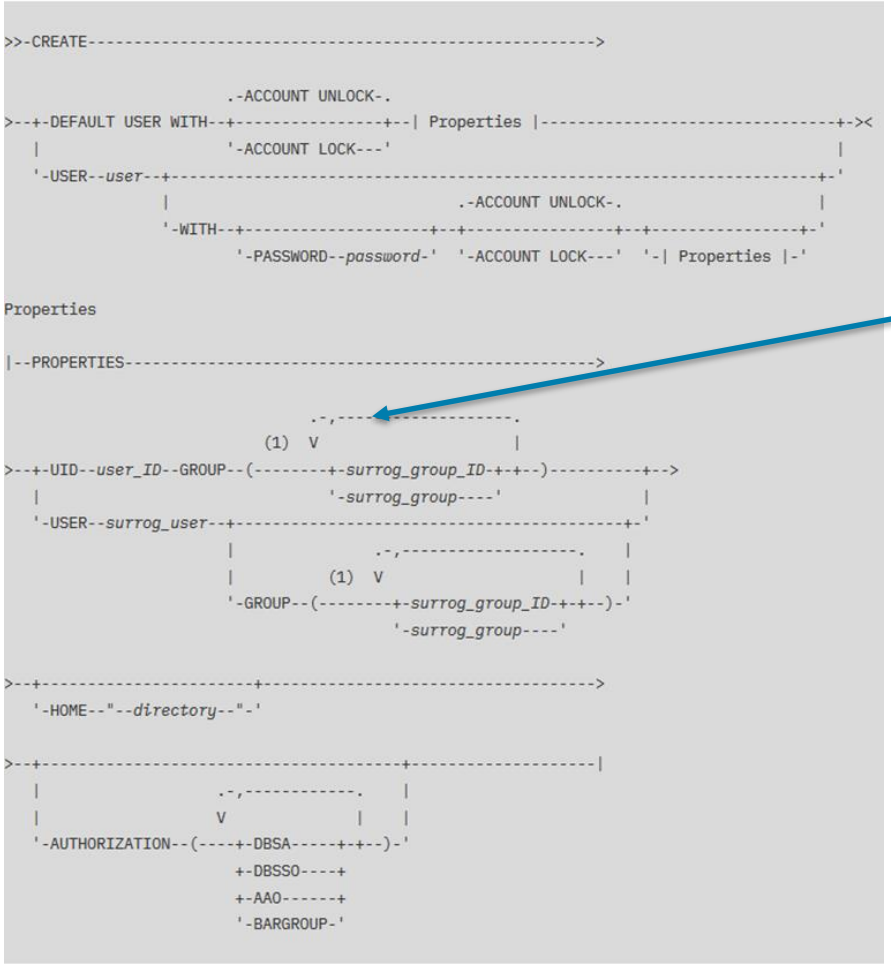
- To map users with the appropriate properties, you must also use DDL such as CREATE USER and ALTER USER to register values in appropriate system tables of the SYSUSER database.

- Depending on the DDL used and the defined table mapping, the following tables will be updated or populated:
  - SYSINTAUTHUSERS
  - SYSUSERMAP
  - SYSSURORGATES
  - SYSSURROGATEGROUPS

CREATE USER Statement

- **CREATE USER statement defines internally authenticated users, or maps externally authenticated users to surrogate user properties required for access to Informix resources.**

- **Syntax**



Notes:

(1) Use this path no more than 16 times

CREATE USER Statement (2)

Element	Description	Restrictions
directory	Path name of directory where user files are stored.	Must be 255 bytes or fewer, and must conform to the rules of your operating system. The *directory* must also: Belong to the mapped *user_ID* and *surrog_group_ID*. Have read, write, and execute permissions for the owner.
password	Password for internal authentication of *user*.	Must be 6 - 32 bytes.
surrog_group	Name of an existing operating system group (surrogate group) that has the permissions to which you want to map *user*. The list of *surrog_group* values must be enclosed in parentheses.	Must be 32 bytes or fewer.You must use one of the surrogates that are specified in the /etc/informix/allowed.surrogates file.

CREATE USER Statement (3)

Element	Description	Restrictions
surrog_group_ID	Group identifier number (surrogate group) to which you want to map the *user*. The list of *surrog_group_id* value or values that you specify must be enclosed in parentheses.	The *surrog_group_ID* cannot be:A group ID with server administrative privileges (DBSA, DBSSO, AAO, and BARGROUP) Group 0 (**root**, sometimes referred to as **wheel** or **system**) Group 80 on Mac OS X (**admin**) A group ID associated with group **bin** or group **sys** Must use one of the surrogates specified in the file /etc/informix/allowed.surrogates
surrog_user	Name of an existing OS user account (surrogate user) on the Informix host computer that has the permissions to which you want to map *user*.	Must conform to the rules of your operating system and one of the surrogates that are specified in file /etc/informix/allowed.surrogates

CREATE USER Statement (4)

Element	Description	Restrictions
user	Authorization identifier of the specific user that you are mapping to user properties.	Cannot be PUBLIC.
user_ID	User identifier number to which to map *user*.	Cannot be that of user **root** or of user **informix**. Must be one of the surrogates that are specified in file /etc/informix/allowed.surrogates.

CREATE USER Statement - Usage

- **Only a DBSA can run the CREATE USER statement**
  - With a non-root installation, the user who installs the server is the equivalent of the DBSA, unless the user delegates DBSA privileges to a different user
- **The USERMAPPING configuration parameter must be set to a value that enables support for mapped users before users defined by the CREATE USER statement can connect to the database server**
  - DBSA can issue the **CREATE USER** statement to map users to properties that correspond to the appropriate level of authorization
- **Values must be entered in the SYSUSERMAP table of the sysusers database to map users with the appropriate user properties so that the mapped user statements of SQL to work correctly**
- **Execution of the CREATE USER statement can be audited with the CRUR audit code**

CREATE USER Statement – PASSWORD Clause

- **For a root-privileged server, if an OS user is connecting and the USERMAPPING configuration parameter is unset, OS authentication occurs even though the user exists in the database**
- **If the USERMAPPING parameter is set, internal user authentication takes precedence over OS authentication**
  - Mapped users are authenticated internally or externally
  - If a user is created without a password, a mapped user is created
  - If a user is created with a password, an internally authenticated user is created with the properties from the operating system, unless an explicit **PROPERTIES** clause is also specified in the statement
  - If the **CREATE USER** statement contains both the **PASSWORD** clause and **PROPERTIES** clause, the user is an internally authenticated user, but has the surrogate properties that are specified in **PROPERTIES** clause
  - In this case, the surrogate user or group must also be listed in the **/etc/informix/allowed.surrogates** file

CREATE USER Statement – PROPERTIES Clause

- **The PROPERTIES clause can define a new user, and can optionally associate that user with surrogate properties that can include a group and a home directory**

- **CREATE DEFAULT USER is a special case of the CREATE USER statement**
  - **CREATE DEFAULT USER** statement defines the properties that are set for the default user
  - After default user properties are defined, new users can be created with default user properties by omitting the **PROPERTIES** clause
- **Mapped users can connect to the database server with the surrogate user properties if they authenticate with pluggable authentication module (PAM), single sign-on (SSO), or internal authentication**
- **Property values are not applicable to non-root installations but must be specified just like a root-privileged server**
  - Surrogate users and groups in non-root installations are not required in the **allowed.surrogates** file.

CREATE USER Statement – AUTHORIZATION clause

- **The AUTHORIZATION clause grants a subset of administrative privileges. The USERMAPPING configuration parameter must be set to ADMIN to enable this clause**

- **Use of the AUTHORIZATION clause (and of the AUTHORIZATION clause of the ALTER USER or GRANT ACCESS TO PROPERTIES statements) is not recommended**
  - It is likely this syntax will not support role separation in a future release

CREATE USER Statement – HOME and ACCOUNT

- **HOME directory clause**
  - Specifying a directory for the user files with the **HOME** keyword is optional, but in some cases it is highly desirable
    - If a home directory is not specified, an externally authenticated user has the same home directory as the surrogate user account on the Informix host computer
    - If the surrogate user identity that does not have a set home directory, then Informix creates a directory for user files in **$INFORMIXDIR/users**
      - In the latter case, the directory name in **$INFORMIXDIR/users** takes the form **uid.*ID_number*** (for example, uid.101)

- **ACCOUNT LOCK and ACCOUNT UNLOCK keywords**
  - With the **ACCOUNT LOCK** and **ACCOUNT UNLOCK** keywords, a **DBSA** can toggle disabling and enabling the specified user's access to the database server

CREATE USER - Examples

- **Following statement creates a mapped user named joe:**
  - **CREATE USER joe;**
    - If the user **joe** is an OS user
      - **joe** has the operating system properties that are associated with his user name.
    - If the user **joe** is not an OS user and if default user properties are defined,
      - **joe** has the surrogate properties of the default user.
      - If default user properties are not defined, an error is returned.

- **Following statement creates an internally authenticated user named joe with a password of joebar:**
  - **CREATE USER joe WITH PASSWORD "joebar";**
    - If the user **joe** is not an OS user and if default user properties are defined,
      - **joe** has the surrogate properties of the default user.
    - If default user properties are not defined, an error is returned

CREATE USER - Examples

- **Following statement creates an internally authenticated user named phil with a locked account:**
  - **CREATE USER phil WITH PASSWORD "joebar" ACCOUNT LOCK;**
  - If the user **phil** is not an OS user and if default user properties are defined,
    - **phil** has the surrogate properties of the default user.
    - If default user properties are not defined, an error is returned

- **Following statement creates an internally authenticated user named mary with a UID, a group, and a home directory:**
  - **CREATE USER mary WITH PASSWORD "joebar" PROPERTIES UID 44567 GROUP(1234) HOME "/home/pd/osuser";**

CREATE USER - Examples

- **Following statement creates a mapped user named bill with a surrogate user name of foo_os:**
  - **CREATE USER bill WITH PROPERTIES user "foo_os";**
    - The user bill has the properties of the operating system user **foo_os**.

- **Following statement creates a user, internally named PUBLIC, with the properties of the surrogate user tmp:**
  - **CREATE DEFAULT USER WITH PROPERTIES USER "tmp";**
    - Other users created without surrogate properties will have these properties.

Creating Surrogates for Mapped Users (1)

- **Specify operating system (OS) user names, user IDs, group names, and group IDs in the allowed.surrogates file to control which OS users and groups can act as surrogates for mapped users**

- **Create a file named allowed.surrogates in the /etc/informix directory**
  - **allowed.surrogates** must be owned by **root** instead of **informix**
    - Must not have execute permissions and only the file owner can have write permission.
    - Enter the OS user names, user IDs, OS group names, group IDs, ranges of user IDs, and ranges of group IDs that you want to allow as surrogates.
    - Enter comma-separated OS user names, user IDs, and ranges of user IDs after entering the user: label
      - **users:user1,user2,105,104,300,400..500**
    - Enter comma-separated OS group names, group IDs, and ranges of group IDs after entering the group: label.
      - **groups:ifx_dbsa,group1,group2,root,1,10..20**

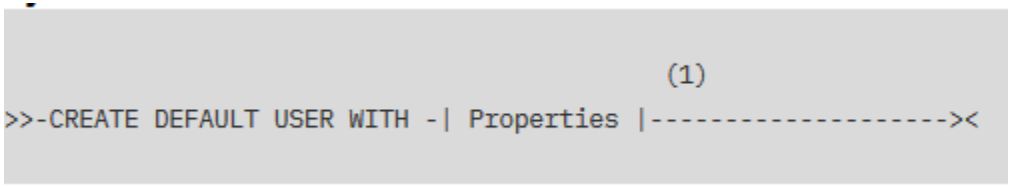
Creating Surrogates for Mapped Users (2)

- **The group and user labels are case-insensitive, and can be pluralized:**
  - Entries are separated by commas
  - Ranges of user IDs and group IDs are inclusive, with the upper and lower ranges separated by two periods
  - You must specify both an upper and lower limit for ranges
  - Comment lines begin with # and are ignored
  - Blank lines are also ignored

  - If **allowed.surrogates** is formatted incorrectly, then user mapping is disabled and an error is logged in the online log file
    - If a user name or group name cannot be identified, the name is logged in the online log file and otherwise ignored, and the cache is cleared
  - Following is an example of an **allowed.surrogates** file entry specifies user user1, user 40, users 45-50, and group 10 as acceptable surrogates:
    - **#Surrogate IDs**
    - **USERS:user1,40,45..50**
    - **GROUP:10**

CREATE DEFAULT USER statement (UNIX, Linux)

- **Use the CREATE DEFAULT USER statement to define the properties set of the default internally authenticated user:**
  - This statement is an extension to the ANSI/ISO standard for the SQL language.
- **Syntax**

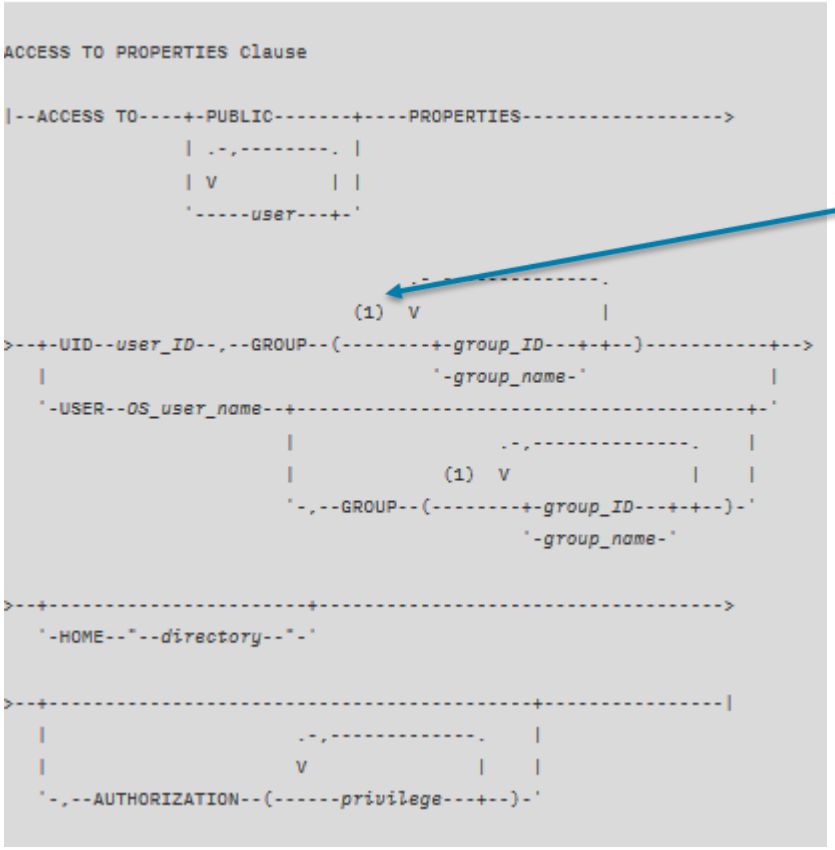


- **CREATE DEFAULT USER is a special case of the CREATE USER statement.**
  - After the **CREATE DEFAULT USER** statement is executed to define default user properties, you can use the **CREATE USER** statement (but omitting the **PROPERTIES** clause) to create new users who have default user properties.
  - Only a **DBSA** can issue the **CREATE DEFAULT USER** statement.
  - With a non-root installation, the user who installs the server is the equivalent of the DBSA, unless the user delegates DBSA privileges to a different user.

Surrogate User Properties (UNIX, Linux)

- Use the ACCESS TO PROPERTIES clause of the GRANT statement to map users to surrogate user properties required for access to Informix resources.
- Only a DBSA can map externally authenticated users to valid surrogate user properties.
- If the USERMAPPING configuration parameter is set to enable support for mapped users, a DBSA issues the GRANT ACCESS TO PROPERTIES statement to map users to properties that correspond to the appropriate level of authorization.
- Mapped users can connect to Informix with the surrogate user properties if they authenticate with a pluggable authentication module (PAM) or single sign-on (SSO).

GRANT Statement – ACCESS TO PROPERTIES



Notes:

(1) Use this path no more than 16 times

ACCESS TO PROPERTIES Clause (2)

Element	Description	Restrictions
directory	Path name of directory where user files are stored.	Must be 255 bytes or fewer, and must conform to the rules of your operating system. The *directory* must also: Belong to the mapped *user_ID* and *group_ID*. Have read, write, and execute permissions for the owner. Not have PUBLIC write permissions
group_id	Group identifier number to which you want to map *user*. The list of *group_id* value or values that you specify must be enclosed in parentheses.	The *group_ID* cannot be: • A group ID with server administrative privileges (DBSA, DBSSO, AAO, and BARGROUP) • Group 0 (**root**, sometimes referred to as **wheel** or **system**) • Group 80 on Mac OS X (**admin**) • A group ID associated with group **bin** or group **sys** The group ID must be present in /etc/informix/allowed.surrogates file.

ACCESS TO PROPERTIES Clause (3)

Element	Description	Restrictions
group_name	Name of an existing operating system group having the permissions to which you want to map *user*. The list of *group_name* values must be enclosed in parentheses.	Length cannot exceed 32 bytes. The group name must be present in /etc/informix/allowed.surrogates file.
privilege	•Administrative privilege to assign *user*. Valid values are as follows: •DBSA •DBSSO •AAO •BARGROUP The *privilege* value or values must be enclosed in parentheses.	The USERMAPPING configuration parameter must be set to ADMIN to grant server administrative privileges with the AUTHORIZATION keyword.
user	Authorization identifier of a specific user that you are mapping to user properties.	Must be an authenticated authorization identifier

ACCESS TO PROPERTIES Clause (4)

Element	Description	Restrictions
user_ID	User identifier number to which to map *user*.	Cannot be that of user **root** or of user **informix**. Must be one of the surrogates that are specified in file /etc/informix/allowed.surrogates.
OS_user_name	Name of an existing OS user account on the Informix host computer having the permissions to which you want to map *user*.	Must conform to the rules of your operating system.The user name must be present in /etc/informix/allowed.surrogates file.

ACCESS TO PROPERTIES Clause – Usage (1)

- **The best practice is to map *user* to a specific OS user name that is reserved as a surrogate user identity only.**
- **Add groups associated with the surrogate user identity with the GROUP keyword, and change the home directory with the HOME keyword.**
  - If the operating system administrator has specified acceptable surrogates in the **/etc/informix/allowed.surrogates** file, you can only map users to those specified OS users or groups.
- **If you map *user* to a user ID number, then remember to not create a user account on the Informix host computer with the same number.**
- **The USERMAPPING configuration parameter must be set to ADMIN in order to assign *user* a server administrative privilege with the ADMINISTRATOR keyword.**

ACCESS TO PROPERTIES Clause – Usage (2)

- The PUBLIC and AUTHORIZATION keywords cannot be used together in the same statement, because the PUBLIC group cannot be granted server administrator privileges.

- Specifying a directory for the user files with the HOME keyword is optional, but in some cases it is highly recommended. When an externally authenticated user is mapped to a surrogate user name but no HOME directory is specified in the GRANT ACCESS TO statement, then the mapped user has the same home directory as the user account on the Informix host computer.

- When a user is mapped to a surrogate user identity with no set home directory, then Informix creates a directory for user files in $INFORMIXDIR/users. In the latter case, the directory name in $INFORMIXDIR/users takes the form uid.ID_number (for example, uid.101).

ACCESS TO PROPERTIES Clause – Examples

- **The syntax and explanations in this section are examples for the following environment, where the acronym GID abbreviates group ID number, and the acronym UID abbreviates user ID number:**
  - There is a user **fred** with an OS account on the Informix host computer.
  - User **fred** has database server access with UID 3000, GID 3000 (users), auxiliary group 200 (staff), and home directory **/home/fred**.
  - On the same computer, there exists an OS account for user **dbuser**. This account is locked so that **dbuser** cannot log in.
  - The **dbuser** account exists only for the purpose of surrogate user mapping. It has UID 3050, GID 4000 (**ifx_user**), and home directory **/home/dbuser**.
  - The group **ifx_user** has GID 4000, with users **bill** and **eileen**.
  - The **administrator** setting up mapped users knows that there is no entry for UID 101 in **/etc/passwd** (or its equivalent) and no entry for GID 10011 or 10101 in **/etc/group** (or its equivalent) .
  - User **bob** does not have OS account on the Informix host computer but can authenticate through PAM or LDAP. Database server is configured to accept authentication through the PAM or LDAP module.
- **The USERMAPPING parameter in the onconfig file is set to ADMIN.**

ACCESS TO PROPERTIES Clause – Examples

- **The syntax and explanations in this section are examples for the following environment, where the acronym GID abbreviates group ID number, and the acronym UID abbreviates user ID numbers (cont'd):**
  - The **USERMAPPING** parameter in the **onconfig** file is set to **ADMIN**.

- **Mapping an externally authenticated user to a surrogate user name:**
  - The administrator maps **bob** to the database server access privileges that already exist for user **fred** by issuing the following **GRANT** statement:
    - **GRANT ACCESS TO bob PROPERTIES USER fred;**

- **Granting Informix access to all externally authenticated users:**
  - In this environment, the purpose of the user **dbuser** account on the Informix host computer is to grant database server access to mapped users.
  - In a situation where there are many mapped users and they do not need to know about the user files created in the home directory, the administrator might find it efficient and sufficiently secure to map **PUBLIC** to the dbuser surrogate user identity.

ACCESS TO PROPERTIES Clause – Examples

- **Granting Informix access to all externally authenticated users (cont'd):**
  - The administrator can map all authenticated users (**PUBLIC**) to the privileges established for **dbuser** with the following **GRANT ACCESS** statement:

- **GRANT ACCESS TO PUBLIC PROPERTIES USER dbuser;**

- **Mapping an externally authenticated user to a UID-GID pair:**
  - The administrator maps **bob** to a surrogate user identity that consists of a UID-GID pair that enables database server access by running the following statement:
    - **GRANT ACCESS TO bob PROPERTIES UID 101, GROUP (10011);**

  - Because no specific directory was specified, a directory under **$INFORMIXDIR/users** will be created with the name **uid.101** and this path will be used as the home directory. The UID 101 and GROUP (10011) are anonymous because they do not have entries in the respective **/etc** directories that designate UIDs and GIDs that can access Informix .

ACCESS TO PROPERTIES Clause – Examples

- **Alternatively, the administrator can map bob to a surrogate user identity that is a combination of an anonymous UID and to an explicit group, such as in the following example:**

- **GRANT ACCESS TO bob PROPERTIES 101, GROUP (ifx_user);**

- **Because the ifx_user group has members bill and eileen, the group is not anonymous.**

- **Mapping an externally authenticated user to a surrogate user identity that has server administrative privileges:**
  - In the following example, the administrator grants **DBSA** privileges to **bob**:
  - **GRANT ACCESS TO bob PROPERTIES USER fred, GROUP (ifx_user), AUTHORIZATION (dbsa);**

ACCESS TO PROPERTIES Clause – Examples

- User bob is assigned UID 3000 (fred) and GIDs 3000 (users), 200 (staff), and the extra group 1000 (ifx_user).

- The administrative role granted to bob could be different by replacing dbsa with a different privilege (DBSSO, AAO, or BARGROUP).
- If the USERMAPPING parameter were set to BASIC in the onconfig file, then bob would not be granted DBSA privileges by this statement.
- If USERMAPPING were set to OFF, then bob would not be able to connect to the database server at all.

ACCESS TO PROPERTIES Clause – Examples

- The following GRANT statements are examples of valid ACCESS TO PROPERTIES clauses, using hypothetical values.
- These examples do not represent the entire syntax and possible semantics of the ACCESS TO PROPERTIES clause:

```
GRANT ACCESS TO bob PROPERTIES USER fred;

GRANT ACCESS TO PUBLIC PROPERTIES USER dbuser;

GRANT ACCESS TO bob PROPERTIES USER dbuser HOME
"/home/dbuser/bob";

GRANT ACCESS TO bob PROPERTIES UID 101, GROUP (10011);

GRANT ACCESS TO bob PROPERTIES 101, GROUP (ifx_user);

GRANT ACCESS TO bob PROPERTIES USER fred, GROUP (ifx_user),
AUTHORIZATION (DBSA);
```

CREATE DEFAULT USER statement (UNIX, Linux)

- The USERMAPPING configuration parameter must be set to a value (ADMIN or BASIC) that enables support for mapped users before default users who were created with the CREATE DEFAULT USER statement can connect to the database server.

- A DBSA can issue the CREATE DEFAULT USER statement to map default users to properties corresponding to an appropriate authorization level.

- The USERMAPPING configuration parameter must be set to ADMIN to enable a default user to have a server administrative privilege with the AUTHORIZATION keyword, where AAO, BARGROUP, DBSA, and DBSSO are the keyword options for specific administrative privileges.

CREATE DEFAULT USER statement (UNIX, Linux)

- **Must also enter values in the SYSUSERMAP table of the sysusers database to map users with the appropriate user properties, so that the mapped user statements of SQL can work correctly**

- **Cannot specify a password, or account lock, or account unlock information in CREATE DEFAULT USER**
  - This is equivalent to **GRANT ACCESS TO PUBLIC PROPERTIES**
    - The equivalent syntax to this is **DROP DEFAULT USER;**

- **To alter the properties of the default internally authenticated user, you can issue the ALTER DEFAULT USER WITH PROPERTIES statement**

- **Execution of CREATE DEFAULT USER can be audited with the CRUR audit code, same mnemonic as used for CREATE USER**

DROP USER statement (UNIX, Linux)

- DBSA only uses the DROP USER to remove an inactive internal user.
- Syntax

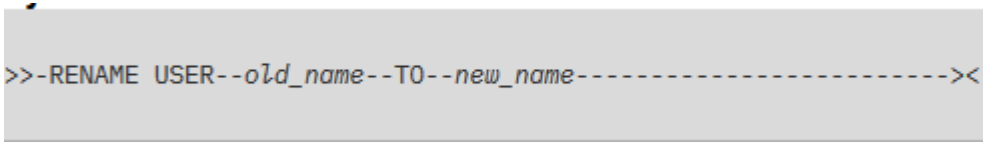
- user
  - Authorization identifier of an inactive specific user that you are dropping.
  - Must be an existing authorization identifier

- With a non-root installation, user installing the server is the DBSA equivalent, unless the user delegates DBSA privileges to a different user.
- DROP USER execution can be audited with the DRUR audit code.
- The following statement drops the user bill:
  - DROP USER bill;

RENAME USER statement (UNIX, Linux)

- **Use the RENAME USER to change the name of an internal user of a non-root installation of the database server.**

- **Syntax**



- **old_name**
  - Authorization identifier of a specific user that you are renaming.
    - Must be an existing authorization identifier
- **new_name**
  - Authorization identifier of a specific user.
    - Cannot be an existing authorization identifier
- **Usage**
  - Only a **DBSA** can run the **RENAME USER** statement.
    - With a non-root installation, the user who installs the server is the equivalent of the **DBSA**, unless the user delegates **DBSA** privileges to a different user.

RENAME USER statement (UNIX, Linux)

- **Use the RENAME USER to change the name of an inactive internal user of a non-root installation of the database server.**

- **Syntax**

```
>>-RENAME USER--old_name--TO--new_name-------------------------><
```

- **old_name**
  - Authorization identifier of an inactive specific user that you are renaming
    - Must be an existing authorization identifier
- **new_name**
  - Authorization identifier of a specific user.
    - Cannot be an existing authorization identifier
- **Usage**
  - Only a **DBSA** can run the **RENAME USER** statement.
    - With a non-root installation, the user who installs the server is the equivalent of the **DBSA**, unless the user delegates **DBSA** privileges to a different user.

RENAME USER statement (UNIX, Linux)

- **Execution**
  - Does not transfer any database or table level privileges granted to the old user name to the new user name
  - Can be audited with the **RNUR** audit code

- **USERMAPPING configuration parameter set to BASIC or ADMIN**

- **Must also enter values in the SYSUSERMAP table of the sysusers database to map users with the appropriate user properties so that the mapped user statements of SQL to work correctly**

- **Following statement renames user bill to bob:**
  - **RENAME USER bill TO bob;**

ALTER USER statement (UNIX, Linux)

- The ALTER USER statement is used to change one or more of the properties, including the password, user ID, surrogate group, administrative authorization, and home directory, and to enable or disable the account of an internally authenticated user, or of the default internally authenticated user.

ALTER USER statement (UNIX, Linux)

Notes:

(1) Use this path no more than 16 times

ALTER USER Statement (2)

Element	Description	Restrictions
directory	Path name of directory where user files are stored.	Must be 255 bytes or fewer, and must conform to the rules of your operating system. The *directory* must also: Belong to the mapped *user_ID* and *surrog_group_ID*. Have read, write, and execute permissions for the owner.
password	Password for internal authentication of *user*.	Must be 6 - 32 bytes.
surrog_group	Name of an existing operating system group (surrogate group) that has the permissions to which you want to map *user*. The list of *surrog_group* values must be enclosed in parentheses.	Must be 32 bytes or fewer.

ALTER USER Statement (3)

Element	Description	Restrictions
surrog_group_ID	Group identifier number (surrogate group) to which you want to map the *user*. The list of *surrog_group_id* value or values that you specify must be enclosed in parentheses.	The *surrog_group_ID* cannot be: • A group ID with server administrative privileges (DBSA, DBSSO, AAO, and BARGROUP) • Group 0 (**root**, sometimes referred to as **wheel** or **system**) • Group 80 on Mac OS X (**admin**) • A group ID associated with group **bin** or group **sys**
surrog_user	Name of an existing OS user account (surrogate user) on the Informix host computer that has the permissions to which you want to map *user*.	Must conform to the rules of your operating system and one of the surrogates that are specified in file /etc/informix/allowed.surrogates
user	Authorization identifier of the specific user that you are mapping to user properties	Must be an authenticated authorization identifier

ALTER USER Statement (4)

Element	Description	Restrictions
user	Authorization identifier of the specific user that you are mapping to user properties.	Cannot be PUBLIC.
user_ID	User identifier number to which to map *user*.	Cannot be that of user **root** or of user **informix**. Must be one of the surrogates that are specified in file /etc/informix/allowed.surrogates.

ALTER USER statement – Usage (1)

- **Only DBSA can run the ALTER USER statement.**
- **With a non-root installation, the user who installs the server is the equivalent of the DBSA, unless the user delegates DBSA privileges to a different user.**

- **The USERMAPPING configuration parameter:**
  - Must be set to **ADMIN** or **BASIC** to enable support for mapped users before users created with **CREATE USER** can connect to the database server
  - Must be set to **ADMIN** to enable the **AUTHORIZATION** clause.

- **Must also enter values in the SYSUSERMAP table of the sysusers database to map users with the appropriate user properties so that the mapped user statements of SQL to work correctly.**
- **Mapped users can connect to Informix with the surrogate user properties if they authenticate with pluggable authentication module (PAM) or single sign-on (SSO).**

ALTER USER statement – Usage (2)

- **The best practice is to map a user to a specific surrog_user reserved as a surrogate user identity only**
  - Add groups associated with the surrogate user identity with the **GROUP** keyword, and change the home directory with the **HOME** keyword
- **ALTER USER does not affect any active operations with the same surrogate user or user ID**
  - Only subsequent operations that require authentication are affected.
- **Can add a password for a user with the ADD keyword only if that user does not have a password:**
  - To change an existing password, use the **MODIFY** option in **ALTER USER**.
- **The total number of groups after the ALTER USER operation cannot exceed 16, which is the maximum number of allowed groups.**

ALTER USER statement – Usage (3)

- **An ALTER USER statement can only add a home directory with the ADD keyword if no home directory exists.**
  - To modify an existing home directory, use the **MODIFY** keyword.
- **In a single ALTER USER statement, a specific property can only be specified once.**
  - Cannot drop a **GROUP** property and add a **GROUP** property in the same statement.
- **After the ALTER USER statement, the user must have either one USER property or one UID property.**
- **Execution of ALTER USER can be audited with the ALUR audit code.**

ALTER USER statement – Examples (1)

- **Replace a USER property with a UID property**
  - Following statement replaces the **USER** property with a **UID** property for the user **bill**:
  - **ALTER USER bill DROP USER, ADD UID 1360;**

- **Change and add properties**
  - Following statement changes a **UID** property, adds the **DBSA** group, and adds a home directory for the user **bill**:
  - **ALTER USER bill MODIFY UID 1361, ADD GROUP (dbsa), ADD HOME "/u/user1";**

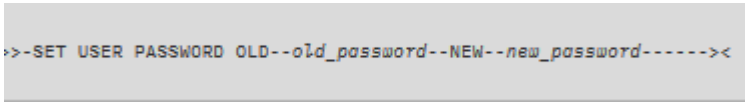
- **Unlock an account and drop an authorization property**
  - Following statement unlocks the account and drops the **DBSSO** authorization for the user **bill**:
  - **ALTER USER bill ACCOUNT UNLOCK DROP AUTHORIZATION (dbsso);**

ALTER USER statement – Examples (2)

- **Drop a home directory**
  - Following statement drops the home directory for the user **bill**:
  - **ALTER USER bill DROP HOME;**

SET USER PASSWORD statement (UNIX, Linux)

- **Use the SET USER PASSWORD to change a user password for database server access if a user is internally authenticated.**
- **Syntax**



- **new_password**
  - New quoted string 6 – 32 byte password for internal user authentication.
- **old_password**
  - Existing quoted string 6 – 32 byte password for internal user authentication.
- **Usage**
  - A **DBSA** cannot use this to change the password of another user.
  - To change the passwords of other users, a **DBSA** can use **ALTER USER.**

- **Execution of SET USER PASSWORD can be audited with the PWUR audit code.**

SET USER PASSWORD statement (UNIX, Linux)

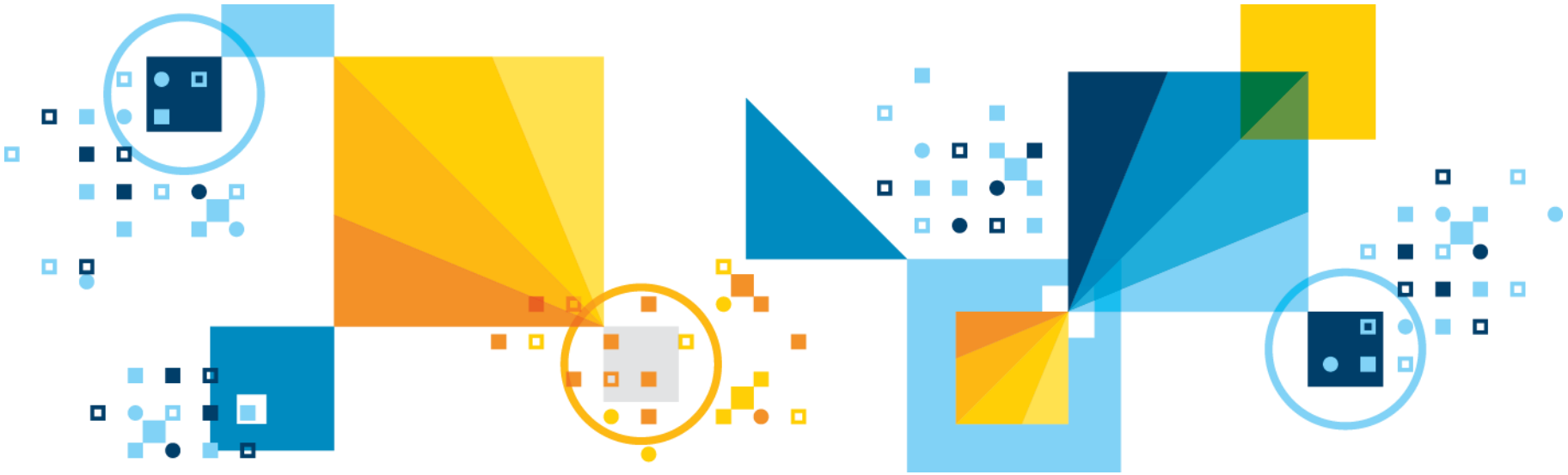
- Following statement changes the password from joebar to joefoo:
  - SET USER PASSWORD OLD 'joebar' NEW 'joefoo';

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Two Factor Authentication – Pluggable Authentication Module (PAM)



What is PAM ?

- A flexible mechanism for authenticating users.
- Since the beginnings of UNIX, user authentication has been accomplished via the user entering a password.
- Since then, a number of new ways of authenticating users have become popular. Including more complicated replacements for the /etc/passwd file, and hardware devices such as Smart cards etc..
- The problem is that each time a new authentication scheme is developed, it requires all the necessary programs (login, ftpd etc...) to be rewritten to support it.
- PAM provides a way to develop programs that are independent of authentication scheme. These programs need "authentication modules" to be attached to them at run--time in order to work.

How to Use PAM

- Define a PAM service.
- Tell the Informix server to use that service for authentication.
- Tell the Informix server whether a challenge response is required.

Defining a PAM Service

- **On Linux there is a directory named /etc/pam.d**
- **Services are defined using text files in that directory**
- **On other UNIX variants, /etc/pam.conf is a text file:**
  - The first column in the file is the name of the service.

Define a PAM Service (1)

- **For each service there are up to four functions**
  - Authentication (Auth)
  - Account management (account)
    - This is functionality such as expired password, account lockout
  - Session Management (Session)
  - Password management (password)

- **Informix only uses auth and account**

Define a PAM Service (2)

- **Generally speaking, it is not necessary to write PAM modules.**
  - A quick look on a Linux machine shows 50+ predefined PAM modules.
- **Many PAM services are also predefined**
  - Example “login” and “rlogin”

PAM Configuration on the Database Server

- **For Informix, PAM configuration is done in the sqlhosts, in the $INFORMIXDIR/etc/sqlhosts.xxx file**

- **Column 5 in the sqlhosts file, the following entries:**
  - **– S=4,**
  - **pam_serv=service**,
  - **pamauth=[password | challenge]**

  - **service** □□ entry in pam.conf or pam.d
  - **pamauth=password** will never attempt to communicate with client

- **Sample line in the sqlhosts file looks like:**

  olserver onsoctcp toru 8749 s=4,pam_serv=login,pamauth=password

System Configuration

- **On UNIX**
  - **/etc/pam.conf** file
  - Service name on each line
  - Multiple lines as needed
- **LINUX**
  - **/etc/pam.d** directory
  - Filename in **pam.d** is service name

- **File layout looks like:**
  - service module_type control_flag module_path options

Service Module Types

- auth – authenticate user and setup user credentials
- account – determine if user account is valid
- session – setup and terminate login session
- password – manage or change authentication tokens

- Informix uses only auth and account

Control Flag Types

- **Binding – save failure as required failure – keep going**
  - Return immediately on success
- **Optional – save failure as optional failure – keep going**
- **Required – save failure as required failure – keep going**
- **Requisite -- save failure as required failure**
  - Return failure immediately.
- **Sufficient – save failure as optional failure**
  - Return immediately on success.
- **See man pam.conf**

O/S provided PAM modules

- **Modules are provided for UNIX passwd authentication**
- **Kerberos – userid/password**
- **pam_deny.so**
  - Configured for ‘other’ service
- **Many others**
  - http://www.kernel.org/pub/linux/libs/pam/Linux--PAM--html/sag--modulereference.html
- **pam_debug.so**
- **pam_stack.so**

Sample PAM configuration – Solaris pam.conf

ifx_pam	auth	requisite	pam_authtok_get.so.1
ifx_pam	auth	required	pam_dhkeys.so.1
ifx_pam	auth	required	pam_unix_auth.so.1
ifx_pam	auth	required	/usr/informix/sqldist/lib/pam/pam_daveds.so
Ifx_pam	account	optional	/usr/informix/sqldist/lib/pam/pam_daveds.so

- IDS uses auth and account service
- If no configuration is specified for a given service, PAM uses 'other' service
- If no pathname is specified, lib is located in /lib/security
- Be careful to differentiate for 32 vs 64 bits.
  - `auth  required  /lib/security/$ISA/pam_deny.so`

- Your lib must have suitable permissions for root execution:

```
-rwxr-xr-x 1 root bin    4256 Mar 30 10:58 pam_daveds.so
```

Sample PAM Configuration -- sqlhosts

```
davedspam    ontlitcp lx-rama 8743
s=4,pam_serv=ifx_pam,pamauth=challenge
davedspamp ontlitcp lx-rama 8744
s=4,pam_serv=login,pamauth=password
```

- **s=4 indicates PAM**
- **pam_serv is service name from pam.conf (pam.d)**
- **pamauth is password or challenge**
  - **Challenge** promises that the pam module can ask for more
  - **Password** will never go back to the user for more

- **Be careful of "other" service**

How to write your own PAM module

- **Shared lib with relocatable code**
- **Correct permissions/ownership -- root 755**
- **pam_authenticate (auth) will call pam_sm_authenticate**
- **pam_acct_mgmt (account) will call pam_sm_acct_mgmt**
- **_sm_ = service module**

- **http://www.kernel.org/pub/linux/libs/pam/Linux--PAM--html/**
  - The System Administrators' Guide
  - The Module Writers' Guide
  - The Application Developers' Guide

Test programs

- Check UNIX password configuration
- Check PAM configuration

Check UNIX password Configuration

```
/*****************************************************************
 * Program to simulate IDS UNIX authentication.
 * Dave Desautels, IBM, 2007
 *
 * Usage: getsec <username>
 *
 *****************************************************************/

#include <pwd.h>
#include <stdio.h>
#include <errno.h>
#include <shadow.h>

int
main(int argc, char *argv[])
{
  int retval = 0;
  struct passwd *pw;
  struct spwd *spw;

  pw = getpwnam(argv[1]);
  printf("User: %s  Password: %s\n", pw->pw_name, pw->pw_passwd);

  spw = getspnam(pw->pw_name);
  if (spw)
  {
    printf("Shadow pw: %s\n", spw->sp_pwdp);
  }
  else
  {
    printf("getspnam failed, errno: %d\n", errno);
    retval = 1;
  }

  return retval;
}
```

Check PAM configuration

```
/*****************************************************************
 * Program to simulate IDS PAM Usage.
 * Dave Desautels, IBM, 2007
 *
 * Usage: pam_test <username> <PAMservice>
 *
 *****************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <pwd.h>
#include <security/pam_appl.h>

#define USER    (argv[1])
#define SERVICE (argv[2])

int conv_func(int num_msg, struct pam_message *msg[],
         struct pam_response **resp, void *appdata_ptr)
{
   int i;
   struct pam_message *msgs = *msg;
   struct pam_response *aresp = calloc(num_msg, sizeof(*aresp));
   char resp_buf[PAM_MAX_RESP_SIZE];

   for (i=0; i<num_msg; i++)
   {
      aresp[i].resp_retcode = 0;
      fputs(msgs[i].msg, stderr);
      fgets(resp_buf, sizeof(resp_buf), stdin);
      resp_buf[strlen(resp_buf)-1] = '\0';
      aresp[i].resp = strdup(resp_buf);
   }
   *resp = aresp;
   return PAM_SUCCESS;
}
```

Cont‘d

```
int main(int argc, char *argv[])
{
   int rc = 0, retval = 0;
   struct passwd *pw;
   struct pam_conv conv = {conv_func, NULL};
   pam_handle_t *pamh = NULL;

   pw = getpwnam(USER);
   if (pw)
      printf("Name: %s\nUID: %d\nGID: %d\nHome: %s\n",
        pw->pw_name, pw->pw_uid, pw->pw_gid, pw->pw_dir);
   else
   {
      printf("User %s does not exist.\n", USER);
      rc = 1;
      goto outta_here;
   }

   retval = pam_start(SERVICE, USER, &conv, &pamh);
   if (retval != PAM_SUCCESS)
   {
      printf("Error from pam_start\n%s\n", pam_strerror(pamh, retval));
      rc = retval;
      goto pam_cleanup;
   }

   printf("Pam Handle: 0x%x\n", pamh);

   retval = pam_authenticate(pamh, 0);
   if (retval != PAM_SUCCESS)
   {
      printf("Error from pam_authenticate: %d\n%s\n", retval, pam_strerror(pamh, retval));
      rc = retval;
      goto pam_cleanup;
   }
```

Cont'd

```
   retval = pam_acct_mgmt(pamh, 0);
   if (retval != PAM_SUCCESS)
   {
      printf("Error from pam_acct_mgmt: %d\n%s\n", retval, pam_strerror(pamh, retval));
      rc = retval;
      goto pam_cleanup;
   }


if (!rc)
   printf ("User %s is authorized for service %s!\n", USER, SERVICE);


pam_cleanup:
   retval = pam_end(pamh, 0);
   if (retval != PAM_SUCCESS)
   {
      printf("Error from pam_end: %d\n%s\n", retval, pam_strerror(pamh, retval));
      rc = retval;
   }


outta_here:
   return rc;
}
```

Sample PAM Module

```
#include <security/pam_appl.h>
#include <security/pam_modules.h>
#include <stdlib.h>
#include <strings.h>

#define NUM_MSGS 2
struct pam_message msgs[NUM_MSGS] = {
 {PAM_PROMPT_ECHO_OFF, "What is your favorite color? "},
 {PAM_PROMPT_ECHO_OFF, "What is your pet's name? "} };

int
pam_sm_authenticate(pam_handle_t *pamh, int flags, int argc, const char **argv)
{
   int retval;
   struct pam_conv *conv;
   struct pam_message *pam_msg = &msgs[0];
   struct pam_response *pam_resp;
   int i;

   retval = pam_get_item(pamh, PAM_CONV, (void*)&conv);
   if (conv)
   {
      retval = conv->conv(NUM_MSGS, &pam_msg, &pam_resp, conv->appdata_ptr);
   }

   if (retval == PAM_SUCCESS)
   {
      for (i=0; i<NUM_MSGS; i++)
      {
         if(strcmp(pam_resp[i].resp, "reject") ==0)
            retval = PAM_PERM_DENIED;
         free(pam_resp[i].resp);
      }
      free(pam_resp);
   }

   return retval;
}
```

PAM Test Program

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <pwd.h>
#include <security/pam_appl.h>


#define USER    (argv[1])
#define SERVICE (argv[2])
#define RHOST   (argv[3])


int conv_func(int num_msg, struct pam_message *msg[],
         struct pam_response **resp, void *appdata_ptr)
{
   int i;
   struct pam_message *msgs = *msg;
   struct pam_response *aresp = calloc(num_msg, sizeof(*aresp));
   char resp_buf[PAM_MAX_RESP_SIZE];


   for (i=0; i<num_msg; i++)
   {
      aresp[i].resp_retcode = 0;
      fputs(msgs[i].msg, stderr);
      fgets(resp_buf, sizeof(resp_buf), stdin);
      resp_buf[strlen(resp_buf)-1] = '\0';
      aresp[i].resp = strdup(resp_buf);
   }
   *resp = aresp;
   return PAM_SUCCESS;
}
```

PAM Test Program

```
int main(int argc, char *argv[])
{
   int rc = 0, retval = 0;
   struct passwd *pw;
   struct pam_conv conv = {conv_func, NULL};
   pam_handle_t *pamh = NULL;


   if (argc != 3 && argc != 4)
   {
      printf("Usage: %s <user> <service> [<rhost>]\n", argv[0]);
      return 1;
   }


   pw = getpwnam(USER);
   if (pw)
      printf("Name: %s\nUID: %d\nGID: %d\nHome: %s\n",
        pw->pw_name, pw->pw_uid, pw->pw_gid, pw->pw_dir);
   else
   {
      printf("User %s does not exist.\n", USER);
      rc = 1;
      goto outta_here;
   }


   retval = pam_start(SERVICE, USER, &conv, &pamh);
   if (retval != PAM_SUCCESS)
   {
      printf("Error from pam_start\n%s\n", pam_strerror(pamh, retval));
      rc = retval;
      goto pam_cleanup;
   }
   printf("Pam Handle: 0x%x\n", pamh);


   retval = pam_set_item(pamh, PAM_RUSER, USER);
   if (retval != PAM_SUCCESS)
   {
      printf("Error from pam_set_item - ruser: %d\n%s\n", retval, pam_strerror(pamh, retval));
      rc = retval;
      goto pam_cleanup;
    }
```

PAM Test Program

```
 if (RHOST)
   {
     retval = pam_set_item(pamh, PAM_RHOST,
     RHOST); if (retval != PAM_SUCCESS)
     {
        printf("Error from pam_set_item - rhost: %d\n%s\n", retval, pam_strerror(pamh,
        retval)); rc = retval;
        goto pam_cleanup;
     }
   }


   retval = pam_authenticate(pamh,
   0); if (retval != PAM_SUCCESS)
   {
      printf("Error from pam_authenticate: %d\n%s\n", retval, pam_strerror(pamh,
      retval)); rc = retval;
      goto pam_cleanup;
   }


   retval = pam_acct_mgmt(pamh,
   0); if (retval != PAM_SUCCESS)
   {
      printf("Error from pam_acct_mgmt: %d\n%s\n", retval, pam_strerror(pamh,
      retval)); rc = retval;
      goto pam_cleanup;
   }


   if (!rc)
      printf ("User %s is authorized for service %s!\n", USER, SERVICE);


pam_cleanup:
   retval = pam_end(pamh,
   0); if (retval !=
   PAM_SUCCESS)
   {
      printf("Error from pam_end: %d\n%s\n", retval, pam_strerror(pamh,
      retval)); rc = retval;
   }

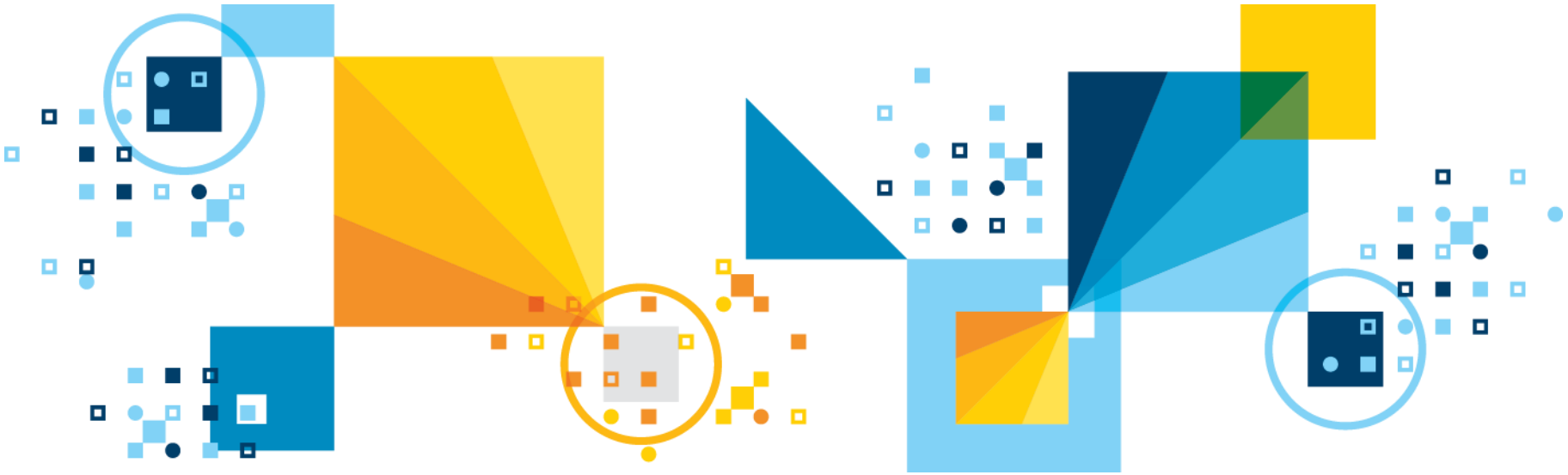

outta_here:
   return rc;
}
```

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

SSL – Encrypted Connections Client to Server



Why Encrypt over the wire?

- **Privacy reasons**
  - A simple sniffer can read data if no encryption
- **Regulatory Compliance**

What is SSL?

- **(S)ecure (S)ocket (L)ayer**
  - A communication protocol, similar to TCP, but has encryption enabled over the wire.

Why use SSL?

- Gives you a clear standard
- Less configuration files
- Allows certificates to be managed by established 3rd partners

Setting up SSL SQLHOSTS

- **SSL communication protocol**
  - **drsocssl** protocol for supporting SSL communication with DRDA clients
  - **onsocssl/olsocssl** protocol for supporting SSL communication with SQLI clients. SSL will also be supported with server to server communication (ISTAR, HDR, ER, SDS/RSS)

- **Example**
  - **portland_on drsocssl pinchy portland_serv**

Setting up SSL - ONCONFIG

- **SSL specific parameters**
  - **SSL_KEYSTORE_LABEL**
    - Specifies label of server digital certificate in keystore. If not configured, the server will use the default label in keystore for SSL communication
  - e.g. **SSL_KEYSTORE_LABEL ids_label**

- **Changes to existing parameters**
  - **NETTYPE** – Describes connection parameters such as number of poll threads, max connections and class of virtual process for poll threads for connection protocols
  - **NETTYPE** protocol, poll threads, connections, VP class
  - Specify the protocol as iiippp
    - where: iii=[ipc|ipc|soc|tli]
  - ppp=[shm|str|tcp|spx|imc|ssl]
    - e.g. **NETTYPE socssl, 3, 50, NET**

Setting up SSL - ONCONFIG

- **All SSL encryption/decryption operations are performed on encrypt VP. Encrypt VPs can be configured via VPCLASS parameter**
  - e.g. **VPCLASS encrypt,num=5**

- **SSL and non-SSL connection protocols can be configured for a single instance using server aliases**

- **DBSERVERNAME - menlo_on**
- **DBSERVERALIASES – lenexa_on, portland_on**

where **menlo_on** is **onsocssl**, **lenexa_on** is **onsoctcp** and **portland_on** is **drsocssl** connection protocol

Setting up SSL – Keystores and Digital Certificates

- **IBM's Global Security Kit (GSKit) will be installed as part of IDS and CSDK installations**

- **GSKit contains iKeyman utility that can be used to create keystores and manage digital certificates needed for SSL communication**

A PSA on GSKIT

- **GSkit is an IBM standard**
  - This means most products in IBM have adopted it.
- **Informix's install script for GSKIT is pretty good**
  - Not true of all products
- **Make sure the product with the most recent version of GSKIT is installed last.**

Setting up SSL – Keystores and Digital Certificates

- **The keystore for server is password protected. Password is stored encrypted in the stash file (also created by iKeyman utility)**

- **One keystore per server instance. It stores server's digital certificate and root CA certificates of other servers its connecting to (as in ISTAR, HDR, ER, SDS/RSS)**

- **Location and name of server keystore and its password stash file is predefined:**
  - **$INFORMIXDIR/ssl/<servername>.kdb**
  - **$INFORMIXDIR/ssl/<servername>.sth**
  - The ownership/permissions of above files must be **informix:informix/600**
- **<servername> is value of DBSERVERNAME onconfig parameter**

Setting up SSL – Keystores and Digital certificates

- **Password is optional for client keystore.**
- **Client keystore stores root CA certificates of all servers the client is connecting to. SQLI and DRDA clients can share same keystore**
- **Location and name of client keystore and its password stash file can be configured via new configuration file:**
  - **$INFORMIXDIR/etc/conssl.cfg**
- **New client configuration parameters:**
  - **SSL_KEYSTORE_FILE** – Specifies fully qualified filename of client keystore
  - **SSL_KEYSTORE_STH** – Specifies fully qualified filename of client stash file
- **If conssl.cfg does not exist or if any of above parameters are not configured, the keystore and stash file will default to:**
  - **$INFORMIXDIR/etc/client.kdb** and **$INFORMIXDIR/etc/client.sth**
- **The ownership/permissions of above files must be informix:informix/664**

Setting up SSL – Keystores and Digital certificates

- Prerequisites for iKeyman utility (gsk8cmd, gsk8ikm)
- IBM JDK/JRE 1.3.1, 1.4.1 or higher with JCE PKS Security packages

- Environment for iKeyman utility:

```
export JAVA_HOME=<JDK/JRE installation>
export PATH=$JAVA_HOME/jre/bin:$PATH
export CLASSPATH=<GSKit installation>/classes/cfwk.zip:<GSKit
installation>/classes/ gsk8cls.jar:$JAVA_HOME/jre/lib/ext/ibmpkcs11.jar
```

- GSKit also has a non--java utility (gsk8capicmd) for administering keystores.

Setting up SSL – Keystores and Digital Certificates

- **Sample commands for creating keystore and self--signed test certificates using iKeyman command line utility:**

- **Server Keystore**

  gsk8cmd --keydb --create --db menlo_on.kdb --pw snoopy --type cms --stash

  gsk8cmd --cert --create --db menlo_on.kdb --pw snoopy --label ids_label --dn "CN=menlo.ibm.com,O=ibm,C=US" --size 1024 --default_cert yes

  gsk8cmd --cert --extract --db menlo_on.kdb --format ascii --label ids_label --pw snoopy --target ids_label.cert

  – where **DBSERVERNAME** is **menlo_on**

  SSL_KEYSTORE_LABEL is **ids_label**

- **Client Keystore**

  gsk8cmd --keydb --create --db client.kdb --pw snoopy --type cms --stash

  gsk8cmd --cert --add --db client.kdb --pw snoopy --label ids_label --file ids_label.cert --format ascii.

RECAP of SSL Setup

- **sqlhosts for client and server**
  - **onsocssl/drsocssl**
- **onconfig for server**
  - **SSL_KEYSTORE_LABEL**
  - **NETTYPE** for **socssl**
  - **VPCLASS** for **encrypt** VP
- **conssl.cfg for client**
  - **SSL_KEYSTORE_FILE**
  - **SSL_KEYSTORE_STH**
- **Keystores and digital certificates for client and server**
- **Once setup is complete, initialize server and all communication between client and server or between servers on onsocssl/drsocssl port will be encrypted using SSL protocol**

Creating a Self Signed SSL Setup

- **Problem**

- **A means to have a single certificate for all boxes, so that there is reduced management costs and makes clients easy to connect to multiple Informix servers using SSL.**

Self Signed is the Solution

- Solution : SSL keystores are platform independent.
- So all we need to do is create one server
- keystore and one client keystore using the “default” label , and then copy this to every machine.

Use Case : The raptor Files

- We have a machine called raptor that needs to use SSL but needs a self signed setup, what do we do? (note machine name raptor, and sqlexec_ssl is in /etc/services)

Use Case : The raptor Files

- **First setup the informix server for that.**
  - Put the following in **$ONCONFIG**

    VPCLASS encrypt,num=3

    NETTYPE socssl,3,100,NET

    DBSERVERALIASES raptorssl
  - Next modify the **SQLHOSTS** FILE

    raptorssl onsocssl raptor sqlexec_ssl

Use Case : The raptor Files

- **First setup the informix server for that.**
  - Put the following in **$ONCONFIG**

    VPCLASS encrypt,num=3

    NETTYPE socssl,3,100,NET

    DBSERVERALIASES raptorssl
  - Next modify the **SQLHOSTS** FILE

    raptorssl onsocssl raptor sqlexec_ssl

Use Case : The raptor Files

- **Create the server and certificate (must be in /usr/informix_engine/ssl**

- **Run the following commands:**

  gsk8capicmd --keydb --create –db raptor.kdb –pw raptor --type cms --stash

  gsk8capicmd --cert --create –db raptor.kdb –pw raptor --label default --dn "CN=rxp,O=ibm,C=US" -- size 1024 --default_cert yes

  gsk8capicmd --cert --extract --db rxp_s05575_us.kdb --format ascii --label default –pw raptor --target default.cert

Use Case : The raptor Files

- **Create the client keystore (must be done in /usr/informix_engine/etc)**

```
gsk8capicmd --keydb --create --db client.kdb --type cms --stash
gsk8capicmd --cert --add --db client.kdb --label default --fille
../ssl/default.cert --format ascii
onmode -cky
```

Additional Considerations

- **To get a client working on another machine:**
  - copy **/usr/informix_engine/etc/client*** to **$INFORMIXDIR/etc** of the box you want to be able to connect.

- **To get another server working with SSL:**
  - Copy the contents of **/usr/informix_engine/ssl** on **raptor** to the server you want to enable
  - Rename all the files from **raptor** to the **DBSERVERNAME** of the box you moved to

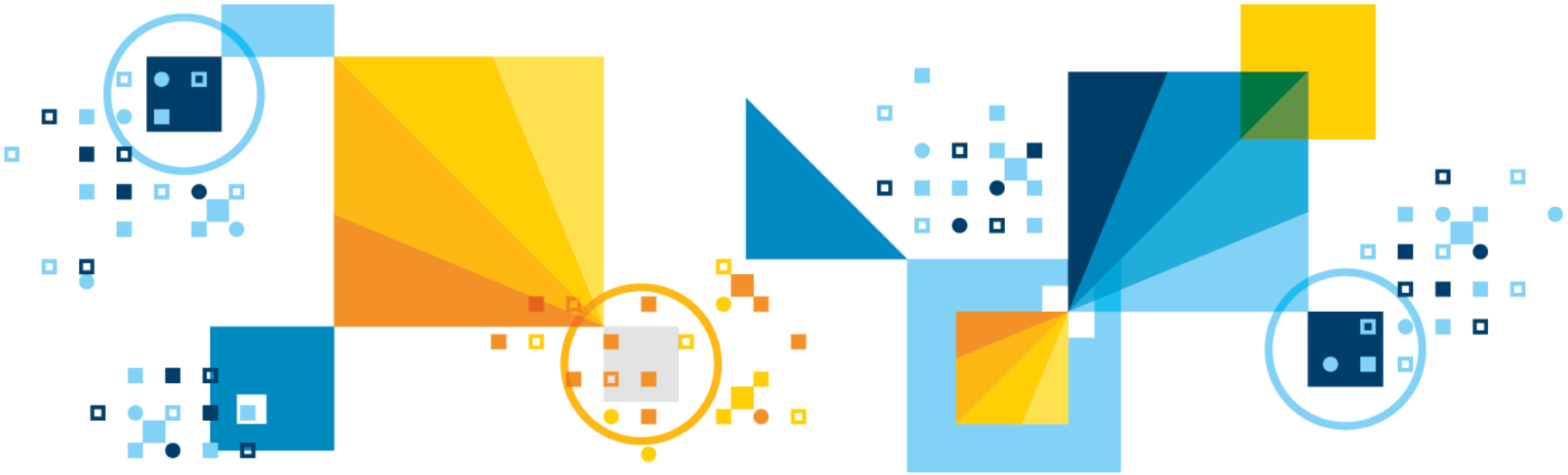
- **onmode -cky**

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

onsecurity - Secure Install Directory



onsecurity

- **The Informix onsecurity utility checks the security of a Unix/Linux file, directory, or path. It also troubleshoots the security problems if any are detected.**
  - Obviously, these objects and issues have to be **Informix** related.

- **Used to:**
  - Check whether a path leading to a directory or a file is secure.
  - Generate diagnostic output that explains the nature of the security problem.
  - Generate a script that can be run user **root** to remedy the security problems.
    - Use the script as generated or modify to meet your environment's security requirements.
  - Specify particular users, groups, or directories, normally not trusted, can be trusted by the Informix utilities; but in special circumstances only and add this information to files in the **/etc/informix** directory.
- **Helps to keep unwanted viruses away from your Informix critical files and directories**

onsecurity

- **When this command is run to check on the Informix installation path, you most often receive a message that the path is secure. If the path is secure, you are not required to do any further work with the utility for the path.**
  - Many customers load this into system scheduler to be run periodically.
    - Every 5 minutes is a good idea … does not interfere with operations.

- **onsecurity does not change file permissions. It supports an extensive set of options to specify how the problem is fixed, by request, generates script that user root runs to modify permissions or settings.**

- **Changes to file or directory permissions are an indirect result from user root running the script that onsecurity generates; onsecurity itself does not make these changes.**

onsecurity

- onsecurity has a number of options, a few of which are shown below

- The following example shows the onsecurity execution output on a secure path:

```
$ onsecurity /usr/informix/12.10.FC8 # /usr/informix/12.10.FC8
resolves to /work4/informix/Operational/12.10.FC8 (path is trusted)
```

- In the preceding example, the specified path /usr/informix/12.10.FC8 traverses at least one symbolic link to end up at the actual directory /work4/informix/Operational/12.10.FC8, but the whole path is secure.

onsecurity

- **The following example shows the output from running onsecurity on a path that is not secure:**

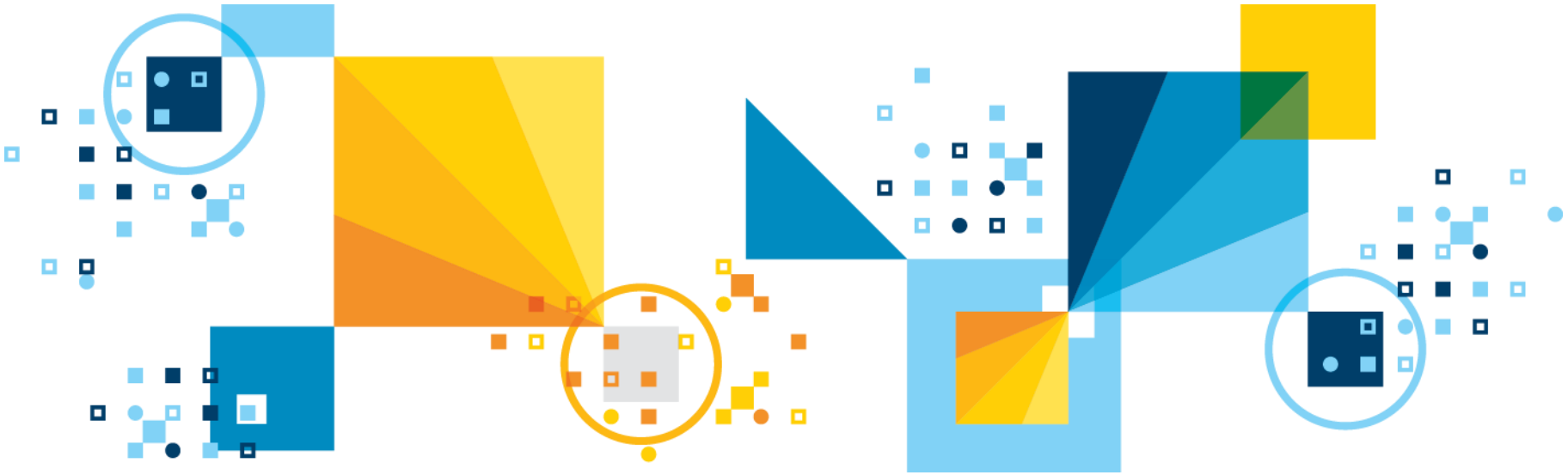
```
$ onsecurity /work/informix/ids-12
# !!! SECURITY PROBLEM !!!
# /work/informix/ids-12 (path is not trusted)
# Analysis: # User              Group      Mode Type Secure Name
#              0 root              0 root       0755   DIR  YES      /
#              0 root              0 root       0755   DIR  YES      /work
#            203 unknown        8714 ccusers    0777   DIR   NO      /work/informix
#            200 informix        102 informix   0755   DIR   NO      /work/informix/ids-12
#
# Name: /work/informix # Problem: owner <unknown> (uid 203) is not trusted
# Problem: group ccusers (gid 8714) is not trusted but can modify the directory
# Problem: the permissions 0777 include public write access
```

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

REST with HTTPS



Agenda

- REST Architecture and Supported Functions
- HTTPS and REST
- **HTTP and SSL**

REST Architecture

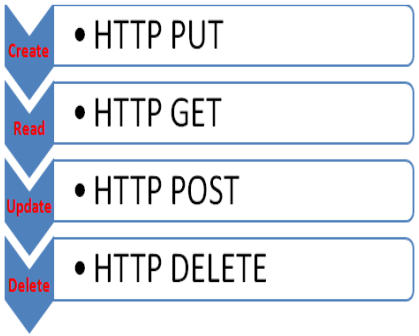
- **Distributed communication architecture**
- **Widely popular in cloud environments**
- **REST**
  - An architectural style for web based communication
  - Permits clients to communicate with servers in a unique manner
  - Represents resources (databases in this case) as URI's
  - Architecture uses **HTTP** protocol
  - A set of operations (**GET**/**POST**/**PUT**/**DELETE**) permit manipulation of resources
- **RESTful architectures are stateless**
  - Server does not maintain any client context between transactions
  - Transaction contains all information necessary to satisfy the particular request.
  - Makes **REST**ful architectures more reliable and helps to expand their scalability.

The strength of REST
REST is an architectural style, not a protocol or an implementation. REST has some core principles, but in the end, it's an abstraction, not a specific implementation.
(Source: http://www.ibm.com/developerworks/library/os-understand-rest-ruby/)

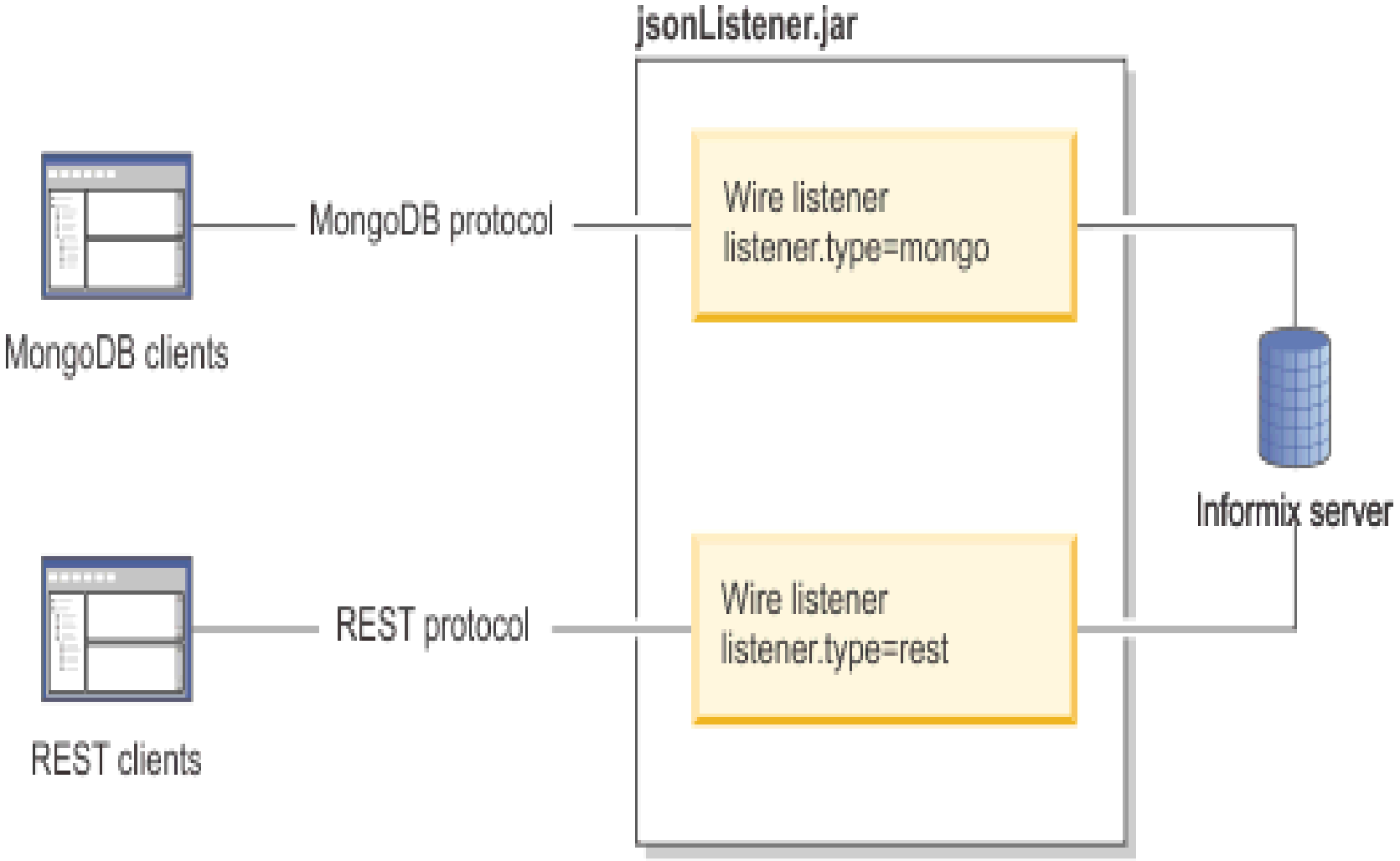
Access Informix from REST API Clients

- **Directly connect applications or devices that communicate through the REST API to Informix**
  - No client side drivers needed, freedom from client dependency
  - Web applications can connect seamlessly to the database using HTTP protocol
  - Create connections by configuring the wire listener for the REST API
  - Use MongoDB and SQL queries against JSON and BSON document collections, traditional relational tables, and time series data
  - The REST API uses MongoDB syntax and returns JSON documents
  - Widely popular in Cloud / IOT architectures
  - Simplify web application development in Cloud environments

- **Subset of the HTTP protocol (GET / POST / DELETE / PUT) supported**
  - **POST** method maps to mongo db insert or create command
  - **GET** method maps to mongo db query command
  - **PUT** method maps to mongo db update command
  - **DELETE** method maps to mongo db delete command



Access Informix from REST API Clients (contd)



Wire Listener & REST (1)

- **The wire listener is a mid-tier gateway server that enables communication between MongoDB applications and the Informix® database server.**
- **The wire listener is provided as an executable JAR file that is named $INFORMIXDIR/bin/jsonListener.jar.**
  - The JAR file provides access to the MongoDB API and REST API.
- **You can connect to a JSON collection by using the REST API.**
- **When a client is connected to the wire listener by using the REST API, each database is registered; events such as create or drop a database.**
- **If a REST request refers to a database that exists but is not registered, the database is registered and a redirect to the database root returned.**

Wire Listener & REST (2)

- **The JSONListener.properties file has an optional parameter called listener.type It specifies the type of wire listener to start:**
  - The default is **mongo** which connects the wire listener to the MongoDB API
    - **listener.type=mongo**
- **To connect to a REST API, connect to the wire listener, connect the wire listener to the REST API using the following parameter value which must be specified to use the REST API:**
  - **listener.type=rest**
- **There are some new REST related optional parameters for the JSONListener.properties file which may be necessary for use**

Multiple wire listeners configuration (1)

- **You can run multiple wire listeners at the same time to access both Mongo and REST data, by creating a properties file for each:**
  - Create each properties file in the **$INFORMIXDIR/etc** directory using the **$INFORMIXDIR/etc/jsonListener-example.properties** file as a template
  - Customize each properties file and assign a unique name:
    - The **url** parameter **must** be specified, either in each individual properties file or in the file that is referenced by the include parameter.
      - Optional: Specify the **include** parameter to reference another properties file. The path can be relative or absolute.
    - If you have multiple properties files, you can avoid duplicating parameter settings in the multiple properties files by specifying a subset of shared parameters in a single properties file, and the unique parameters in the individual properties files.
  - Start the wire listeners.

Multiple wire listeners configuration (2) - Example

- The same url, authentication.enable, and security.sql.passthrough parameters are used to run two separate wire listeners:

- Create a properties file named shared.properties that includes the following parameters
  - :url=jdbc:informix-sqli://localhost:9090/sysmaster: INFORMIXSERVER=lo_informix1210; authentication.enable=true security.sql.passthrough=true

- Create a properties file for use with the MongoDB API that is named mongo.properties, with the parameter setting include=shared.properties included:
  - include=shared.properties listener.type=mongo listener.port=27017

- Create a properties file for use with the REST API that is named rest.properties, with the parameter setting include=shared.properties included:
  - include=shared.properties listener.type=rest listener.port=8080

Multiple wire listeners configuration (3) - Example

- **Start the wire listeners by using the command line:**

  java -jar jsonListener.jar -start
  -config json.properties
  -config rest.properties

HTTP: POST

- The POST method maps to the MongoDB insert or create command.

Method	Path	Description
POST	/	Create a database
POST	/databaseName	Create a collection databaseName – database name
POST	/databasename/collectionName	Create a document databaseName – database name collectionName – collection name

HTTP: POST – Create a database

- With the locale specified.
- Request: Specify the POST method:
  - POST / Data:
- Specify database name mydb and an English UTF-8 locale:
  - {name:"mydb",locale:"en_us.utf8"}
- Response: The following response indicates that the operation was successful:
  - Response does not contain any data.

HTTP: POST – Collection Creation

- Creates a collection in the mydb database.
- Request: Specify the POST method and the database name as mydb:
  - POST /mydb
- Data: Specify the collection name as bar:
  - {name:“bar”}
- Response: The following response indicates that the operation was successful:
  - {"msg":"created collection mydb.bar","ok":true}

HTTP: POST – Relational Table Creation

- This example creates a relational table in an existing database.
- Request: Specify the POST method and stores_mydb as the database:
  - POST /stores_mydb
- Data: Specify the table attributes:
  - { name: "rel", columns: [{name:"id",type:"int",primaryKey:true,}, {name:"name",type:"varchar(255)"}, {name:"age",type:"int",notNull:false}]}
- Response: The following response indicates that the operation was successful:
  - {msg: "created collection stores_mydb.rel" ok: true}

HTTP: POST – Insert a Single Document

- Inserts a document into an existing collection.
- Request: Specify the POST method, mydb database, and people collection:
  - POST /mydb/people
- Data: Specify John Doe age 31:
  - {firstName:"John",lastName:"Doe",age:31}
- Response: Because the _id field was not included in the document, the automatically generated _id is included in the response. Here is a successful response:
  - {"id":{"$oid":"537cf433559aeb93c9ab66cd"},"ok":true}

HTTP: POST – Insert Multiple Documents

- This example inserts multiple documents into a collection.
- Request: Specify the POST method, mydb database, and people collection:
  - POST /mydb/people
- Data: Specify John Doe age 31 and Jane Doe age 31:
  - [{firstName:"John",lastName:"Doe",age:31}, {firstName:"Jane",lastName:"Doe",age:31}]
- Response: Here is a successful response:
  - {ok: true}

HTTP: GET

- **The GET method maps to the MongoDB query command.**

Method	Path	Description
GET	**/**	List all databases
GET	**/databaseName**	List all collections in a database **databaseName** – database name
GET	**/databasename/collectionName?*queryParameters***	Query a collection **databaseName** – database name **collectionName** – collection name ***queryParameters*** - The query parameters. The supported Informix ***queryParameters*** are: **batchSize**, **query, fields**, and **sort**. These map to the equivalent MongoDB **batchSize**, **query**, **fields**, and **sort** parameters.

HTTP: GET – List All Databases on the Server

- **Specify the GET method and forward slash (/):**
  - **GET /**
- **Data: None.**
- **Response: Here is a successful response:**
  - **["mydb" , "test“]**

HTTP: GET – List All Collections in a Database

- **Request: Specify the GET method and mydb database:**
  - **GET /mydb**
- **Data: None.**
- **Response: Here is a successful response:**
  - **["bar"]**

HTTP: GET – Show Sorted Data from a Collection

- **This example sorts the query results in ascending order by age.**
- **Request: Specify the GET method, mydb database, people collection, and query with the sort parameter.**
  - The sort parameter specifies ascending order **(age:1),** and filters id (_id:0) and last name (**lastName:0**) from the response
  - **GET /mydb/people?sort={age:1}&fields={_id:0,lastName:0}**
- **Data: None.**
- **Response: The first names are displayed in ascending order with the _id and lastName filtered from the response:**
  - **[{"firstName":"Sherry","age":31}, {"firstName":"John","age":31}, {"firstName":"Bob","age":47}, {"firstName":"Larry","age":49}]**

HTTP: PUT

- **The PUT method maps to the MongoDB update command.**

Method	Path	Description
PUT	**/databasename/collectionName *?queryParameters***	Update a document **databaseName** – database name **collectionName** – collection name **queryParameters -** The supported Informix ***queryParameters*** are **query**, **upsert**, and **multiupdate.** These map to the equivalent MongoDB query, insert, and multi query parameters, respectively.**-**

HTTP: PUT – Document Update in a Collection

- **Update the value for Larry in an existing collection, from age 49 to 25:**
  - **[{"_id":{"$oid":"536d20f1559a60e677d7ed1b"},"firstName":"Larry","lastName":"Doe","age":49},{"_id":{"$oid":"536d20f1559a60e677d7ed1c"},"firstName":"Bob","lastName":"Doe","age":47}]**
- **Request: Specify the PUT method and query the name Larry:**
  - **PUT /?query={name:"Larry"}**
- **Data: Specify the MongoDB $set operator with age 25:**
  - **{"$set":{age:25}}**
- **Response: Here is a successful response:**
  - **{"n":1,"ok":true}**

HTTP: DELETE

- **The DELETE method maps to the MongoDB delete command.**

Method	Path	Description
DELETE	**/**	Delete all databases
DELETE	**/databaseName**	Delete a database **databaseName** – database name
DELETE	**/databasename/collectionName**	Delete a collection **databaseName** – database name **collectionName** – collection name
DELETE	**/databasename/collectionName ?queryParameter**	Delete a document **databaseName** – database name **collectionName** – collection name ***queryParameter*** - The query parameter. The supported Informix ***queryParameter*** is **query**. This maps to the equivalent MongoDB query parameter.

HTTP: DELETE (1) – Database Deletion

- Delete a database called mydb.
- Request: Specify the DELETE method and the mydb database:
  - DELETE /mydb
- Data: None.
- Response: Here is a successful response:
  - {msg: "dropped database"ns: "mydb"ok: true}

HTTP: DELETE (2) – Collection deletion

- This example deletes a collection from a database.
- Request: Specify the DELETE method, mydb database, and bar collection:
  - DELETE /mydb/bar
- Data: None.
- Response: Here is a successful response:
  - {"msg":"dropped collection""ns":"mydb.bar""ok":true}

The Wire Listener and REST

- **The wire listener is a mid-tier gateway server that enables communication between MongoDB, REST API, and MQTT clients and the Informix database server.**

- **Wire listener is a Java application and an executable JAR file, $INFORMIXDIR/bin/jsonListener.jar, included with the database server:**
  - JAR file provides access to the MongoDB and REST API's, and MQTT protocol.

The Wire Listener and Supported features

- **MongoDB API access**
  - Connect to a JSON collection via a MongoDB API via a MongoDB Wire Protocol
  - When a MongoDB client is connected to the wire listener and requests a connection to a database, the wire listener creates a connection.
- **REST API access**
  - Connect to a JSON collection by using the REST API.
    - If a client connects to the wire listener thru the REST API, each database is registered.
    - The wire listener registers to receive session events such as create or drop a database.
    - If a REST request refers to a database that exists but is not registered, the database is registered and a redirect to the root of the database is returned.
- **MQTT protocol access**
  - Connect to a JSON collection via the MQTT protocol.
  - When an MQTT client publishes data to the wire listener, the wire listener creates a connection to the database for inserting the data.
- **The wire listener connection properties file, default named jsonListener.properties, defines every operational characteristic.**

The Wire Listener and Supported Features

- **When you create a database or a table through the wire listener, automatic location and fragmentation are enabled**
- **Databases are stored in the dbspace that is chosen by the server**
- **Tables are fragmented among dbspaces that are chosen by the server**
  - More fragments are added when tables grow
- **The default logging mechanism for the wire listener is Logback**
  - Logback is pre-configured and installed along with the JSON components

Configuring the Wire Listener for the First Time

- **Configure the wire listener by specifying an authorized user and customizing the wire listener configuration file.**

- **Before you begin**
  - The wire listener .JAR file is included in the database server installation.
  - If you created a server instance as a part of the Informix installation process, the wire listener is configured with default properties and started:
    - Wire listener configuration file, **$INFORMIXDIR/etc/jsonListener.properties**, is created.
  - **ifxjson** user, which has **REPLICATION** privilege group access, is created and added to the **url** parameter in the wire listener configuration file.
    - This user ID is used by the wire listener to connect to Informix.
  - Wire listener is started and connected to the MongoDB API and database server.

- **To use the REST API or MQTT protocol, or make other changes, edit the wire listener configuration file and restart the wire listener.**

Configuring the Wire Listener for the First Time

- **Choose an authorized user whom must have access to the databases and tables that are accessed through the wire listener.**
  - Windows:
    - Specify an operating system user.
  - UNIX or Linux:
    - Specify an operating system user or a database user.
    - For example, here is the argument to create a database user in UNIX or Linux:
      - **CREATE USER userID WITH PASSWORD 'password' ACCOUNT unlock PROPERTIES USER daemon;**

Configuring the Wire Listener for the First Time (optional)

- **If you want to shard data, grant the user REPLICATION privilege by running the admin or task SQL Admin API command with the grant admin argument**
  - User **ifxjson** has **REPLICATION** privilege:
    - **EXECUTE FUNCTION task('grant admin','userID','replication');**

- **Create a wire listener configuration file in $INFORMIXDIR/etc with the .properties file extension. Use the file $INFORMIXDIR/etc/jsonListener-example.properties as a template.**

- **Customize the wire listener configuration file to your needs:**
  - To include parameters in the wire listener, uncomment the row and customize the parameter
  - The **url** parameter is required
  - All other parameters are optional

Configuring the Wire Listener for the First Time

- **Review the defaults for the following parameters and verify that they are appropriate for your environment; minimally:**
  - **mongo.api.version**
  - **authentication.enable**
  - **listener.type**
  - **listener.port**
  - **listener.hostName**

Configuring the Wire Listener for the First Time

- **If you are using a Dynamic Host Configuration Protocol (DHCP) on your IPv6 host, you must verify that the connection information between JDBC and Informix is compatible.**

- **Connect from the IPv6 host through an IPv4 connection by using the following steps:**
  - Add a server alias to the **DBSERVERALIASES** configuration parameter for the wire listener on the local host; for example:
    - **lo_informix1210.**
  - Add an entry to the **sqlhosts** file for the database server alias to the loopback address 127.0.0.1:
    - **ol_informix1210 onsoctcp 127.0.0.1 9090**
  - In the wire listener configuration file, update the **url** entry with the wire listener alias. For example:
    - **url=jdbc:informix-sqli://localhost:9090/sysmaster:**
    - **INFORMIXSERVER=ol_informix1210;**

The Wire Listener Configuration File

- **Contains the settings that control the wire listener and the connection between the client and database server.**
  - It's default name is **$INFORMIXDIR/etc/jsonListener.properties**; it can be renamed, but its suffix must be **.properties**.

- **If a server instance is created during the installation process, a configuration file named jsonListener.properties is automatically created with default properties; otherwise this file must be manually created:**
  - Use **$INFORMIXDIR/etc/jsonListener-example.properties** as a template
  - Make a copy, don't edit the template

- **In the configuration file that is created during installation, and in the template file, all of the parameters are commented out by default:**
  - To enable a parameter, uncomment the row and customize the parameter

Wire Listener Configuration File – Default Sample When Created by the Installer (Windows):

```
listener.port=6462
url=jdbc:informix-sqli://localhost:5853/sysmaster:INFORMIXSERVER=lo_informix1210_5
```

url Parameter

- **url**
  - Required parameter specifies the host name, port number, user ID, and password that are used in connections to the database server.
    - Must specify the **sysmaster** database in the **url** parameter. That database is used for administrative purposes by the wire listener.

```
>>-url=--jdbc:informix-sqli://hostname:portnum--/sysmaster:----->

>--+-----------------------------------------+------------><
   '-USER=userid;--PASSWORD=password--NONCE=value-'
```

  - Include additional JDBC properties in the **url** parameter such as **INFORMIXCONTIME**, **INFORMIXCONRETRY**, **LOGINTIMEOUT**, and **IFX_SOC_TIMEOUT**.

url Parameter (cont'd)

- **hostname:portnum**
  - The host name and port number of your computer.
  - For example, **localhost:9090.**

- **USER=userid**
  - Optional attribute, specifies the user ID used in Informix database server connections.
  - If you plan to use this connection to establish or modify collection shards by using the Informix sharding capability, the specified user must be granted the **REPLICATION** privilege group access.
  - If the user ID and password is not specified, the JDBC driver uses O/S authentication and all wire listener actions are run by using the user ID and password of the O/S user who runs the wire listener **start** command.

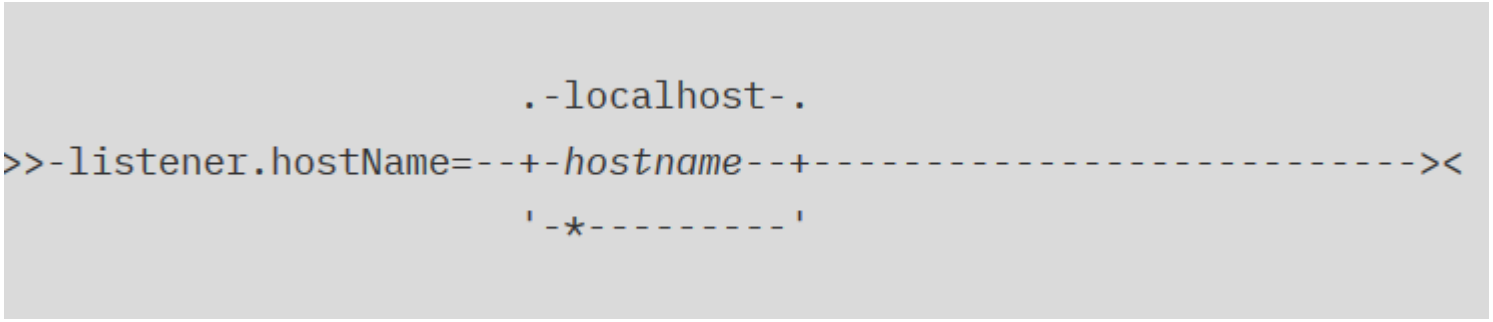
- **PASSWORD=password**
  - Optional attribute, specifies the password for the specified user ID.

url Parameter (cont'd)

- **NONCE=value**
  - Optional attribute, specifies a 16-character value that consists of numbers and the letters a, b, c, d, e, and f.
  - Property triggers password encoding when a pluggable authentication module is configured for the wire listener.
  - Applicable only if the **db.authentication** parameter is set to **informix-mongodb-cr.**

Other Parameters Worth Setting Initially – listener.hostName

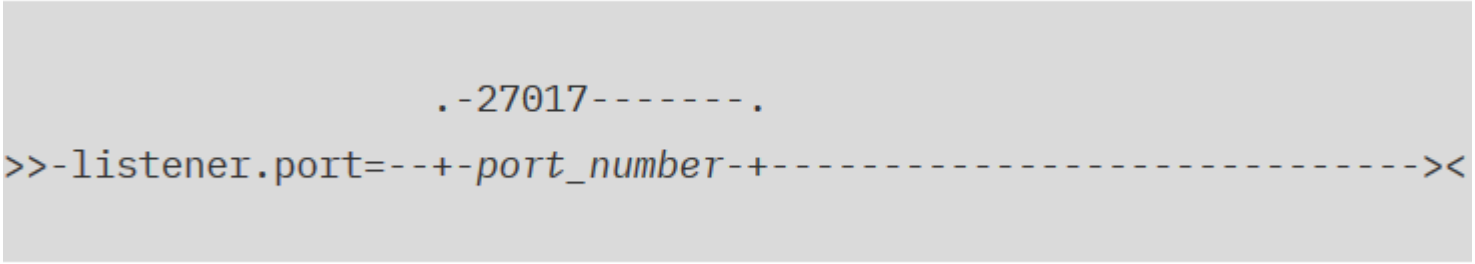
- **listener.hostName**
  - Optional parameter specifies the host name of the wire listener.
  - The host name determines the network adapter or interface that the wire listener binds the server socket to.
  - If you enable the wire listener to be accessed by clients on remote hosts, turn on authentication by using the **authentication.enable** parameter.



  - **localhost**
    - Bind the wire listener to the localhost address.
    - (default) Wire listener not accessible from clients on remote machines.
  - ***hostname***
    - The host name or IP address of host machine where the wire listener binds to.
  - * The wire listener can bind to all interfaces or addresses.

Other Parameters Worth Setting Initially – listener.port

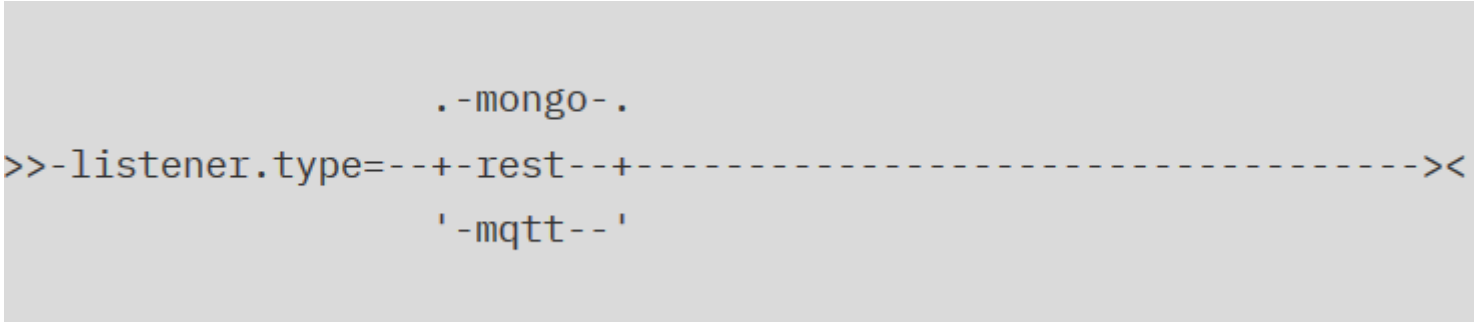
- **Optional parameter specifies the port number to listen on for incoming connections from clients:**
  - Value can be overridden from the command line by using the **-port** argument.
  - The default value is **27017**.
    - This is the default port for a MongoDB database connection.
  - A port number less than **1024** requires the user that starts the wire listener to



Other Parameters Worth Setting Initially – listener.type

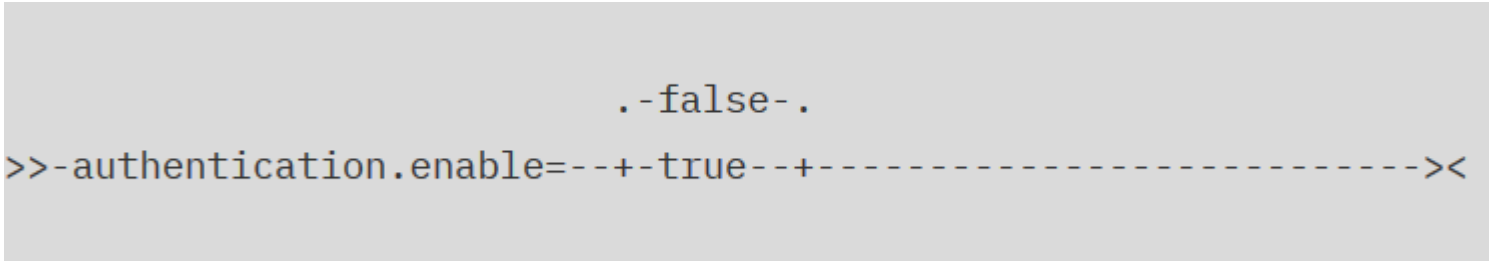
- **Optional parameter specifies the type of wire listener to start.**

- **There are presently three wire listener types supported:**
  - **mongo**
    - Connect the wire listener to the **MongoDB API**.
    - This is the default value.
  - **rest**
    - Connect the wire listener to the **REST API.**
  - **mqtt**
    - Connect the wire listener to the **MQTT** protocol.



Other Parameters Worth Setting Initially – authentication.enable

- **Optional parameter indicates whether to enable user authentication.**
- **Choose to authenticate users through the wire listener or in the database server.**



- **false**
  - Do not authenticate users.
  - This is the default value.
- **true (recommended)**
  - Authenticate users.
  - Use the **authentication.localhost.bypass.enable** parameter to control the type of authentication.

Other Parameters Worth Setting Initially – authentication.localhost.bypass.enable

- **Set this only if authentication.enable=true**

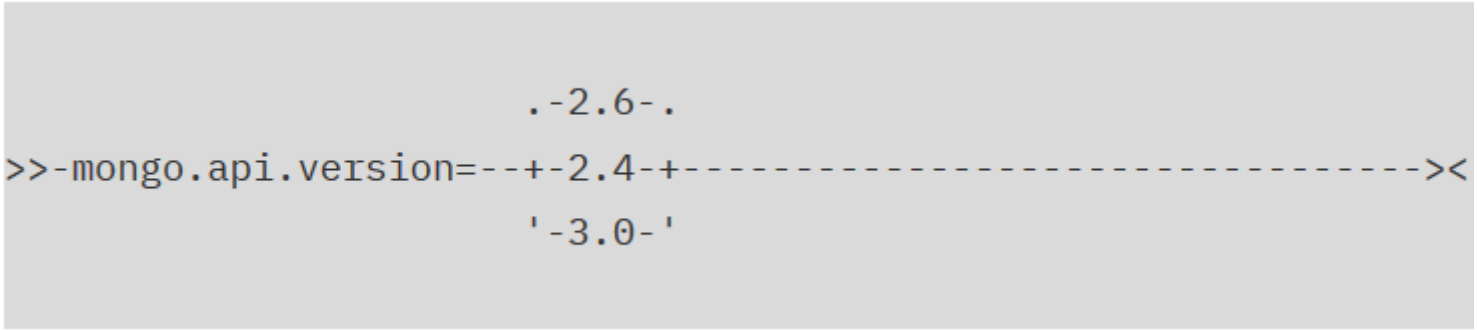
- **Optional parameter indicating whether to grant full administrative access if you connect from the localhost to the Informix admin database, and the admin database (similar to the MongoDB admin database) contains no users**

```
                                              .-true--.
>>-authentication.localhost.bypass.enable=--+-false-+----------><
```

- **true**
  - Grant full administrative access to the user.
  - This is the default value.
- **false**
  - Do not grant full administrative access to the user.

Other Parameters Worth Setting Initially – mongo.api.version

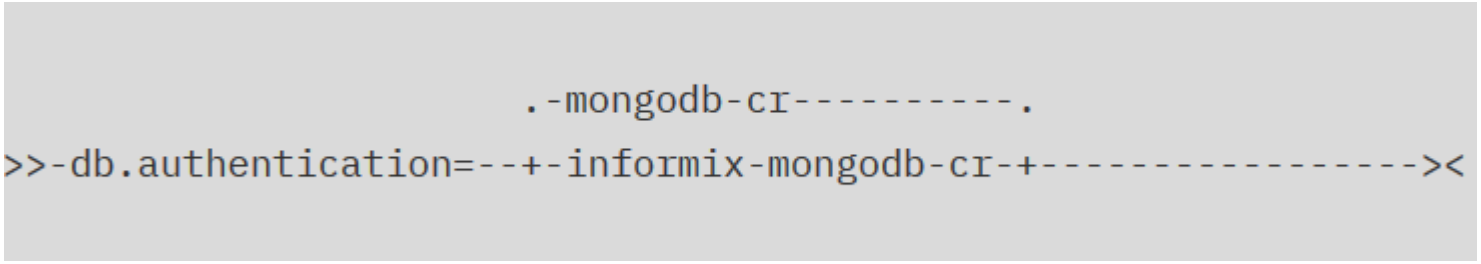
- This optional parameter specifies the MongoDB API version with which the wire listener is compatible
- Version affects authentication methods and MongoDB commands



- Do not set mongo.api.version=3.0 if you want to use the REST API or database server authentication

db.authentication

- **Optional parameter specifies the user authentication method.**



- **mongdb-cr**
  - Authenticate through the wire listener with a MongoDB authentication method.
  - The MongoDB authentication method depends on the setting of the **mongo.api.version** parameter.
  - Earlier versions of MongoDB than 3.0 do not authenticate users by default, which is a big security risk.

- **informix-mongodb-cr**
  - Authenticate through the database server with a pluggable authentication module (**PAM**). This is the preferred method.

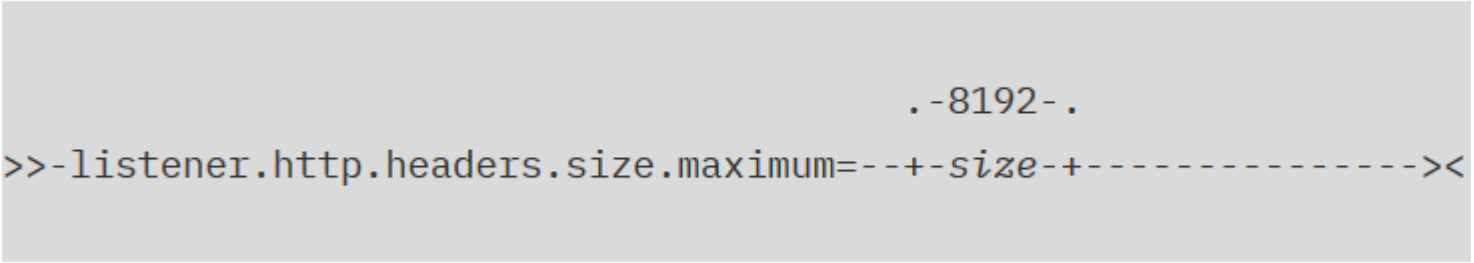
listener.rest.cookie.domain

- **Optional parameter specifies the name of the cookie that is created by the REST wire listener.**
  - If not specified, the domain is the default value as determined by the Apache Tomcat web server.
  - Might want to set this if not planning to use Apache Tomcat.

```
>>-listener.rest.cookie.domain=--+------------------+----------><
                                 '-rest_cookie_name-'
```

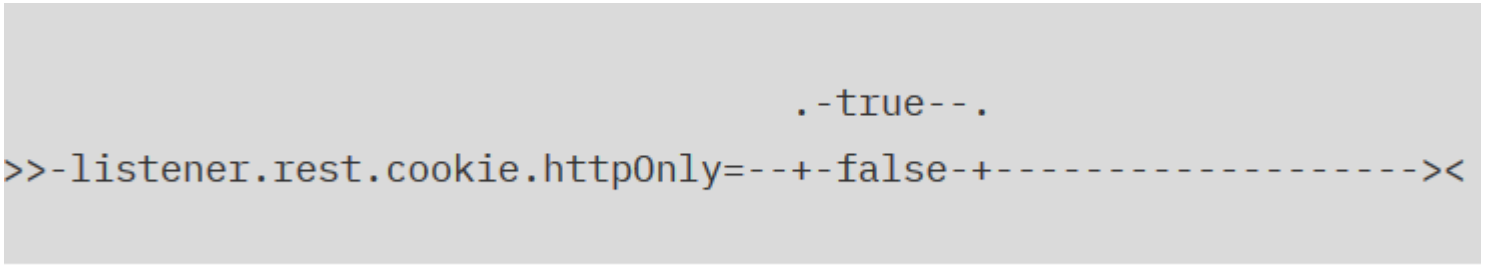
listener.http.headers.size.maximum

- **Optional parameter specifies the maximum size in bytes of headers in incoming HTTP requests:**
  - The default is **8192** bytes.



listener.rest.cookie.httpOnly

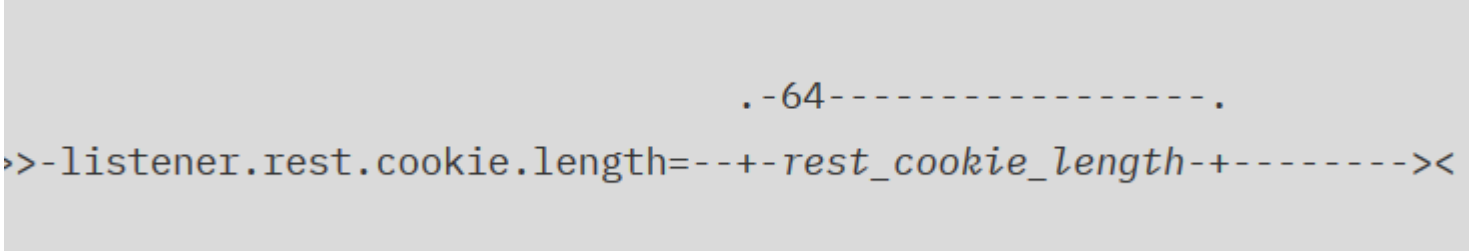
- **Optional parameter indicates whether to set the HTTP-only flag.**



- **true**
  - Set the HTTP-only flag.
  - This flag helps to prevent cross-site scripting attacks.
  - This is the default value.
- **false**
  - Do not set the HTTP-only flag.

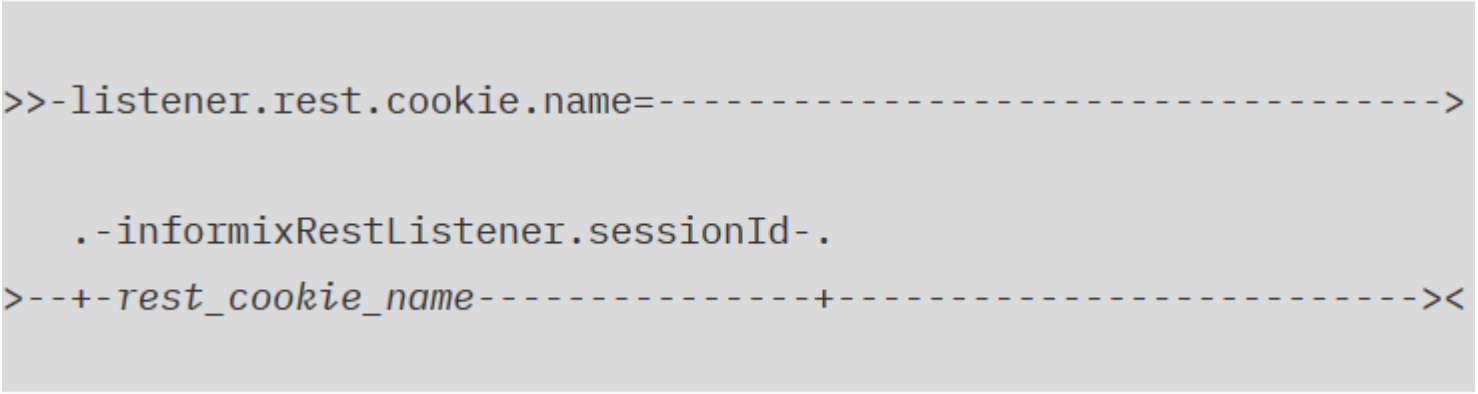
listener.rest.cookie.length

- **Optional parameter specifies the length, in bytes, of the cookie value that is created by the REST wire listener, before Base64 encoding**
  - The default value is **64 bytes**.



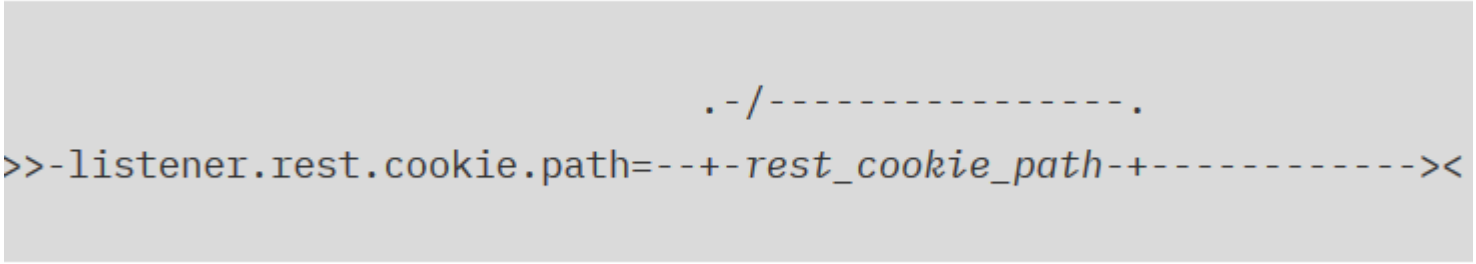
listener.rest.cookie.name

- **Optional parameter specifies the name of the cookie that is created by the REST wire listener to identify a session**
  - Default value is:
    listener.rest.cookie.name=informixRestListener.sessionId



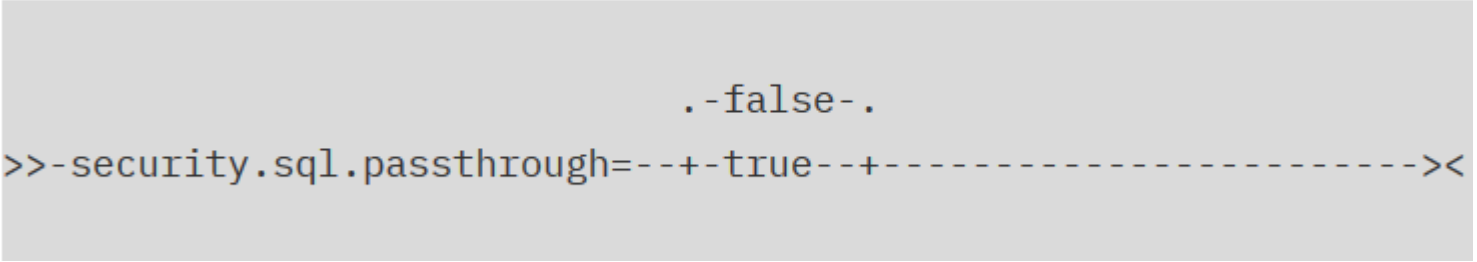
listener.rest.cookie.path

- **Optional parameter specifies the path of the cookie that is created by the REST wire listener**
  - The default value is: **listener.rest.cookie.path=/.**



security.sql.passthrough

- **Optional parameter indicates whether to enable support for issuing SQL statements by using JSON documents.**
  - Only if you want to issue SQL statements to Informix within the MongoDB environment.



- **false**
  - Disable the ability to issue SQL statements by using the MongoDB API.
  - This is the default.
- **true**
  - Allow SQL statements to be issued by using the MongoDB API.

User Authentication with the Wire Listener

- **Authenticate users with MongoDB authentication or with the database server, through a pluggable authentication module (PAM).**
- **Use the following types of authentication methods:**
  - MONGODB-CR challenge-response
    - The wire listener authenticates users with the MongoDB challenge-response authentication method outside of the database server environment
    - Create users with the MongoDB API **create user** commands
    - Clients connect to the wire listener as MongoDB users and the wire listener authenticates the users
    - The wire listener connects to the database server as the user that is specified by the **url** parameter in the wire listener configuration file
    - The database server cannot access MongoDB user account information
    - For MongoDB version 2.4, user information and privileges are stored in the **system_users** collection in each database
    - For MongoDB version 2.6 and later, user information and privileges are stored in the **system.users** collection in the **admin** database
    - If MongoDB version is being upgraded and users exist, user schema must be upgraded

User Authentication with the Wire Listener

- **Use the following types of authentication methods with the wire listener (cont'd):**
  - SCRAM-SHA-1 two-step authentication (**Cannot use** SCRAM authentication with the REST API or the MQTT protocol)
    - **SCRAM-SHA-1** is only available when the **mongo.api.version=3.0** parameter is set in the wire listener configuration file.
    - The wire listener authenticates users with the **SCRAM-SHA-1** authentication method outside of the database server environment.
    - You create users with the MongoDB API **create user** commands.
    - User info and privileges are stored in **system.users** in the **admin** database
    - Clients connect to the wire listener as MongoDB users and the wire listener authenticates the users.
    - Wire listener connects to the database server as the user specified by the **url** parameter in the wire listener configuration file.
    - Database server cannot access MongoDB user account information.

User Authentication with the Wire Listener

- **Use the following types of authentication methods with the wire listener (cont'd):**
  - Database server authentication with a PAM (UNIX, Linux)
    - PAM implements the **MONGODB-CR** challenge-response method
    - Wire listener connects to the database server using the user and password provided by clients and database server authenticates a user thru PAM
    - Database server controls all user accounts and privileges
    - Audit user activities and configure fine-grained access control
    - Types of authentication used depend on the type of client and MongoDB version

User Authentication with the Wire Listener – Mongo Clients

Authentication type	MongoDB 2.4	MongoDB 2.6	MongoDB 3.0	Details
MONGODB-CR	Yes	Yes	No	Here
SCRAM-SHA-1	No	No	Yes	The user schema must be at MongoDB version 2.6 or later.
PAM	Yes	Yes	No	Here

User Authentication with the Wire Listener – REST Clients

Authentication type	MongoDB 2.4	MongoDB 2.6	MongoDB 3.0	Details
MONGODB-CR	Yes	Yes	No	• Here • HTTP clients authenticate using the HTTP basic authentication method.
SCRAM-SHA-1	No	No	No	SCRAM is not supported.
PAM	Yes	Yes	No	• Here • HTTP clients authenticate using the HTTP basic authentication method.

User Authentication with the Wire Listener – MQTT Clients

Authentication type	MongoDB 2.4	MongoDB 2.6	MongoDB 3.0	Details
MONGODB-CR	Yes	Yes	Yes	• Here • The MQTT CONNECT packet must include the database name as a prefix of the user name, in the following format: "*database_name.user_name*".
SCRAM-SHA-1	No	No	No	SCRAM is not supported.
PAM	Yes	Yes	No	• Here • The MQTT CONNECT packet must include the database name as a prefix of the user name, in the following format: "*database_name.user_name*".

Configuring MongoDB Authentication for the Wire Listener

- **Before you begin**
  - If you are upgrading your MongoDB version and you have existing MongoDB users, you must upgrade your user schema.
- **Set the following parameters in the wire listener configuration file:**
  - Enable authentication:
    - Set **authentication.enable=true**
  - Specify MongoDB authentication:
    - Set **db.authentication=mongodb-cr**
  - Specify the MongoDB connection pool:
    - Set **database.connection.strategy=mongodb-cr**
  - Set the MongoDB version:
    - Set **mongo.api.version** to the version that you want
  - Optional
    - Specify the authentication timeout period:
    - Set the **listener.authentication.timeout** parameter to the number of milliseconds for authentication timeout
- **Restart the wire listener**

Configuring MongoDB Authentication for the Wire Listener

- If necessary, upgrade your user schema by running the MongoDB authSchemaUpgrade command in the admin database on the MongoDB command line:

- For example:

```
use admin
db.runCommand({authSchemUpgrade : 1})
```

- The authSchemaUpgrade command upgrades the user schema to the MongoDB version that is specified by the mongo.api.version parameter.

Configuring MongoDB Authentication for the Wire Listener

- **To add authorized users**
  - Start the wire listener with authentication turned off:
    - **Set authentication.enable=false** in the wire listener configuration file

- **Add users:**
  - MongoDB version 2.4
    - Run the **addUser** command per user in each database
  - MongoDB version 2.6 and 3.0
    - Run the **createUser** command for each user

- **Turn on authentication:**
  - **set authentication.enable=true** in the wire listener configuration file.

- **Restart the wire listener.**

Configuring PAM Authentication for the Wire Listener (1)

- PAM authentication can be used for MongoDB, REST, or MQTT clients
- Configure the database server to authenticate wire listener users with a pluggable authentication module (PAM)
- Create a user for the wire listener for PAM connections
- Wire listener uses the PAM user to look up system catalog-related information before sending client connection requests to the database server for authentication
- The database server authenticates the client users through PAM

Configuring PAM Authentication for the Wire Listener (2)

- **Configure PAM authentication for MongoDB, REST, or MQTT clients:**
  - Set the **IFMXMONGOAUTH** environment variable.
    - For example:
      - **setenv IFMXMONGOAUTH 1**
  - Create a PAM service file named **/etc/pam.d/pam_mongo** and has the following:
    - **auth required *$INFORMIXDIR*/lib/pam_mongo.so file=mongohash**
    - **account required *$INFORMIXDIR*/lib/pam_mongo.so**
  - Replace ***$INFORMIXDIR*** with the **$INFORMIXDIR** environment variable value.
  - On AIX 64-bit, create a symbolic link named **64** that points to **$INFORMIXDIR/lib** by running the following command:
    - **cd $INFORMIXDIR/lib; ln -s . 64**

Configuring PAM Authentication for the Wire Listener (3)

- **To configure PAM authentication for MongoDB, REST, or MQTT clients (cont'd):**
  - Edit the **sqlhosts** file to add a connection that uses PAM
    - Include the **s=4** option
    - Specify the PAM service **pam_mongo** with the **pam_serv** option
    - Specify the password authentication mode with the **pamauth** option
    - For example:

      ol_informix1210 onsoctcp myhost 40000
      s=4,pam_serv=pam_mongo,pamauth=password
  - Enable connections for mapped users by setting the **USERMAPPING** configuration parameter to **BASIC** or **ADMIN** in the **onconfig** file:
  - Set up mapping to an operating system user that has no privileges:
    - For example: on a typical Linux system, the user **nobody** is appropriate:
    - Add the following line to the **/etc/informix/allowed.surrogates** file:

      users:nobody

Configuring PAM Authentication for the Wire Listener (4)

- **To configure PAM authentication for MongoDB, REST, or MQTT clients (cont'd):**
  - Restart the database server
  - Create a PAM user for the wire listener
    - The user must be internally authenticated and map to the user **nobody**
    - For example:
      - Create a user that is named **mongo** by running the following SQL in the **sysmaster** database:

```
database sysmaster;
CREATE USER 'mongo' WITH PASSWORD 'aPassword' PROPERTIES USER 'nobody';
GRANT CONNECT TO 'mongo';
```

  - Verify the user creation by running the following statement:

```
SELECT * FROM sysuser:sysmongousers WHERE username='mongo';
```

    - The result of the query shows the user and hashed password:

```
username   mongo
hashed_password  bbb8f9630d5c6e094b9aedd945893faf
```

Configuring PAM Authentication for the Wire Listener (5)

- **To configure PAM authentication for MongoDB, REST, or MQTT clients (cont'd):**
  - Set the following parameters in the wire listener configuration file
    - Enable authentication:
      - Set **authentication.enable=true**
    - Specify PAM authentication:
      - Set **db.authentication=informix-mongodb-cr**
    - Specify the PAM connection pool:
      - Set **database.connection.strategy=informix-mongodb-cr**
    - Set the MongoDB version:
      - Set **mongo.api.version=2.6** or **mongo.api.version=2.4**

        The PAM authentication method is not compatible with MongoDB version 3.0
    - Optional.
      - Specify the authentication timeout period parameter:

        Set the **listener.authentication.timeout** to the number of milliseconds for authentication timeout

Configuring PAM Authentication for the Wire Listener (6)

- **To configure PAM authentication for MongoDB, REST, or MQTT clients (cont'd):**
  - Set the following parameters in the wire listener configuration file
    - Specify the mapped user and password for connections and specify to encode and hash the password on the **url** parameter:
      - Include the **NONCE** property set to any 16 character string that contains only the digits 0-9 and the lower-case characters a-f (extended grep: [0-9a-f]{16})
      - For example:

        url=jdbc:informix-sqli://10.168.8.135:40000/sysmaster:USER=mongo;
        PASSWORD=aPassword;NONCE=0123456789abcdef

  - Restart the wire listener
  - Create users that the database server authenticates with PAM by running the SQL statement **CREATE USER**
    - If there are existing MongoDB users, they must be re-created in the database server

REST, Wire Listener and SSL

- **The Wire listener can be configured for SSL as well:**
  - The database server must be configured for SSL connection first
  - See **here** in this presentation as to how to do this

- **Wire listener must use the same public key certificate file as the database server**

- **To configure SSL connections between the wire listener and the database server:**
  - Use the **keytool** utility that comes with your Java runtime environment to import a client-side keystore database and add the public key certificate to the keystore:

    C:\work>keytool -importcert -file *server_keystore_file* -keystore *client_keystore_name*

    - The ***server_keystore_file*** is the name of the server key certificate file.

REST, Wire Listener and SSL

- **Edit the wire listener properties file:**
  - To update the **url** property to database server **SSL** port
  - Add the **SSLCONNECTION=true** property to the end of the URL:

- **Start the listener with the javax.net.ssl.trustStore and javax.net.ssl.trustStorePassword system properties set:**

```
java -Djavax.net.ssl.trustStore="client_keystore_path“
     -Djavax.net.ssl.trustStorePassword="password"
     -jar jsonListener.jar
     -config jsonListener.properties
     -logfile  jsonListener.log -start
```

  - ***client_keystore_path*** is the full path and file name of the client keystore file.
  - The *password* is the keystore password.

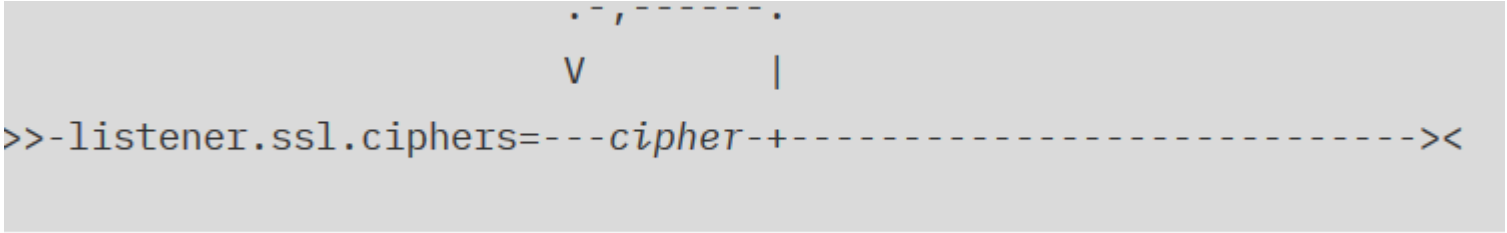
listener.ssl.algorithm

- **This optional parameter specifies the Service Provider Interface (SPI) for the KeyManagerFactory that is used to access the network encryption keystore:**
  - On an Oracle Java Virtual Machine (JVM):
    - This value is typically **SunX509**.
  - On an IBM JVM:
    - This value is typically **IbmX509.**
  - Default value is no SPI.
- **Important**
  - Do not set this property if you are not familiar with Java Cryptography Extension (JCE).

```
>>-listener.ssl.algorithm=SPI----------------------------------><
```

listener.ssl.ciphers

- **Optional parameter specifies a list of Secure Sockets Layer (SSL) or Transport Layer Security (TLS) ciphers to use with network encryption.**
  - Default value is no ciphers
    - Which means that the default list of enabled ciphers for the JVM are used.
    - Do not set this property if you are not familiar with Java Cryptography Extension (JCE) and the implications of using multiple ciphers.
    - Consult a security expert for advice.



- **Include spaces between ciphers.**
- **For example:**

listener.ssl.ciphers=TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA, TLS_EMPTY_RENEGOTIATION_INFO_SCSV

listener.ssl.key.enable

- **Optional parameter enables SSL or TLS network encryption on the socket for client connections**
  - **false**
    - Disable network encryption
    - This is the default
  - **true**
    - Allow network encryption

```
>>-listener.ssl.key.alias=alias--------------------------------><
```

listener.ssl.key.alias

- **Optional parameter specifies the alias, or identifier, of the entry into the keystore.**
  - Default value is no alias, which indicates that the keystore contains one entry.
  - If the keystore contain more than one entry and a key password is needed to unlock the keystore, set this parameter to the alias of the entry that unlocks the keystore.

- **Parameter is effective when the listener.ssl.enable is true.**

```
>>-listener.ssl.key.alias=alias--------------------------------><
```

listener.ssl.key.password

- **Optional parameter specifies the password to unlock the entry into the keystore, identified by the listener.ssl.key.alias parameter.**
  - Default value is no password
    - Which means to use the keystore password
    - If the entry into the keystore requires a password that is different from the keystore password, set this parameter to the entry password

```
>>-listener.ssl.key.password=password--------------------------><
```

- **This is effective when the listener.ssl.enable parameter is true.**

listener.ssl.keyStore.file

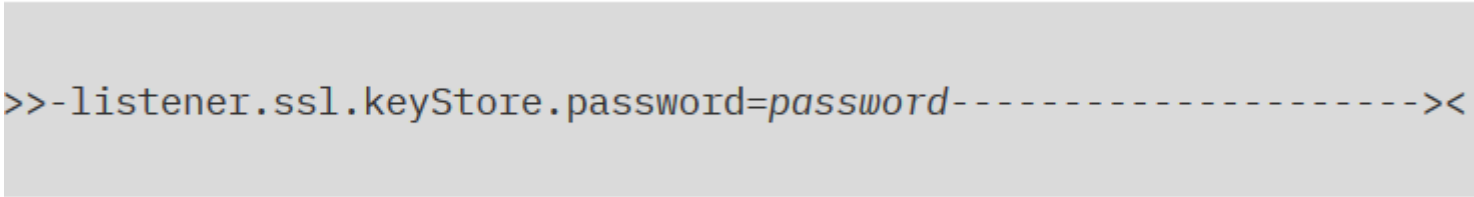
- **Optional parameter specifies the fully-qualified path and file name of the Java keystore file to use for network encryption**
  - Default value is no file

```
>>-listener.ssl.keyStore.file=file_path------------------------><
```

- **Parameter is effective when the listener.ssl.enable parameter is true**

listener.ssl.keyStore.password

- **Optional parameter specifies the password to unlock the Java keystore file for network encryption**
  - The default value is no password



- **Parameter is effective when the listener.ssl.enable parameter is true**

listener.ssl.keyStore.type

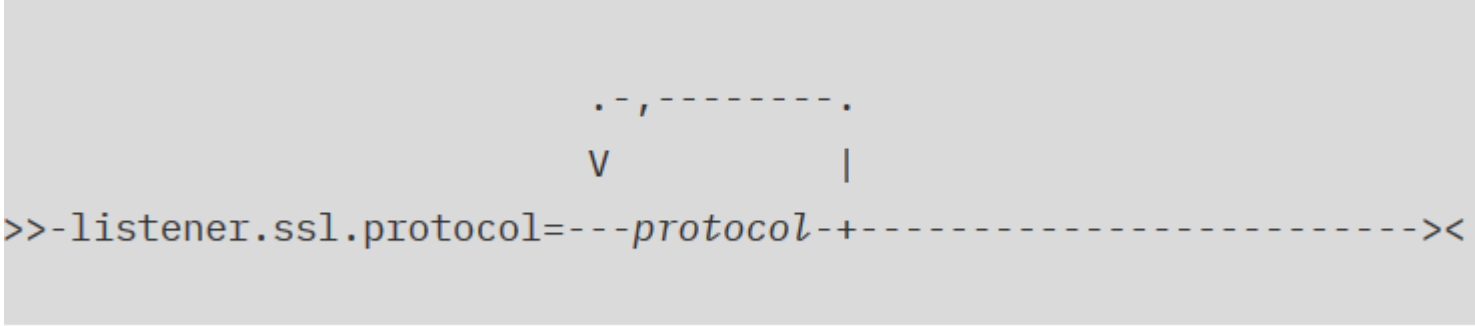
- **Optional property specifies the provider identifier for the network encryption keystore SPI:**
  - Default value is JKS.
  - Do not set this property if you are not familiar with Java Cryptography Extension (JCE).

```
>>-listener.ssl.keyStore.type=SPI------------------------------><
```

- **Parameter is effective when the listener.ssl.enable parameter is true.**

listener.ssl.protocol

- **Optional parameter specifies the SSL or TLS protocols.**
  - The default value is TLS.

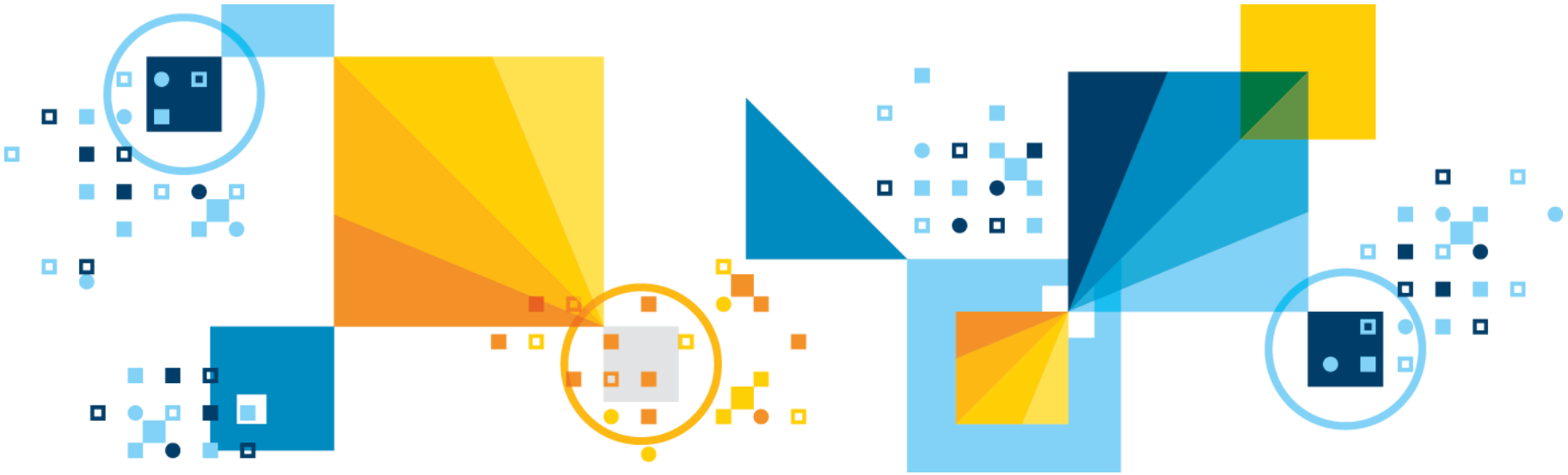


Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Mapped Users



Mapped Users

- **Mapped users are external users not logging in on the database server that are mapped by the DBSA to an O/S level profile on the database server for processing connection requests:**
  - User/password validated in an authentication layer outside the database server.
  - DBSA configures the database server to authenticate users by checking their credentials with a hashed password stored in the database server.
  - Requires external users to attempt to connect through Kerberos single sign-on (SSO), or a Pluggable Authentication Module (PAM), for internal authentication to be mapped to an OS-level profile.

- **Sometimes running an SQL statement requires the database server to interact with the OS, typically to read or write a file, or to run a program through the SPL SYSTEM statement.**

- **When interaction with the OS is required, the database server must be provided OS credentials to manage the file or run the program.**

Mapped Users

- **Users can be mapped to one of the following surrogate user identities:**
  - A UID and GID pair defined in the database server
  - An existing OS user account on the database server host computer

- **After a user authenticates, whenever the database server interacts with the OS on behalf of the user, the surrogate user properties specified by the user mapping are invoked.**

- **The simplest mapping is identity mapping when the user name maps directly to the OS properties of a user with the same name. If you are the OS Administrator, use the /etc/informix/allowed.surrogates file to specify which surrogate users and groups can be used so that mapped users are not granted owner access to sensitive systems, such as databases, print spoolers, email, or the operating system, itself.**

Mapped Users

- **The allowed.surrogates file is not used or read by non-root installations of the database server, because the database server does not perform operations or run commands as the user who started the session.**

- **The CREATE USER and GRANT ACCESS TO PROPERTIES SQL statements can create complex mappings of surrogate properties, including:**
  - user ID
  - user name
  - surrogate groups
  - home directory
  - authorization privilege (DBSA, DBSSO, AAO, or BARGROUP)

Mapped users

- **The CREATE USER and ALTER USER statements associate OS-level privileges by mapping users to OS properties and storing this information in a series of Informix system catalog tables.**

- **Users can be mapped by the DB-Access utility and the IBM® OpenAdmin Tool (OAT) for Informix® GUI. After a DBSA sets the USERMAPPING configuration parameter in the onconfig file, and maps externally authenticated users to surrogate properties in tables of the SYSUSER database, it is possible for the mapped users to connect to the database server without a local OS account.**

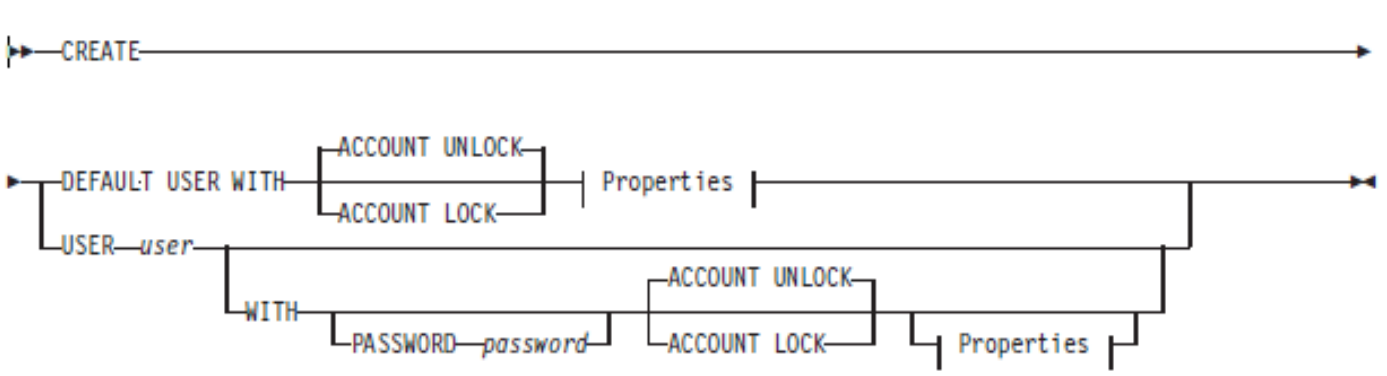
- **In order to enable mapped users functionality, the USERMAPPING configuration parameter must be set to either BASIC or ADMIN.**
  - **BASIC** - Enables mapping of users to any user/group except *informix*
  - **ADMIN** - Enables mapping of users to any user/group including informix

SQL Operations for Mapped Users

- **After the USERMAPPING configuration parameter is set to BASIC or ADMIN, use the following DDL operations on mapped users:**
  - **ALTER USER**
  - **CREATE USER**
    - Sets up user accounts with passwords
  - **DROP USER**
  - **GRANT ACCESS TO PROPERTIES**
  - **SET USER PASSWORD**
    - Users can change their own password
  - **RENAME USER**
  - **REVOKE ACCESS**

CREATE USER statement

- **CREATE USER statement:**



- **CREATE USER associates OS-level privileges by mapping users to OS properties.**
  - e.g. **UID**, **GROUP**, **HOME** directory

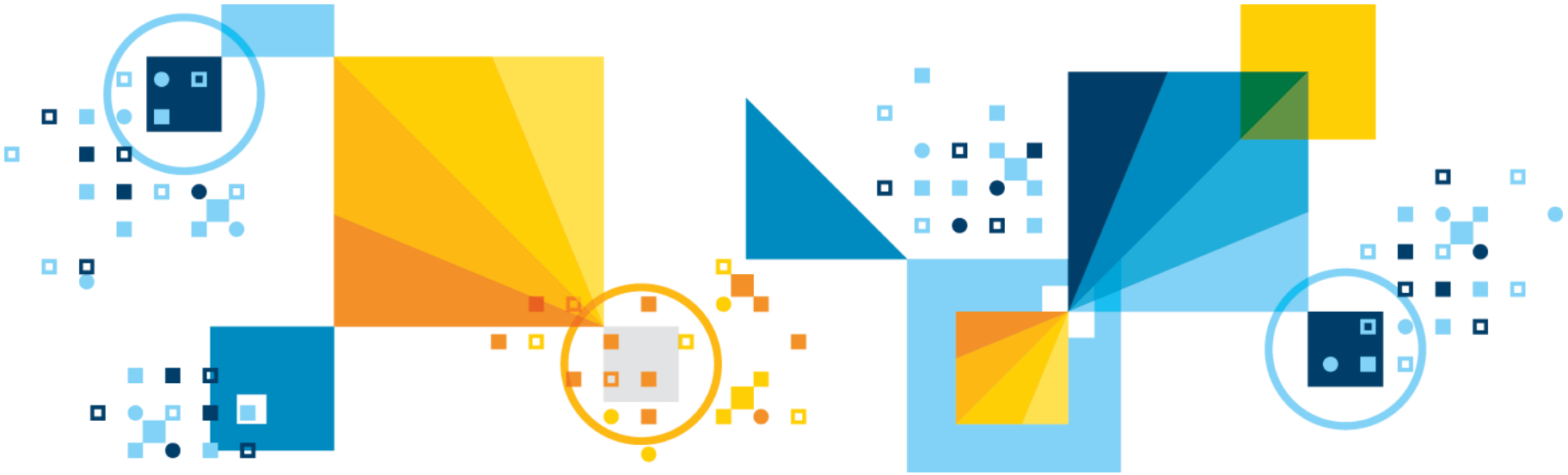
- **OS credentials are used when Informix server interacts with the OS on behalf of Database users.**
  - e.g. **SPL SYSTEM()** , **SET EXPLAIN**

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

LBAC – Label Based Access Control



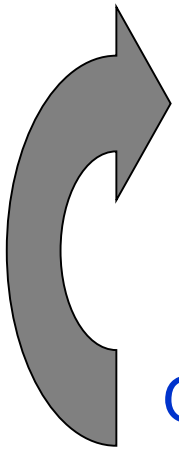
What is LBAC?

- **Form of Mandatory Access Control.**
- **Data is Labeled.**
- **Users are granted labels.**
- **Based on a User and Data Label comparison, users can access data:**
  - A predefined rule set, commonly referred to as IDSLBACRULES.
  - User access labels must meet or exceed data labels for a user to get access.
  - DBSECADM can grant exceptions to this.
- **Available on Enterprise Edition only.**

LBAC Demonstration

User Label - Public

SecurityLabel	Col1	Col2	Col3
Public			
Public			
Public			

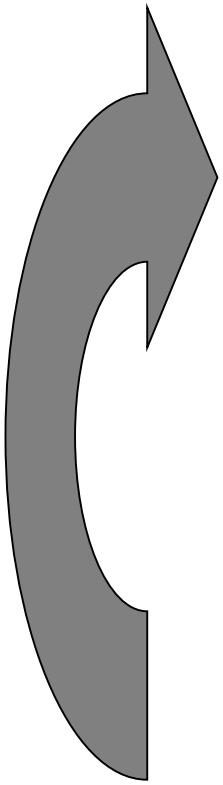


Only the rows with “Public” access are returned

SELECT * FROM Table1

LBAC Demonstration

User Label - Confidential



SecurityLabel	Col1	Col2	Col3
Public			
Confidential			
Public			
Confidential			
Confidential			
Public			

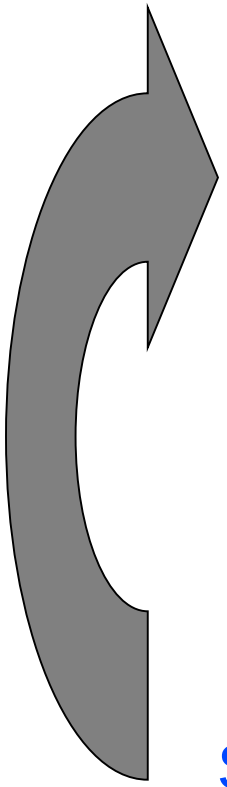
Rows with “Public” or “Confidential” access are returned

SELECT * FROM Table1

LBAC Demonstration

User Label - Secret

SecurityLabel	Col1	Col2	Col3
Secret			
Public			
Confidential			
Secret			
Public			
Confidential			
Secret			
Confidential			
Public			



All Rows with “Public”,“Confidential” or Secret Access are returned

SELECT * FROM Table1

Security Policies, Components, Labels

- **Security Policy:**
  - Database object that protects the table.
  - Composed of security label components.
- **Security Label Component:**
  - Array.
  - Set.
  - Tree.
- **Security Label:**
  - Always associated with a security policy.
  - Includes one value for each component in the security policy
    - A value is a list of zero or more of the elements allowed by that component.

Creating Security Policies

- **A Security Policy is created from the Security Components:**
  - Up to 16 components.

- **CREATE SECURITY POLICY company COMPONENTS level, department, region;**
  - This policy has three components.
  - Labels for this policy will have value (zero or more elements) for each of these components.

Array Component

- **Ordered list of elements:**
  - Up to 64 elements.
- **First one is the highest security level.**
- **Only one element allowed in a label for a component.**
- **You can read data that is less than or equal to your level.**
- **You can write data only equal to your level.**

CREATE SECURITY LABEL COMPONENT level ARRAY [‘Secret’, 'Confidential', 'Public'];

Set Component

- **Non-ordered set of elements:**
  - Up to 64 elements.
- **One or more elements in a label for a component.**
- **You can read or write data if your label contains all the elements in the data label.**

```
CREATE SECURITY LABEL COMPONENT department SET
{'Marketing', 'Product Development', 'Quality Assurance'};
```

Tree Component

- **Hierarchical set of elements:**
  - Up to 64 elements.
- **You can have one or more elements in a label.**
- **You can read or write data if your label contains any of the elements in the data label or the ancestor of one such element.**

CREATE SECURITY LABEL COMPONENT region TREE ('Entire Region' ROOT, 'East' UNDER 'Entire Region', 'West' UNDER 'Entire Region');

Creating Security Labels

- **A Security Label specifies the value for each component in a Security Policy.**
- **Example:**

```
CREATE SECURITY LABEL company.director
        COMPONENT level ‘Secret’,
        COMPONENT department 'Product Development',
                            'Quality Assurance‘,
        COMPONENT region ‘Entire Region’;
```

  - Single element in the labels for components level and region.
  - Multiple elements in the label for component department.

Granting Security Labels, Exemptions

- **After a Security Label is created, it can be GRANTED to a user:**
  - The same label can be granted to many users.
  - One **READ** label and one **WRITE** label can be granted to a user for a given security policy
    - If the labels are different, the **READ** label must dominate the **WRITE** label.

- **Exemptions:**
  - A user can be granted an exemption to bypass one or more access rules for a component type in a security policy.

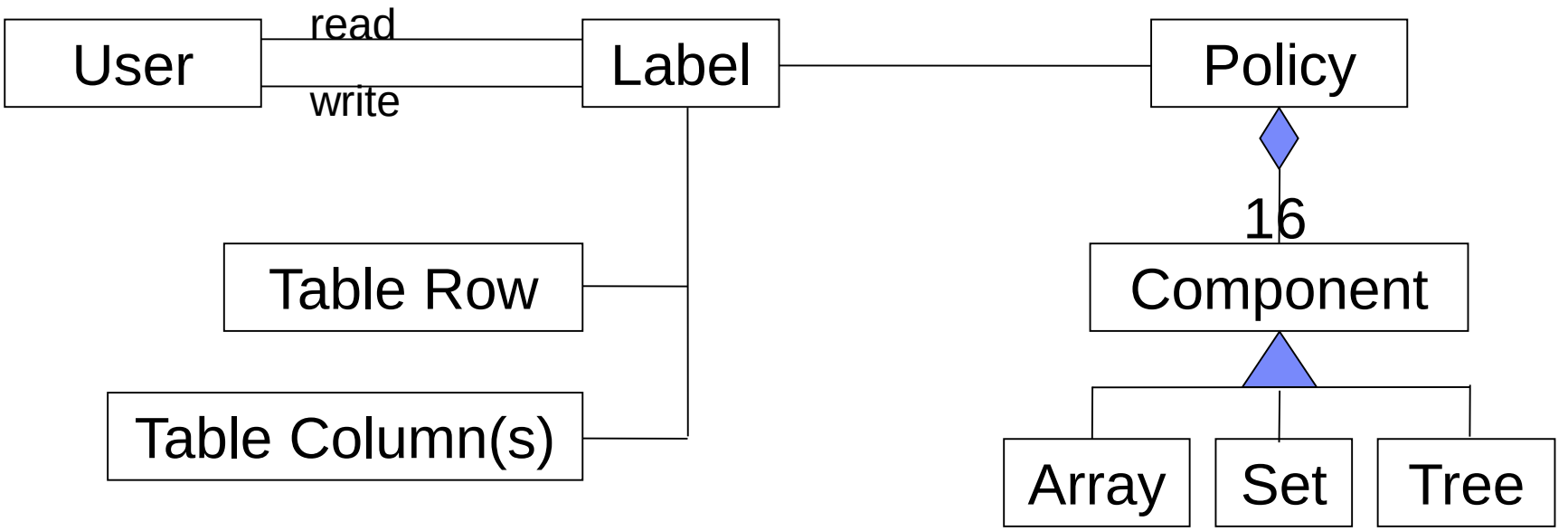
Protecting a Table

- **Protecting rows (Row level protection granularity)**
  - Attach a security policy to a table.
  - Add a security label column using the new **IDSSECURITYLABE**L data type.
- **Protecting columns (Column level protection granularity):**
  - Attach a security policy to a table.
  - Attach a security label to one or more columns.

CREATE TABLE T1 (C1 IDSSECURITYLABEL, C2 int, C3 char (10) COLUMN SECURED WITH director) SECURITY POLICY company;

ALTER TABLE T2 ADD (C1 IDSSECURITYLABEL), MODIFY (C2 int COLUMN SECURED WITH manager), ADD SECURITY POLICY company;

LBAC Overview Diagram



Summary of Read Access Rules

- **Applied when data is read. Data is read on SELECT, UPDATE and DELETE operations:**
  - **IDSLBACREADARRAY**
    - Each array component of the user security label must be greater than or equal to the array component of the data security label, i.e. users can only read data at or below their levels.
  - **IDSLBACREADSET**
    - Each set component of the user security label must include the set component of the data security label.
  - **IDSLBACREADTREE**
    - Each tree component of the user security label must include at least one of the elements in the tree component of the data security label (or the ancestor of one such element).

Summary of Write Access Rules

- **Applied when the data is written. Data is written on INSERT, UPDATE and DELETE operations:**
  - **IDSLBACWRITEARRAY**
    - Each array component of the user security label must be equal to the array component of the data security label, i.e. users can write data only at their levels.
  - **IDSLBACWRITESET**
    - Each set component of the user security label must include the set component of the data security label.
  - **IDSLBACWRITETREE**
    - Each tree component of the user security label must include at least one of the elements in the tree component of the data security label (or the ancestor of one such element).

SQL Functions

- **SECLABEL_BY_COMP**
  - A built-in function that can be used in insert and update operations to provide the row security label of a data row by providing its individual components

    ```
    INSERT INTO T1
    VALUES (SECLABEL_BY_COMP('company', 'Director:Marketing:West'),
    1, 'xyz')
    ```

- **SECLABEL_BY_NAME**
  - A built-in function that can be used in insert and update operations to provide the row security label of a data row by providing its name

    ```
    UPDATE T1 SET C1 = SECLABEL_BY_NAME('company', 'manager')
    ```

- **SECLABEL_TO_CHAR**
  - A built-in function that can be used in select operations to retrieve the row security label column

    ```
    SELECT SECLABEL_TO_CHAR('company', C1), C2, C3 FROM T1
    ```

SETSESSIONAUTH privilege

- SET SESSION AUTHORIZATION statement allows a DBA to assume the identity of another user.
- The DBA can see other users data in protected table(s).
- New privilege SETSESSIONAUTH prevents unauthorized access.
- Only users who are granted SETSESSIONAUTH privilege can use the SET SESSION STATEMENT.
- During conversion from an older server to IDS 11, the SETSESSIONAUTH privilege is granted to DBA for backward compatibility.

DBSECADM

- **Database security administrator.**
- **Server level role.**
- **Can be granted by DBSA only.**
- **Responsibilities:**
  - Create, drop, alter and rename security label components.
  - Create, drop and rename security policies.
  - Create, drop and rename security labels.
  - Attach, detach policies to/from tables.
  - Grant and revoke security labels.
  - Grant and revoke policy exemptions.
  - Grant and revoke **setsessionauth** privilege.
- **NOTE: Informix ID CAN grant DBSECADM to him/herself****

**IDS docs say this can not happen, but it is allowed

Restrictions

- **Tables that cannot be protected:**
  - Virtual Table Interface (VTI) tables.
  - Tables with Virtual Index Interface (VII) indexes.
  - Temp tables.
  - Typed tables.
  - Hierarchical tables.

- **A security label column cannot have:**
  - Referential constraints.
  - Check constraints.
  - Primary Key or Unique constraints if the security label column is the only column in the constraint.
  - Column protection.
  - Encryption.

Utilities

- **dbschema/dbexport/dbimport:**
  - User must be granted **DBSECADM** role if a database contains LBAC objects.
  - User must have the necessary labels or exemptions if all rows in protected tables are to be exported/imported.
  - Data unloaded is that where the users column/row or both label dominates.
- **onload/onunload:**
  - Cannot be used with LBAC.

- **HPL (Express mode only):**
  - Users must have all exemptions to bypass the security policy.
  - **.rej** and **.flt** files should be restricted permission files.

- **All other load/unload utilities:**
  - Users must have the necessary labels or exemptions if all rows in protected tables are to be loaded/unloaded.

Utilities

- DBSECADM required for LBAC protected table level backup/restore with onbar/ontape.
- Point in time table or entire table restores require appropriate LBAC permissions.
- archecker not supported with LBAC for table level restore.
- LBAC user read and write access to objects required for onbar/ontape operations.

Migrations

- **Legacy IDS data servers (pre 11.10) not supporting LBAC automatically grant SETSESSIONAUTH privilege for PUBLIC to DBA privileged users in the migration process on those IDS servers that do support LBAC (11.10 and higher):**
  - Should convert the 11.10+ server to one supporting LBAC security policies to remove the **SETSESSIONAUTH** privilege from all DBAs and enable the **DBSECADM** role to grant this privilege.

Other Considerations

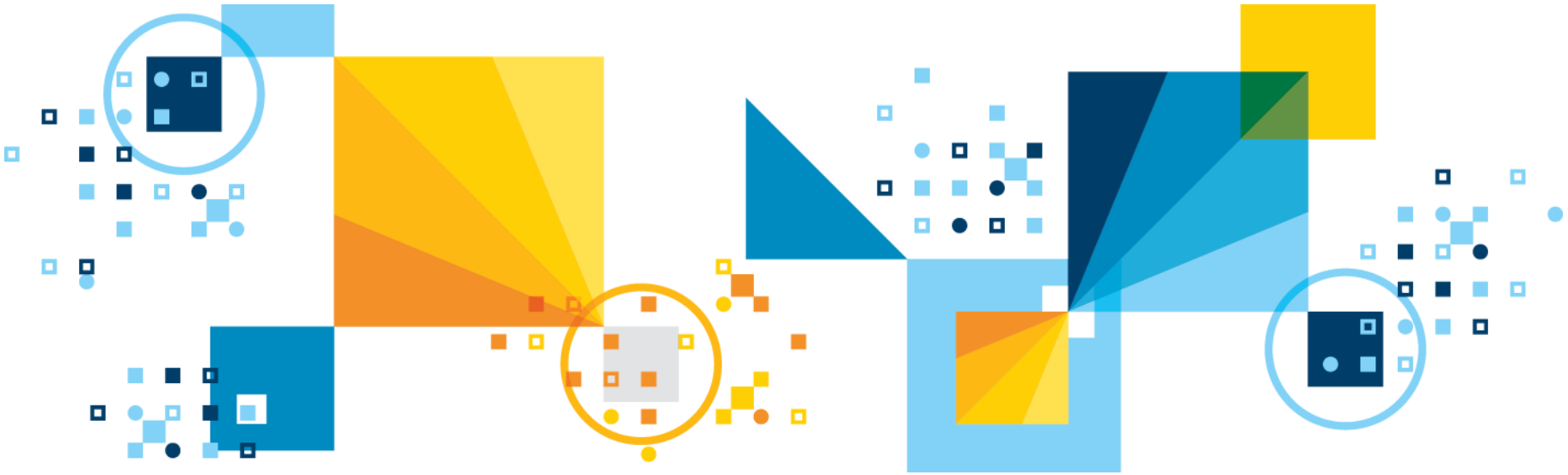
- **Fully functional in a HA environment:**
  - LBAC objects created on a primary replicated to secondary.
  - Tables equally protected on primary and secondary.
  - SDS, HDR, RSS
- **Distributed queries fully supported:**
  - Users from non-LBAC protected servers querying LBAC protected servers must have LBAC permissions on the LBAC protected server to see the data.
- **Synonyms and views using tables protected by LBAC are supported, locally and remotely.**
- **Enterprise Replication is not supported for LBAC operations as of 11.50.xC6.**
- **oncheck and onlog and some DDL operations can expose data otherwise protected by LBAC.**
- **Tables in Insert into ..... select * from operations should be protected by the same LBAC policies to prevent unauthorized data observation.**

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Access, Permissions and Roles



Database Access

- Who can connect to your database?
- Who can create a database?
- Who can access the database objects?
- What actions are allowed for a certain user?

Database Access

- Authorization Process
- Database Privileges
  - CONNECT
  - RESOURCE
  - DBA
- GRANT/REVOKE
- Roles

Authorization process

- **Authorization is the mechanism which allows an authenticated user to access various resources based on the user's identity.**

- **Authorized users can:**
  - Perform database actions.
  - Access certain database objects.

- **Authorization to use a database is also known**
  - As an access privilege.

Database Privileges

- Informix provides three levels of Database access privileges:
  - CONNECT
  - RESOURCE
  - DBA
- DBCREATE_PERMISSION onconfig parameter can be used to control who can create a database.
- When you create a database, you are the database administrator, or DBA of that database.
- The database remains inaccessible to other users until you, as DBA, grant database privileges.
- User informix has the privilege required to alter the tables of the system catalog, including the systables table.

Connect

- Connect to the database with the CONNECT statement or another connection statement.
- Execute SELECT, INSERT, UPDATE, DELETE statements, and execute SPL routine, provided the user has the necessary table-level privileges.
- Create views, provided the user has SELECT privilege on the underlying tables.
- Create temporary tables and create indexes on the temporary tables.

Resource

- All CONNECT plus
- Create new tables
- Create new indexes
- Create new UDRs
- Create new data types

DBA

- **All Resource permission, plus:**
  - Grant any database-level privilege, including the **DBA** privilege, to another user.
  - Grant any table-level privilege to another user or to a role.
  - Grant a role to a user or to another role.
  - Revoke a privilege whose grantor you specify as the revoker in the **AS** clause of the **REVOKE** statement.
  - Restrict the Execute privilege to DBAs when registering an UDR.
  - Execute the **SET SESSION AUTHORIZATION** statement.
  - Create any database object.
  - Create tables, views, and indexes to be owned by other user.
  - Alter, drop, or rename database objects regardless of who owns them.
  - Execute the **DROP DISTRIBUTIONS** option of the **UPDATE STATISTICS** statement
  - Execute **DROP DATABASE** and **RENAME DATABASE** statements.

GRANT / REVOKE

- Use the GRANT statement to grant privileges on a database, table, view, or procedure, or to grant a role to a user or another role.

- Use the REVOKE statement to revoke privileges on a database or database object, or to revoke a role from a user or from another role.

Roles

- A Role is a database feature that lets the DBA standardize and change the access privileges of many users by treating them as members of a class.
- User-defined roles cannot be granted the database-level privileges CONNECT, RESOURCE, or DBA.
- Use CREATE ROLE statement to define a role.

Table Access

- What kind of access do users have to the contents of the database?
- How do you restrict access to certain information in a database?

Table Access

- Table Privileges
- GRANT / REVOKE

Table Privileges

- When you create a table with the CREATE TABLE statement, you are the table owner and receive all table-level privileges.
- Cannot transfer ownership to another user, but can grant table-level privileges to another user or to a role.
- RENAME TABLE statement can change both the name and the ownership of a table.
- A user with the database-level DBA privilege receives all table-level privileges on every table in that database.

Table Privileges

Privilege	Effect
INSERT	Lets you insert rows
DELETE	Lets you delete rows
SELECT	Lets you access any column in SELECT statements. You can restrict the Select privilege to one or more columns by listing the columns.
UPDATE	Lets you access any column in UPDATE statements. You can restrict the Update privilege to one or more columns by listing the columns.
REFERENCES	Lets you define referential constraints on columns. You must have the Resource privilege to take advantage of the References privilege. (You can add, however, a referential constraint during an ALTER TABLE statement without holding the Resource privilege on the database.) You need only the References privilege to indicate cascading deletes. You do not need the Delete privilege to place cascading deletes on a table. You can restrict the References privilege to one or more columns by listing the columns.
INDEX	Lets you create permanent indexes. You must have the Resource privilege to use the Index privilege. (Any user with the Connect privilege can create an index on temporary tables.)

Table Privileges

Privilege	Effect
ALTER	Lets you add or delete columns, modify column data types, add or delete constraints, change the locking mode of the table from PAGE to ROW, or add or drop a corresponding ROW data type for your table. It also lets you enable or disable indexes, constraints and triggers, as described in "SET Database Object Mode statement" on page 2-737. You must have the Resource privilege to use the Alter privilege. In addition, you also need the Usage privilege for any user-defined data type affected by the ALTER TABLE statement.
UNDER	Lets you create sub-tables under a typed table.
ALL	Provides all privileges listed above. The PRIVILEGES keyword is optional.

GRANT / REVOKE

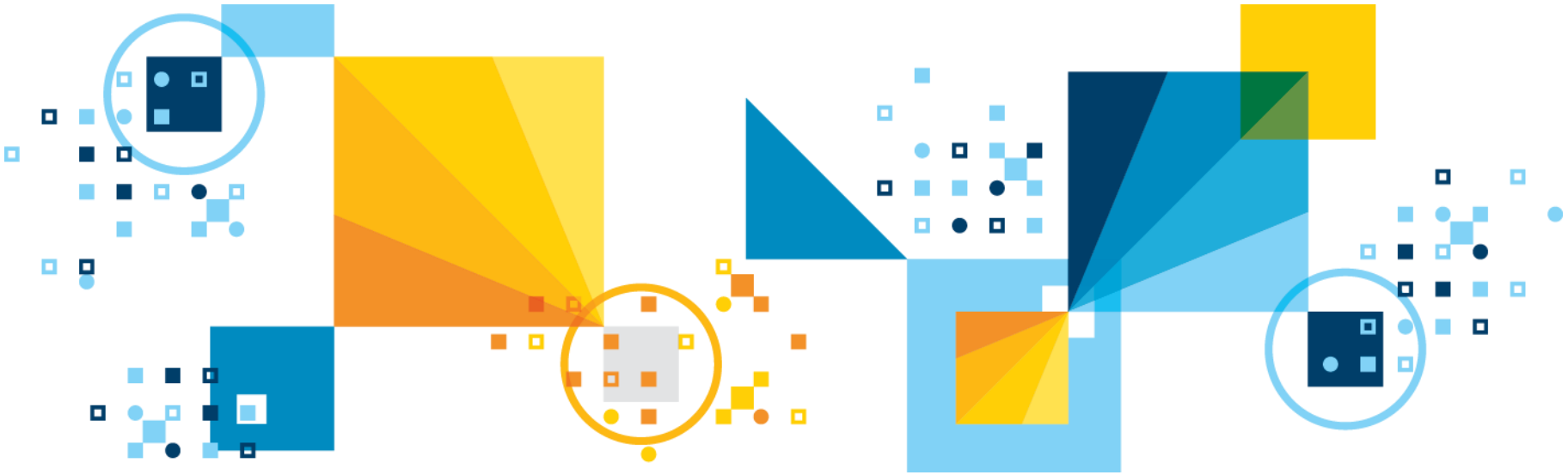
- Use GRANT / REVOKE statements to assign/revoke Table-level privileges.
- You can narrow the scope of a SELECT, UPDATE, or REFERENCES privileges by specifying the columns to which the privilege applies.
- Specify the keyword PUBLIC as user if you intend the GRANT statement to apply to all users.

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Locking at All Levels



Locks

- **What follows on the next several slides are database things not previously discussed herein that you would not do everyday, but could if you decided to do so as part of a server/data maintenance outage for the purposes of updating/deleting/modifying/masking/anonymizing your data.**

- **Don't do these things on a multi-user live system. Bad things can happen and backups will need to be restored; time, labor and money will be lost.**

- **Do a verified full database backup (called a level-0 backup in Informix parlance), first:**
  - **ontape –s –L 0** and **archecker –tdvs** # backup and verification (2 steps)
  - **onbar –b -v** # backup and verification (1 step, 2 processes)
    http://www-01.ibm.com/support/docview.wss?uid=swg21154059

Locks

- **Database server level locks of a kind are supported.**
- **The command line onmode -j option puts the database server into the administration mode and allows only the DBSA group and the user informix to connect to the server**
  - The -j option allows a DBSA to have the server in a fully functional mode to perform maintenance
    - To both database and database server
- **The -j -U option also enables the DBSA to grant individual users access to the database server in administration mode. Once connected, these named users can execute any SQL or DDL command**
- **When the server is changed to administration mode, all sessions for users other than user informix, the DBSA group users, and those identified in the onmode -j -U command lose their database server connection**
  - This is assured exclusivity to do really big data/server maintenance jobs, locks out anyone not needed.

Locks

- **Sounds ludicrous, but exclusively locking objects to assure no changes are made while you do an update or delete of data are an integral part of assured security; while an exclusive lock is held by an authorized user, no one else can update the object.**

- **Database level exclusive locking in SQL:**

```
DATABASE stores_demo@sfp_instance EXCLUSIVE;
```

- **If an opportunity arises, you can always RENAME DATABASE as well if you want to take an entire database offline (because, in theory only you know its name and in reality, connections happen to databases first), as long as no one is connected to it when you rename it (onstat –g sql to find out):**

```
database sysmaster;
rename database stores_demo@sfp_instance to
stores_demo1@sfp_instance;
```

  - -329 errors for subsequent connection attempts by other users

Locks

- **Tables**
  - Can be locked in either exclusive mode or in shared mode; if you are doing operations at a table level and no one else is online, you can consider exclusive locks. Shared locks run the risk of others reading the data you are updating/deleting
    - Must have permissions to access the table.
    - **LOCK TABLE customer IN EXCLUSIVE MODE;**
    - **UNLOCK TABLE customer;**
  - Locking tables exclusively when the server is live multiuser is more problematic as it runs the danger of incomplete transactions and transaction rollbacks.

Locks

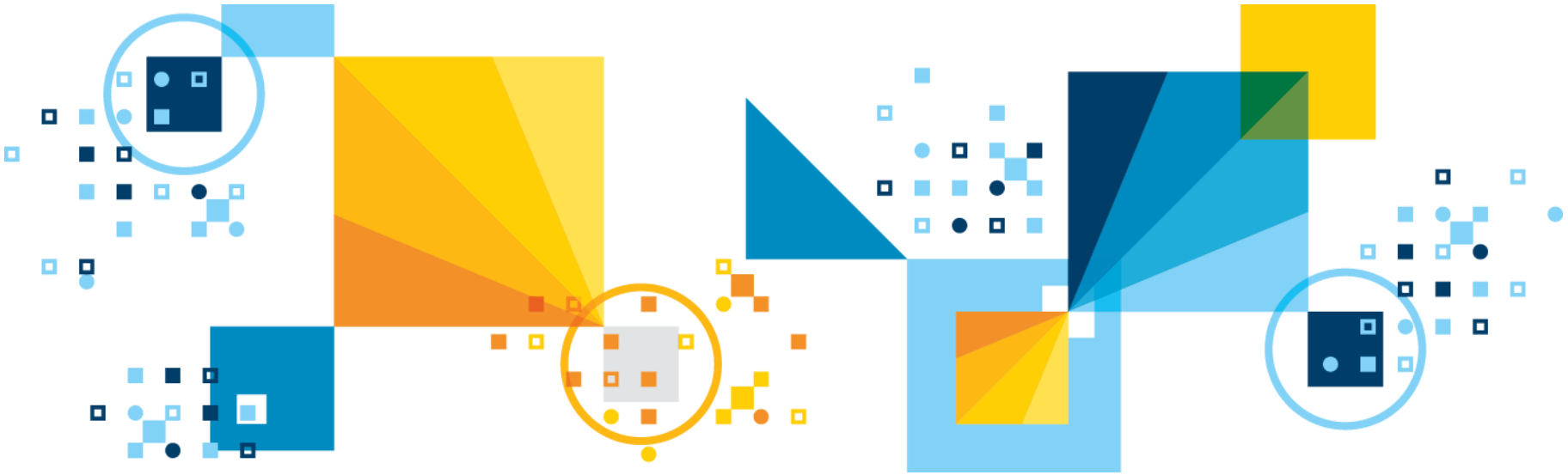
- **Tables**
  - Locking tables exclusively when the server is live multiuser is more problematic as it runs the danger of incomplete transactions and transaction rollbacks.
  - Better may be **REPEATABLE READ** isolation levels which locks all records in between a **BEGIN WORK** …. **COMMIT WORK** statement for the duration of the transaction and guarantees absolute exclusivity:
    - **SET ISOLATION REPEATABLE READ;** **(SET TRANSACTION REPEATABLE READ;)** if an ansi db
    - **SET LOCK MODE TO WAIT;**
    - Be careful with the number of records to be worked on with **repeatable read** isolation level; these are exclusive locks held until the transaction is complete and the effect on other users can be severe with long transaction rollbacks possible.
      - Less is better over multiple transactions – best practice.
  - Also, you can rename a table if no one is online
    - **RENAME TABLE customer to customer1;**
      - Make sure, if needed, you rename it back to the original table name when done

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Informix Data Sunset Capabilities



Data Sunset Capabilities

- **Support for online roll on/roll off of user defined table/index fragment organization of structured and unstructured data for easy sunset by, for example: date/time/place/range/list/partition of user-defined id or key columns**

- **User defined storage, user-defined data types, and unstructured data storage is a hallmark of Informix, with the key field based fragmentation (expression, range, list, round robin, and partition) capabilities of Informix lend themselves to easy disposal via target tables and retrieval of otherwise not easily visible data.**

- **Cascading deletes for easy complete deletion of user data across primary and secondary tables in relational database settings.**

- **Online index builds for situations not previously thought of needing indexes in relation to user data**

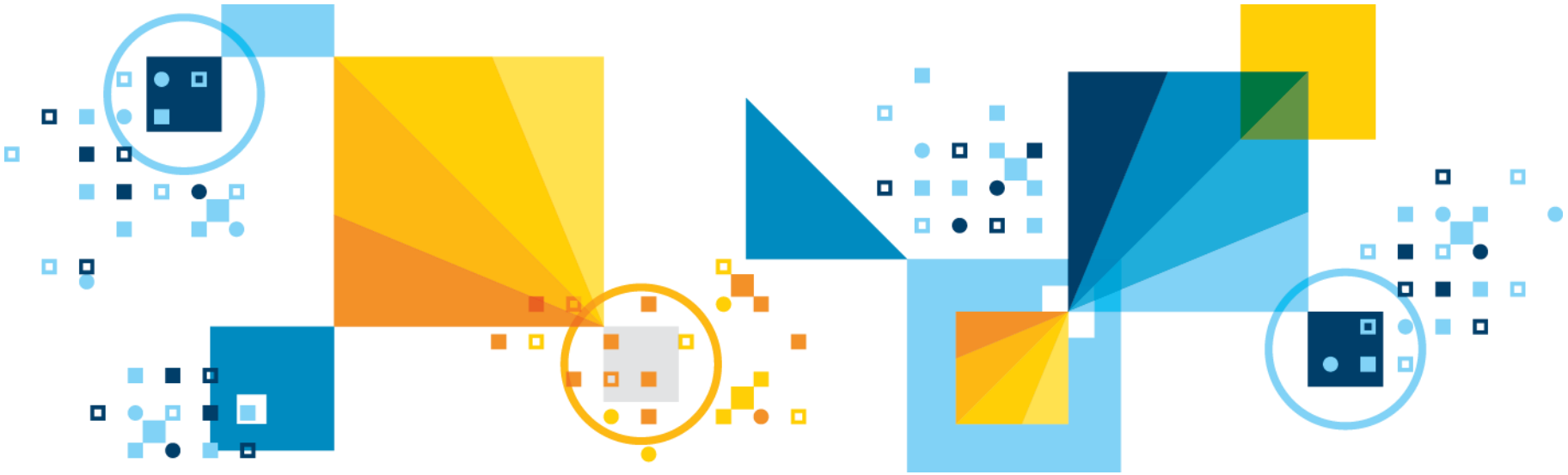
Data Sunset Capabilities

- **Exact location inference of Informix data possible via host name/server instance name/database name/owner name/table name combo.**
  - Not spatial/geospatial.
- **Physical disk location of data possible via host name/server instance name/database name/dbspace name/chunk device name/table name combo**

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Informix on Cloud and GDPR



Agenda

- **Privacy Shield Certification**
- **Informix on Cloud & Encryption at Rest**
- **Private cloud (ICIAE), fully Softlayer compliant with the standards of IBM Cloud via VPN.**
  - Same service as Informix on Cloud except via Virtual Private Network (VPN)

IBM Approved for Privacy Shield with EEA Personal Data

- US Privacy Shield Approval by US Department of Commerce
- Informix on Cloud is now Privacy Shield certified

"IBM's application to join the Privacy Shield has now been approved by the Department of Commerce. This means that the 140+ XaaS offerings listed at the bottom of our Privacy Shield Privacy Policy, available here, are now Privacy Shield certified. That is, when clients choose to host data in the United States via the listed offerings, the transfers are covered without any further action by the client or IBM. European Model Clauses are not required by law in these cases, although we are always willing to put European Model Clauses in place as clients request them."

- References to Safe Harbor in any and all Cloud agreements may now be eliminated and replaced with Privacy Shield.

IBM Approved for Privacy Shield with EEA Personal Data

- **"As the Privacy Shield only applies to personal data transferred from European Economic Area (EEA) to the United States, this Statement only applies to personal data from the EEA that is hosted in the United States through the Privacy Shield-Certified Cloud Services. This Policy does not apply when clients choose to have their offering content hosted in other countries."**

Informix on Cloud

- **Hosted Service via Softlayer, billed monthly, over public Internet.**
- **Purchased thru IBM Cloud and a salesperson, or via credit card at IBM Cloud.**
- **Customer is responsible for everything once the instance is provisioned and initial passwords are changed.**
- **Default single instance provisioned, additional available.**
- **Encryption at Rest enabled from the beginning default.**
  - This can be turned off by the user.

Softlayer Data Center Availability

- **Currently:**
  - London
  - Dallas
  - Sydney/Melbourne
  - Chennai (new)

- **More as time and need permits**

Optional Services

- **IBM Informix on Cloud Jump Start Remotely Delivered Setup – expires after 90 days.**

- **IBM Informix on Cloud Accelerator Remotely Delivered Setup – expires after 1 year.**

- **These setup services deliver 50 hours of remote consulting time so that organizations get expert guidance with setup activities and quickly get productive use of one or more software as a service (SaaS) offerings. Organizations can easily obtain expert assistance to get up and running at the time of SaaS signup.**

IBM Approved for Privacy Shield with EEA Personal Data

- US Privacy Shield Approval by US Department of Commerce
- Informix on Cloud is now Privacy Shield certified

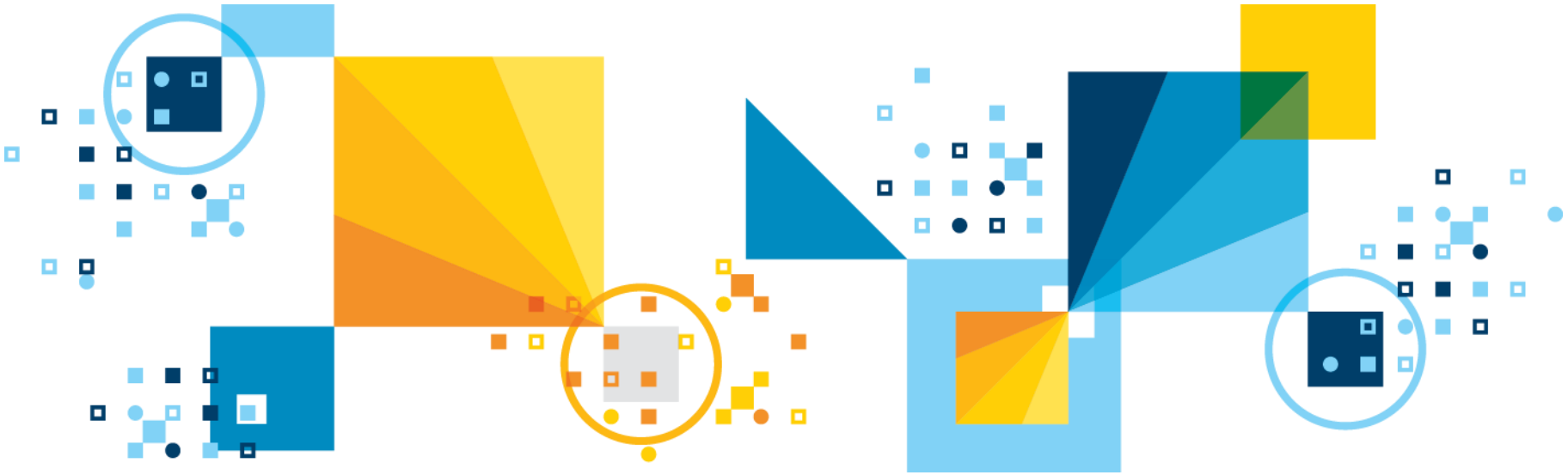
"IBM's application to join the Privacy Shield has now been approved by the Department of Commerce. This means that the 140+ XaaS offerings listed at the bottom of our Privacy Shield Privacy Policy, available here, are now Privacy Shield certified. That is, when clients choose to host data in the United States via the listed offerings, the transfers are covered without any further action by the client or IBM. European Model Clauses are not required by law in these cases, although we are always willing to put European Model Clauses in place as clients request them."

- References to Safe Harbor in any and all Cloud agreements may now be eliminated and replaced with Privacy Shield.

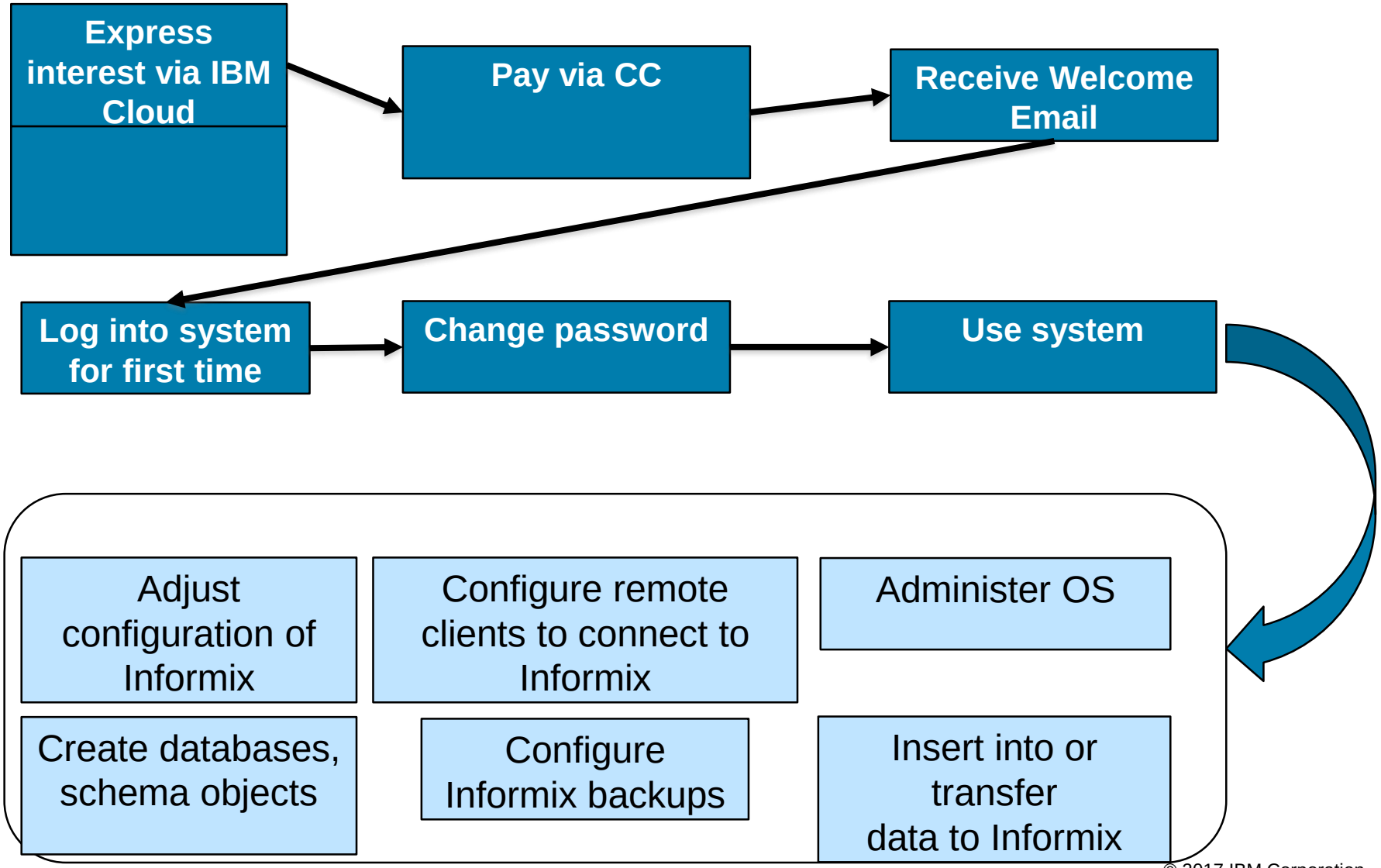
IBM Approved for Privacy Shield with EEA Personal Data

- **"As the Privacy Shield only applies to personal data transferred from European Economic Area (EEA) to the United States, this Statement only applies to personal data from the EEA that is hosted in the United States through the Privacy Shield-Certified Cloud Services. This Policy does not apply when clients choose to have their offering content hosted in other countries."**

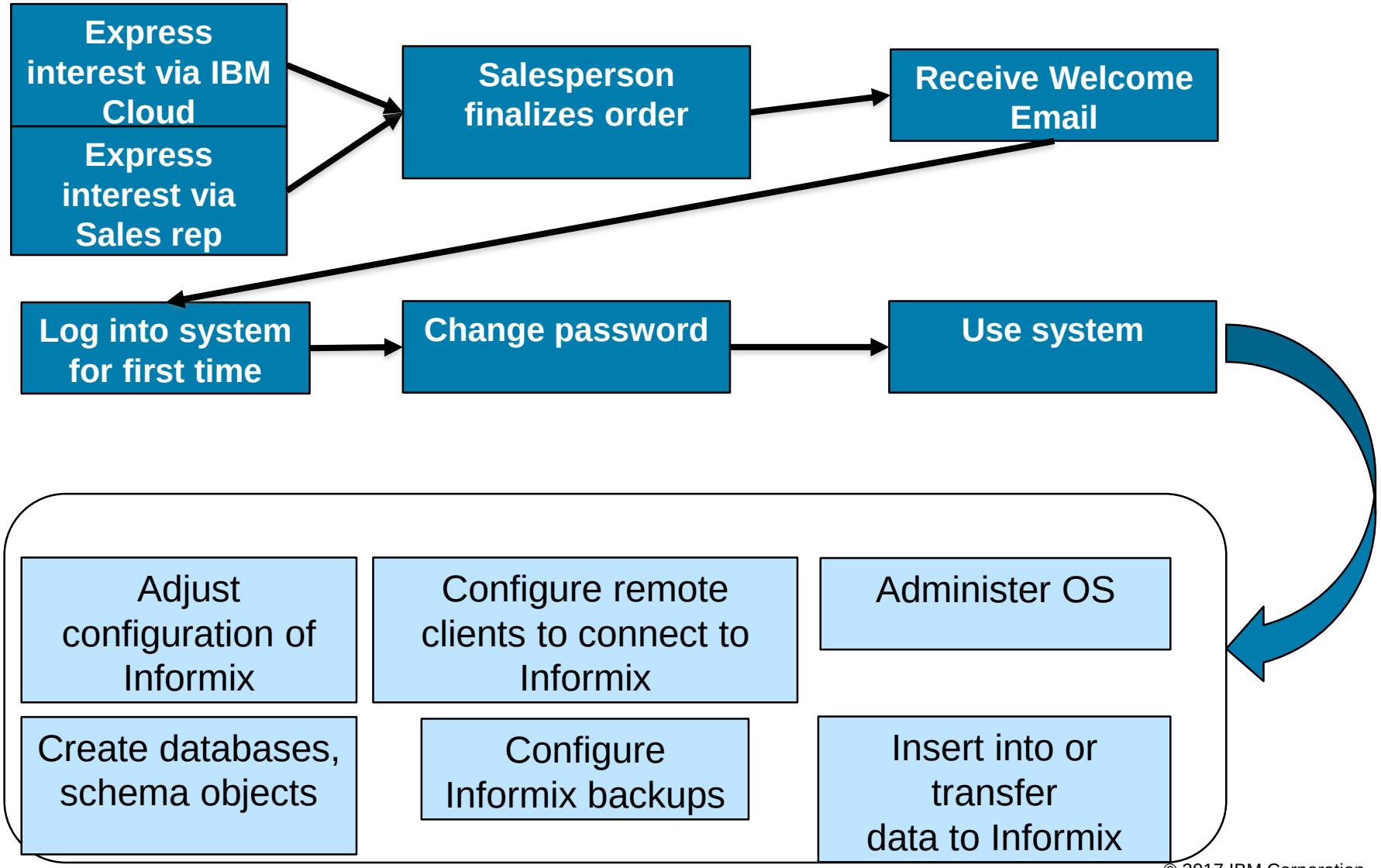
Customer Experience



What does a customer experience via credit card?



What does a customer experience via salesperson?



Related Questions

- **How long does it take from expressing interest to logging in for the first time?**
  - Days
- **What decisions must a customer make?**
  - T-shirt size
  - What SoftLayer datacenter(s)
  - How many machines
  - Number of initial virtual provisioned instances on a machine
  - Requested delivery date
- **Customer has no choice about version of Informix, OS, etc.**
- **After a system is delivered who administers it?**
  - The customer and only the customer.
  - IBM doesn’t even log in!

Customer Experience: Discovering the offering

Customer is approached by salesperson or finds it in IBM Cloud catalog and can either pay monthly later or with a credit card

	PLAN	FEATURES	PRICING
✓	**Small**	**Private 2 x 2.0 GHz Cores** 8GB RAM 1x100GB (SAN), 1x500 GB (SAN); 100GB at 500 IOPS 1 Gbps Network	$1,250.00 USD/Instance
		IBM Informix on Cloud Small plan gives you an Informix server that is installed into a virtual server configuration. The Small plan is suitable for development, customization, and functional testing operations. This plan requires a minimum charge of 30 days of usage. After the initial 30 days you may cancel any time, however you will be billed for usage until the end of that month.	
	Medium	**Private 4 x 2.0 GHz Cores** 16GB RAM 1x100GB (SAN), 1x1TB (SAN); 100GB at 1200 IOPS 1 Gbps Network	$2,200.00 USD/Instance
	Large	**Private 8 x 2.0 GHz Cores** 32GB RAM 1x100GB (SAN), 1x2TB (SAN); 100GB at 1600 IOPS 1 Gbps Network	$4,000.00 USD/Instance
	Extra Large	**Bare metal server 12 x 2.4 GHz Xeon Cores** 128GB RAM 2x800GB SSD configured with RAID 1 (~800GB), 6x1.2TB SSD configured with RAID 10 (~3.5TB) 10 Gbps Redundant Network	$8,000.00 USD/Instance

IBM Informix on Cloud

IBM

PUBLISH DATE
09/13/2016

AUTHOR
IBM

TYPE
Service

LOCATION
US South

VIEW DOCS

IBM Informix on Cloud offering provides a Informix database on IBM's SoftLayer global cloud infrastructure. It offers customers the rich features of an on-premise Informix deployment without the cost, complexity, and risk of managing their own infrastructure

- **Optimized for OLTP and fully configurable**

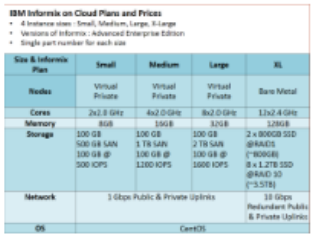
  IBM Informix server comes with a pre-configured instance optimized for online transaction processing applications. IBM Informix on Cloud also allows customers the flexibility to create their own instances for analytic or mixed workloads.

- **Contact sales to order**

  To order, contact your Americas Call Centers, local IBM representative, or your IBM Business Partner. To identify your representative or partner call 800-426-4968. For more information, contact the Americas Call Centers. Phone: 800-IBM-CALL (426-2255) Fax: 800-2IBM-FAX (242-6329).

- **Reduces time to value**

  Use of this offering reduces the time required for provisioning and deploying Informix so more resources can be devoted to developing new solutions and innovation.



IBM Informix on Cloud Plans and Prices

- 4 instance sizes - Small, Medium, Large, X-Large
- Versions of Informix - Advanced Enterprise Edition
- Single part number for each size

Size & Informix Plan	Small	Medium	Large	XL
Nodes	Virtual Private	Virtual Private	Virtual Private	Bare Metal
Cores	2x2.0 GHz	4x2.0 GHz	8x2.0 GHz	12x2.4 GHz
Memory	8GB	16GB	32GB	128GB
Storage	100 GB 500 GB SAN 100 GB @ 500 IOPS	100 GB 1 TB SAN 100 GB @ 1200 IOPS	100 GB 2 TB SAN 100 GB @ 1600 IOPS	2 x 800GB SSD @RAID1 (~800GB) 8 x 1.2TB SSD @RAID 10 (~3.5TB)
Network	1 Gbps Public & Private Uplinks			10 Gbps Redundant Public & Private Uplinks
OS	CentOS			

Pick a plan

Monthly prices shown are for country or region: United States

Plan	Features	
✓ **Small**	Nodes: Virtual Private Cores: 2 x 2.0 GHz	$50.00 USD/Monthly

Add Service

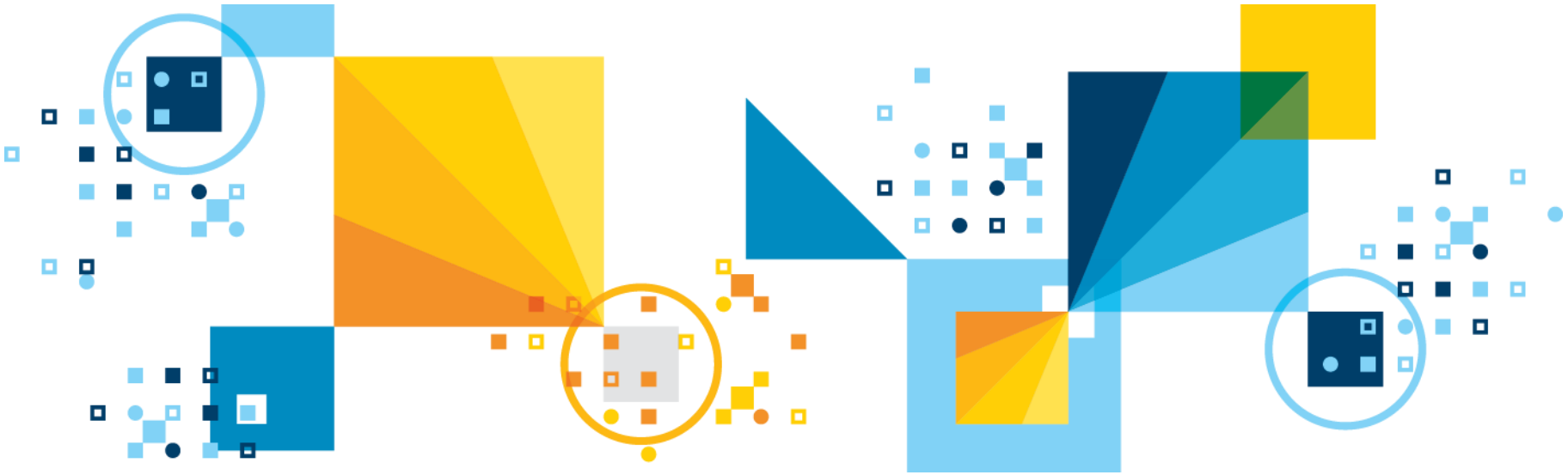
Space:
dev

Service name:
IBM Informix on Cloud-q0

Selected Plan:
Small

REQUEST

Machine Details

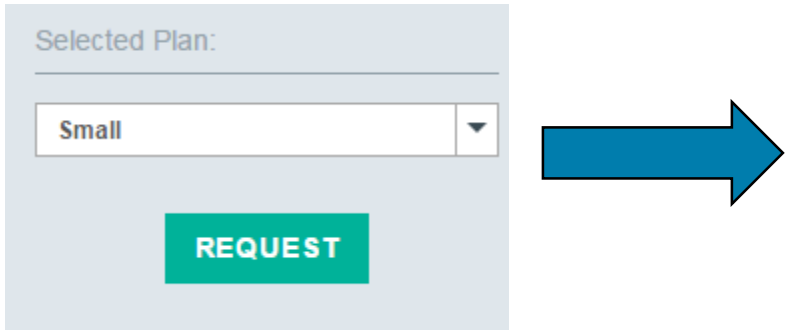


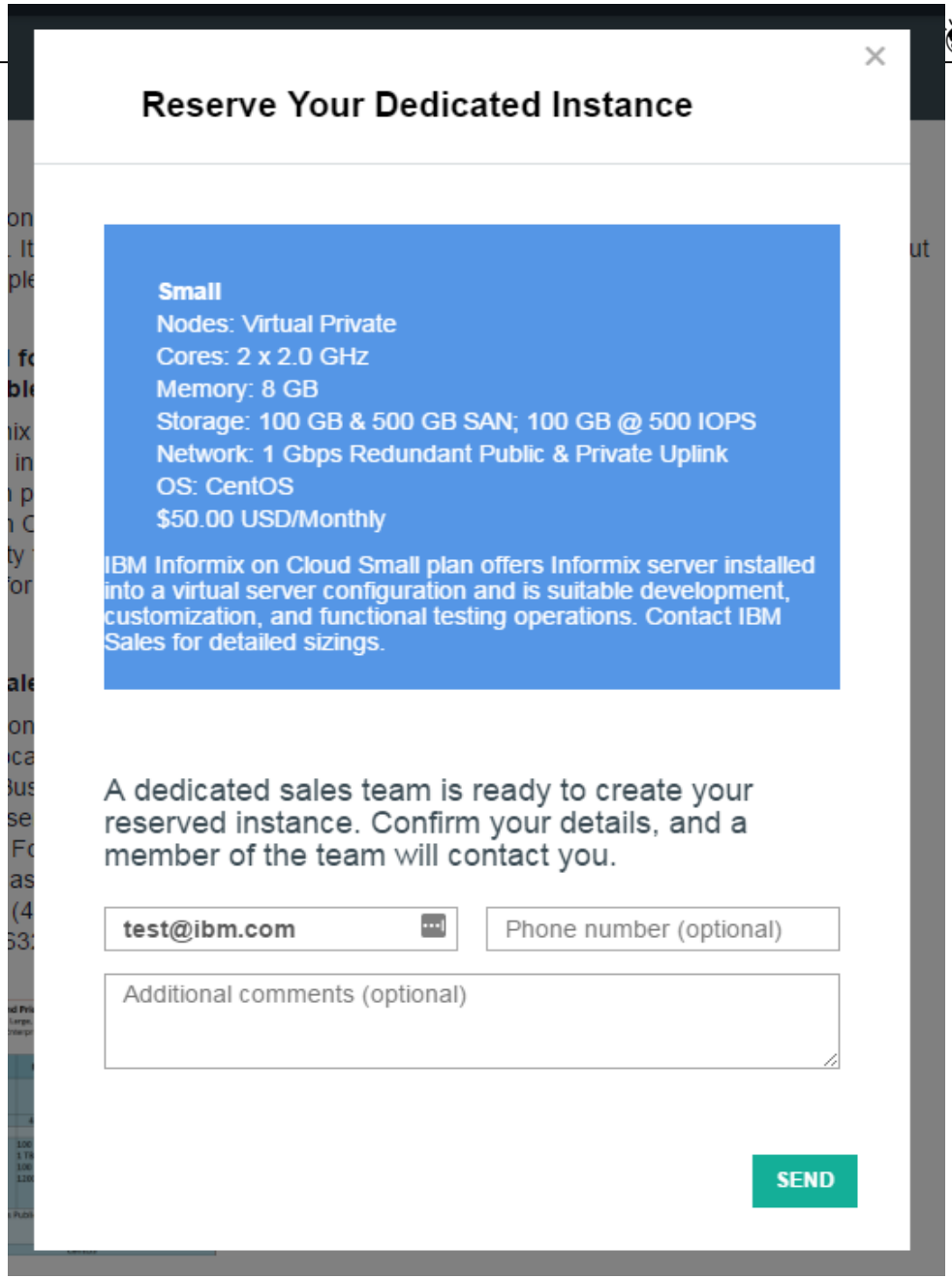
T-shirt sizes

Size & Informix Plan	Small	Medium	Large	XL
Nodes	Virtual Private	Virtual Private	Virtual Private	Bare Metal
Cores	2x2.0 GHz	4x2.0 GHz	8x2.0 GHz	12x2.4 GHz
Memory	8GB	16GB	32GB	128GB
Storage	100 GB 500 GB SAN 100 GB @ 500 IOPS	100 GB 1 TB SAN 100 GB @ 1200 IOPS	100 GB 2 TB SAN 100 GB @ 1600 IOPS	2 x 800GB SSD @RAID1 (~800GB) 8 x 1.2TB SSD @RAID 10 (~3.5TB)
Network	1 Gbps Public & Private Uplinks			10 Gbps Redundant Public & Private Uplinks
OS	CentOS			

- Monthly prices:

  small: $1,250 medium: $2,200 large: $4,000 xl: $8,000





Customer Receives Interest Confirmation Email

Details of your Informix Hosted request
Codename: Bluemix to: Nicholas Geib
08/11/2016 11:58 AM
Hide Details

From: "Codename: Bluemix" <no-reply@admin.ibmcloud.com>
To: Nicholas Geib/Lenexa/IBM@IBMUS
Please respond to "Codename: Bluemix" <no-reply@admin.ibmcloud.com>

Security: To ensure privacy, images from remote sites were prevented from downloading. Show Images

Hi NICHOLAS GEIB,

Thank you for your request to reserve an instance of Informix Hosted.

Your Reserve Instance Details:

Service: Informix Hosted

Plan: Small

Details: Informix Hosted on Cloud Small plan offers Informix server installed into a virtual server configuration and is suitable development, customization, and functional testing operations. Contact IBM Sales for detailed sizings.

Estimated Price: 50.00 USD

We'll be in touch

Thank you for sharing your contact information, our team will be in touch as soon as we can to get you up and running.

If you prefer to get in touch with us first, Contact us

Thank you!

Informix Hosted Team

Sales Purchase Contacts Customer

- **Lead emails go to Tomas Escobar, WW Informix Database Sales are forwarded appropriately.**
  - Understands the customer's needs
  - Gathers required information for the order
  - Ultimately enters quote in SQO system

- **Or the customer can download on a PAYGO basis from IBM Cloud**

Order Fulfilled; Customer Receives Welcome Email

WELCOME TO IBM Informix on Cloud!

We are excited to partner with you on this journey. Please leverage the resources outlined in this document to get up and running as quickly as possible.

LOG IN AND GET SET UP

Your new service is deployed and ready to use. For your first access to the provisioned system please log in using the following information::

Server Ip Address	Username	Password	Plan
<ip>	root	<pass>	Advanced-Small

This user has sudo/admin privileges. You will be required to change the password after you log in. Informix has been configured for SQLI, DRDA, Mongo, and REST clients to communicate over SSL. Please refer to our documentation for details on configuring a client to use SSL.

GET SUPPORT

Your IBM Customer Number is undefined
If you experience technical Issues, please contact our first-class Support Team: - View documentation:
https://console.ng.bluemix.net/docs/services/InformixOnCloud/InformixOnCloud.html

- Email us:support@bluemix.net
- Open a ticket : https://www.ng.bluemix.net/docs/troubleshoot/getting_customer_support.html
- Look for answers in our Support Forum: https://developer.ibm.com/answers/smartspace/bluemix/

PROVIDE FEEDBACK

Our Client Success Team is dedicated to making sure that you get the most out of your purchase. Please contact them with your questions or feedback: cdscsm@us.ibm.com.

Customer Logs In To Machine For First Time

```
$ ssh root@169.44.89.187
root@169.44.89.187's password:
You are required to change your password immediately (root enforced)
Changing password for root.
(current) UNIX password:
New password:
Retype new password:
[root@myinformixhost ~]#
```

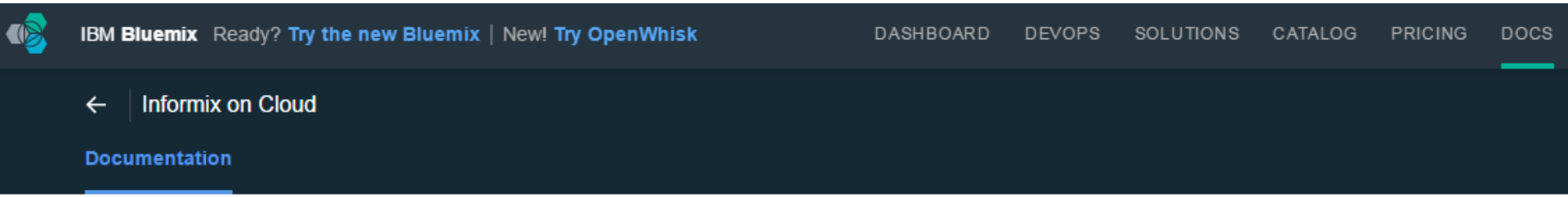
- **They must change the password immediately**
- **IBM doesn't know the password anymore. The system is entirely in the hands of the customer.**

Customer Uses the System

- The system is the customer's. They have full power and responsibility to be the system administrator, database administrator, network administrator, etc.
- Who takes backups of the system? The customer.
- Who configures client connectivity to the system? The customer.
- Who transfers data to/from the system? The customer.
- Who upgrades Informix? The customer.

Customer Needs Help

- **Documentation about the service:**
  - Getting started
- **Forums:**
  - Stack Overflow
  - IBM developerWorks dW Answers:
- **Support:**
  - Opening a Ticket



Getting help and support

If you have any questions or issues while using IBM Informix on Cloud for Bluemix, you can get help by doing one or more of the following:

- Navigate through or post questions on one of the forums.
  - When using forums to ask a question, make sure to tag your question so that it is seen by the IBM Bluemix development teams.
  - If you have technical questions about deploying an app with Informix on Cloud, post your question on Stack Overflow and tag your question with "bluemix" and "informixoncloud".
  - For questions about the service and getting started instructions, use the IBM developerWorks® dW Answers forum. Include the "bluemix" and "informixoncloud" tags.
  - See Getting help for more details about using the forums.
- Open a support ticket.
  - For information about opening an IBM support ticket, or about support levels and ticket severities, see Contacting support.

Details of an IBM Informix on Cloud Machine

- **OS: CentOS 7 (currently 7.2.1511)**

- **Informix version: 12.10.FC9AEE**
  - We plan to fulfill orders using the latest available Informix release. So at some point new orders will be fulfilled using 12.10.FC10 (Q4 in 2017)
- **File System:**
  - / for os
  - /data for dbspaces, logs, etc.
- **OS accounts:**
  - root, only customer knows password
  - informix, initially configured without a password
- **Accessible from public internet only:**
  - No VPN into SoftLayer (customer lacks SoftLayer account)
  - No private network

Details of an IBM Informix on Cloud Machine

- **Firewall ala iptables**
  - Allows traffic to SSH (22), SQLI over SSL (9089), DRDA over SSL (9091),
  - Mongo over SSL (27018), REST over SSL aka HTTPS (443)

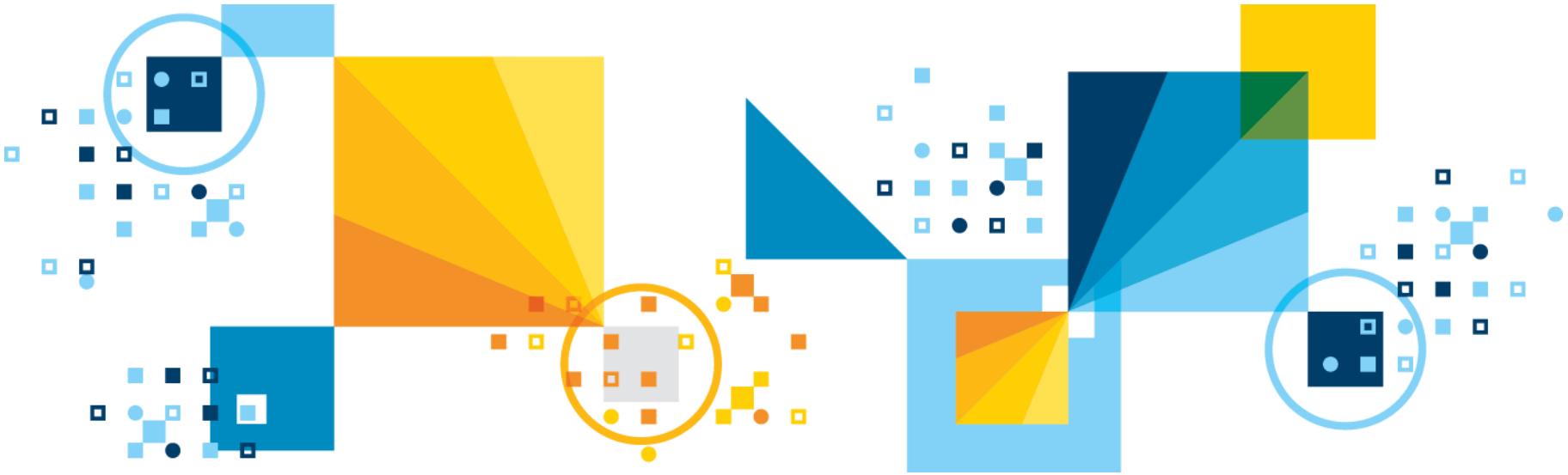
- **Informix and WL configured with a self-signed certificate**
  - Passwords to keystore in /home/informix/passwords

- **Informix also listens for non-SSL traffic: SQLI (9088), DRDA (9090), Mongo (27017), REST aka HTTP (80)**

- **Product locations**
  - $INFORMIXDIR /home/Informix/server
  - WL /opt/ibm/wire_listener
  - IWA /opt/ibm/iwa Only .tar; not configured

- **Plog, llogs, sysadmin each in own dedicated dbspace**

Configuring Remote Client Connectivity



Basic server side setup for client connectivity

- ssh root@169.46.137.242
- FYI: assume 169.46.137.242 is the IP address of host @ cloud.

- create user satyan with password 'blue4you' account unlock properties user nobody authorization(dbsa);

- create database db1 with log;
- grant dba to satyan;
- grant connect to satyan;

Import the SSL certificate to client

- cd /xyz/mykeystores

- sftp root@169.46.137.242
- get /home/informix/client_ssl/selfsigned_ssl.cert

Create the SSL keystore

- **userid informix gsk8capicmd_64 -keydb -create -db ol_hosted.kdb -pw blue4you -type cms –stash**

- **userid informix gsk8capicmd_64 -cert -add -db ol_hosted.kdb -pw blue4you -label selfsigned_ssl -file selfsigned_ssl.cert -format asci**

UPDATE $INFORMIXDIR/etc/conssl.cfg

- SSL_KEYSTORE_FILE /xyz/mykeystores/ol_hosted.kdb
- SSL_KEYSTORE_STH /xyz/mykeystores/ol_hosted.sth

cat $INFORMIXSQLHOSTS

- ol_hosted onsoctcp 169.46.137.242 9088
- ol_hostedssl onsocssl 169.46.137.242 9089

- ol_hosted_drda drsoctcp 169.46.137.242 9090
- ol_hosted_drdassl drsocssl 169.46.137.242 9091

Informix on Cloud Part Numbers & Costs – June 2017

D1Q49LL	IBM INFORMIX ON CLOUD ADVANCED LARGE PAYGO INSTANCE PAY PER USE	Tier 1	>=1	4000.00 PA->
D1Q47LL	IBM INFORMIX ON CLOUD ADVANCED SMALL PAYGO INSTANCE PAY PER USE	Tier 1	>=1	1250.00 PA->
D1Q48LL	IBM INFORMIX ON CLOUD ADVANCED MEDIUM PAYGO INSTANCE PAY PER USE	Tier 1	>=1	2200.00 PA->
D1Q4ALL	IBM INFORMIX ON CLOUD ADVANCED XLARGE PAYGO INSTANCE PAY PER USE	Tier 1	>=1	8000.00 PA->
D1Q0ELL	IBM INFORMIX ON CLOUD ADVANCED XLARGE INSTANCE PER MONTH	Tier 1	>=1	8000.00 PA->
D1Q0DLL	IBM INFORMIX ON CLOUD ADVANCED LARGE INSTANCE PER MONTH	Tier 1	>=1	4000.00 PA->
D1Q06LL	IBM INFORMIX ON CLOUD ACCELERATOR REMOTELY DELIVERED ENGAGEMENT SET UP	Tier 1	>=1	12700.00 PA->
D1Q05LL	IBM INFORMIX ON CLOUD JUMP START REMOTELY DELIVERED ENGAGEMENT SET UP	Tier 1	>=1	12700.00 PA->
D1Q0CLL	IBM INFORMIX ON CLOUD ADVANCED MEDIUM INSTANCE PER MONTH	Tier 1	>=1	2200.00 PA->
D1Q0BLL	IBM INFORMIX ON CLOUD ADVANCED SMALL INSTANCE PER MONTH	Tier 1	>=1	1250.00 PA->
D1Q4BLL	IBM INFORMIX ON CLOUD BLUEMIX PAYGO SERVICE LEVEL AGREEMENT	Tier 1	>=1	0.00 PA->

ICIAE – Part Numbers and Costs – VPN Access for Informix

D1QZALL	IBM CLOUD INTEGRATED ANALYTICS SAN STORAGE 250 GIGABYTES PER MONTH	Tier 1	>=1	191.00 PA->
D1QZBLL	IBM CLOUD INTEGRATED ANALYTICS SAN STORAGE 500 GIGABYTES PER MONTH	Tier 1	>=1	346.00 PA->
D1Q3CLL	IBM CLOUD INTEGRATED ANALYTICS LARGE SERVER INSTANCE PER MONTH	Tier 1	>=1	804.00 PA->
D1Q3DLL	IBM CLOUD INTEGRATED ANALYTICS SAN STORAGE 100 GIGABYTES PER MONTH	Tier 1	>=1	98.00 PA->
D1Q3BLL	IBM CLOUD INTEGRATED ANALYTICS MEDIUM SERVER INSTANCE PER MONTH	Tier 1	>=1	508.00 PA->
D1Q37LL	IBM CLOUD INTEGRATED ANALYTICS STANDARD SECURITY APPLIANCE INSTANCE PER MONTH	Tier 1	>=1	1400.00 PA->
D1Q34LL	IBM CLOUD INTEGRATED ANALYTICS ENVIRONMENT INSTANCE PER MONTH	Tier 1	>=1	1240.00 PA->
D1Q35LL	IBM CLOUD INTEGRATED ANALYTICS VPN CONNECTIVITY INSTANCE PER MONTH	Tier 1	>=1	762.00 PA->
D1Q3ALL	IBM CLOUD INTEGRATED ANALYTICS SMALL SERVER INSTANCE PER MONTH	Tier 1	>=1	346.00 PA->
D1Q38LL	IBM CLOUD INTEGRATED ANALYTICS ENTERPRISE SECURITY APPLIANCE INSTANCE PER MONTH	Tier 1	>=1	2650.00 PA->
D1Q39LL	IBM CLOUD INTEGRATED ANALYTICS EXTRA SMALL SERVER INSTANCE PER MONTH	Tier 1	>=1	202.00 PA->

Informix on Cloud on ICIAE now available via Sales Person

Informix on Cloud – PayGO Editions via IBM Cloud

	PLAN	FEATURES	PRICING
✓	**Small**	**Private 2 x 2.0 GHz Cores** 8GB RAM 1x100GB (SAN), 1x500 GB (SAN); 100GB at 500 IOPS 1 Gbps Network	$1,250.00 USD/Instance
		IBM Informix on Cloud Small plan gives you an Informix server that is installed into a virtual server configuration. The Small plan is suitable for development, customization, and functional testing operations. This plan requires a minimum charge of 30 days of usage. After the initial 30 days you may cancel any time, however you will be billed for usage until the end of that month.	
	Medium	**Private 4 x 2.0 GHz Cores** 16GB RAM 1x100GB (SAN), 1x1TB (SAN); 100GB at 1200 IOPS 1 Gbps Network	$2,200.00 USD/Instance
	Large	**Private 8 x 2.0 GHz Cores** 32GB RAM 1x100GB (SAN), 1x2TB (SAN); 100GB at 1600 IOPS 1 Gbps Network	$4,000.00 USD/Instance
	Extra Large	**Bare metal server 12 x 2.4 GHz Xeon Cores** 128GB RAM 2x800GB SSD configured with RAID 1 (~800GB), 6x1.2TB SSD configured with RAID 10 (~3.5TB) 10 Gbps Redundant Network	$8,000.00 USD/Instance

Informix on Cloud via IBM Cloud – via a Salesperson

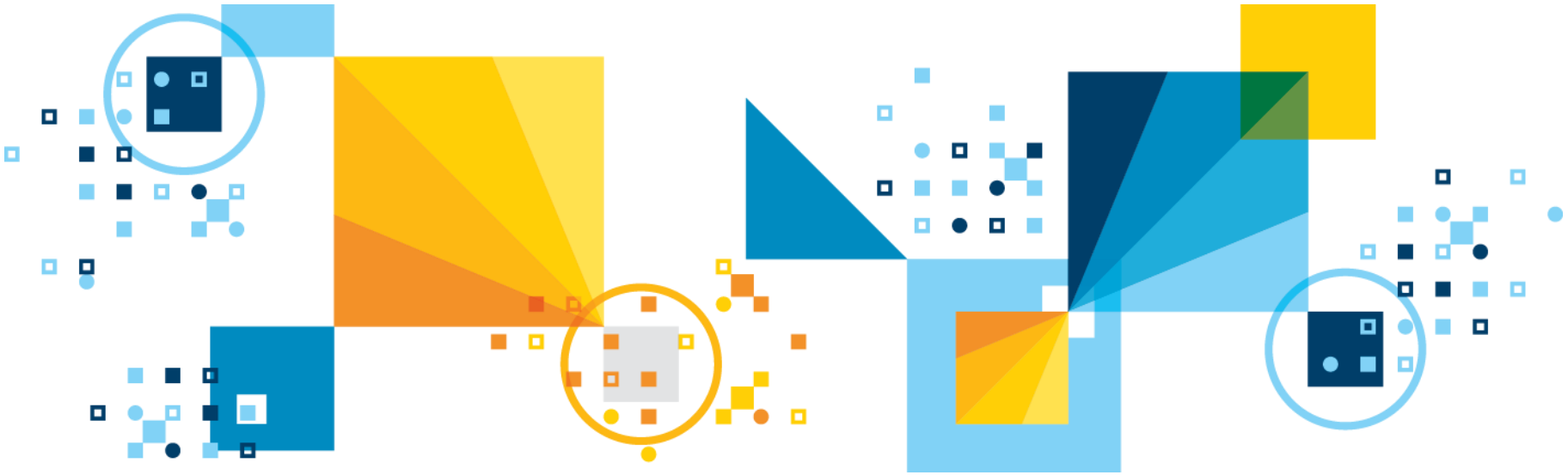
Small (Contact IBM Sales)	**Order with IBM Sales assistance** Private 2 x 2.0 GHz Cores 8GB RAM 1x100GB (SAN), 1x500 GB (SAN); 100GB at 500 IOPS 1 Gbps Network	$1,250.00 USD/Monthly
Medium (Contact IBM Sales)	**Order with IBM Sales assistance** Nodes: Virtual Private Cores: 4 x 2.0 GHz Memory: 16 GB Storage: 100 GB & 1 TB SAN; 100 GB @ 1200 IOPS Network: 1 Gbps Redundant Public & Private Uplink OS: CentOS	$2,200.00 USD/Monthly
Large (Contact IBM Sales)	**Order with IBM Sales assistance** Private 8 x 2.0 GHz Cores 32GB RAM 1x100GB (SAN), 1x2TB (SAN); 100GB at 1600 IOPS 1 Gbps Network	$4,000.00 USD/Monthly
Extra Large (Contact IBM Sales)	**Order with IBM Sales assistance** Bare metal server 12 x 2.4 GHz Xeon Cores 128GB RAM 2x800GB SSD configured with RAID 1 (~800GB), 6x1.2TB SSD configured with RAID 10 (~3.5TB) 10 Gbps Redundant Network	$8,000.00 USD/Monthly

Questions

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Appendix A - All GDPR Articles



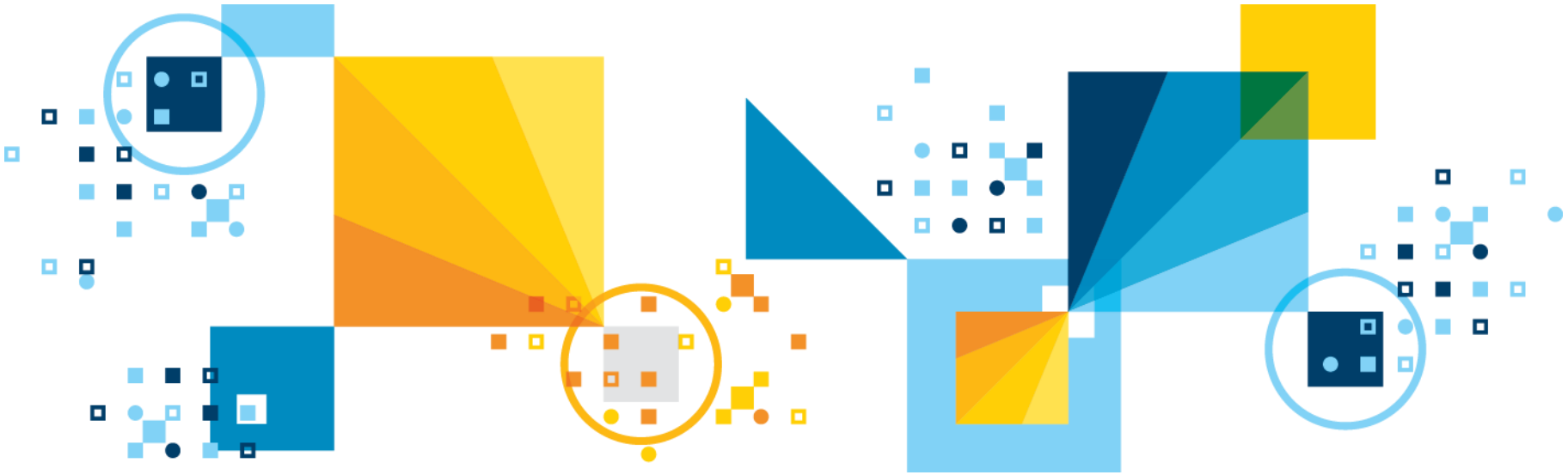
Appendix A – All of the GDPR Articles - Complete

- Below, in the speaker notes, are all of the GDPR Articles as currently constituted as of June 30, 2017.
- The source for these, as first verbatim published, is the English Language edition of the Official Journal of the European Union, dated 4.5.2016 Pages 32-88.
- If these should change in the future, this slide will become obsolete.
- These will apply in law in full May 25, 2018, per Article 99.

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Appendix B - All GDPR Recitives



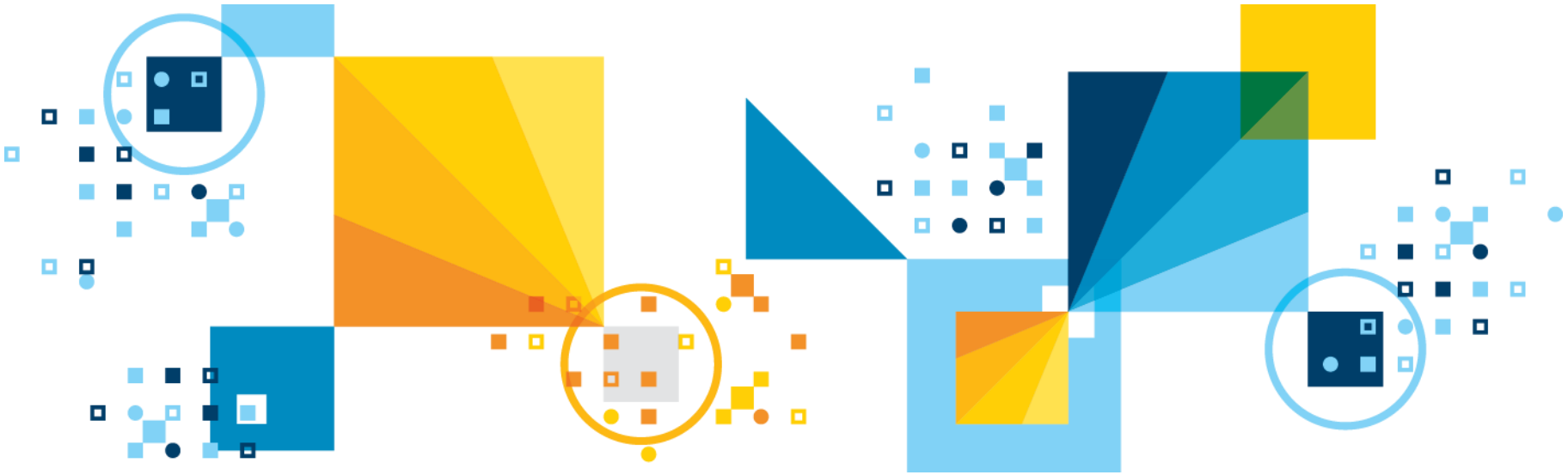
Appendix B - GDPR Recitives – In Full

- These are the Regulations of the General Provision Articles found in the speaker notes below.
- The source for all of these, as was first verbatim published, is the English Language edition of the Official Journal of the European Union, dated 4.5.2016 Pages 1-31.
- If these should change in the future, this slide will become obsolete.
- These will apply in law in full May 25, 2018, per Article 99.
- There are 173 Recitives, or regulations, the two words used interchangably. So in the notes below, “(173)”, at the beginning of a paragraph, marks Recitive 173. Similar for “(78)”, and so on.

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Appendix C – GSKit Setup



Global Security Kit - Overview

- **IBM Global Security Kit (GSKit) is a library and set of command-line tools that provides SSL implementation along with base cryptographic functions (symmetric and asymmetric ciphers, random number generation, hashing, and so on) and key management:**
  - Underlying cryptographic library, IBM Crypto for C, is FIPS certified.

- **Some products expose and even require a user to use GSKit utilities for certain tasks, while others wrap GSKit's capabilities in their own interfaces.**
  - GSKit is used by many IBM software products for its security, usability, and FIPS certification.
  - GSKit 7 and GSKit 8 are by default installed by Informix as part of the Informix Dynamic Server installation.

- **GSKit is a component and not a stand-alone product. It is not obtainable independent of the products that ship it. GSKit support and updates are provided as part of other products' support and updates.**

GSKit Setup

- **GSKit supports two installation methods: global and local**
  - Both types of installations may be present on a system at the same time.

- **On a global installation, a single GSKit instance is shared by multiple products. In this configuration, GSKit libraries and executable files are placed in a common location on the system outside of the product's installation directory. If more than one product uses the same GSKit version, these products will not create multiple copies of GSKit, but instead share the single global copy.**

- **On a local installation, each product has its own, private version of GSKit. In this configuration, GSKit files are placed somewhere within the product's directory structure and their location may or may not be documented. If a global installation exists on the system, it is ignored by the product, which uses only its local installation of GSKit.**

GSKit Setup

- **Different major versions of GSKit (for example, version 7 and version 8) are separate and can coexist as global installations.**

- **This presentation discusses GSKit versions 7 and 8 only, and not any prior versions.**

- **All examples are given for GSKit 8.**
  - Unless noted otherwise, the same commands and options that are provided in the examples also work in version 7.

Finding GSKit

- **The GSKit command-line tool is named as follows:**
  - **gsk**<**version**>**capicmd**[**_64**]
    - Where <**version**> is the GSKit major version (either 7 or 8).
    - The **_64** suffix is added on 64-bit platforms.

- **Determine if there is a global installation of GSKit:**
  - On UNIX or Linux, enter one of the following commands, **gsk7capicmd_64** or **gsk8capicmd_64**, on the command line.
    - If anything other than an error message is returned, GSKit is installed and ready to use.
  - On Windows, open the Registry Editor and look for one of the following keys:
    - **HKEY_LOCAL_MACHINE\SOFTWARE\IBM\gsk8\CurrentVersion\InstallPath** or **HKEY_LOCAL_MACHINE\SOFTWARE\IBM\gsk7\CurrentVersion\InstallPath** These keys indicate where GSKit is installed.
  - Search the product's install directories or the entire file system/disk for files and directories containing "**gsk**." There are two subdirectories, **lib** and **bin**, containing GSKit shared libraries and binaries.

Environment to Run GSKit – Unix and Linux

- **For global installations of GSKit, no configuration is needed. The command-line tool is already on the executable path, and the libraries are in their standard system location. The GSKit commands can be run from any terminal window.**

- **For local installations of GSKit, add its shared libraries directory to your environment:**

  export <Shared library path environment variable>=<GSKit library path>

  export PATH=$PATH:<GSKit binary path>

- **The shared library path variable name depends on your platform:**

Platform	Variable Name
AIX	LIBPATH
HP-UX	SHLIB_PATH
Linux/Solaris	LD_LIBRARY_PATH

Environment to Run GSKit – Unix and Linux, Windows

- **For example, to set the environment on Linux, use:**

  export LD_LIBRARY_PATH=/path/to/gskit/lib

  export PATH=$PATH:/path/to/gskit/bin

- **For Windows:**

  – Add both library and binary paths to the PATH environment variable.
    - You can do this either in a command-line window for a single session, or change the global settings.

  – To add the paths using a command line, type:

  set PATH=C:\path\to\IBM\gsk7\bin;C:\path\to\IBM\gsk7\lib;%PATH%

OpenSSL

- **Installing OpenSSL**
  - Some tasks in this presentation use OpenSSL.
  - See https://www.openssl.org/ for instructions on obtaining OpenSSL, and follow the OpenSSL instructions to install on your system.

Key Database Preparation

- **GSKit stores public and private keys and certificates in a key database. A key database consists of a file with a .kdb extension and up to three other files with .sth, .rdb, and .crl extensions.**

- **Your product may have already created a key database. If so, look at the product documentation to find its location. If you don't already have a key database, you need to create and initialize a new one.**

- **To create and initialize a new key database, run the following command (depending on your version):**
  - Version 7:
    - **gsk7capicmd_64 -keydb -create -db <filename>.kdb -pw <password> -stash**
  - Version 8:
    - **gsk8capicmd_64 -keydb -create -populate -db <filename>.kdb -pw <password> -stash**
- **(over)**

Key Database Preparation

- **To create and initialize a new key database (cont'd)**
  - **-db** indicates the file name for the new key database
  - **-pw** indicates the password to use to protect the key database file
  - **-populate** in version 8 is optional and tells GSKit to populate the key database with a number of predefined trusted CA certificates
    - Version 7 always populates the new key database with the predefined trusted CA certificates
  - **-stash** tells GSKit to save the specified key database password locally in the .sth file so that it doesn't have to be entered on the command line in the future

- **In the example scenarios in this tutorial, the following key database names are used:**
  - **server.kdb**: Server key database
  - **client.kdb**: Client key database
  - **ca.kdb**: Certificate Authority key database

Managing and Creating Self-Signed Certificates

- **A self-signed certificate consists of a public/private key pair and a certificate for the public key that is signed by the private key.**
  - Also known as a "root" certificate as it can be used to create a Certificate Authority.

- **Self-signed certificates can also be used in simple scenarios when both the client and the server are known to each other and can exchange certificates securely out-of-band.**

- **To generate a self-signed certificate and store it in the key database, use the following command:**

```
gsk8capicmd_64 -cert -create -db server.kdb -stashed -dn
"CN=myserver,OU=mynetwork,O=mycompany,C=mycountry" -expire 7300 -
label "My self-signed certificate" -default_cert yes
```

Managing and Creating Self-Signed Certificates - Parameters

- **To generate a self-signed certificate (cont'd)**
  - **-db** Specifies the key database where the self-signed certificate should be stored
  - **-dn** Specifies the distinguished name to use on the public key certificate
  - **-expire** Indicates the number of days the certificate is valid
  - **-label** A name to use for the self-signed certificate within the key database
  - **-default_cert** (optional) Makes the newly created certificate the default

Installing the Certificate On Client Systems

- **For the clients to trust a certificate, its public part needs to be distributed to the clients and stored in their key databases.**

- **The process for doing this is:**
  - Extract the public part to a file using the following command:
    - **gsk8capicmd_64 -cert -extract -db server.kdb -stashed -label "My self-signed certificate" -format ascii -target mycert.arm**
      - **-db** specifies the server key database that contains the certificate to be shared with clients.
      - **-label** specifies the certificate's label within the key database.
      - **-target** specifies the file name where the exported certificate should be stored.
  - Distribute **mycert.arm** to the clients.
  - Add the new certificate to the clients' key database as follows:
    - **gsk8capicmd_64 -cert -add -db client.kdb -stashed -label "Server self-signed certificate" -file mycert.arm -format ascii -trust enable**
      - **-db** specifies the name of the client's key database file.
      - **-label** specifies the label to be used for the certificate inside the key database file.
      - **-file** specifies the file containing the certificate to be imported.

Creating a Certificate Authority (CA) via GSKit

- **Initialize the CA key database and create the CA certificate**
- **For example:**
  - **gsk8capicmd_64 -keydb -create -db ca.kdb -pw mypass -stash**
  - **gsk8capicmd_64 -cert -create -db ca.kdb -stashed -dn CN=CA,O=CA,C=US -expire 7300 -label "CA cert" -default_cert yes -ca true**
    - **-db** specifies the file name to be used for the CA's key database file.
    - **-pw** specifies the password to use to protect the key database file.
    - **-expire** specifies the number of days before the certificate expires.
    - **-dn** specifies the distinguished name to use on the CA certificate.
    - **-label** specifies the name to be used for the CA certificate in the key database file.
- **Extract the CA's root certificate. This certificate must be installed at both the clients and servers:**
  - **gsk8capicmd_64 -cert -extract -db ca.kdb -stashed -label "CA cert" -format ascii -target ca.arm**
    - **-db** specifies the file name of the CA's key database file.
    - **-label** specifies the CA's certificate label in the key database file.
    - **-target** specifies the file that is stored in the exported CA certificate.

Issuing a Server Certificate With a CA

- **For clients to verify a server's identity, the CA must issue a signed server certificate to the server.**
- **The CA's root certificate must be added to the server's key database and marked as trusted, as follows:**
  - **gsk8capicmd_64 -cert -add -db server.kdb -stashed -label "My CA root" -file ca.arm -format ascii -trust enable**
    - **-db** specifies the name of the server's key database file.
    - **-label** specifies the label to use for the CA's root certificate in the database file.
    - **-file** specifies the file that contains the CA's root certificate.
- **At the server, create a server certificate request as follows:**
  - **gsk8capicmd_64 -certreq -create -db server.kdb -stashed -label "My CA signed certificate" -dn "CN=host.mycompany.com,OU=unit,O=company" -file cert_request.arm**
    - **-db** specifies the name of the server's key database file.
    - **-label** specifies the label to use for the server certificate in the key database file.
    - **-dn** specifies the distinguished name to use on the certificate.
    - **CN** specifies the DNS name of your server, which is necessary for an SSL client to validate the certificate.

Issuing a Server Certificate With a CA

- **You can also request a subject alternative name (SAN) extension by using -san_dnsname or -san_ipaddr options (not supported in version 7). For example:**
  - **gsk8capicmd_64 -certreq -create -db server.kdb -stashed -label "My CA signed certificate" -dn "CN=host.mycompany.com,OU=unit,O=company" -san_dnsname "host1.mycompany.com,host2.mycompany.com" -san_ipaddr "10.10.10.1,10.10.10.2" -file cert_request.arm**

- **The certificate request must be transported to the CA, and the CA must sign the certificate as follows:**
  - **gsk8capicmd_64 -cert -sign -file cert_request.arm -db ca.kdb -stashed -label "CA cert" -target cert_signed.arm -expire 364**
    - **-file** specifies the file that contains the certificate request.
    - **-db** specifies the name of the CA's key database file.
    - **-label** specifies the label of the CA's root certificate that should be used to sign the certificate request.
    - **-target** specifies the file to be used for the signed server certificate.

Issuing a Server Certificate With a CA

- **If a SAN extension was requested in the server certificate request, you can either use the -preserve option to keep the requested values or override them by specifying your own -san_dnsname or -san_ipaddr options with the -sign command (not supported in version 7).**
  - If you use both **-preserve** with **-san_dnsname** or **-san_ipaddr**, the values are merged with the ones requested. For example
    - **gsk8capicmd_64 -cert -sign -file cert_request.arm -db ca.kdb -stashed -label "CA cert" -target cert_signed.arm -expire 364 -preserve -san_dnsname "host3.mycompany.com" -san_ipaddr "10.10.10.3"**

- **The server must receive the signed certificate from the CA and set it as the default for communicating with clients as follows:**
  - **gsk8capicmd_64 -cert -receive -db server.kdb -stashed -file cert_signed.arm -default_cert yes**
    - **-db** specifies the name of the server's key database file.
    - **-file** specifies the name of the file that contains the signed server certificate.

Distributing the CA root certificate to clients

- **For your clients to validate the signed certificate that they receive from the server during an SSL connection, they must trust your Certificate Authority.**
  - This is achieved by installing the CA root certificate on the clients.

- **Transfer the CA root certificate to clients. (See the ca.arm file created above.)**
- **Add the CA root certificate to the client key database and enable trust as follows:**
  - **gsk8capicmd_64 -cert -add -db client.kdb -stashed -label "My CA root" -file ca.arm -format ascii -trust enable**
    - **-db** specifies the client's key database file to store the CA's root certificate.
    - **-file** specifies the file that contains the CA's root certificate.

Using a Third-party Certificate Authority (CA)

- **Install the CA root certificate**
- **Instead of setting up its own CA, a company may use a third-party certificate authority to sign its server certificates. The client and server must have access to the third-party CA's root certificate to verify the server certificates that are signed by the third-party CA.**

- **If the server is going to use certificates from a third-party CA whose root certificate is not shipped with GSKit, the third-party CA's root certificate must be imported to both the server and the clients' key database files as follows:**
  - Obtain the CA root certificate
    - The process for this varies depending on the third-party CA's procedures
    - Third-party CAs often make their root certificates available for download

Using a Third-party Certificate Authority (CA)

- **Add the third-party's root CA certificate to both server and client key databases and mark it as trusted as follows:**
  - **gsk8capicmd_64 -cert -add -db server.kdb -stashed -label "Some CA root" -file ca.der -format binary -trust enable**
  - **gsk8capicmd_64 -cert -add -db client.kdb -stashed -label "Some CA root" -file ca.der -format binary -trust enable**

- **The example above uses a third-party CA root certificate that is in a binary format. If the certificate is in an ASCII format, use the -format ascii option.**
  - **-db** specifies the name of the key database to import the third-party CA root certificate into.
  - **-label** specifies the label to use for the third-party CA root certificate inside the key database file.
  - **-file** specifies the file that contains the third-party CA root certificate.

Requesting a Certificate Using a Signing Request

- **In this scenario, GSKit creates a certificate request, the third-party CA signs the certificate in the request, and GSKit imports the signed certificate into the server key database.**

- **Generate a server certificate request using the server's key database file:**
  - **gsk8capicmd_64 -certreq -create -db server.kdb -stashed -label "Some CA signed certificate" -dn "CN=host.mycompany.com,O=company,C=country" -file cert_request.arm**
    - **-db** specifies the name of the server's key database file.
    - **-label** specifies a label to refer to the newly created certificate in the key database file.
    - **-dn** specifies the distinguished name to be used on the server's certificate.
    - **-file** specifies the file to contain the exported certificate signing request.
    - **CN** specifies the server DNS name; necessary for SSL client certificate validation.

Requesting a Certificate Using a Signing Request

- You can also request SAN extension by using -san_dnsname or -san_ipaddr options (not supported in version 7).

- For example:
  - gsk8capicmd_64 -certreq -create -db server.kdb -stashed -label "Some CA signed certificate" -dn "CN=host.mycompany.com,OU=unit,O=company" -san_dnsname "host1.mycompany.com,host2.mycompany.com" -san_ipaddr "10.10.10.1,10.10.10.2" -file cert_request.arm

Requesting a Certificate Using a Signing Request

- **In this scenario, GSKit creates a certificate request, the third-party CA signs the certificate in the request, and GSKit imports the signed certificate into the server key database.**
- **Generate a server certificate request using the server's key database file:**
  - **gsk8capicmd_64 -certreq -create -db server.kdb -stashed -label "Some CA signed certificate" -dn "CN=host.mycompany.com,O=company,C=country" -file cert_request.arm**
    - **-db** specifies the name of the server's key database file.
    - **-label** specifies a label to refer to the newly created certificate in the key database file.
    - **-dn** specifies the distinguished name to be used on the server's certificate.
    - **-file** specifies the file to contain the exported certificate signing request.
    - **CN** specifies the server DNS name; necessary for SSL client certificate validation.

Requesting a Certificate Using a Signing Request

- You can also request SAN extension by using -san_dnsname or -san_ipaddr options (not supported in version 7).

- For example:
  - gsk8capicmd_64 -certreq -create -db server.kdb -stashed -label "Some CA signed certificate" -dn "CN=host.mycompany.com,OU=unit,O=company" -san_dnsname "host1.mycompany.com,host2.mycompany.com" -san_ipaddr "10.10.10.1,10.10.10.2" -file cert_request.arm

Requesting a Certificate Using a Signing Request

- **Send the certificate request (that is, the cert_request.arm file) to the CA. The process for submitting a certificate signing request varies among CA companies. Often the signing request can be submitted using a web form.**

- **The CA then returns the signed certificate. In the scenario below, the assumption is that the signed certificate is in a file that is called cert_signed.arm and is in an ASCII format.**

- **To receive the signed certificate into the server's key database file and set it as the default for communicating with clients:**
  - **gsk8capicmd_64 -cert -receive -db server.kdb -stashed -file cert_signed.arm -default_cert yes**
    - **-db** specifies the name of the server's key database file.
    - **-file** specifies the name of the file that contains the signed certificate.

Requesting a Certificate Without a Signing Request

- **Some Certificate Authorities do not accept signing request files. Instead, they generate the signing request internally on behalf of the requesting server and then sign it as one transaction.**
  - The CA then returns to the server two files, one containing the private key for the server to use and one containing the signed server certificate.
  - In this example, the assumption of the two files is as follows:
    - **host.mycompany.com.crt:**
      - This is the file that contains the signed server certificate.
    - **host.mycompany.com.key**:
      - This is the file that contains the server's private key
    - To use these files, they must be converted to an industry standard format called PKCS12 before they can be imported into a key database.

Requesting a Certificate Without a Signing Request

- **Three steps:**
  - Use OpenSSL to convert the two files into a PKCS12 file as follows:
    - **openssl pkcs12 -export -in host.mycompany.com.crt -inkey host.mycompany.com.key -out host.mycompany.com.p12 -name "CA signed"**
      - Command prompts you to enter a password
      - This password is only used temporarily so it can be any arbitrary password. In this example, the password is set to **abc** **(see below)**
      - **-in** specifies the file that contains the signed server certificate
      - **-inkey** specifies the file that contains the server's private key
  - Import the certificate from the PKCS12 file to the server's key database file as follows:
    - **gsk8capicmd_64 -cert -import -db host.mycompany.com.p12 -pw abc -target server.kdb**
      - **-db** specifies the name of the PKCS12 file
      - **-pw** specifies the password that protects the PKCS12 file
      - **-target** specifies the name of the server's key database file

        You are prompted for the password that protects the target database file.

Requesting a Certificate Without a Signing Request

- **Three steps (cont'd):**
  - Make the imported certificate the default certificate to use for communications as follows:
    - **gsk8capicmd_64 -cert -setdefault -db server.kdb -stashed -label "CA signed"**
      - **-db** specifies the name of the server's key database file.
      - **-label** specifies a label of the imported certificate.

GSKit Security Considerations

- **Protecting private keys**

- **If an attacker obtains access to the private keys, the associated certificates can't be trusted, which compromises the servers that depend on them.**

- **Protect the key database file by:**
  - Using a strong password for the key database file.
  - Protecting the stored password file (the **.sth** file) using the file system's security mechanisms if you use the GSKit stashed password feature.
    - For example, you can set file permissions to restrict access to this file to certain users.
  - Restricting file system access to the key database file (the **.kdb** file) so that it is only readable by the users that run an application that uses the key database.

Verifying the Identity of Certificate Requesters

- If you manage your own Certificate Authority, ensure that any certificate signing request comes from an identity that is authorized to access the resource the requested certificate is for.

- The trustworthiness of certificates issued by the Certificate Authority is only as good as the process used to verify the identity of the requester.

Tips and Tricks - Listing Key Database Contents

- **To get a short list (labels only) of all certificates in a key database, use the following command:**
  - **gsk8capicmd_64 -cert -list -db server.kdb -stashed**
    - **-db** specifies the name of the key database file

- **To get detailed information about a particular certificate, use the following command:**
  - **gsk8capicmd_64 -cert -details -db server.kdb -stashed -label "My certificate"**
    - **-db** specifies the name of the key database file
    - **-label** specifies the label of the certificate in the database

Tips and Tricks - Switching Between Certificates

- **A server's key database file can have multiple server certificates in it. However, only one certificate, which is known as the default certificate, can be used by the server at a time.**

- **Change which certificate is the default certificate using the following command:**
  - **gsk8capicmd_64 -cert -setdefault -db server.kdb -stashed -label <certificate's label>**
    - **-db** specifies the name of the server's key database file
    - **-label** specifies the label of the certificate to make the default.

- **Because client key database files have only the server certificates in them and no private keys, none of the certificates in a client key database can be set as the default.**

Tips and Tricks - Verifying the Server Certificate via Browser

- If your GSKit key database file is being used to implement SSL on a web server, connect to the server with a web browser using an https://server:port syntax.

- A security warning may appear if the browser doesn't have the signing CA certificate or the self-signed certificate used by the server. But most browsers let you display information about the certificate currently being used by the server.

Verifying the Server Certificate Using OpenSSL

1. **Connect to the server and display the certificates using the -showcerts parameter of the OpenSSL as follows:**
   - **openssl s_client -connect server:port**
     - **-connect** specifies the server domain name and port that the server is listening on

2. **From the command's output, copy everything from "BEGIN CERTIFICATE" to "END CERTIFICATE," including those two lines. In the example below, assume the command output is copied to a file called server.cert**

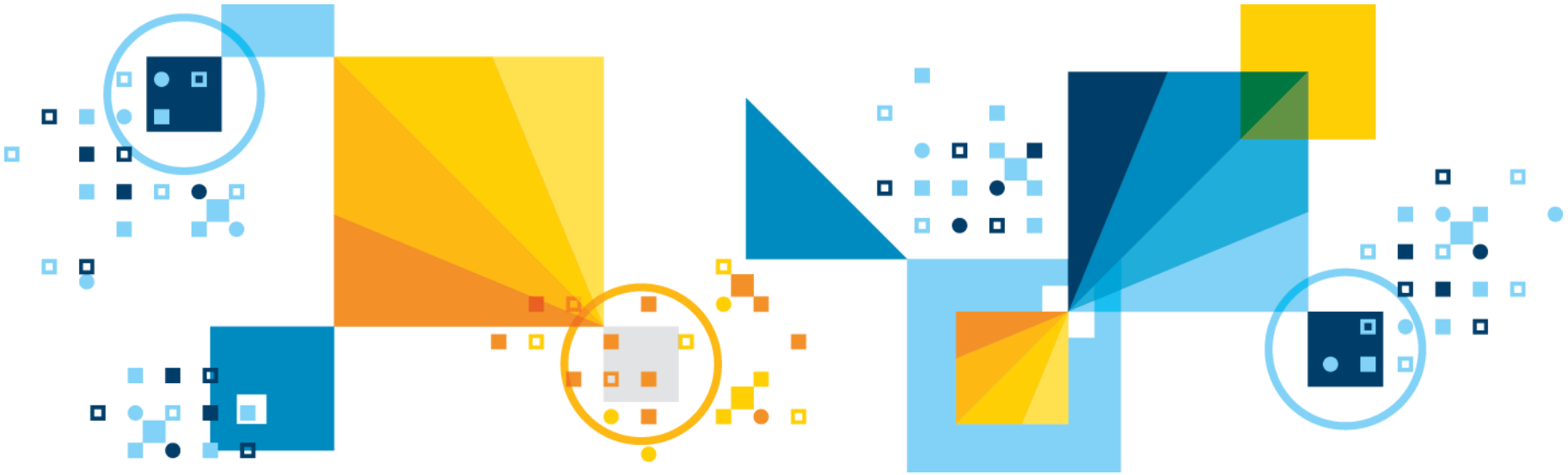
3. **Use OpenSSL to display the certificate as follows:**
   - **openssl x509 -in server.cert -noout -text**
     - **-in** specifies the name of the file that contains the output from the **-showcerts** command.

- **If you add the -showcerts option in step 1, you get the full certificate chain. You can repeat steps 2 and 3 for each certificate in the chain to analyze them.**

Scott Pickett – WW Informix Technical Sales

June 27, 2017

Appendix D - GDPR In a Little More Depth



Additional Data Elements Considered to be SPI Under Other Country Law (1)

- **In some countries, local law also provides that other data elements not listed above should be treated as SPI:**
  - If a particular country is not listed below, then there are no additional data elements that should be considered SPI for that country.

- **Australia –**
  - Membership in a professional or trade association (e.g.: International Association of Privacy Professionals (IAPP) or the Australian Corporate Lawyers Association)
- **Peru –**
  - Income
  - Private Life/Personal Status
  - Information relating to the intimate sphere. The 'intimate sphere' does not only include sexual activities, but includes all data that (i) an individual discloses only to a small circle of people, and (ii) have significant emotional importance.
- **Canada –**
  - Financial data including Salary / Income / Compensation

Additional Data Elements Considered to be SPI Under Other Country Law (2)

- **Denmark –**
  - Personality Test Data
  - Substantial social problems and other matters of a purely, private nature (e.g.: long term unemployment, accidents with significant personal or social consequences, suicides and attempted suicide, family disputes, separation and divorce application, issues relating to adoption, an employee's positive alcohol or drug test and expulsion from work/school.)
- **Finland -**
  - Social welfare needs of a person or the benefits, support or other social welfare assistance received by the person, social affiliation, illness or handicap or treatment or other comparable measures directed.
- **Greece**
  - Membership in societies pertaining to: race or national origin, political opinions, religious or philosophical beliefs, trade-union membership, health, social welfare and sex life, criminal proceedings or convictions
- **India**
  - Password

Additional Data Elements Considered to be SPI Under Other Country Law (3)

- **Israel**
  - Financial condition
  - Vocational qualifications/education information
  - Personality Test Data
  - Intimate Relations
  - Private Life/Personal Status

- **Philippines**
  - Marital status
  - **Vocational qualifications/education information**
  - **Tax return information**

- **Portugal -**
  - Private Life Personal Status Data

Additional Data Elements Considered to be SPI Under Other Country Law (4)

- **Slovakia –**
  - Data resulting out of personality tests or data concerning an individual's personality; such as kind of temperament, decision-making and communication style, attitudes towards matters such as work and recreation, taste, emotions, personality traits
- **Switzerland –**
  - Information relating to the intimate sphere. The 'intimate sphere' does not only include sexual activities, but includes all data that (i) an individual discloses only to a small circle of people, and (ii) have significant emotional importance.
  - Data resulting out of personality tests or data concerning an individual's personality; such as kind of temperament, decision-making and communication style, attitudes towards matters such as work and recreation, taste, emotions, personality traits

Personal Information Requirements for Notices-Content (1)

- **All countries which have enacted comprehensive privacy laws require organizations to be transparent and provide notices to individuals when collecting their PI.**

- **While some countries have local nuances, the following elements are common across these laws and must be included in a notice:**
  - The type of personal information that is collected - if the data being collected is not obvious, you should list the type of information you will be collecting by name.
  - Remember, only collect PI which is needed for your business purpose.
  - How it will be used - here, describe the purposes of collection and all the uses that will be made of the data, with an emphasis upon those that aren't obvious to the individual. This is often displayed in short information pop-ups.
- **Why and to whom the PI will be disclosed - be clear about who will have access to PI and why.**
- **Access to PI should be limited to those with a need-to-know to accomplish the purposes identified in the notice.**

Personal Information Requirements for Notices-Content (2)

- **Europe has particularly stringent requirements when it comes to the notice content. If you process PI from Europe, ensure that your notice includes:**
  - The name and contact details of the legal entity responsible for the data processing (so the name of the Data Controller);
  - The contact details of the Data Controller's Privacy Officer, where available or other contact;
  - All of the intended purposes for which PI will be processed. If the PI turns out to be eventually used for a purpose not originally included in the notice, individuals must be notified before their data is processed for this new purpose;
  - The legal basis for processing:
    - (a) If legitimate interest, a description of the legitimate interest must be given and the right to object to such must be highlighted
    - (b) Consent, provided the right to withdraw consent at any time is highlighted
    - (c) Contractual performance, legal obligation, vital interests of any natural person, or public interest
  - The recipients or categories of recipients of the PI;

Personal Information Requirements for Notices-Content (3)

- The rights of the individual to access their PI, have their PI rectified or erased, the right to restrict or object to the processing as well as the right to data portability, as applicable
- Whether the PI is subject to international transfers and the safeguards in place and a means to obtain a copy of them
- An explanation of how long PI will be retained
- The right to lodge a complaint with the supervisory authority
- If the data is not collected from the individual, the source of the data and, if applicable whether the data comes from a publicly accessible source
- As applicable, the existence of any automated decision-making including profiling, and meaningful information about the logic involved, along with the significance of the envisaged consequences of such processing on the individual.

Notice After Collection & Tracking

- **Where PI is obtained from someone other than the individual, a privacy notice must still be provided to the individual. It must be so:**
  - Within a reasonable period after obtaining the PI, but at the latest within one month, having regard to the specific circumstances in which the PI is processed; or
  - If the data is to be used for communication with the individual, at the latest at the time of the first communication to that individual;
  - If a disclosure to another recipient is envisaged, at the latest when the PI is first disclosed.

- **The privacy notice should also mention whether and how individuals will be notified of changes in the privacy notice (material changes must be brought to the attention of individuals by the most logical mean to reach them (e.g. e-mail; phone call; pop-up to website visitors).**
- **In addition, there must be a process to track the specific version of a privacy notice that was provided to an individual.**

Lawful Bases for Processing Personal information in the EU

- **You must have a valid lawful basis to process personal information. Under GDPR, there are six available legal bases for processing.**
  - Consent
  - Contractual necessity
  - Legal Obligation
  - Vital Interest
  - Public Interest
  - Legitimate Interest

- **Which legal basis is most appropriate will depend on your processing purpose and the relationship with the Data Subject.**

- **The applicable legal basis must be determined before you begin the processing of personal information.**

Consent (1)

- **The Data Subject gives consent to the processing of personal information for a specific purpose.**
- **For Consent to be valid it must be…**
  - Freely given
    - The Data Subject must not be under undue pressure to consent.
    - They must be able to refuse or withdraw consent at any time.
    - Consent is not freely given if it is a pre-condition of a service.
- **Informed**
  - The Data Subject must be provided with sufficient information to allow them to understand what they are consenting to.
  - Data Subjects should be made aware—using clear and plain language—of the purposes for which their personal information will be used, and with whom it will be shared.
- **Unambiguous**
  - Consent must be given by a statement or clear affirmative action, like checking a box to opt-in.
  - Silence, pre-checked boxes, inactivity, or failure to opt-out does not constitute valid consent.

Consent (2)

- **Specific**
  - Consent must be given for each purpose separately.
  - For processing activities for multiple purposes, consent should provide granular options.
  - Consent must be distinguishable from other T&Cs.
  - Sensitive personal information (or Special Category of Data (SCD under GDPR) requires explicit consent.
  - Consents given under current data protection laws may be further used under GDPR if they are in line with GDPR provisions.

Contractual necessity

- The processing is permitted if it is necessary for the performance of a contract with the Data Subject, or to take steps at the request of the Data Subject before entering into a contract.

- For example, entering into an employment agreement would entail processing personal information such as payroll information. Employees' consent is not needed in order to properly process such personal information.

Legal Obligation

- **The processing is necessary for the Data Controller to comply with a legal obligation (not including contractual obligations).**

- **The legal obligation must follow from EU law or member state law to which the Data Controller is subject.**

Vital interest

- **The processing is necessary to protect someone's life.**

Public Interest

- **The processing is necessary for the Data Controller to perform a task in the public interest, or for official functions. The task or function must have a clear basis in law.**

Legitimate Interest

- **The processing is necessary for the legitimate interest of the Data Controller or of a third party.**

- **Legitimate interest legal basis cannot be applied when such interests are overridden by the interests and fundamental rights and freedoms of the Data Subject, which require protection of their personal information.**

- **It is essential to identify the legitimate interest of the Data Controller or the third party and balance it with the interests of the Data Subject.**

EU Countries- Legal Basis for Processing Personal Information

- **Under EU law, processing of PI is only permitted if it can be supported by a "legal basis".**

- **A legal basis is a justification a company must have to collect, use, share or otherwise process PI.**
  - If no such compelling justification exists, PI may not be processed.

- **There are several legal basis available under EU law to justify processing, and depending on whether PI or certain types of SPI are processed, the legal basis available are different.**

Possible Legal Basis Available – Processing PI (1)

- Necessary to comply with a legal obligation to which a company is subject (e.g. for mandatory tax reporting to governments or for laws relating to safety at work)

- Necessary for the performance of a contract with the individual or to take preparatory steps to enter into such a contract (e.g. entering into an employment agreement would entail processing PI such as payroll information which would be supported by this legal basis; same for processing PI necessary to onboard a new employee).

- Necessary to protect the vital interest of the individual or another person. For this legal basis to be used, the life of the individual whose information is processed must be threatened. This may occur in situations where employees are travelling to extremely dangerous areas or with incidents on a company's premises. This legal basis should be used in a restrictive manner.

Possible Legal Basis Available – Processing PI (2)

- **Necessary to satisfy the legitimate interests of a company or a third party.**
  - This covers a broad range of data processing activities such as management activities, processing in relation to contractual matters of customers, and measures to protect a company's assets and interests, in connection with which a company has a legitimate interest.
  - This legal basis is more complex and will be covered in further detail below.

- **Performed after express consent has been obtained from the individual to whom the PI pertains. To be valid:**
  - Express consent needs to meet a specific form.
  - It must also be capable of being freely given (or denied) and withdrawn.
  - As a result, consent is not likely to be the preferred legal basis used in support of an organization's data processing activities.
  - Rather, other legal basis such as "legitimate interest" or the "performance of a contract" are likely to be leveraged. See detailed section below.

Legitimate Interest as Legal Basis

- **Legitimate interest can be relied upon as a legal basis when the "interest" being claimed is not overridden by the interests or fundamental rights of the Individual:**
  - In other words, one needs to look at the interest of the company (e.g. the need to protect its network and data) versus the interest and fundamental rights of individuals (e.g. the right not to be tracked online or be monitored) and determine whether the latter overrides the former.

- **This "balancing of interests" must be formally documented as follows, and the related documentation may need to be provided when requested by individuals, other stakeholders, or Data Protection Authorities**

Legitimate Interest as Legal Basis – Criteria (1)

- **Legitimate interest must be made explicit and must be sufficiently articulated to be able to balance it against the interests of the individuals:**
  - The interest also needs to be legitimate, which means that the way it is pursued must meet legal requirements, which includes all other aspects of data privacy laws (e.g. proportionality, minimization, transparency, accuracy);

- **Intended processing must be necessary in relation to that interest**

- **Interests at stake (both risks and consequences) for individuals must also be documented.**
  - This does not only include the intended consequences for the individuals, but also risks (e.g. loss of data, discrimination, reputational damage, unexpected or inaccurate predictions regarding behavior, career and financial damage, irritation, harassment)

Legitimate Interest as Legal Basis – Criteria (2)

- Reasonable expectations of the individuals affected and the level of transparency around the initiative must also be taken into account and documented.

- If the balancing of interest exercise doesn't result in a situation where one party's interest clearly prevails upon another's or in a situation where, the interest of the individuals weighs heavier, mitigation measures must be taken, such as functional separation, anonymization, aggregation or pseudonymisation, transparency, additional individual ability to exercise some level of control over data processing and opt-out.

- Activities such as direct marketing (however, with an opt-out), fraud prevention, transmission of PI within the group for administrative purposes, network and IT security, and reporting of criminal activity or threats can be recognized as legitimate interests.

Legitimate Interest as Legal Basis – Criteria (3)

- Important to remember that transparency is key, and that when using legitimate interest as a basis to process PI, the notice presented to individuals describing a company's processing activities must include a description of the legitimate interest being pursued (e.g. a company processes your data in Claims in order to be able to invoice our customers for work done by you and to maintain auditable records).

- Keep in mind that individuals have the right to object to processing of their data based on legitimate interest on grounds related to their specific situation. In such a case, a company would be required to demonstrate that it has compelling interests for the processing to continue.

Legitimate Interest as Legal Basis – Processing of SPI (1)

- **Specific rules apply to the following categories of SPI in the EU for data revealing:**
  - Racial or ethnic origin,
  - Political opinions,
  - Religious or philosophical beliefs,
  - Trade union membership;
  - The processing of genetic data,
  - Biometric data for the purpose of uniquely identifying a natural person;
  - Data concerning personal health
  - Data concerning a person's sex life or sexual orientation.

- **Processing of such SPI is only allowed with the express consent of the individual, with the exception of some narrowly described situations in which SPI can be processed.**
- **Local EU laws may also set further requirements, or completely even forbid the processing of these SPI data elements.**

Legitimate Interest as Legal Basis – Processing of SPI (2)

- **The below narrowly described exceptions which do not require express consent to be obtained for the above data elements are (to the extent relevant for a company):**
  - The processing is necessary in relation to one's employment, social security and protection, as far as authorized by local law
  - The processing is necessary for legal claims
  - The processing is necessary for preventive or occupational medicine, working capacity of employees, on the basis of local law or pursuant to a health professional and the processing is for the purposes mentioned in the law and under the responsibility of a professional under the obligation of professional secrecy

Consent (1)

- **EU law has strict requirements for consent. If these are not met, the consent will not be deemed to be valid. These requirements are as follows:**
  - Consent must be obtained prior to the PI's collection, use or disclosure.
  - Consent for the processing of general PI, defined:
    - This is PI that is NOT data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership; the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person; and data concerning health or a person's sex life or sexual orientation.
    - Requires the individual's affirmative and unambiguous action.
      - This can be accomplished by ticking a box when visiting an Internet website, choosing technical settings or another statement or conduct which clearly indicates in the specific context the individual's active acceptance of the proposed processing of its PI (also known as "**opt-in**").
    - Silence, pre-ticked boxes or inactivity (known as "opt-out") do not constitute valid consent under EU law.
    - References to General Terms and Conditions with a remark that "by using the application or tool" the user agrees to these Terms and Conditions is also not sufficient.
    - The affirmative action rather has to be clearly connected to the consent for the specific processing of PI.

Consent (2)

- **These requirements are as follows (con'td):**
- **Consent must be given in these cases:**
  - Consent for the processing of data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership
  - Processing of genetic data, biometric data for the purpose of uniquely identifying a natural person
  - Data concerning health or a person's sex life or sexual orientation) has to be given expressly/explicitly and for a specific purposes (e.g. through an explicit "I agree" or "I consent" provided for a specific and single purpose).
  - Consent such as a conduct indicating individual's acceptance is not valid for this type of data.

- **Additionally, <u>each specific kind of SPI</u> has to be explicitly set out in the request for consent (e.g. religious opinion, trade union membership, racial origin) as well as the specific processing purpose(s) <u>for each</u>.**

Consent (3)

- **Some other considerations for consent to be valid:**
  - Where consent is requested in conjunction with other matters, the request for consent must be clearly distinguishable in its appearance from these other matters.
  - Language used to request consent from individuals must be clear and plain and easily accessible.
  - For consent to be valid, it must be provided freely by the individual.
  - Individual needs to be able to refuse or withdraw consent without having disadvantages.
  - Additionally, the performance of a contract should not be made conditional on an individual's consent to the processing of PI that is not absolutely required for performing the contract.

Consent (3)

- **Consent can't be used as a legal basis where there is a clear power imbalance between an individual and the Controller (e.g. employment relationship, children, students)**
  - In employment relationship, consent is generally not accepted as a legal basis and should be used only for situations where there is a real choice, or an additional benefit to the employee, which holds no consequences for them:
    - For example: participation in a BYOD program and processing of PI in conjunction with such a program can be performed using consent as a legal basis to the extent that the program is truly optional and that alternatives are provided to employees.

- **Consent must be informed.**
  - Therefore, an appropriate notice must be provided, and a mention that individuals can withdraw their consent included.

- **Individuals have the right to withdraw consent at any time, and it must be as easy to withdraw consent as it is to give it:**
  - Where consent is withdrawn, PI must be erased without undue delay unless there is another legal basis for processing.

Consent (4)

- **Consent, as well as the withdrawal of consent, needs to be documented and preserved for audit purposes.**

- **There must be a process in place to document the consent (consent language, identification mark if individual and timestamp).**
  - Note that some countries may have additional documentation requirements.

Further Processing

- **Where PI is to be processed for a new purpose, one must consider whether this new purpose is compatible with the original purpose for which the PI was obtained, taking into account the following factors:**
  - Any link between the original purpose and the new purpose
  - The context in which the data has been collected (and the Individuals notified) and the intended further processing (e.g. if the original processing was based on consent for a specific purpose)
  - The nature of the PI, in particular whether we're dealing with SPI that in Europe generally requires express consent (see "Processing of SPI" above)
  - The consequences of the further processing on the individual
  - The existence of appropriate safeguards (e.g. encryption or pseudonymisation)

- **Where further processing is intended, a proper notice must be given to the individuals prior to the further processing taking place.**
- **Further processing for incompatible purposes may only be conducted with consent.**

Profiling and Automated Decision-Making (1)

- **Profiling is defined in EU law as "any form of automated processing of personal data consisting of using those data to evaluate certain personal aspects relating to a natural person, in particular to analyze or predict certain aspects concerning that natural person's performance at work, economic situations, health, personal preferences, interests, reliability, behavior, location or movement"**

Profiling and Automated Decision-Making (2)

- **Profiling is not prohibited under such law but come accompanied with additional requirements beyond that which apply to all other types of data processing activities, specifically:**
  - Use appropriate mathematical or statistical procedures for profiling and ensure that such is documented
  - Implement technical and organizational measures appropriate to ensure that the risk of errors and inaccuracies is minimized and that, where they occur, they can be corrected
  - Secure PI in a manner that takes account of the potential risks involved for the interests and rights of the individual and that prevents discriminatory effects on individuals
  - Be transparent.

Profiling and Automated Decision-Making (3)

- **Individuals have in many instances the right to object to profiling based on their specific situation (see above under Legal Basis – Legitimate Interest).**
  - For example, they may object to profiling for direct marketing purposes (e.g. advertising to groups assumed to be interested in given offerings pertaining to credit, travel or food).

Profiling and Automated Decision-Making (4)

- **Automated Decision-Making provisions under EU law give individuals the right not to be evaluated in any material sense (e.g. in connection with offers of employment or other employment related decisions; insurance premiums) based solely on automated processing (including profiling) where such evaluations produce legal effects or significantly affect the individual. There exists exceptions to this rule:**
  - Where the processing is necessary for entering into a contract between the individual and the Controller or for the performance of such (e.g. credit checks; recruitment applications)
  - Where the individual has explicitly consented (see Consent) and appropriate safeguards are in place. In relationships with unequal power (such as employment), consent may not be valid
  - Where processing is authorized by law (these will be narrow provisions and will rarely apply).
  - If one of these exceptions apply, there have to be processes and controls in place for safeguarding the individual's rights: the right to obtain human intervention, to obtain an explanation on the decision reached, to express their point of view and to contest the decision.

Profiling and Automated Decision-Making (5)

- **Furthermore, the individual has to be informed by way of appropriate notice about the existence of automated decisions-making, the logic involved and the significance of such processing for them**

Profiling and Automated Decision-Making (6)

- **To sum up:**
  - Ensure there is a documented legitimate ground to justify the profiling (also consider whether secondary use of existing data is allowed, and notify individuals of such use)
  - Document what the effects are for the individuals:
    - If the effects are negligible, the specific restrictions will probably not apply.
  - Implement internal processes to handle objections (this should be documented as evidence)
    - This may include the right to obtain an explanation, to express the individual's point of view and to contest a result based on profiling, if that result may have significant impact on the individual.
  - Ensure complete and clear information notices to adequately inform individuals about the profiling and related rights to object
  - Use appropriate mathematical or statistical procedures and minimize the risk of inaccuracies (both in the data sources and in the results).
  - Avoid decisions solely automated by ensuring a reasonable level of human intervention, or demonstrate that the automated decision-making is necessary for entering into or performance of a contract.

Profiling and Automated Decision-Making (7)

- **To sum up (cont'd):**
  - Automated decision-making should not include certain types of SPI (as described above under Legal Basis), **nor information pertaining to children.**
  - Engaging in automated decision-making or profiling with possibly material effects for individuals will often require a Data Protection Impact Assessment (DPIA).

GDPR – Seven Rights of Data Subjects (You and Me)

- Access
- Rectification
- Erasure
- Objection
- Restriction of Processing
- Portability
- Not to be subject to a decision based solely on automated processing

Access

- **Under the GDPR, Data Subjects have the right to obtain confirmation whether a Data Controller is or is not processing their personal information (PI), and to be provided with the following information without undue delay and, in any event, within one month of receipt of the request:**
  - Purposes of the processing
  - Categories of PI being processed
  - With whom their PI is shared, including the appropriate safeguards relating to the transfers of data (if any)
  - Data retention period
  - Data sources where PI was not collected directly from the Data Subject
  - Right to request the rectification, or erasure of their PI, or restriction of the processing, or to object to such processing
  - Right to lodge a complaint with the supervisory authority
  - Existence of automated decision-making, including profiling, and meaningful information about the logic involved

Rectification

- **Data Subjects have the right to correct inaccurate personal information.**

Erasure

- Data Subjects have the right to request the deletion or removal of their personal information where there is no compelling reason for its continued processing

- This right implies the secure deletion of personal information in a way that it cannot be restored

- Third-party recipients to whom personal information has been disclosed must be notified about the erasure request, unless it proves impossible or involves disproportionate effort

- In some specific circumstances, the right to erasure does not apply and Data Controllers can therefore refuse to honor the request

Objection

- **Data Subjects have the right to object to:**
  - The processing based on legitimate interest or the performance of a task in the public interest or in the exercise of official authority, including profiling
  - Direct marketing, including profiling
  - The processing for purposes of scientific or historical research and statistics

- **When a Data Subject exercises this right, processing of personal information becomes unlawful and concerned personal information must be deleted, unless the Data Controller demonstrates a compelling legitimate ground for the processing, which overrides Data Subject's interest**

- **Information of the right to object must be provided at the point of "first communication" with the Data Subjects, and within the Privacy Notice. It should be presented clearly and separately from any other information.**

Restriction of Processing

- Data Subjects have the right to block or suppress processing of their personal information in certain circumstances.
- When processing of personal information is restricted, Data Controllers are permitted to store personal information, but not further process it in the future.
- Third-party recipients to whom personal information has been disclosed must be notified about the restriction of processing, unless it proves impossible or involves disproportionate effort.
- Data Subjects must be informed when a Data Controller decides to lift a restriction on processing.

Portability

- **Portability**

- **Data Subjects have the right to receive a copy of their personal information in a commonly used machine-readable format, and to transfer their personal information from one Data Controller to another in a safe and secure way without hindrance to usability.**

- **The right to data portability only applies:**
  - To personal information a Data Subject has provided to a Data Controller
  - Where the processing is based on Data Subject's consent or on the performance of a contract
  - When processing is carried out by automated means

Not to be subject to a decision based solely on automated processing

- **Data Subjects have the right not to be subject to a decision based solely on automated processing, including profiling, which produces a negative legal effect concerning them or similarly significantly affect them.**

- **Data Controllers can only carry out solely automated decision-making with legal or similarly significant effects if the decision is:**
  - Necessary for entering into or performance of a contract between the Data Controller and the Data Subject
  - Authorized by law, or based on Data Subject's explicit consent
- **And:**
  - Specific information about the processing is provided to Data Subjects, including meaningful information about the logic involved and the significance and envisaged consequences for Data Subjects
  - Data Subjects are given the right to challenge and request a review of the automated decision

Data Controllers Obligations (1)

- **Compliance and accountability**
  - Data Controllers are primarily responsible under GDPR for ensuring compliance with the Key Principles.
  - Data Controllers are expected to put into place comprehensive and proportionate governance measures.
- **Data Processors**
  - When appointing a Data Processor, Data Controllers must ensure a written data processing agreement is in place.
  - Data processing agreements must include certain specific terms, as a minimum, to ensure that processing carried out by a particular Data Processor meets the appropriate requirements of the GDPR.
- **Data Subject's rights**
  - Data Controllers must ensure and facilitate the exercise of Data Subject's rights under GDPR.
  - They are also obligated to respond to requests from Data Subjects without undue delay and within one month as from the request.

Data Controllers Obligations (2)

- **Notifications**
  - Data Controllers are required to notify the Data Protection Authority of data breaches within 72 hours, and in some cases, the Data Subject.
  - Data Controllers must also keep a record of any personal information breaches, regardless of whether they are required to notify or not.

- **Privacy by Design and by Default**
  - Under GDPR, Data Controllers have a general obligation to implement technical and organizational measures to show that personal information protection has been considered and integrated into their processing activities.
  - Privacy should not be an afterthought—it should be "baked in" early in the Data Controller's planning and implementation. By default, Data Controllers must ensure only the minimum personal information required for a particular processing is processed.

Data Controllers Obligations (3)

- **Data Impact Privacy Assessment**
  - Data Controllers are required to conduct a Data Privacy Impact Assessment (DPIA) when a new processing activity is likely to result in a high degree of risk for Data Subjects.
  - A DPIA enables Data Controllers to identify and mitigate risks that they might not have otherwise known about.
  - Company's have implemented a process to evaluate compliance of each initiative/application/business process and mitigate risks.

Data Processor's Obligations

- **Follow Data Controller's Instructions**
  - Data Processors are required to process personal information in accordance with the Data Controller's written instructions.
  - They are not allowed to use the personal information in ways that the Data Controller did not request.
- **Legal Obligation to Notify the Data Controller**
  - Data Processors are required to immediately notify Data Controllers if they believe the Data Controller's instructions are in conflict with the requirements of the GDPR, or other EU member state laws.
  - Data Processors are also required to notify Data Controllers of any data breach without undue delay.
- **Seek Approval for Sub-Processors**
  - Data Processors must not appoint a sub-processor (i.e. a subcontractor) without the prior written consent of the Data Controller.
  - When specifically authorized, sub-processors must be subject to the same terms as are set out in the agreement in place between the Data Controller and the Data Processor.

Data Controllers and Data Processors – Duties, Obligations

- **GDPR poses certain duties and obligations for both Data Controllers and Data Processors.**

- **Cooperate with Data Protection Authorities**
  - Upon request, both Data Controllers and Data Processors are required to cooperate with the Data Protection Authorities in the performance of their tasks.

- **Appoint a Data Protection Officer**
  - In certain circumstances, the GDPR requires Data Controllers and Data Processors to appoint a Data Protection Officer (DPO).

- **Implement Security Measures**
  - Data Controllers and Data Processors must implement appropriate technical, procedural, and organizational measures to protect personal information against accidental or unlawful destruction or loss, alteration, unauthorized disclosure, or access.

Data Controllers and Data Processors – Duties, Obligations

- **Maintain Records**
  - The GDPR contains strict requirements for documentation, a duty shared by both Data Controllers and Data Processors. Upon request, these records must be disclosed to the Data Protection Authority.

- **Processing records must include:**
  - Name and contact details of the Data Controller, DPO, and any representatives
  - Purposes of data processing activities
  - Categories of Data Subjects
  - Categories of personal information processed
  - Categories of recipients with whom the personal information may be shared
  - Information regarding cross-border data transfers
  - Data retention dates
  - Technical, procedural, and organizational measures implemented

- **Records must be accurate and complete.**

Data Transfer to Countries Outside of the EU

- **The personal information of EU citizens may only be transferred (without additional safeguards) to countries where the EU considers data protection legislation to be adequate.**
- **The personal information of EU citizens may be transferred to countries outside Europe with adequate protection for personal information:**
  - These are limited, but include Canada, New Zealand, Israel, and Argentina.
- **The personal information of EU citizens may be transferred to non-EU countries that are part of the European Economic Area (EEA):**
  - This includes Iceland, Liechtenstein, and Norway.
- **The personal information of EU citizens may be transferred to Switzerland.**
  - Although it is neither an EU nor an EEA member, it has adequate data protection legislation.
- **For Spain, the personal information of EU citizens may be transferred to other EU member states.**

Allowed EU Personal Information Transfer Mechanisms (1)

- **Several mechanisms can be used to enable transfers of personal information to areas and countries without adequate data protection.**

- **Model Clauses**
  - The most commonly used mechanism for data transfer outside of the EU are contract clauses that have been approved and issued by the European Commission, known as *model clauses*.
  - Model clauses have to be completed on a company by company basis when data is to be transferred from Data Controller within the EU to Data Controller outside the EU, or from Data Controller within the EU to Data Processor outside the EU.
  - By agreeing to the model clauses, both parties agree that they will comply with data protection standards that meet the requirements of the EU Data Protection Directive.
  - Data Controller must provide information about the personal information and sensitive personal information involved, the sorts of individuals, and the purposes for which it will be used.
  - If the contract is between a Data Controller and Data Processor, a summary of the Data Processor's data security arrangement is also required.

Allowed EU Personal Information Transfer Mechanisms (2)

- **Binding Corporate Rules**
  - Binding Corporate Rules is a cross-border transfer mechanism to allow multinational companies to transfer personal information from the European Economic Area (EEA) to their affiliates located outside of the EEA in compliance with the European Data Protection Law.

Privacy Shield Frameworks

- **Privacy Shield frameworks were designed by the U.S. Department of Commerce, European Commission, and Swiss Administration to provide companies with a mechanism to comply with data protection requirements when transferring personal information from the EU and Switzerland to the U.S.**
- **Privacy Shields are ONLY for data transfers to U.S.**
- **There are two types of frameworks:**
  - EU–U.S. and Swiss–U.S.
- **To join either framework—**
  - A U.S. company self-certifies to the U.S. Department of Commerce and publicly commits to comply with the framework's requirements.

Binding Corporate Rules

- **Binding Corporate Rules is a cross-border transfer mechanism to allow multinational companies to transfer personal information from the European Economic Area (EEA) to their affiliates located outside of the EEA in compliance with the European Data Protection Law.**

- **A company's Binding Corporate Rules – Why and What?**
  - To help a company share personal information between a company all around the world, a company has obtained a “Binding Corporate Rules” (“BCR”) certification from the European Data Protection Authorities.
    - The BCRs are a set of internal company policies that set out how personal information should be handled by all company employees around the globe.
  - By having the BCRs in place, company’s personal information will be adequately protected wherever in the world it is accessed or handled by a company.
    - In order to ensure this protection, all company employees must comply with the requirements of our BCRs.
  - These BCRs apply to the company’s own internal personal information, e.g., which we use to employ people, to market and sell our products and services, to fulfill orders and bill customers, etc.

Binding Corporate Rules

- **The company's BCRs are divided into two categories:**
  - Company Employee Information (personal information about the company employees)
  - Company Business Personal Information (personal information the company holds on existing or potential customers, Business Partners, suppliers, etc.)

Reporting Requirements

- When there has been a data breach, the Data Controller must report it to the supervisory authority without undue delay and not later than 72 hours after becoming aware of the breach.

- If a data breach is likely to result in a high risk to the rights and freedoms of the Data Subjects, the Data Controller must also inform the Data Subjects without undue delay.

Legal Disclaimer