Customer

Securing ASE, IQ & SRS for Compliance

Jeff Tallman jeff.tallman@sap.com
SAP ASE Product Management

UKSUG
SAP DATABASE & TECHNOLOGY USER GR

isug tech

Agenda

- User Authentication
- Network Security
- Granular Permissions
- Data Encryption
- DBA’s & Predicated Privileges
- Auditing
- Hot Fixes & Vulnerabilities

Disclaimer

- This presentation outlines our general product direction and should not be relied on in making a purchase decision. This presentation is not subject to your license agreement or any other agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or to develop or release any functionality mentioned in this presentation. This presentation and SAP's strategy and possible future developments are subject to change and may be changed by SAP at any time for any reason without notice. This document is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP assumes no responsibility for errors or omissions in this document, except if such damages were caused by SAP intentionally or grossly negligent.

Getting Started….Today's Goal

A lot of security topics are basic common sense….

- Don't use default (supplied) passwords, null passwords, etc.
- Documenting your security technical implementation
- …we assume you already know this, so many things like this will *not* be mentioned.

Some are easily over looked…..but still simple (in theory)

- E.g. disable functionality not being used…like XP server
- ….we aren't going to talk about this either

We are not going to walk through a detailed security checklist….

- Because no single checklist could likely cover all the various regulations and standards - despite a lot of overlap

The goal is to dig a bit deeper into core security features provided by ASE/SRS/IQ that relate to common regulatory requirements

- E.g. how to set up password complexity rules, auditing, encryption and best practices for each
- Everything in this ppt have been FAQ's from customers I have been asked in the past 18 months
  - ✓ ….and becoming extremely familiar with several more regulations than I ever wished to

Getting started….. Know the rules

There are a plethora of them….

- HIPAA, ESMA MIFID/MIFIR, PCI DSS, DOD DISA STIG, ….
- So many we can’t tell you how to comply with all of them…but there is a lot of commonality

Try to understand what is to be in the database and what is from the app/system level

- E.g. standard may dictate individual logins - that doesn’t necessarily translate to individual DBMS logins

Understand what data sensitivities exist and which need encryption…vs. just sensitive in context

- Not all do - HIPAA has restrictions (for example) on exposing any locations of zip codes of health issues for zip codes containing less than 20000 people.
  - ✓ Doesn’t mean zip code is to be encrypted
    - just means if extracting the data for public consumption, some form of data obfuscation may be necessary.
    - In reality, this likely means obfuscating/aggregating for all….unless you keep zip code census information in the database
  - ✓ Patient name may or may not be sensitive. Otherwise hospital visitation would be fun. On the flip side, psych patient names may be sensitive
  - ✓ The patient’s diagnosis/medical condition is sensitive …..but we don’t encrypt that you broke your leg.
  - ✓ The trick is to restrict access to information - not simply encrypt everything (less storage).
- However, also realize that some things simply ought to be encrypted
  - ✓ SSN’s, credit card numbers, passwords/security question answers, data on disk, etc.
- And some things are not allowed to be stored at all - e.g. credit card security codes.

Which Systems & Which Regulations

- **PCI DSS**
  - Do you accept/process credit cards???

- **HIPAA**
  - Do you store health care related information???
  - Believe it or not, most major corporations in US need to comply with this due to self insurance, etc. (HR systems)

- **<insert country here> Data Privacy Act**
  - Applies to just about any system that stores personally identifiable information or citizen identity numbers
  - Lacking any other regulation, this is the one that will likely drive data encryption requirements

- **Etc. etc. etc.**

- **The point??**
  - Likely more than one applies to your system(s)
  - Get a copy and read the applicable sections at a minimum

U.S. Department of Health and Human Services
Office for Civil Rights

HIPAA Administrative Simplification

UNCLASSIFIED

DATABASE
SECURITY REQUIREMENTS GUIDE (SRG)
TECHNOLOGY OVERVIEW

Version 2, Release 3

22 January 2016

Developed by DISA for the DoD

NIST Special Publication 800-53
Revision 4

Security and Privacy Controls for
Federal Information Systems
and Organizations

JOINT TASK FORCE
TRANSFORMATION INITIATIVE

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-53r4

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

Payment Card Industry (PCI)
Data Security Standard

Requirements and Security Assessment Procedures
Version 3.1
April 2015

Example of vagueness/flexibility…HIPAA on Data Encryption….

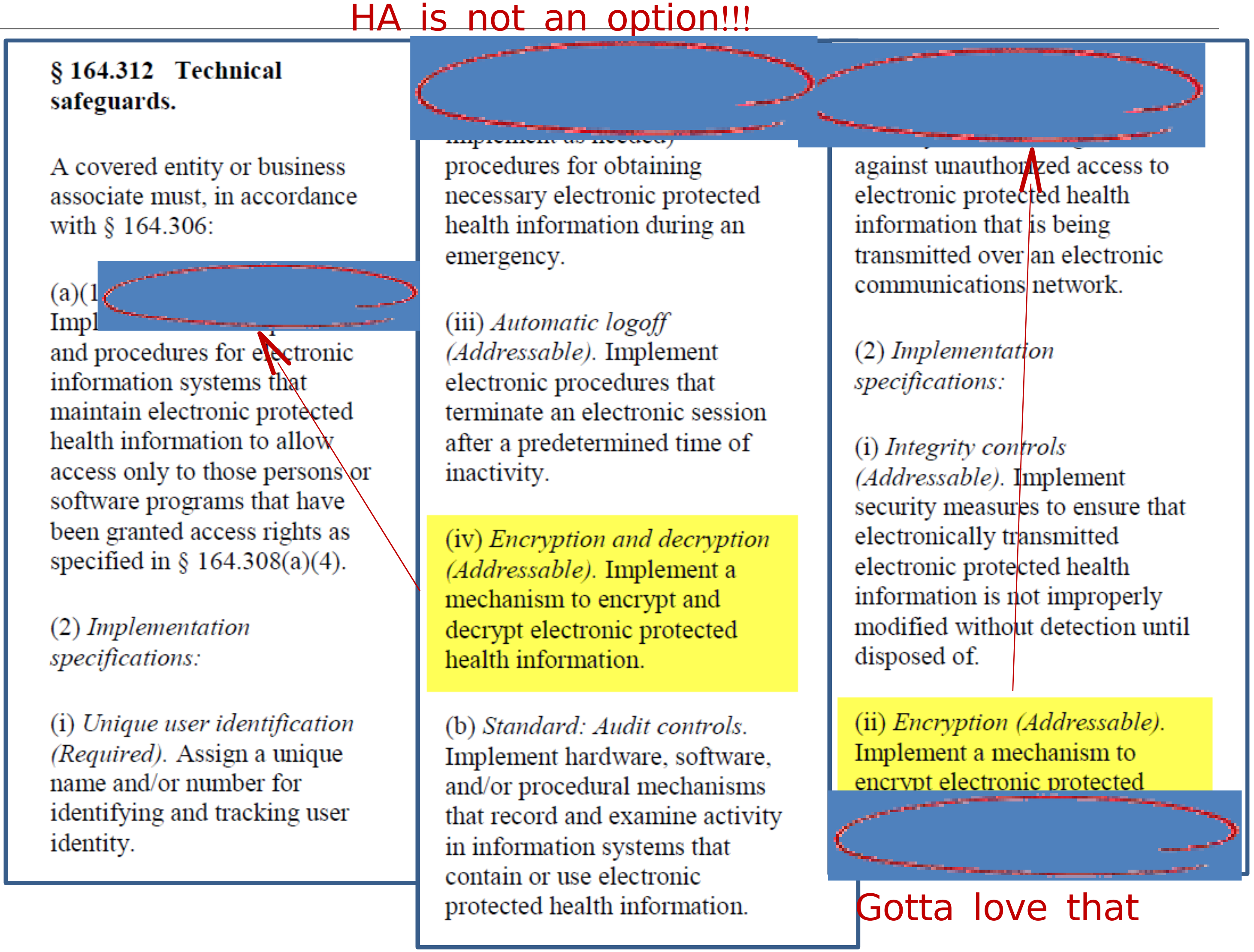
Key is 164.306

- Which says that unauthorized disclosure needs to be prevented
- States that the size of the organization, costs, technical capabilities, etc. are to be taken into consideration
  - ✓ Hence “*Addressable*” vs. “*Required*” with respect to encryption

Reading 164.312

- Data encryption at rest seems to be the requirement
  - ✓ Full database encryption
  - ✓ Encrypted file system would also work
  - ✓ Encryption as *access control* also supports using column encryption on data such as SSN, etc.
- Network encryption is recommended
  - ✓ “…deemed appropriate” could suggest that internal file transfers, ETL, etc. may not need network encryption whereas sending an email to patient might need it
- HA is not an option….2 aspects
  - ✓ Providing care during disasters
  - ✓ Sudden surge in staff during epidemic

HA is not an option!!!



§ 164.312 Technical safeguards.

A covered entity or business associate must, in accordance with § 164.306:

(a)(1 [redacted] Impl and procedures for electronic information systems that maintain electronic protected health information to allow access only to those persons or software programs that have been granted access rights as specified in § 164.308(a)(4).

(2) *Implementation specifications:*

(i) *Unique user identification (Required).* Assign a unique name and/or number for identifying and tracking user identity.

[redacted] procedures for obtaining necessary electronic protected health information during an emergency.

(iii) *Automatic logoff (Addressable).* Implement electronic procedures that terminate an electronic session after a predetermined time of inactivity.

(iv) *Encryption and decryption (Addressable).* Implement a mechanism to encrypt and decrypt electronic protected health information.

(b) *Standard: Audit controls.* Implement hardware, software, and/or procedural mechanisms that record and examine activity in information systems that contain or use electronic protected health information.

[redacted] against unauthorized access to electronic protected health information that is being transmitted over an electronic communications network.

(2) *Implementation specifications:*

(i) *Integrity controls (Addressable).* Implement security measures to ensure that electronically transmitted electronic protected health information is not improperly modified without detection until disposed of.

(ii) *Encryption (Addressable).* Implement a mechanism to encrypt electronic protected [redacted]

Gotta love that

Understand the tradeoffs

·**If application uses a single DBMS login…..**

- Realize that column encryption or even restricting access within the DBMS is useless
- This is true whether the app uses a single login (ala SAP Business Suite) or whether connection pool without proxy is used

·**Nothing is free….it all comes down to $$$ecurity**

- The obvious $$$ is licensing ASE directory services, encryption & IQ advanced security options
- The co$t of extra development & testing efforts for security implementation in application development
- The price of faster disks/networks to offset data/network encryption = $$
- Separation of duties = more people ($$$$$) + more time (coordinating schedules) ($$$$)

·**…but the cost of a single incident is much much much higher**

- The price of lost sensitive data = $$$
- …and not just to the primary victim (e.g. Target, Home Depot, etc.)
  - ✓Every bank had to reissue all their credit cards (although some sued Target to recoup the costs)
  - ✓….think of all the chip card readers that had to be purchased across all the retailers in USA.
- ….or the embarrassment of having to pay ransom for your own data.
  - ✓Yes, this happened to one hospital…just this past year

From DBA’s Perspective

Compliance Regulation covers many facets

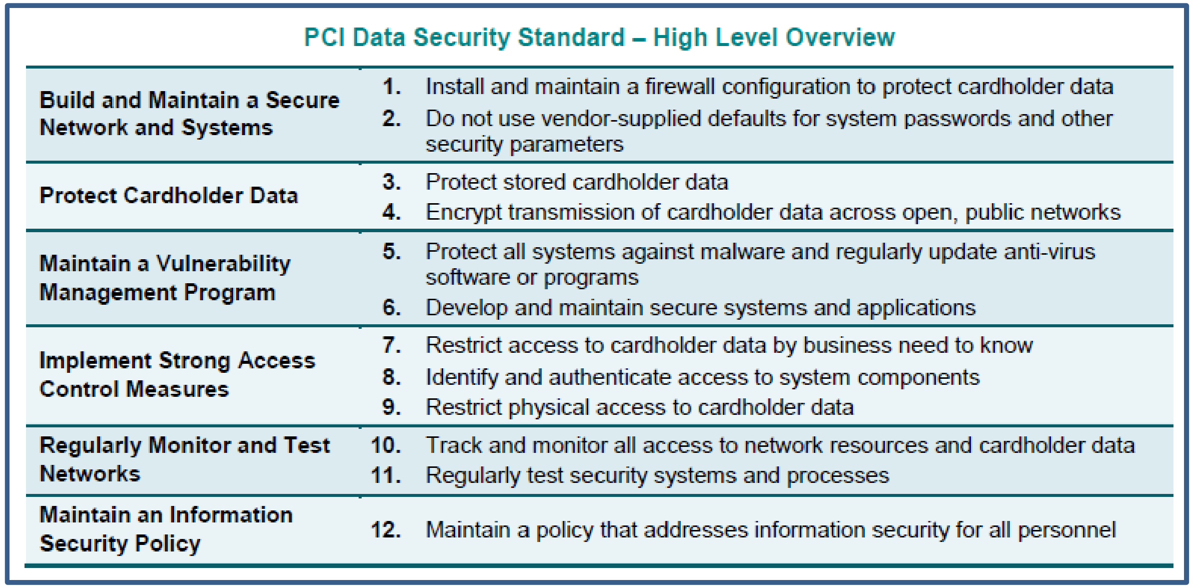
- Virus protection, high availability, etc.
- Many are vague/high level and deal with the overall system vs. DBMS/data security

Likely is a corporate compliance office

- They are the ones on the hook for ensuring compliance
- They likely dictate internal corporate standards and policies to be used to measure compliance with external regulations/standards
  - ✓ Determine the site specifics for complying with the regulatory vagueness

Ought to be a corporate security board

- DBA needs a seat at the table
- Otherwise it is likely that implementations chosen will not be best suited for data processing/retention in DBMS

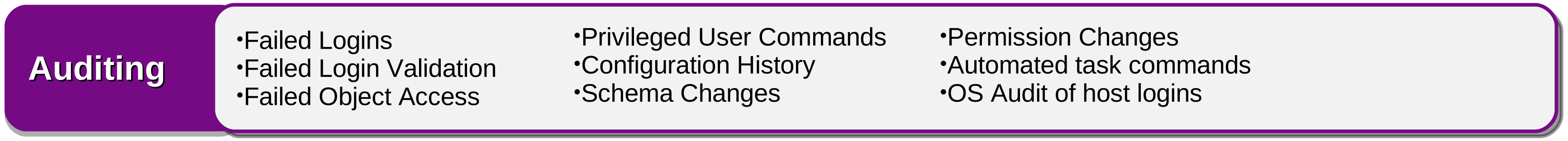


PCI Data Security Standard – High Level Overview

Build and Maintain a Secure Network and Systems	1. Install and maintain a firewall configuration to protect cardholder data 2. Do not use vendor-supplied defaults for system passwords and other security parameters
Protect Cardholder Data	3. Protect stored cardholder data 4. Encrypt transmission of cardholder data across open, public networks
Maintain a Vulnerability Management Program	5. Protect all systems against malware and regularly update anti-virus software or programs 6. Develop and maintain secure systems and applications
Implement Strong Access Control Measures	7. Restrict access to cardholder data by business need to know 8. Identify and authenticate access to system components 9. Restrict physical access to cardholder data
Regularly Monitor and Test Networks	10. Track and monitor all access to network resources and cardholder data 11. Regularly test security systems and processes
Maintain an Information Security Policy	12. Maintain a policy that addresses information security for all personnel

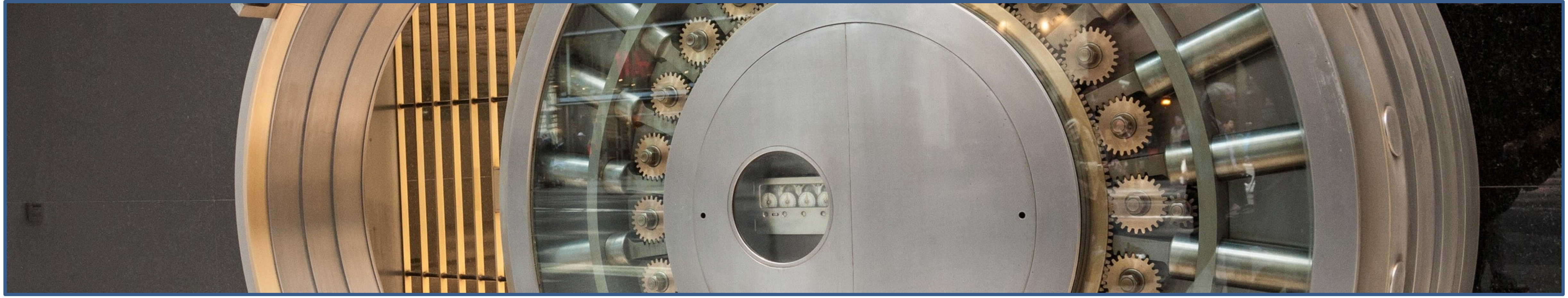
Source: Payment Card Industry (PCI) Data Security Standard Requirements and Security Assessment Procedures v3.1

Security Model Layers for Compliance



Auditing

- •Failed Logins
- •Failed Login Validation
- •Failed Object Access
- •Privileged User Commands
- •Configuration History
- •Schema Changes
- •Permission Changes
- •Automated task commands
- •OS Audit of host logins



User Authentication

- Logins, LDAP Authentication, Two Factor Authentication (2FA)

Regulations, Standards & User Authentication

- **All require individual user authentication**
  - However, industry regulations are often written to the system level which allows for individual logins to the application while using a common login to the DBMS
  - Usually have some rules around preventing unauthorized users from assuming system privileges
  - Technology Security Standards will often cite individual DBMS logins
  - Some regulations have verbiage that suggests that some method of verifying the actual user vs. “stolen password”….which ties to Two Factor Authentication
    - ✓ The most common hacks are due to lost/stolen passwords
    - ✓ First indication might be a user connecting from a different machine/program than allowed app server

- **All have some rules around password complexity and management**
  - Generally suggest centralized account & password management
    - ✓ The problem is ensuring that accounts (such as automated utility accounts & DBA) that are not centrally managed comply with the password complexity requirements
  - Most require automatic locking of the account after so many failed logins

- **Some corporate security standards go a bit further**
  - Dictate that privileged users can only connect from specific hosts

Best Practices for Security Compliance - User Authentication

- **Avoid application logins**
  - If you use a single common login for all users, you will have to implement all security in the application layer
  - In addition, it voids the potential for using encrypted columns, auditing and other features as a single login bypasses any security that depends on ability to distinguish end-user identity.

- **If using a connection pool (assumes an initial application login)….**
  - Use different connection pools for each application
  - Use an app server login that has ‘set session authorization’ (aka proxy authorization)
  - However, do the following
    - ✓Make sure that the appserver login is only used by the app server (use a login trigger to check)
    - ✓Make sure that the appserver login cannot assume any system roles

- **Use LDAP or some other external authentication for password management**
  - Centralized administration
  - Password complexity rules implemented in LDAP

Locking it down….(authentication)

Sp_configure

- secure default login **x** 0 (the default)
- FIPS login password encryption □ 1
- systemwide password expiration □ 90
- check password for digit 9 1
- minimum password length L 12
- maximum failed logins □ 5
- enable logins during recovery □ 0
- use security services □ 1

Optional

- enable pam user auth
- enable ldap user auth
- unified login required

Other password lockdowns (w/o LDAP)

- Sp_passwordpolicy
  - ✓ allow password downgrade **Y** 0
  - ✓ maximum failed logins □ 5
  - ✓ min alpha in password □ 2
  - ✓ min digits in password □ 1
  - ✓ min lower char in password □ 1
  - ✓ min special char in password → □
  - ✓ min upper char in password → □
  - ✓ minimum password length → □□
- Sp_extrapwdchecks/sp_cleanpwdchecks

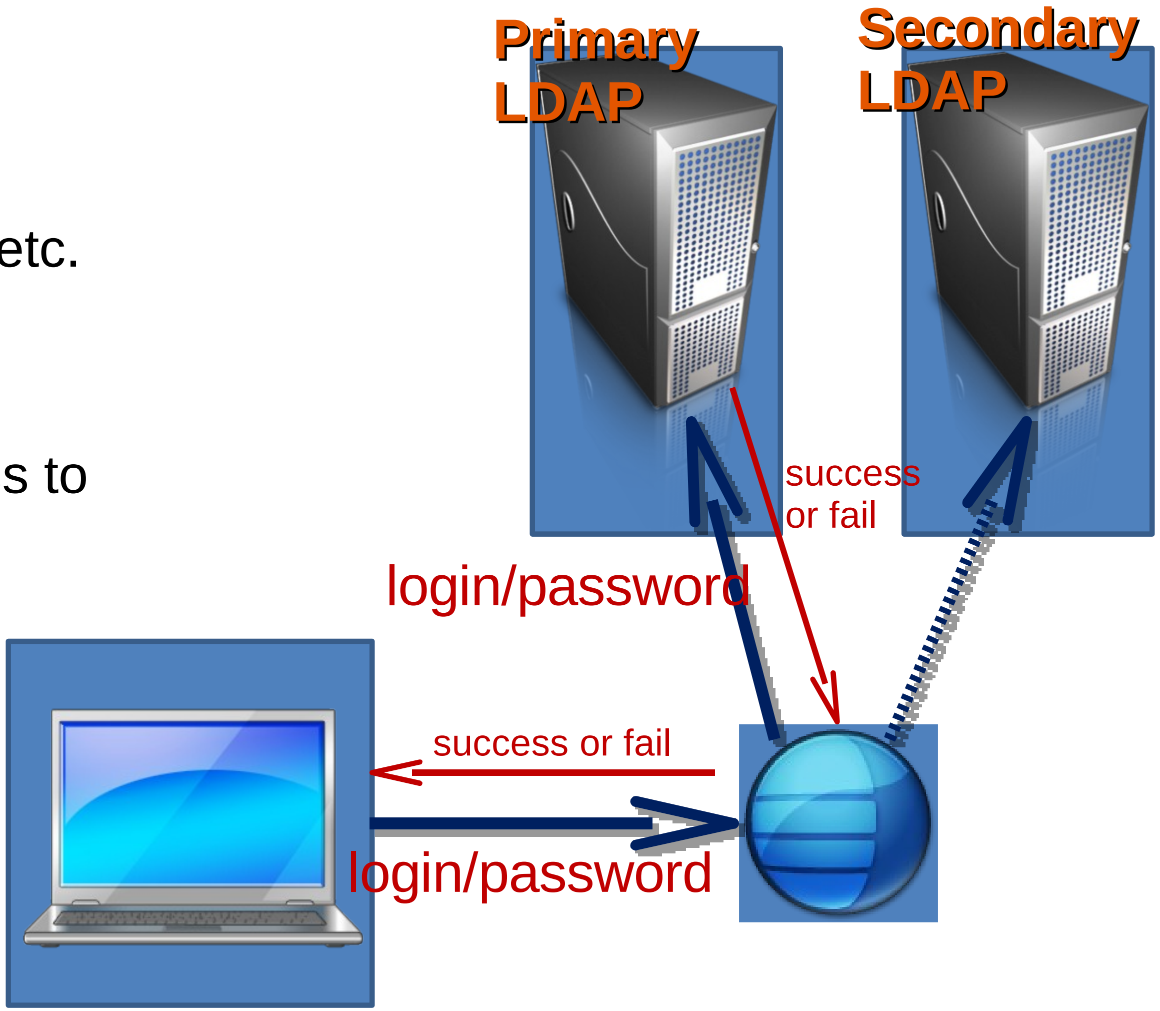
Other authentication lockdowns

- Use login triggers
- Prohibit proxy authorizing from assuming any roles

LDAP/Windows AD Authentication

·**External Authentication**

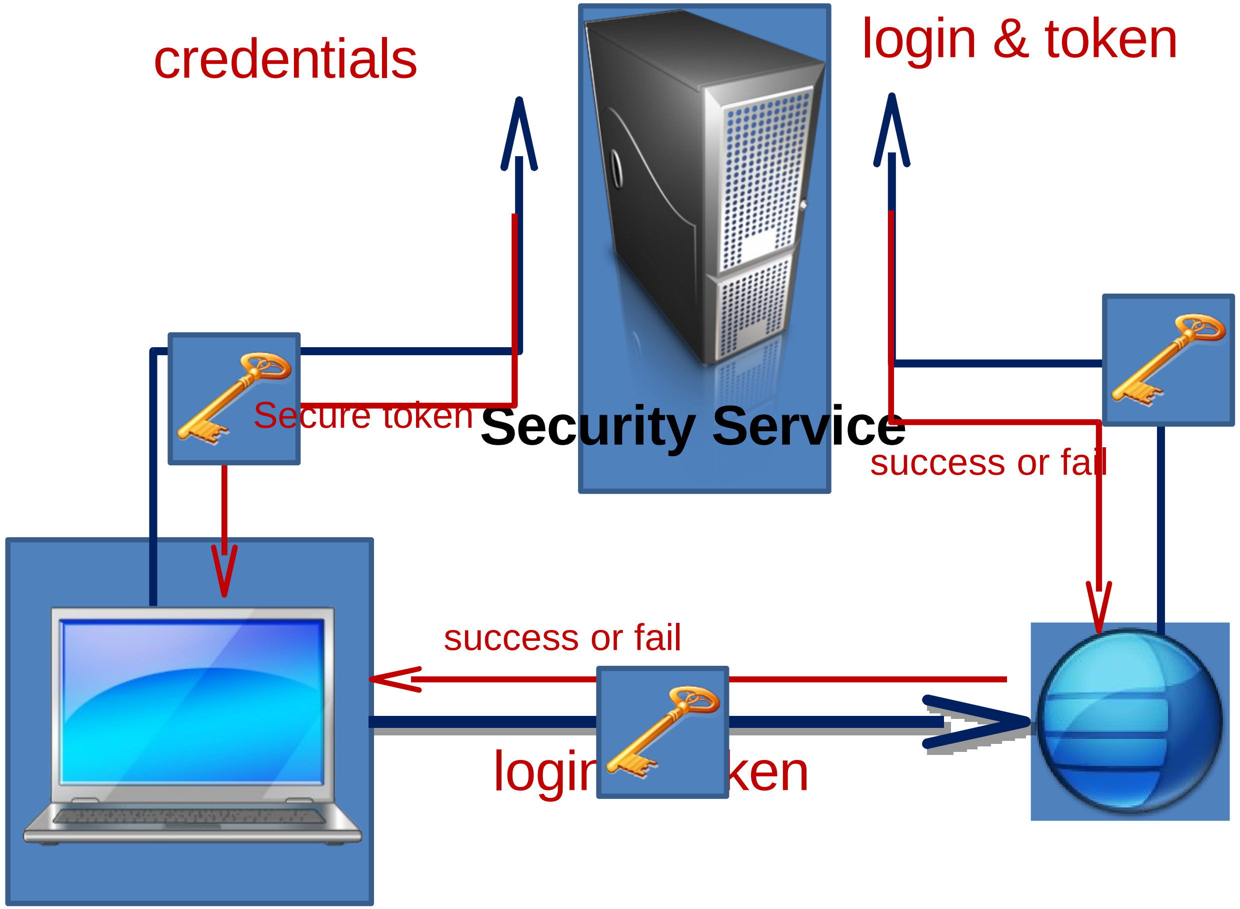
1. ASE receives user & password
2. Syslogins maps to external authentication
3. ASE sends login credentials to LDAP, AD, etc.
4. LDAP server validates credentials
5. LDAP sends success to ASE
6. ASE finishes connection and sends success to client



Kerberos Authentication

·**Kerberos Sequence**

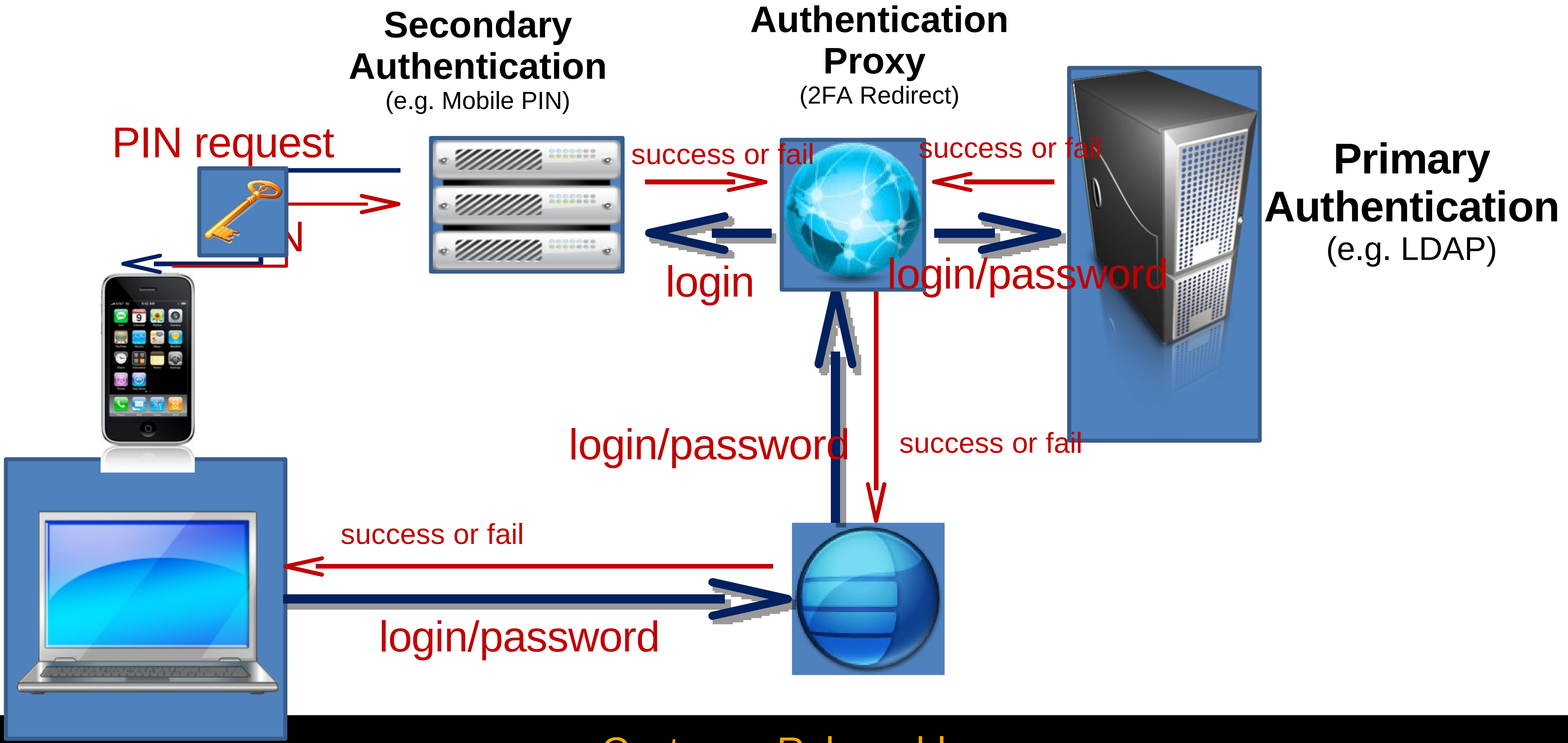
1. User connects to Kerberos server using credentials
2. User receives a secure token in return
3. User connects to ASE using the secure token
4. ASE sends the login & token to Kerberos for verification
5. Kerberos returns success
6. ASE completes connection and returns success to client login request.



Two Factor Authentication (2FA)

Problem: Standard SQL API for login authentication only provides loginname & password (or token)

Solution: Use an authentication proxy to support 2FA or biometrics



Remember “Login Profile”…..not sp_addlogin

- **sp_addlogin is deprecated**
  - A lot of new security features that sp_addlogin & sp_modifylogin didn’t support
  - Use login profiles to set high level security policy settings
  - Use create login to assign to login profile and any specific overrides/additional features
- **Avoid password complexity in create login**
  - Unless for some reason it needs to be set differently
  - Instead use sp_passwordpolicy to set a single system wide
  - Remember, this will only affect users authenticated by ASE - not LDAP, etc.

```
-- create a login trigger that enforces certain
-- appserver logins will only work from certain
-- registered hosts and program names before
-- creating the profile

create login profile applogin_profile
     with default database tempdb
          default language us_english
          login script sp_check_applogin_trigger
          authenticate with ASE
          track lastlogin FALSE
go

create login myappserver
     with password SuperSecret123
          login profile applogin_profile
          -- min password length 12
          -- max failed attempts 10
          exempt inactive lock true
go
```

Password complexity

A few in sp_configure/cfg file

- systemwide password expiration
- check password for digit
- minimum password length
- FIPS login password encryption
- unified login required
- use security services

Sp_passwordpolicy is better

- allow password downgrade
- disallow simple passwords
- enable last login updates
- expire login
- keypair regeneration period
- keypair error retry [wait | count]
- maximum failed logins
- min alpha in password
- min digits in password
- min lower char in password
- min special char in password
- min upper char in password
- minimum password length
- …

Extra password checks

- **Custom rules may be required**
  - Password cannot contain company name, persons name, etc.
  - Password cannot be password123, test123, abcd1234
  - Can’t reuse the last 5 passwords

- **Supported in ASE via 2 custom stored procs that YOU create in master**
  - Sort of like sp_thresholdaction - LCT is already defined to call it
  - Create login, etc. are predefined to call these - <u>you can’t call directly</u>

- **Sp_extrapwdcheck**
  - Custom logic you define - ***<u>must</u>*** use raiserror to flag a failure.
  - If storing a password for last “n” comparison, make sure passwords are stored securely
    - ✓ E.g. use a SHA hash or column encryption

- **Sp_cleanpwdcheck**
  - Custom logic that removes data when a login is dropped (e.g. last N passwords)

AppServer Logins: Controlling Proxy Authorization

- **Remember, you can restrict system roles**
  - Key assumption - using role based security (vs. granting permissions to individual users)
  - The best approach is likely to grant proxy to a role and restrict system roles plus any roles created under granular permissions
    - ✓ Using a role for proxy makes this easier
- **You can also make roles mutually exclusive**
  - However, this doesn’t work with proxy like you would think
  - If you make it mutually exclusive membership, then it fails only if granting excluded role to proxy login
  - If you make it mutually exclusive activation, then only if both roles activated at same time
    - ✓ The set proxy to new user deactivates the appserver_role
    - ✓ As a result, the appserver login could proxy to someone with sa_role if merely mutually exclusive vs. restricted
- **More on proxy & granular permissions later**

```
create role appserver_role
go

grant set proxy to appserver_role
     restrict role system
go

grant role appserver_role to myappserver
go

-- you need to do this for all non-system
-- roles you want active at login time
alter login myappserver
add auto activated roles appserver_role
go
```

Login Triggers

Extremely Useful

- Can implement different query optimization for different apps/ users
- Can enforce host access restrictions or other security policy (login validation!!!)

Tips:

- **Avoid** selecting from master..sysprocesses
  - ✓ It will drive spinlock contention extremely high on Resource->rpssmgr_spin
  - ✓ Use select get_appcontext (‘SYS_SESSION’, ‘<attribute>’) or pssinfo(0, ‘<attribute>’)
- Store any tables used by login trigger in master
  - ✓ If you boot with recover master only trace flag, you will not be able to log in due to missing table
  - ✓ …and only info in master - ***no xp_cmdshell or other complicated logic that delays login processing***
- Not fired for ‘set proxy/session authorization’
  - ✓ So likely not usable for applications that use a single application server connection and proxy - other than to restrict/verify the app server login is from a registered host/ipaddress

Attribute	Where you can get it
username	user_name(), suser_name(), SYS_SESSION
(client) hostname	host_name(), SYS_SESSION
applname	SYS_SESSION
dbname	db_name(), SYS_SESSION
proxy_suserid	SYS_SESSION
client_name	SYS_SESSION (set client_name)
client_applname	SYS_SESSION (set client_applname)
client_hostname	SYS_SESSION (set client_hostname)
client PID	host_id()
ipaddr (client)	pssinfo()
ipport (client)	pssinfo()
LDAP DN	pssinfo()
extusername	pssinfo()
Authentication Mech	@@authmech
SSL cipher	@@ssl_ciphersuite
Tempdb	@@tempdbid

Example - Setting up the table

```
use master
go

create table privileged_access_chk (
     LoginName       varchar(30)  not null,
     RoleName        varchar(30)  not null,
     ApplicationName     varchar(255) not null,
     RegisteredHost      varchar(255) not null,
     RegisteredIPAddr char(15)    not null,
     SSL_required    bit      not null,
          primary key (LoginName, RoleName, ApplicationName, RegisteredHost, RegisteredIPAddr)
)
go

insert into privileged_access_chk values ('sa','sa_role','DBISQL','PHLN00610123A','127.0.0.1',0)
insert into privileged_access_chk values ('sa','sa_role','isql','PHLN00610123A','127.0.0.1',0)
go
```

Example - The Login Trigger

```
use master
go

create or replace procedure sp_privloginchk_trigger
as begin
     declare @cur_hostname    varchar(30),
          @has_role_flag      int
     if (has_role('sa_role', 1) > 0)
     begin
          if exists (select 1 from master..privileged_access_chk
               where LoginName=suser_name()
                 and RoleName='sa_role'
                 and ApplicationName=get_appcontext('SYS_SESSION','applname')
                 and RegisteredHost=host_name()
                 and RegisteredIPAddr=pssinfo(0,'ipaddr'))
               return 0
          else begin
               raiserror 30000 "Login Failed"
               return -9999
          end
     end
     return 0   -- they don’t have a privileged role
end
go
```

Example - Attaching the logins

- A few notes
  - ‘sa’ login doesn’t have the login profile, so it won’t exec the login trigger
    - ✓….but remember, it is supposed to be locked
  - Don’t forget to grant appcontext permissions to public
    - ✓Remember to use the ‘builtin’ keyword to avoid confusion if some genius creates a table with the same name as a builtin function

```
grant select on builtin get_appcontext
to public
```

```
use master
go

create login profile dbalogin_profile
     with default database master
          default language us_english
          login script sp_privloginchk_trigger
          authenticate with ASE
          track lastlogin TRUE
go

create login joe_dba
     with password SuperSecret123
          login profile dbalogin_profile
          exempt inactive lock true
go
```

IQ & Login Authentication

·Similar to ASE, IQ supports external authentication

- Kerberos authentication
- LDAP authentication
- PAM authentication

·For internal (IQ) authentication, IQ supports

- Login policy (create login policy statement)
  - ✓Specifies if external authentication to be used, password expiration, etc.
- User logins (create user statement)
  - ✓Allows login policy to be specified

·IQ CONNECT events implement login triggers

- As with ASE, throw an error to disconnect user
- Unlike ASE, IQ also supports a disconnect event
  - ✓Useful for custom auditing or other functions

```
CREATE LOGIN POLICY Test1
password_life_time=UNLIMITED
max_failed_login_attempts=5;

CREATE USER SQLTester IDENTIFIED BY welcome
LOGIN POLICY Test1
FORCE PASSWORD CHANGE ON;

CREATE EVENT <event-name>
[ TYPE { BackupEnd | "Connect" | ConnectFailed |
   DatabaseStart | DBDiskSpace | "Disconnect" |
   GlobalAutoincrement | GrowDB | GrowLog |
   GrowTemp | IQMainDBSpaceFree | IQTempDBSpaceFree |
   LogDiskSpace | "RAISERROR" | ServerIdle |
   TempDiskSpace }
[ WHERE trigger-condition
     [ AND trigger-condition ], ...]
| SCHEDULE schedule-spec , … ]
…[ ENABLE | DISABLE ]
…[ AT { CONSOLIDATED | REMOTE | ALL } ]
…[ HANDLER
BEGIN
…
END ]
```

IQ & Password Complexity/Security

·IQ supports optional FIPS password encryption

- FIPS server option has to be set
  - ✓Uses SHA256_FIPS instead of SHA256
- LDAP authentication
- PAM authentication

·SSO role (SYS_AUTH_SSO_ROLE)

- Change password permission can be granted
- Supports dual control for password change
  - ✓Each DBA enters a portion of the password
  - ✓On login, user is required to change password

·Password complexity rules set via

- Server options
  - ✓Some are set at command line
- VERIFY_PASSWORD_FUNCTION

```
ALTER LOGIN POLICY <policy-name>
CHANGE_PASSWORD_DUAL_CONTROL=ON

ALTER USER <userID>
IDENTIFIED FIRST BY <password_part1>

ALTER USER <userID>
IDENTIFIED LAST BY <password_part1>
```

Password Options
LOGIN_MODE
MIN_PASSWORD_LENGTH
MIN_ROLE_ADMINS
TRUSTED_CERTIFICATES_FILE
VERIFY_PASSWORD_FUNCTION

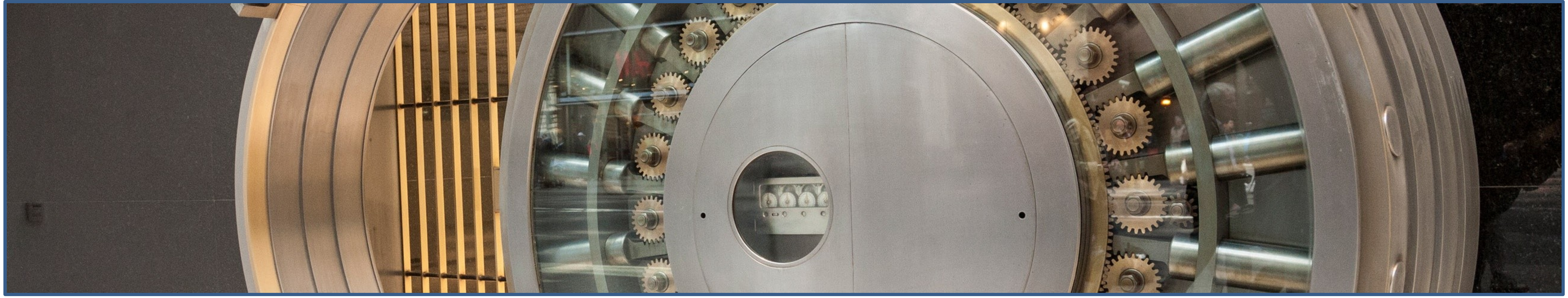
A word about SAP Replication Server

- **Create/Alter user**
  - Sets/resets password

- **Doesn’t support external authentication**
  - No LDAP, etc.
  - However, general users should not be accessing SRS - only DBA’s and possibly developers (development DBA’s)

- **SRS Configuration parameters control**
  - Password complexity
  - Failed login locking
  - Password expiration

SRS Config Param
hide_maintuser_pwd
maintuser_pwd_expiration
min_password_len
max_password_len
password_lowercase_required
password_uppercase_required
password_numeric_required
password_special_required
simple_passwords_allowed
disallowed_prev_passwords
password_expiration
initial_password_expiration
max_failed_logins
password_lock_interval
unused_login_expiration



Network Security

·SSL & Packet Encryption

Regulations, Standards & Network Security

·Many require a minimum of password encryption during login

·Network encryption is optional for most users

- Most regulations only dictate network encryption required when interfacing to external systems
- Many regulations are flexible with respect to whether network encryption is required between internal end-users and system
  - ✓Assumption is they are behind a firewall and therefore less susceptible
  - ✓Not always a good assumption….and well known as bad assumption - pushing corporations wanting end-to-end network encryption.

·Some may specify encryption for privileged users

- …mostly around non-console access

·Many corporations impose a hybrid solution

- Encrypt network between client and middle tier but middle tier to DBMS is unencrypted

Locking it down….

·Sp_configure

- net password encryption reqd 1
- enable ssl G 1
- msg confidentiality reqd 9 1
- msg integrity reqd □ 1

·Other considerations

- Block non-SSL access (except console) for public or 3rd party accessible networks
- Consider HW encryption between app servers and DBMS hosts as alternative to SSL
- Restrict access to DBMS host
  - ✓Only allow ssh & sftp access
  - ✓Restrict access to DBA staff & OS admin staff
  - ✓Restrict access to specified IP’s/hosts
    - – Could include DBA laptops and/or DBA dev/admin hosts
  - ✓Have DBA’s use own login and sudo to sybase user

·How to block unencrypted non-console access:

- Use OS firewall to block TDS ports
- Start ASE with only TDS ports on 127.0.0.1
  - ✓Also backup server
- Only allow SSL port(s) open in OS firewall
- Use login trigger to reject any other connections
  - ✓e.g. not using SSL
  - ✓e.g. not from approved hosts for DBA’s

Minimally, you should enforce network password encryption

- **Sp_configure**
  - ‘net password encryption reqd’ = 1
  - SAP apps have enabled by default
- **Apps should set as connection property**
  - Older apps may leverage OCS.cfg
  - Add -X to isql aliases
- **Check other applications for cfg**
  - SRS has send_enc_password as server config

```
; This is the external configuration definition file.
;

[DEFAULT]
        ; This is the default section loaded by applications that use
        ; the external configuration (CS_EXTERNAL_CONFIG) feature, but
        ; which do not specify their own application name (CS_APPNAME).
        ; Initially this section is empty - defaults from all properties
        ; will be the same as earlier releases of Open

[isql]
        CS_SEC_ENCRYPTION = CS_TRUE
        CS_OPT_QUOTED_IDENT = CS_TRUE
```

Network Security Made Simple - Use SSL (aka TLS)

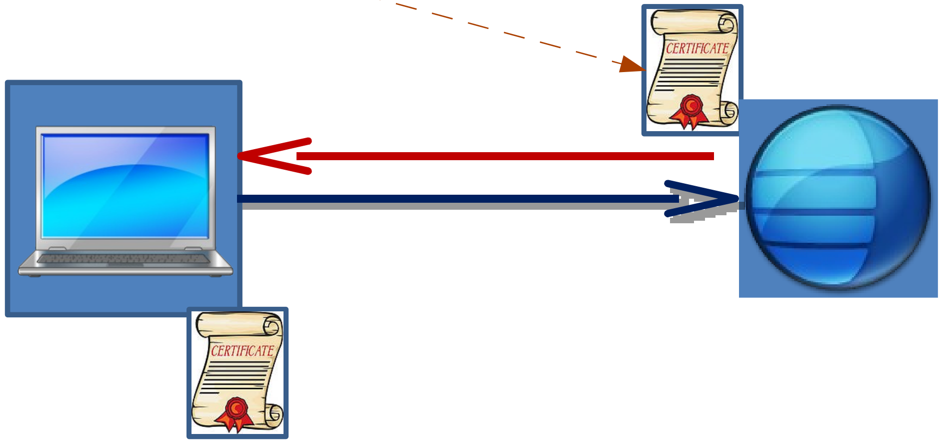
- **Many regulations require it now….**
  - …especially for non-console communications

- **Setting it up….can be a bit fun**
  - You have to get the certificate
  - You have to set the server’s trusted CA list
  - You have to configure an SSL listener
  - You have to load the server certificate in client keystore
  - You have to change your app connection API calls to invoke SSL (or conn props)
  - See Security Admin Guide, section 9 ‘Confidentiality of Data’

- **There is performance overhead**
  - All encryption has overhead
  - 40KB more memory per connection
  - Can be 2x longer round trip time

```
master tcp ether myhostname 30001 ssl="CN= SYBSRV1.mydomain.com"
query tcp ether myhostname 30001 ssl="CN= SYBSRV1.mydomain.com"
master tcp ether 127.0.0.1 30000
query tcp ether 127.0.0.1 30000
```

Common name in certificate must match interfaces file



Server certificate must be in client keystore!!!

Common Issues

·SSL Certificate or Private Key formats

- SSL Certificate must be in PEM format
  - ✓If in PKCS #12 format use openSSL to con[redacted] to PEM (see www.openssl.org)
- The private key must be in PKCS #8 encrypted format.

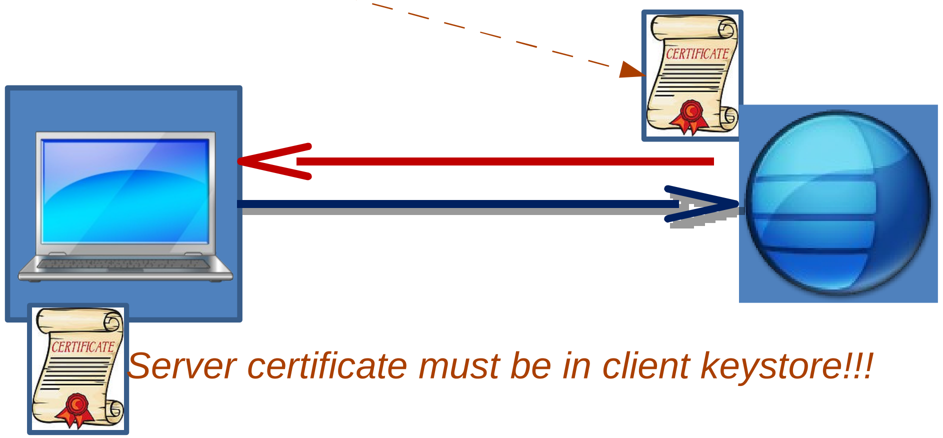
·Other common problems

- Servername in interfaces doesn’t match COMMON NAME (CN)
  - ✓Remember, the actual servername in ASE is irrelevant
- CA not listed in trusted CA file on server
- Forgot to have separate ports/listeners in interfaces file for SSL
- Forgot to load server certificate in client keystore

Separate tcp ports for SSL listener

```
master tcp ether myhostn[redacted]sl="CN= SYBSRV1.mydomain.com"
query tcp ether myhostna             l="CN= SYBSRV1.mydomain.com"
master tcp ether 127.0.0.
query tcp ether 127.0.0.1 30000
```

Common name in certificate must match interfaces file



Server certificate must be in client keystore!!!

```
%JAVA_HOME%\jre\bin\keytool -import -trustcacerts
     -file <absolute path of servername.txt>
     -alias root
     -keystore %JAVA_HOME%\jre\lib\security\cacerts
```

What ASE supports with SSL

·Supported features

- Server identity authentication
- Message integrity
- Message confidentiality
- Dynamic listeners
  - ✓Remember, protocol is ssltcp vs. tcp
  - ✓Also in threaded kernel, engine param is not used

·Unsupported features

- Client identity authentication

·Supported versions

- TLS 1.1+ **Y** ASE 15.7sp137+; 16sp02 pl04+
- TLS 1.0 □ASE 15.x; 16.x
- Note that TLS 1.0 & older SSL versions are deprecated due to vulnerabilities

```
exec sp_configure ‘enable ssl’, 1

exec sp_configure ‘msg confidentiality reqd’, 1

exec sp_configure ‘msg integrity reqd’, 1
```

```
sp_listener 'start',
   'ssltcp:blade1:17251:"CN=ase1.big_server_1.com"'
```

Alternative to SSL

- **SSL has performance penalty**
  - Connection speed up to 2x slower
  - Data transmit speeds up to 2x slower
  - Performance penalty not completely offset by proprietary encryption chips on motherboard
    - ✓ May only help by 40% + limited platforms/versions
  - However, it encrypts entire end-to-end data stream
    - ✓ Application to DBMS

- **Hardware Network Encryption**
  - Advantages
    - ✓ Transparent to applications
    - ✓ Easier implementation
    - ✓ Supports advanced policies
    - ✓ Much better performance
      - – Low latency applications
    - ✓ Likely cheaper (priced per device vs. per core)
  - Disadvantages
    - ✓ Susceptible to sniffing programs on same host as applications or DBMS
    - ✓ Need to purchase high-end units to support 10GbE
      - – Entry level & mid-range only support 100Mbs or 1Gbs

ASE 16SP03 Feature: On Demand Encryption

Future

·Problem

- Password encryption is enforced during login, but if user changes password, the old & new passwords are sent in clear unless using SSL
- Other commands with sensitive data have similar issues (e.g. encryption passwords, etc.)
  - ✓See list
  - ✓Intent is that programmer would invoke on demand encryption before sending these commands and likely disable afterwards

·Encryption added to OpenClient directly

- no SSL necessary
- negotiated symmetric session key between client and server
- AES algorithm with 256 bit keys.
- Support in CTLib - probably JDBC & ODBC
- Isql will use “go encrypt” vs. “go” for command/batch terminator

alter encryption key
alter login
alter role
connect
create cluster
create encryption key
create login
create role
deploy plugin
drop encryption key
dump database
dump transaction
load database
load transaction

set cluster
set encryption passwd
set role
show agent
upgrade server
sp_addexternlogin
sp_addlogin
sp_companion
sp_encryption
sp_extrapwdchecks
sp_ldapadmin
sp_password
sp_ssladmin

ASE 16 SP03 & SSL

Future

- **Currently both ASE & SRS use OpenSSL implementation**
  - FIPS/NIST certified OpenSSL implementation
  - Problem is that with any large code project with code freeze >3 months before GA, the near constant patching of OpenSSL is a problem

- **SAP Common Cryptographic Library**
  - Supports SSL for network security
  - Currently in last/final phase of FIPS/NIST certification
  - Will replace OpenSSL starting with ASE 16 SP03 & SRS release Q2’17
  - IQ will adopt at some point as well (schedule/release unknown)

IQ Supports SSL as well

RSA or FIPS (OpenSSL) supported SSL algorithms

- RSA on all platforms
- Docs say that FIPS only on
  - ✓Server: LinuxAMD64, Solaris Sparc, Solaris AMD64
  - ✓Client; LinuxAMD32, Windows32
- …but then mentions that for FIPS you need to use 64-bit libs on 64 bit Windows, so 32-bit client limitation seems to be a doc bug
- FIPS moving to SAP CCL as soon as FIPS approved and release vehicle defined

Rather than different listeners, IQ uses -ec & -es command line options

- -ec { NONE | SIMPLE | TLS ([FIPS={ Y | N };] IDENTITY=<server-identity-filename>; IDENTITY_PASSWORD=<password>) },
- -es ▯ used with the above to all SSL connections via shared memory connections
- You can also block TDS connections (start_iq -x tcpip(TDS=NO)….)

Password encryption is also supported (EncryptedPassword connection parameter)

A word about SAP Replication Server

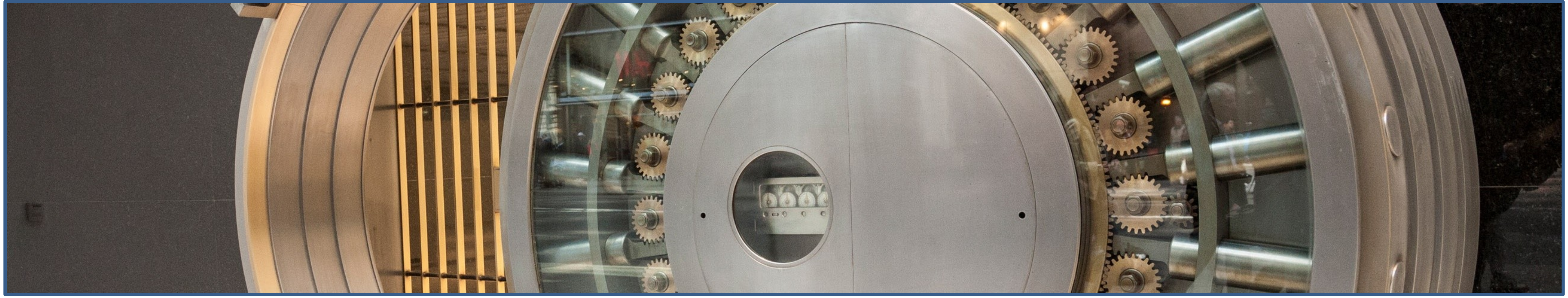
·Can use SSL/security services

- Most are connection level
    - ✓Server default, but can be set per connection
- Controls security between:
    - ✓SRS RSSD
    - ✓SRS SRS (route)
    - ✓SRS RDB

·RepAgents have similar params

- To enable SSL from RepAgent to SRS
    - ✓LTL RepAgent supported today
    - ✓CI RepAgent support in ASE 16sp02 pl05 (Q4'16)
- This completes the end-to-end security
    - ✓RA SRS) SRS 6   RDB
- For HADR, RMA support for SSL is being added as well
    - ✓Covers the non-console DBA access aspect of compliance

SRS Config Param
msg_confidentiality
msg_integrity
msg_origin_check
msg_replay_detection
msg_sequence_check
mutual_auth
security_mechanism
send_enc_password
unified_login
use_security_services
use_ssl



User Permissions

- Granular Permissions, Execute as Owner, etc.

Regulations, Standards & User Permissions

- **Nearly all, if not all, regulations require data access restrictions**
  - Some form of rules around limiting data access to those with “need-to-know”
  - Some may have rules around who has access to the binaries are can update the binaries
  - None restrict access to the schema (as this prevents common reporting tool access)

- **This is strongly reliant on the capability to distinguish different users**

- **Generally it is recognized that persona/role-based access restrictions are best**

- **The technology challenges:**
  - How to do this in a manageable fashion vs. umpteen thousand permissions to manage
  - How to manage personnel changes and changes in access
    - ✓Jill gets promoted to manager….now needs more access than previous
    - ✓Jack fell down …and got fired for being drunk - needs permissions revoked
    - ✓Obviously, a need to be able to ‘push down’ the access controls closer to the business is best
      - However, this would require the application to have some form of application security officer support - which is extremely rare. Consequently, most often what we see is high level, single tier role based security implementations

Best Practices for Security Compliance - User Permissions

- **Use role based security/role based access controls (RBAC)**
  - Grant all permissions, etc. to roles - not individual users
  - Only give the users roles they need (hint - you may need a role hierarchy)

- **Use execute as owner only when security policy allows it**
  - One of many recent enhancements to ASE in past years was the ability to execute a stored procedure either as ‘caller’ (normal) or as ‘owner’.
  - Executing proc as owner would have the equivalent effect (from permissions/statements within the proc) as if it was the owner executing vs. the caller.
  - This can be useful when high cardinality checks for user precludes predicated privileges
    - ✓ E.g. allowing employees to update their own HR records for address, phone numbers, etc. where there isn’t any decent column for which a predicated privilege could be easily implemented (e.g. loginname isn’t in HR table).

- **Use Predicated Privileges to implement row level access controls**

- **After ASE 16.0 SP03, enable ‘restrict owner permission’**

Restrict Owner Permission

- **Similar to ‘restricted decrypt permission’….**
  - Added in ASE 15.0+
  - By default, with encrypted columns, the object owner has decrypt permission
    - ✓DBA’s with sa_role become ‘dbo’ in any database and could see the data
  - Enabling the ‘restricted decrypt permission’…
    - ✓Blocks ‘owner’ from being able to decrypt data
    - ✓Only allows the SSO to grant decrypt permission
    - ✓SSO can grant decrypt permission ‘with grant option’ to allow others to grant decrypt

- **ASE 16 SP02 PL05 (or SP03) will add ‘restrict owner permission’**
  - By default, the object owner has full DML permissions on the object
  - By enabling this, object owner no longer has DML permissions
    - ✓As with restrict decrypt permission, it will likely fall to the SSO and be grantable
  - Keep in mind that this may make debugging queries more difficult
    - ✓Although there still is that ‘setuser’ command……(for Suite users, this is pretty normal)
    - ✓

Future

Row Level Access Controls - the Old Way

ASE 12.5 added Access Rules, along with ACF

- Rules were bound to a column
- Rules were limited to values in the row or system functions except via SQL UDF
- Rules were applied universally to selects/inserts/updates/deletes
- Required object ownership to define (and create object permission)

Example:

- Based on using “access rules”

```
create access rule StateCode_hash_rule as
    proc_role('sa_role')=1 or StateCode_hash(@state,suser_name())=1
go
exec sp_bindrule StateCode_chk_rule, "residential_customer.state"
go
```

- Limited to just columns in table unless you used SQLJ or SQL UDF
- Hard to combine rules (create [and | or] access rule….no precedence or order)

Predicated Privileges (1)

- **ASE 15.7 ESD #2+ adds “Predicated Privileges”**
  - Just like RLAC/FGAC, it requires Security/Directory Services option (no change there)
  - Must be enabled with sp_configure ‘enable predicated privileges’
  - You will need to tune sp_configure ‘permission cache entries’
    - ✓ I like to set to 50 to start with anyhow….
    - ✓ ….but with predicated privileges, you will need to add 1 for each table that requires more than 1 privilege (e.g. combining grants)
  - Companion to, but not to be confused with ‘Granular Permissions’ (system privileges)

- **Predicated Privileges**
  - Grant with a where clause
  - Any legal subquery expression
  - Can have different expressions for inserts vs. updates vs. deletes vs. selects

Predicated Privileges (2)

- **Example Syntax**

❖ Full Syntax:

```
grant {all [privileges] | permission_list}
    on table_name [correlation_name] [(column_name_list)]
    [where search_conditions
    [as pred_name]]
to {public | name_list | role_list}
[with grant option]
```

❖ Example

```
grant select on rs_statdetail
    where counter_id in (select counter_id from rs_statcounters
                            where module_name in ('SERV','RSH'))
to public
-- yes, that WAS a subquery to a different table…it works…yeeehhhaawwww
-- just be careful as there could be performance implications with joins
-- if multiple predicated tables involved
```

Predicated Privileges (3)

- **It also applies to roles**
- Full syntax:

```
grant role role_name
[where pred_expression]
to {username | login_profile_name }
```

- Example:

```
-- Bob is our night-time DBA...make sure he does nothing goofy during the day
grant role oper_role
where datepart(hour, current_time()) not between 8 and 18
to Bob
```

Full Predicated Privileges Syntax & Debugging

```
grant {all [privileges] | permission_list}
on table_name [<correlation_name>] [(<column_name_list>)]
[where search_conditions
[as pred_name]]
to {public | <name_list> | <role_list>}
[with grant option]

grant role <role_name> [where <pred_expression>]
to {<username>< | ><login_profile_name >}

revoke {all [privileges] | [all] <permission_list>} on <table_name> (<column_list>)
[with { <pred_name> | {all |no} predicates}]
from {public | <name_list> | <role_list>}

set show_transformed_sql {on | off}
set show_permission_source {on | off}
exec sp_helptext <pred_name>
```

OS Level Permissions

- **Enforce sudo for ‘sybase’ user**
  - Have DBA’s connect to host as themselves and sudo to ‘sybase’ user
  - This is especially required if compliance regulation requires auditing of file changes

- **Make $SYBASE accessible only by ‘sybase’ user**
  - Some compliance regulations require this to prevent access to binaries
  - Install a separate copy of OCS for applications to link to as well as ‘public’ interfaces on the host
    - ✓ This would allow the server to have a ‘private’ interfaces to support dynamic listeners

- **Make sure data devices are owned and only accessible by ‘sybase’ user**
  - Don’t forget the devices you create in /tmpfs

IQ & Object Permissions

- **Similar to ASE - supports object & column permissions**
  - Doesn’t support execute as owner permission - but that can be a good thing from security POV
  - Similar to ASE, recommend a Role Based Access Control
    - ✓ Grant all permissions to roles
    - ✓ Grant roles to users

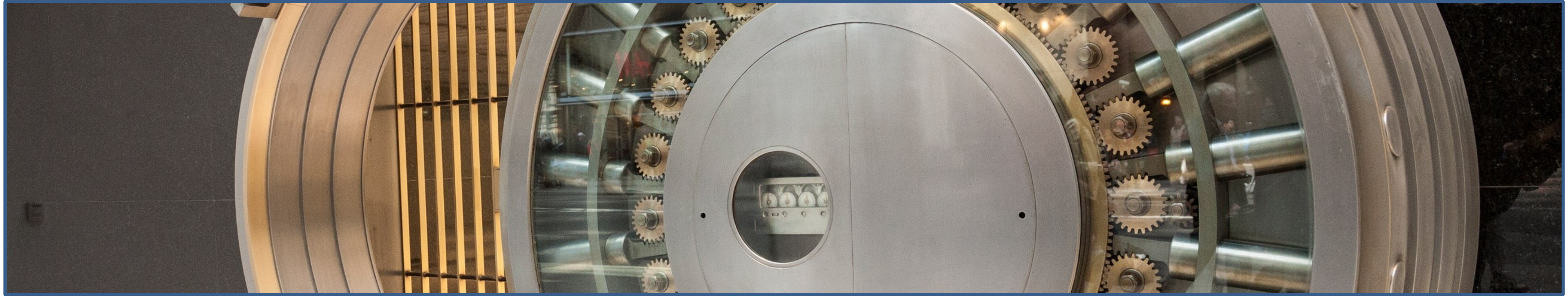
- **Doesn’t support row level access controls**
  - No predicated privileges
  - No RLAC rules
  - The workaround is to create views

A word on SRS

- **Supports a limited set of permissions:**
  - `grant {sa | create object | primary subscribe | connect source} to user`
  - Many commands require ‘sa’ (e.g. suspend/resume)
  - ‘admin’ commands (admin who, admin stats, etc.) can be run by anyone

- **For ops staff & monitoring tools, create accounts in both SRS and RSSD**
  - Don’t grant any permissions - they can monitor via admin who and SRS MC
  - Create account in RSSD with select only

- **For developers, consider**
  - Create object and/or primary subscribe to allow them to maintain repdefs/subscriptions
  - Create account in RSSD with select only



Data Security

- Full Database Encryption, Encrypted Columns, Secure Wipe, Backup Security

UKSUG
SAP DATABASE & TECHNOLOGY USER GROUP

Regulations, Standards & Data Encryption

- **Most current regulations requires some form of data encryption**
  - Often expressed as protecting “data at rest”

- **Many have identified specific data elements….**
  - …that need to be encrypted
  - …that need to be obfuscated if public access is allowed or data disseminated

- **This is an area where the most mistakes are made**
  - Usually due to lack of familiarity with the exact regulatory requirement
  - DBA’s use column encryption where not needed/bad fit - in lieu of full database encryption
  - Primarily view data masking/obfuscation from a production development lifecycle issue
    - ✓ This may be a corporate standard, but not specifically called out by *any* common regulations
    - ✓ In fact, HIPAA is fine with internal employees and even allows 3rd party (vendor/partner) access when legal agreements exists (and there is a need to do so)

Best Practices for Data Security

- **Read and understand the data security requirements for the applicable standards**

- **Only grant access to database to users who need it & permissions on tables to roles that require it**

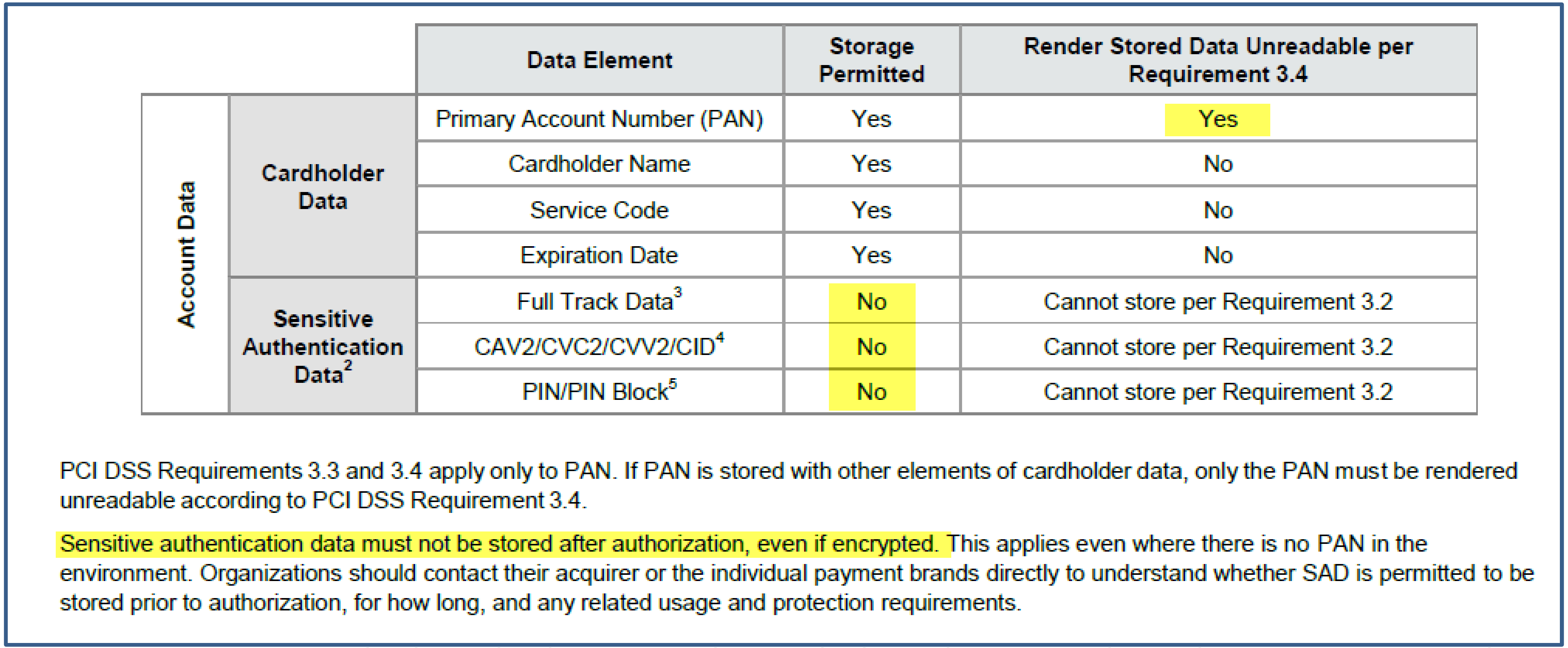
- **Enable residual data removal on ‘people’ tables (e.g. customers)**
  - …and any related tables that use personal identifiers as fkeys (e.g. SSN)
  - …or similar tables such as ‘programs/projects’, etc.

- **Minimally, use full database encryption (ASE 16+)**
  - Protects data at rest/storage
  - Protects backups
  - Does NOT protect sensitive data elements from authorized system users

- **Use Encrypted Columns (w/ masking) on truly sensitive data elements**
  - These should items such as bank account numbers, SSN, etc.
  - Only grant decrypt permission to those who absolutely need to read the data
    - ✓ A customer may need to insert their credit card info, but they do not need be able to read it - if it is wrong, just have them re-enter the information.

- **Implement a data masking/obfuscation solution for public access if required**
  - Don’t confuse data masking for public access with data encryption for data security
  - This doesn’t have to be the same system - create a separate ‘public’ system maintained via replication, ETL, etc.

PCI DSS Example



		Data Element	Storage Permitted	Render Stored Data Unreadable per Requirement 3.4
Account Data	Cardholder Data	Primary Account Number (PAN)	Yes	Yes
		Cardholder Name	Yes	No
		Service Code	Yes	No
		Expiration Date	Yes	No
	Sensitive Authentication Data[2]	Full Track Data[3]	No	Cannot store per Requirement 3.2
		CAV2/CVC2/CVV2/CID[4]	No	Cannot store per Requirement 3.2
		PIN/PIN Block[5]	No	Cannot store per Requirement 3.2

PCI DSS Requirements 3.3 and 3.4 apply only to PAN. If PAN is stored with other elements of cardholder data, only the PAN must be rendered unreadable according to PCI DSS Requirement 3.4.

Sensitive authentication data must not be stored after authorization, even if encrypted. This applies even where there is no PAN in the environment. Organizations should contact their acquirer or the individual payment brands directly to understand whether SAD is permitted to be stored prior to authorization, for how long, and any related usage and protection requirements.

Source: Payment Card Industry (PCI) Data Security Standard Requirements and Security Assessment Procedures v3.1

Net net: Card account number must be encrypted if stored, but cardholder name & expiration date does not have to be. However, the CVV or PIN (debit or chip+PIN) can not be stored (and now you know why iTunes always forces you to enter the CVV).

Residual Data Removal

·**Residual data on deleted objects could be visible to users <u>with</u> privileges**

·**Some tasks which result in residual data**

- Drop database, Alter/Drop/Truncate table, Drop index, Delete row
- Space deleted – but data still resides there

·**Why residual data**

- Erasing the data as part of the transaction could slow down transaction
- Rows deleted would need page to be re-written to clear data
- Pages/Extents in dropped objects would need to be zero'd

·**Options provided to enable/disable this feature at different granular levels**

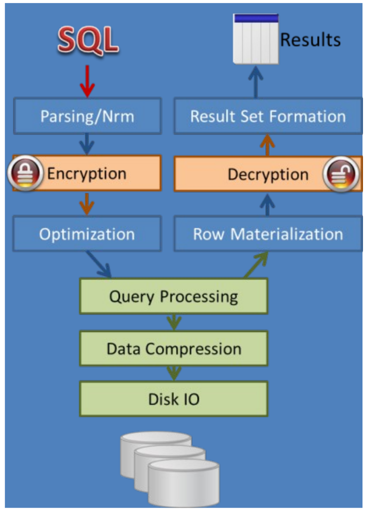
- Supported levels
  - ✓Database level □ sp_dboption <dbname>, "erase residual data", true
  - ✓Session level □ set erase_residual_data {on | off}
  - ✓Table level □alter table <tablename> set erase residual data {on | off}
- Once set at database level, all tables will have this on, unless explicitly turned off (session or table)

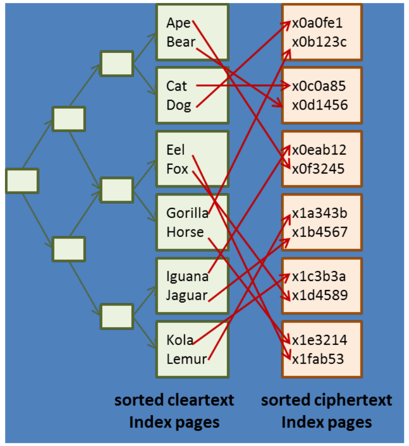
Column Encryption in ASE

- **Column Encryption (ASE 12.5.4 and later)**
  - Totally secure – encrypted on disk, in memory, in log, etc…..
  - Each column can be encrypted with separate keys
    - ✓ Assumes different users with different access requirements
    - ✓ Prevents inadvertent disclosure to authorized users of system but not authorized for data
  - Column decrypt permission with data masking (unique to ASE)
    - ✓ e.g. ###-##-####

- **Impacts on performance**
  - Good news: Indexable encryption for Pkey and Fkey columns
    - ✓ Unique to ASE vs. Oracle and other competitive implementations
  - Bad news:
    - ✓ Range queries (due to ciphertext sorting) and other qp issues
    - ✓ Blocks compression effectiveness (if a SALT/IV is used)

- **Certified with SAP applications….but….**
  - Only provides disk level protection as SAP uses common login





Works extremely well on account numbers, employee id, SSN, health test results, etc. (equi-SARGS)

Doesn't work well on Date of Birth, Last Names, etc. due to range queries – nor also on ORDER BY/GROUP BY columns that are indexed (queries still work but could be slower or a lot slower)

Column Encryption Best Practices

·**Keep keys in separate database**

- Separate storage of keys & data are a must
  - ✓ Use a separate keydb for each database/app
- Also allows you control of who has access to key repository (and can grant access to them)
  - ✓ Check to make sure all users in keydb really need access
  - ✓ Use user aliasing for key ownership (next slide)

·**Always use a decrypt default/data mask**

- Avoids application failure on error without decrypt permission (ASE doesn't expose ciphertext)

·**Use SALT/IV appropriately**

- Don't use on high domain columns such as SSN, bank account #'s where they might be searched for as SALT/IV prohibits indexing
- Definitely use on low domain columns (<20-50K distinct values)

·**Avoid encrypting primary keys**

- Yes, this is doable…and indexable
- But if a user doesn't have decrypt, it is hard to lookup detail records in child tables if the master pkey is encrypted
- This could have an impact on schemas
  - ✓

```
create encryption key [[<database>.]
[<owner>].]<keyname>
[as default] [for <algorithm_name>]
[with [{{passwd {char_literal |
system_encr_passwd} | master key}]
[key_length <num_bits>]
[init_vector {null | random}][pad {null | random}]
[[no] dual_control]}]
```

```
create table emp (
    name char(50),
    ssn char(11)
          encrypt with keydb.hr_admin.ssn_key
          decrypt_default '000-00-0000',
...
)
go
grant select on emp to public
go
grant decrypt on emp(ssn) to hr_role
go
```

Simplifying Key Management for Column Encryption

·**Remember:**

- Key Custodian Role
  - ✓ Creates a key using create encryption key
  - ✓ Grants schema developers ability to use key via ‘grant select on <keyname>’
- Schema developer
  - ✓ Creates table with encrypted column specifying key
- Both have to be users in the key database
  - ✓ You don’t necessarily want to alias key custodians as dbo

·**Problem:**

- Development
  - ✓ Bob is key custodian in dev/test - creates key and grants select to Jane
  - ✓ Jane is schema developer - uses key
- Issue 1 - Development moves to prod (Bob & Jane users don’t exist)
- Issue 2 - Bob/Jane leave company

·**Solution**

- In key database, alias Bob to ‘key_mgmt’ user and alias Jane to ‘app_schema’ user
- Key_mgmt & app_schema logins are locked

·**Note: A similar issue exists with full database encryption, but keys are in master**

A walk through scenario (1)

- **Assume we have 2 databases**
  - Pubs2_encr **v** pubs2 database with encrypted columns
  - Pubs2_encr_keys L contains column encryption keys

- **The preparation steps**
  - Add users and alias logins
  - Grant manage column encryption key to the user (must be granted to user or role, not login)
  - Create a master key for the key database
    - ✓Yes, master keys are database specific
  - Create a copy of the master key for the database SSO

```
use pubs2_encr
go
exec sp_addalias 'jane_dba','dbo'
go

use pubs2_encr_keys
go
exec sp_adduser 'pubs2_encr_sso'
exec sp_adduser 'pubs2_encr_schema'
exec sp_addalias 'bob_sso', 'pubs2_encr_sso'
exec sp_addalias 'jane_dba', 'pubs2_encr_schema'
go

grant manage column encryption key to pubs2_encr_sso
go

create encryption key master
for AES with passwd 'ThisIsASpecialKeyForEncryptingKeys1234'
go [encrypt]

alter encryption key master
with passwd 'ThisIsASpecialKeyForEncryptingKeys1234'
add encryption with passwd 'ThisKeyIsForPubs2EncrSSOs5678'
for user 'pubs2_encr_sso'
go [encrypt]
```

A walk through scenario (2)

Now the database SSO has to create the keys

- Bob_sso logs in and is aliased to pubs2_encr_sso

The steps

- Set the master key password on to enable access to master key
- Create the column encryption key
- Grant select on the key to the database schema development DBA

Notes:

- The key is owned by pubs2_encr_sso
- Key protection is also via pubs2_encr_sso as master key is owned by dbo

```
use pubs2_encr_keys
go

set encryption passwd 'ThisKeyIsForPubs2EncrSSOs5678'
for key master
go [encrypt]

create encryption key ssn_key
for AES with keylength 256 init_vector null pad null
go

grant select on ssn_key to pubs2_encr_schema
go
```

A walk through scenario (3)

- **Now the database schema DBA can use the key in schema to encrypt columns**
  - Jane_dba logs in and is aliased to dbo

- **The steps**
  - Create table with decrypt default
  - Grant normal permissions (without decrypt) to normal roles
  - Grant decrypt to roles who need it

- **Do you see the pkey issue???**
  - How do you look up titles for a specific author (or sales) without decrypt permission?
    - ✓ Only way is to have an alternate key (e.g. au_lname + au_fname must be unique and have a unique index defined)
    - ✓ This still could have an impact due to always joining with parent table
    - ✓ This fails to return rows in ASE 16SP02

```
use pubs2_encr
go

create table authors (
            au_id   char(11)       not null
                encrypt with pubs2_encr_keys.pubs2_encr_sso.ssn_key
                decrypt_default '000-00-0000',
            au_lnamevarchar(40)         not null,
            au_fnamevarchar(20)         not null,
            phone   char(12)       not null,
            address varchar(40)         null,
            city    varchar(20)         null,
            state   char(2)        null,
            country varchar(12)         null,
            postalcode   char(10)       null,
               primary key (au_id)
)
lock datarows
go
```

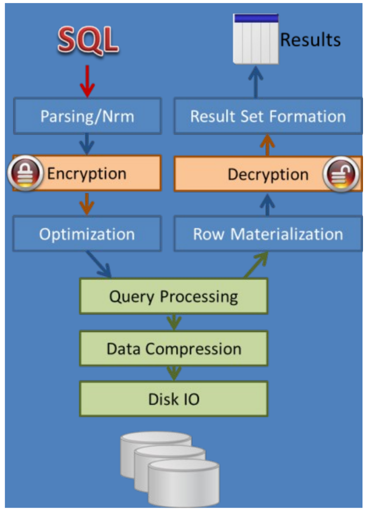
Impact of column encryption on Query Optimization/Processing

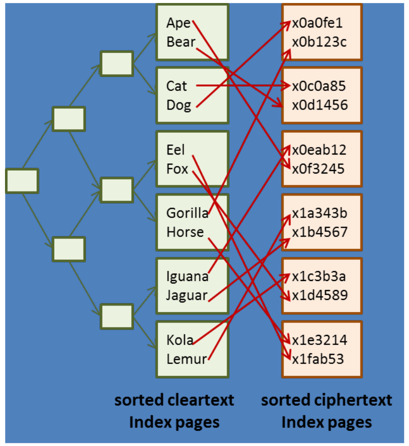
- **Since it is encrypted in memory and disk, some queries are impacted**
  - Many of these were discussed back in 12.5.3/12.5.4 release days….but it has been awhile….
    - ✓ Range scans (including LIKE predicates)
    - ✓ Order by/Group by (where ASE would use an index to do a sort avert
    - ✓ In some cases, the datatype translation of literal values
  - These still are problems for ASE (and Oracle and MSSQL and DB2)….
    - ✓ ….except they don't support indexing them at all

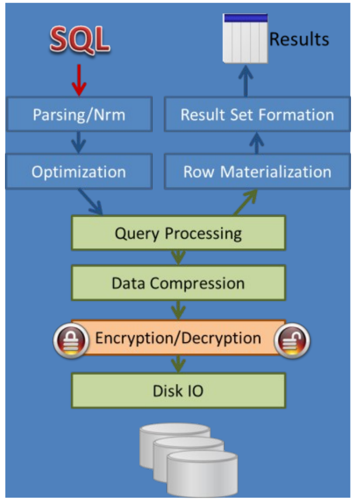
- **To avoid the problems**
  - Only use column encryption on truly sensitive data - e.g. SSN, account numbers, etc.
  - Use database encryption for names and other general information that just needs it at rest
  - Use SALT/IV only on low cardinality data (e.g. medical test results)

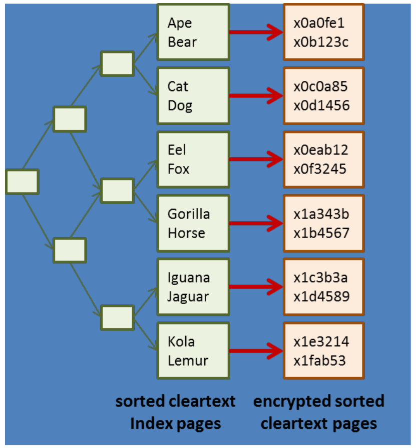
Full Database Encryption

- **Full Database Encryption (new in ASE 16)**
  - Provides protection for an entire database, WITHOUT affecting existing applications
    - ✓ All data, indexes, and transaction logs in database are encrypted
    - ✓ Backed up database in encrypted form
    - ✓ All authorized database users can see data
  - No impact on range queries or compression
  - Encryption is at page level
    - ✓ as pages are written to disk, and decryption before they are loaded into memory
    - ✓ Will be after/before Compression/Decompression
  - Can be used with column encryption
  - Dual key control with automatic startup
- **Can be implemented online w/o user impact**
  - Can be suspended/resumed for long run times









Database Encryption

- **Uses Database Encryption Key to encrypt a database – symmetric key**
  - User has to create the key before database encryption
      - **create encryption key key_name [for AES] for database encryption**
      - **[with {[master key] [keylength 256] [init_vector random] [[no]dual_control]}**
    - Default is init_vector random (mandatory)
    - Example:
      - **Create encryption key test_key for database encryption with master_key**
      - **Create encryption key test_key for database encryption with dual_control**
  - Master key and dual master key will be used to protect test_key

- **Create database with encryption**
  - Create a new encrypted database
      - **Create [temporary] database database_name encrypt with key_name**
  - Alter an existing database to be encrypted
      - **Alter database database_name encrypt with key_name**

Notes about full database encryption

- **Fully online operation**
  - User queries and DML are not affected
- **Can be paused/resumed**
  - …and at times, it needs to be
- **Why would you pause:**
  - Database backups (including incremental)
  - Shut down the server (for maintenance or static config changes)
  - …others (see documentation)
    - Database encryption is implemented within the server
    - Anything that reads the raw pages needs to pause encryption (e.g. backup server)
    - If the server is going to have an outage, you should pause encryption first
      - It will recover from a crash, but kinder/gentler operations avoids problems
- **Backup master database**
  - Ideally, you should be doing this every day anyhow
  - Database Encryption Keys are kept here….
    - You lose master…and you really lost the data
    - Especially if you didn’t do the key recovery steps ahead of time

```
alter database <database_name>
{[encrypt with <key_name>
| decrypt [<with key_name>]]
   [parallel <degree_of_parallelism>]
| resume [encryption | decryption
   [parallel <degree_of_parallelism>]]
| suspend [encryption | decryption] }
```

ASE Data Encryption/Data Security vs. MS SQL

Data Security Feature	ASE 16	MS SQL 2016	ORACLE 12
Transparent full database encryption	✓	✓	✓
Column encryption	✓	✓	✓
Transparent column encryption (usable by SAP/older apps, etc.)	✓	✗	✓
Decrypt permission on encrypted columns	✓	✗	✗
Dynamic masking/redacting of cipher text on encrypted columns	✓	✗	🛠
Encrypted cols in indexes/joins	✓	✗	✗
DBA can be prohibited from viewing encrypted column data	✓	! WHAT'S NEW	🛠
Dual key control	✓	✗	✓
Export encrypted data	✓	✗	✓
Row level access (using UDF or ACF)	✓	! WHAT'S NEW	🛠
Predicated Privileges (grant with where clause)	✓	✗	

HIPAA Definition of Sensitive Data (1)

§ 164.514 Other requirements relating to uses and disclosures of protected health information.

- (a) *Standard: De-identification of protected health information.* Health information that does not identify an individual and with respect to which there is no reasonable basis to believe that the information can be used to identify an individual is not individually identifiable health information.
- (b) *Implementation specifications: Requirements for de-identification of protected health information.* A covered entity may determine that health information is not individually identifiable health information only if:
  - ✓ (1) A person with appropriate knowledge of and experience with generally accepted statistical and scientific principles and methods for rendering information not individually identifiable:
    - (i) Applying such principles and methods, determines that the risk is very small that the information could be used, alone or in combination with other reasonably available information, by an anticipated recipient to identify an individual who is a subject of the information; and
    - (ii) Documents the methods and results of the analysis that justify such determination; or
  - ✓ (2)(i) The following identifiers of the individual or of relatives, employers, or household members of the individual, are removed:

HIPAA Definition of Sensitive Data (2)

·1. Names.

·2. All geographic subdivisions smaller than a state, including street address, city, county, precinct, ZIP Code, and their equivalent geographical codes, except for the initial three digits of a ZIP Code if, according to the current publicly available data from the Bureau of the Census:

- a. The geographic unit formed by combining all ZIP Codes with the same three initial digits contains more than 20,000 people.
- b. The initial three digits of a ZIP Code for all such geographic units containing 20,000 or fewer people are changed to 000.

·3. All elements of dates (except year) for dates directly related to an individual, including birth date, admission date, discharge date, date of death; and all ages over 89 and all elements of dates (including year) indicative of such age, except that such ages and elements may be aggregated into a single category of age 90 or older.

·4. Telephone numbers.

·5. Facsimile numbers.

·6. Electronic mail addresses.

·7. Social security numbers.

·8. Medical record numbers.

.

·9. Health plan beneficiary numbers.

·10. Account numbers.

·11. Certificate/license numbers.

·12. Vehicle identifiers and serial numbers, including license plate numbers.

·13. Device identifiers and serial numbers.

·14. Web universal resource locators (URLs).

·15. Internet protocol (IP) address numbers.

·16. Biometric identifiers, including fingerprints and voiceprints.

·17. Full-face photographic images and any comparable images.

·18. Any other unique identifying number, characteristic, or code, unless otherwise permitted by the Privacy Rule for re-identification.; and

·19 and (ii) The covered entity does not have actual knowledge that the information could be used alone or in combination with other information to identify an individual who is a subject of the information.

.

Why this discussion???

- **Because it is best to read and understand what the regulations say themselves….**
  - ❖ And what they don't say

- **What HIPAA was NOT requiring was that this info had to be encrypted (vs. other data)**
  - ❖ More than one customer assumed names had to be encrypted….and paid the price on searches

- **What HIPAA was establishing was the rules for _*de-identification*_ (data masking) for public access**
  - ❖ In other words, there was no issue with valid users of the system from seeing the data
  - ❖ The intent was to prevent unauthorized disclosure of personal information when data was being de-identified for public disclosure
    - ✓ E.g. avoid repeating the internet CEO's 2014 controversy with 'distressed babies' on earnings call
      - – Sooooo…..who violated HIPAA - the CEO who made it public? ….or the HR person who give him the details?
      - – Remember, HIPAA said the violation is if someone could _*statistically*_ figure out who is being referred to…..not just saying it was "Bob"
  - ❖ Obfuscation might not be enough - sometimes proper masking of the data requires aggregation to a level that an sensitive data can not be statistically determined
    - ✓ E.g. if HR told CEO only that $## million was spent on healthcare broken down into a single level of categories such as Cancer, Diabetes, Auto Immune, Emergency Health, Mental Health, Maternity/OB, etc.

Encrypting Data in SAP IQ

·Column Encryption

- Requires using AES_ENCRYPT() and AES_DECRYPT() functions (not transparent as with ASE)
  - ✓Column has to be varchar or varbinary and length >32
  - ✓DATALENGTH(ciphertext) = (((DATALENGTH(plaintext)+ 15) / 16) + 1) * 16
- Uses ENCRYPTED keyword for column spec in LOAD TABLE command
- AES-128 bit (FIPS-197)
- Unlike ASE, there are not “Key” first class objects (this is similar to MSSQL)
  - ✓App needs to manage the keys to avoid having users lose them (or forget them) and to ensure the same key is used universally for the same column(s)
  - ✓It is possible to have different rows of data encrypted with different keys, but think about the SELECT impacts
    - – You would need a case statement and some way of knowing which key to use with which rows
  - ✓Nothing comparable to SALT/IV, but remember, IQ storage is different
    - – Values are stored a single time + compression on column removes repeating values (reduces effectiveness of statistical attacks)
    - – To be totally safe - use full database encryption with column encryption

·Full database encryption

- Implemented at create database time
  - ✓CREATE DATABASE … ENCRYPTED…. Option
  - ✓AES (128-bit), AES 256-bit or AES-256 FIPS algorithms
- No explicit decrypt syntax - e.g. no ALTER DATABASE syntax
  - ✓You will need to extract the data, recreate the database and reload the data

A word on SRS

- **SRS doesn't inherently have any data encryption for data storage**
  - However, data is transient, so exposure risk is smaller than with ASE - but still a potential problem
  - If data encryption is required, best to have SRS use encrypted file systems or other encrypted devices via OS or other software encryption outside of SRS (e.g. Vormetric)
    - ✓ The combination of an encrypted file system + SSL should meet most security requirements as both data at rest as well as data in motion is protected
    - ✓ Only issue is that other than for encrypted columns, SRS DBA's could still gain access to the data via dumps of the queues (sysadmin dump_queue) or SRS tracing

- **SRS replicates ciphertext for encrypted columns**
  - But only for encrypted columns - full database encryption would be still in the clear
  - This only works between ASE databases with the same keys
    - ✓ This can be a bit of an issue as there is no way to replicate encrypted cols in decrypted form
    - ✓ In addition, if you change the keys, the keys have to be changed in all locations simultaneously with the queues drained.
  - Replicating data from encrypted columns to non-ASE databases (e.g. IQ) or to ASE's with different encryption keys is ***not*** supported.

Using SRS as a data obfuscator

- **There are offline/batch data obfuscation tools**
  - ❖ Generally focused on providing test data for application development
  - ❖ Typically requires a bit of setup to maintain primary/foreign key relations

- **Regulations may require public disclosure**
  - ❖ European Securities and Market Authority Regulatory technical and implementing standards – Annex I (MiFID II / MiFIR)
    - ✓ RTS 2 Annex II: Details of transactions to be made available to the public
  - ❖ This is an example of a near constant public data disclosure that data obfuscation tools don’t work well with

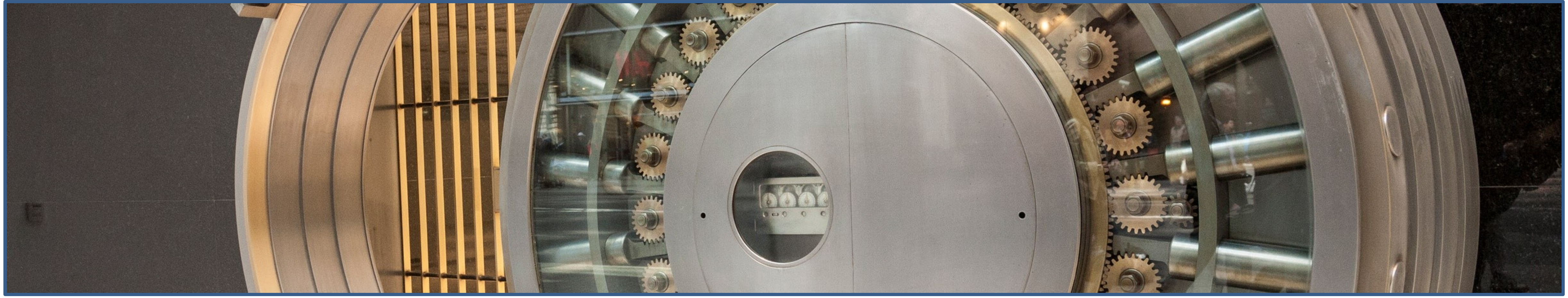
- **SRS has been used in the past where requirements are more real-time**
  - ❖ E.g. to provide public access to court data or other mandated data access requirements with implied data currency being maintained.
  - ❖ Leverages:
    - ✓ Ability of SRS to exclude sensitive columns from replication
    - ✓ Exploit fstrings to truncate zip codes, last 4 of social, etc.
    - ✓ Exploit fstrings to provide lookup/translation tables to replace or one way hash sensitive data elements
      - – Replace patient ID/SSN with SHA/MD5 hashed form or fully obfuscated with ‘####’
    - ✓ Exploit request functions allow public modifications without storing data in public system
  - ❖ ETL tools could also work

esma European Securities and Markets Authority

Regulatory technical and implementing standards – Annex I

MiFID II / MiFIR

28 September 2015 | ESMA/2015/1464



DBA’s & Privileged Users

- Predicated Privileges, Roles, etc.

Regulations, Standards & Privileged Users

- **Nearly all, if not all, regulations have rules around privileged user access**
  - ❖ Many have to do with auditing what they do….
  - ❖ ….or ensuring the security of their network or validating their credentials
  - ❖ Most specify the separation of duties (and preventing “super users” with all privileges)
  - ❖ Some require that the administrators can not view the data

- **An oft neglected aspect is….**
  - ❖ the aspect that automated maintenance tools often use privileged user accounts
  - ❖ Other users often know the tool credentials and could use them to avoid the audit traceability of commands outside the scope of the tools normal operations
  - ❖ …a quick audit failure if they look for this

DBA, SSO & sa accounts

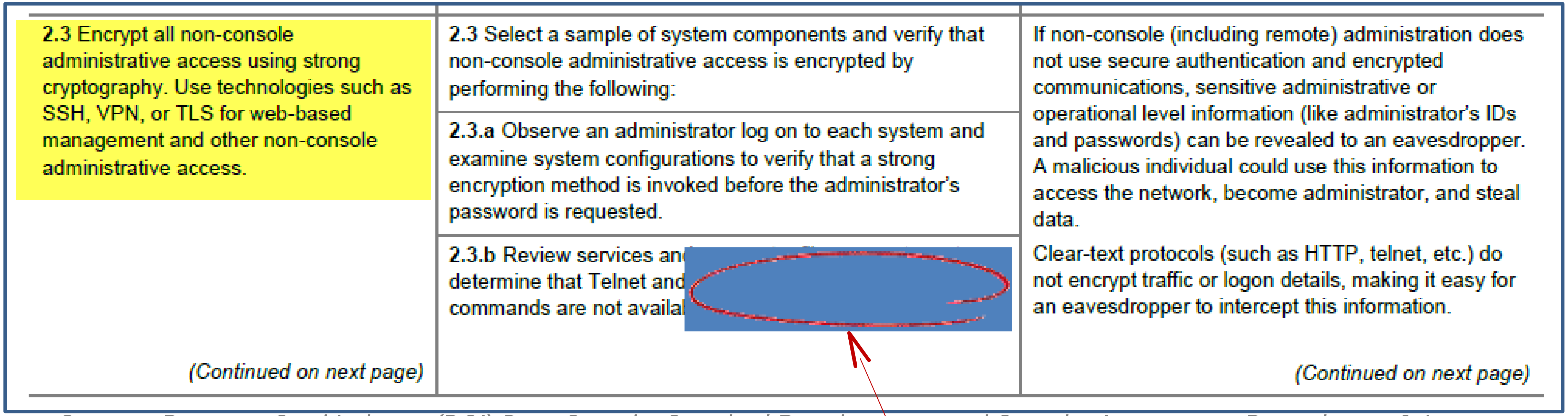
- **Lock the ‘sa’ account**
  - You will need to unlock it for certain upgrades, so ….
  - When sso unlocks the account, they can also reset the password in case you forgot it

- **Encrypt non-console network communications**
  - E.g. DBA (sa & sso) tools need to support SSL

- **Use granular permissions and roles to create tiers as well as support for automated processes**

- **DBA & SSO accounts**
  - Joe DBA and Mary SSO may be both application users as well as privileged system admins
  - Create separate accounts - joe_dba, mary_sso
  - Whether these accounts need to use LDAP or not is a up to the site/compliance adherence
    - ✓ Advantages
      - DBA’s can use a common password across systems administering
      - 2FA and other more secure login features can be implemented for privileged account access
      - No need to set ASE password complexity rules
    - ✓ Disadvantage
      - If external authentication mechanism fails, you will need to reboot ASE with -p switch to allow sso to connect
    - ✓ You may want to leave at least one SSO account that doesn’t use it so that an SSO can connect and rescue the system if all LDAP servers are down…..or to fix LDAP URL location changes, etc.

Another PCI DSS Example



2.3 Encrypt all non-console administrative access using strong cryptography. Use technologies such as SSH, VPN, or TLS for web-based management and other non-console administrative access.	2.3 Select a sample of system components and verify that non-console administrative access is encrypted by performing the following:	If non-console (including remote) administration does not use secure authentication and encrypted communications, sensitive administrative or operational level information (like administrator's IDs and passwords) can be revealed to an eavesdropper. A malicious individual could use this information to access the network, become administrator, and steal data.
	2.3.a Observe an administrator log on to each system and examine system configurations to verify that a strong encryption method is invoked before the administrator's password is requested.	
	2.3.b Review services and [redacted] determine that Telnet and [redacted] commands are not availa[redacted]	Clear-text protocols (such as HTTP, telnet, etc.) do not encrypt traffic or logon details, making it easy for an eavesdropper to intercept this information.
(Continued on next page)		*(Continued on next page)*

Source: Payment Card Industry (PCI) Data Security Standard Requirements and Security Assessment Procedures v3.1

In other words, no non-SSL TDS ports (except localhost)

Think about it - DBA may be changing passwords, setting system security keys, etc - via remote terminal sessions.

Granular Permissions (ASE 15.7 ESD #2)

·**Normally ASE has a few defined system roles**

- sa_role, sso_role, oper_role

·**Problem**

- Some sites need to restrict actions of junior DBA’s, outsourced DBA’s or 3rd party apps

·**Granular Permissions**

- Provides ~50 DBA actions as separate grantable options
- Essentially makes sa_role more limited
    - ✓You will need to grant ‘sa’ permissions to do things that previously it could with sa_role
- Intent is that ‘sa’ would be only user with sa_role/sso_role
    - ✓You would then create multiple levels of sa/sso roles and grant as needed
        - Backup_role, recovery_role, dbcc_role
        - Manage_logins_role
        - hw_resource_role (e.g. ability to change caches, add/remove engines from thread pools, etc.)
    - ✓You could also grant dbcc and other permissions to schema owners
        –

sa_role permissions (by default)

- **allow exceptional login**
- **checkpoint any database**
- **connect**
- **create database**
- **dbcc checkalloc any database**
- **dbcc checkcatalog any database**
- **dbcc checkdb any database**
- **dbcc checkindex any database**
- **dbcc checkstorage any database**
- **dbcc checktable any database**
- **dbcc checkverify any database**
- **dbcc fix_text any database**
- **dbcc indexalloc any database**
- **dbcc reindex any database**
- **dbcc tablealloc any database**
- **dbcc textalloc any database**
- **dbcc tune**

- **dump any database**
- **kill any process**
- **load any database**
- **manage any database**
- **manage any ESP**
- **manage any execution class**
- **manage any thread pool**
- **manage cluster**
- **manage data cache**
- **manage disk**
- **manage dump configuration**
- **manage lock promotion threshold**
- **manage resource limit**
- **manage server**
- **manage server configuration**
- **manage server permissions**

map external file

mount any database

online any database

own any database

quiesce any database

select on get_appcontext

select on list_appcontext

select on rm_appcontext

select on set_appcontext

set switch

set tracing any process

show switch

Shutdown

unmount any database

Other permissions (by default)

·Sso_role

- alter any object owner (in any database)
- change password
- decrypt any table (in any database)
- manage any encryption key (in any database)
- manage any login
- manage any login profile
- manage any remote login
- manage any user (in any database)
- manage auditing
- manage roles
- manage security configuration
- manage security permissions
- select on authmech
- show switch
- set tracing any process
- update any security catalog (in any database)

·oper_role

- checkpoint any database
- dump any database
- load any database
- manage dump configuration
- online any database
- use any database

·Key custodian

- manage any encryption key

replication_role

- checkpoint any database
- dump any database
- load any database
- manage replication (in any database)
- monitor server replication
- online any database
- quiesce any database
- truncate any table (in any database)
- truncate any audit table (in sybsecurity)
- (set ciphertext on)
- (set triggers off)
- (set disable_ri_check on)
- (set dml_on_computed on)
- (set replication off)

Automated processes & tiered DBA/SSO’s (1)

DBA Role	Granular permissions			
DB Administration Roles				
dbcc_role	dbcc checkalloc any database dbcc checkcatalog any database dbcc checkdb any database dbcc checkindex any database	dbcc checkstorage any database dbcc checktable any database dbcc checkverify any database dbcc fix_text any database	dbcc indexalloc any database dbcc reindex any database dbcc tablealloc any database dbcc textalloc any database	manage checkstorage report checkstorage
pnt_role	dbcc tune kill any process checkpoint any database set switch	show switch manage any execution class manage any statistics reorg any table	manage any thread pool manage disk manage lock promotion threshold manage resource limit	manage server monitor qp performance manage abstract plans select any system catalog
backup_role	checkpoint any database	dump any database	quiesce any database	manage dump configuration
recovery_role	checkpoint any database quiesce any database mount any database	load any database online any database create database	manage any database unmount any database	identity_insert any table identity_update any table
dbmaint_role	manage any statistics	reorg any table	select any system catalog	
svradm_role	allow exceptional login connect dbcc tune kill any process manage any ESP	manage any thread pool manage cluster manage disk manage server manage server configuration	set switch show switch shutdown map external file	select on get_appcontext select on list_appcontext select on rm_appcontext select on set_appcontext
DB Security Roles				
useradm_role	change password manage any login manage any login profile	manage any remote login manage roles	manage any user (in any database) update any security catalog (in any database)	show switch set tracing any process
ssoadm_role	manage security permissions manage auditing	manage security configuration alter any object owner (in any database)	decrypt any table (in any database) manage any encryption key (in any database)	select on authmech
devsec_role	manage any object permission	manage any user	manage database permissions	
devtest_role	setuser			

Automated processes & tiered DBA/SSO's (2)

DBA Role	dbcc_role	pnt_role	mon_role	backup_role	recovery_role	dbmaint_role	svradm_role	sybase_ts_role	useradm_role	ssoadm_role	devsec_role	devtest_role
junior_dba	s		s	s		s						
senior_dba												
performance_dba		s	s					s				
access_sec_sso												
system_sec_sso									s	s	s	
app_developer (in dev/test)												

Working with granular permissions and proxy authorization

- **What you might thing works**
  - In mistaken theory one might assume now that someone with appserver_role can't assume the identity of netapp_backup and thereby perform a database dump.
- **But…..**
  - When you proxy to another user, all roles of previous user are deactivated - so the exclusive activation check misses
  - Since myappserver login doesn't actually have backup_role, the exclusive membership check misses as well
  - Result, your appserver can do database dumps…..arrrrghhhhh

```
create role backup_role

-- this idea doesn't work…
alter role backup_role
     add exclusive membership appserver_role
alter role backup_role
     add exclusive activation appserver_role
alter role backup_role
     add exclusive membership sa_role
alter role backup_role
     add exclusive activation sa_role

grant dump any database to backup_role
grant checkpoint any database to backup_role
grant quiesce any database to backup_role

create login netapp_backup
     with password SuperSecret123
          default database tempdb

grant role backup_role to netapp_backup
alter login netapp_backup
     add auto activated roles backup_role
```

The real solution

- **Fix the role**
  - Disconnect appserver logins
  - Revoke & regrant the proxy authorization with list of granular permission roles

```
revoke set proxy from appserver_role

grant set proxy to appserver_role
   restrict role system, backup_role, …
```

- **Tips…..**
  - Keep the list of roles reasonable
  - If using tiers of roles, the level 1 (bottom) roles are the ones to list

A neat trick for replication server (or ETL tools)

- **Problem**
  - Maintaining permissions for maintenance user on all the tables is a nightmare

- **Old school solution**
  - Alias maint_user to ‘dbo’
  - But then maint user can do some things really not too desirable
    - ✓ Like granting others permissions

- **Secure solution**
  - Granular permissions

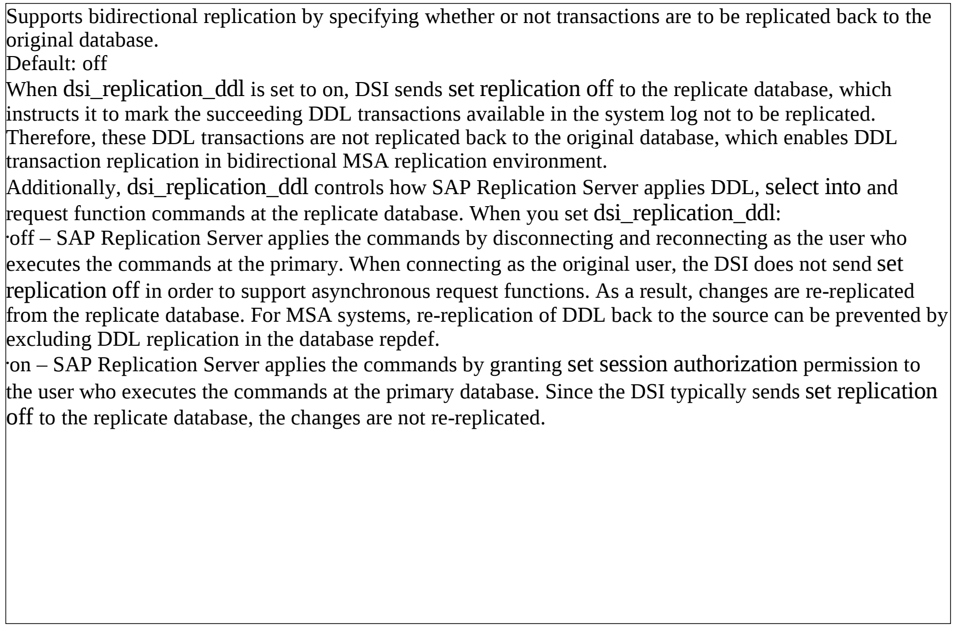
```
-- needs granular permissions enabled
grant delete any table to replication_role
grant execute any function to replication_role
grant execute any procedure to replication_role
grant identity_insert any table to replication_role
grant identity_update any table to replication_role
grant insert any table to replication_role
grant select any system catalog to replication_role
grant select any table to replication_role
grant truncate any table to replication_role
grant update any table to replication_role
```

Question - how do we replicate DDL??
…..do we need to grant create any object, manage and permissions to replication_role ??
….and if so, doesn’t that undo all this work???

So, how do we replicate DDL

- **Answer**
  - Depends on the setting for dsi_replication_ddl
  - If “ON”, SRS simply uses *set proxy* to become the user and executes the DDL
    - ✓ Advantage: Password sync not required
    - ✓ Disadvantage: Maint user has proxy authorization
  - If “OFF”, SRS disconnects as maint user and reconnects as original user from source
    - ✓ Advantage: Maint user doesn’t need proxy authorization
    - ✓ Disadvantage: password synchronization required
    - ✓ Problem: DDL might be re-replicated in bidirectional systems
      - You will need to turn off DDL replication in repdefs

- **This is where the trick of restricting roles becomes handy**
  - Grant set proxy but restrict the entire list of granular permissions as well as other roles
  - Problems:
    - ✓ If replicating encryption keys, keycustodian_role may be necessary
    - ✓ If replicating master, sso_role may be necessary
    - ✓ Yet another good reason to use LDAP (avoid password sync)



Supports bidirectional replication by specifying whether or not transactions are to be replicated back to the original database.

Default: off

When dsi_replication_ddl is set to on, DSI sends set replication off to the replicate database, which instructs it to mark the succeeding DDL transactions available in the system log not to be replicated. Therefore, these DDL transactions are not replicated back to the original database, which enables DDL transaction replication in bidirectional MSA replication environment.

Additionally, dsi_replication_ddl controls how SAP Replication Server applies DDL, select into and request function commands at the replicate database. When you set dsi_replication_ddl:

- off – SAP Replication Server applies the commands by disconnecting and reconnecting as the user who executes the commands at the primary. When connecting as the original user, the DSI does not send set replication off in order to support asynchronous request functions. As a result, changes are re-replicated from the replicate database. For MSA systems, re-replication of DDL back to the source can be prevented by excluding DDL replication in the database repdef.
- on – SAP Replication Server applies the commands by granting set session authorization permission to the user who executes the commands at the primary database. Since the DSI typically sends set replication off to the replicate database, the changes are not re-replicated.

```
revoke set proxy from replication_role

grant set proxy to replication_role
   restrict role sa_role, sso_role,
         mon_role, backup_role, …
```

IQ & System Privileges

- **IQ natively supports a ‘granular permissions’ scheme**
  - A lot of distinct system privileges (see subsequent slides)

- **Impersonation (Proxy authorization) is a bit different**
  - User needs ‘set user’ system privilege
  - User then can execute ‘setuser’ command
  - Although it can be restricted from admin roles, the best syntax is to grant permission only on application roles
  - All impersonations are audited automatically
  -

```
GRANT SET USER ( <target_users_list>
| ANY
| ANY WITH ROLES <target_roles_list> )
TO <userID [,...]>
[ WITH ADMIN [ ONLY ] OPTION | WITH NO
ADMIN OPTION ]
```

```
GRANT SET USER
(ANY WITH ROLES <target_roles_list>)
TO <user_ID [,...]>

GRANT SET USER (ANY WITH ROLES Marketing1,
Marketing2) TO MarketingAppRole
```

IQ Granular System Privileges (1)

ACCESS SERVER LS
ALTER ANY INDEX
ALTER ANY MATERIALIZED VIEW
ALTER ANY OBJECT
ALTER ANY OBJECT OWNER
ALTER ANY PROCEDURE
ALTER ANY SEQUENCE
ALTER ANY TABLE
ALTER ANY TEXT CONFIGURATION
ALTER ANY TRIGGER
ALTER ANY VIEW
ALTER DATABASE
ALTER DATATYPE
BACKUP DATABASE
CHANGE PASSWORD
CHECKPOINT
COMMENT ANY OBJECT
CREATE ANY INDEX
CREATE ANY MATERIALIZED VIEW
CREATE ANY OBJECT
CREATE ANY PROCEDURE
CREATE ANY SEQUENCE
CREATE ANY TABLE
CREATE ANY TEXT CONFIGURATION
CREATE ANY TRIGGER
CREATE ANY VIEW
CREATE DATATYPE
CREATE EXTERNAL REFERENCE
CREATE MATERIALIZED VIEW
CREATE MESSAGE
CREATE PROCEDURE
CREATE PROXY TABLE
CREATE TABLE
CREATE TEXT CONFIGURATION
CREATE VIEW
DEBUG ANY PROCEDURE
DELETE ANY TABLE
DROP ANY INDEX
DROP ANY MATERIALIZED VIEW
DROP ANY OBJECT
DROP ANY PROCEDURE
DROP ANY SEQUENCE
DROP ANY TABLE
DROP ANY TEXT CONFIGURATION
DROP ANY VIEW
DROP CONNECTION
DROP DATATYPE
DROP MESSAGE
EXECUTE ANY PROCEDURE
LOAD ANY TABLE
INSERT ANY TABLE
MANAGE ANY DBSPACE
MANAGE ANY EVENT
MANAGE ANY EXTERNAL ENVIRONMENT
MANAGE ANY EXTERNAL OBJECT
MANAGE ANY LDAP SERVER
MANAGE ANY LOGIN POLICY
MANAGE ANY MIRROR SERVER
MANAGE ANY OBJECT PRIVILEGES
MANAGE ANY SPATIAL OBJECT

IQ Granular System Privileges (2)

MANAGE ANY STATISTICS
MANAGE ANY USER
MANAGE ANY WEB SERVICE
MANAGE AUDITING
MANAGE MULTIPLEX
MANAGE PROFILING
MANAGE REPLICATION
MANAGE ROLES
MONITOR
READ CLIENT FILE
READ FILE
REORGANIZE ANY OBJECT
SELECT ANY TABLE
SERVER OPERATOR
SET ANY PUBLIC OPTION
SET ANY SECURITY OPTION
SET ANY SYSTEM OPTION
SET ANY USER DEFINED OPTION
SET USER
TRUNCATE ANY TABLE
UPDATE ANY TABLE
UPGRADE ROLE
USE ANY SEQUENCE
VALIDATE ANY OBJECT
WRITE CLIENT FILE
WRITE FILE

```
-- to get a list of privileges for a role
CALL sp_displayroles( 'SYS_SPATIAL_ADMIN_ROLE' );
```

IQ & Roles

·Similar to ASE with some differences

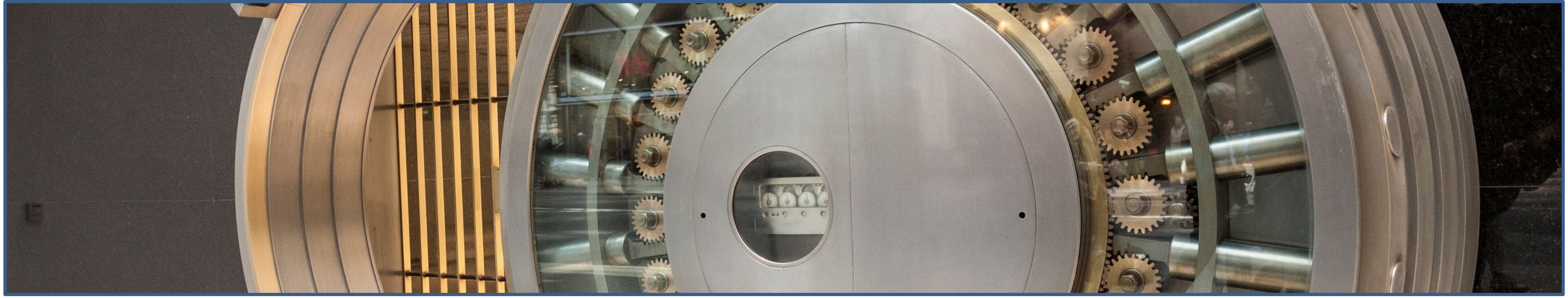
- All roles are automatically enabled
- No role password capability
- Unlike ASE, IQ already defines a lot of system roles that help implement tiered DBA access
  - ✓E.g. predefined backup and user admin roles, etc.

·Proxy authorization is a bit different

- User needs ‘set user’ system privilege
- User then can execute ‘setuser’ command
- Although it can be restricted from admin roles, the best syntax is to grant permission only on application roles

```
GRANT ROLE role_name [, …]
TO <grantee [, …]>
[ {WITH NO ADMIN | WITH ADMIN [ ONLY ] } OPTION ]
[ WITH NO SYSTEM PRIVILEGE INHERITANCE ]
```

IQ System Roles	
dbo	SYS_AUTH_WRITEFILE_ROLE
diagnostics	SYS_AUTH_WRITEFILECLIENT_ROLE
PUBLIC	SYS_AUTH_READFILE_ROLE
rs_systabgroup	SYS_AUTH_READFILECLIENT_ROLE
SA_DEBUG	SYS_AUTH_PROFILE_ROLE
SYS	SYS_AUTH_USER_ADMIN_ROLE
SYS_AUTH_SA_ROLE	SYS_AUTH_SPACE_ADMIN_ROLE
SYS_AUTH_SSO_ROLE	SYS_AUTH_MULTIPLEX_ADMIN_ROLE
SYS_AUTH_DBA_ROLE	SYS_AUTH_OPERATOR_ROLE
SYS_AUTH_RESOURCE_ROLE	SYS_AUTH_PERMS_ADMIN_ROLE
SYS_AUTH_BACKUP_ROLE	SYS_REPLICATE_ADMIN_ROLE
SYS_AUTH_VALIDATE_ROLE	SYS_RUN_REPLICATE_ROLE
	SYS_SPATIAL_ADMIN_ROLE



Auditing

- Setting up auditing, common auditing requirements, etc.

Regulations, Standards & Auditing

·**Most of the auditing requirements center on….**

- Auditing privileged user commands/history
- Auditing configuration changes
- Auditing security anomalies - such as failed logins
- Business auditing around actions by corporate executives, etc.

·**Most don’t specify auditing as a general practice for normal operations…**

- …except they often state that the capability needs to exist in case security decides to audit a suspected user behavior

There is a big distinction

- **Regulatory auditing generally falls into three categories**

- **1) Business auditing/controls**
  - E.g. stock sales by principals or company officers, employee discount sales, etc.
  - Generally, this type of auditing has to be done by the application

- **2) Administrative actions auditing/controls**
  - Normal administrative actions due to privileged access

- **3) Early threat detection**
  - Failed logins, failed access to data as early indications of security holes/exposures

- **Understand that the best DBMS auditing can only cover the latter 2…**

- **…further understand that auditing normal activity by end users via DBMS auditing is *NOT* a good idea**
  - The volume of information easily overwhelms any security review capability
  - As a result, some requirements such as tracking who modified a record likely are best implemented in the application as well (which includes the table schema with last_updated_by fields, etc.)
    - ✓ Rationale is that often this data needs to be visible to end-users so they can call and discuss data maintenance issues - vs. sso only accessible audit trails

Setting up auditing - correctly (1)

·Create the database devices

- You will need to have n+1 devices where “n” is the number of audit tables you wish to have
- The minimum number of audit tables recommended is 3 …the maximum is 8

·Adding space later ….can be fun

- You can’t simply extend by adding another device (except the log)
- You need to extend the existing devices via disk resize command
- As a result, the BEST devices are likely file system devices (but make sure using DIRECTIO)
- This is why I do this manually vs. auditinit with resource file

```
use master
go
disk init
     name = 'audit_data01',
     physname = 'D:\Servers\JT1\devices\audit_data01.dat',
     skip_alloc = true,
     size = '128M', dsync = false, directio = true
go

…

disk init
     name = 'audit_data04',
     physname = 'D:\Servers\JT1\devices\audit_data04.dat',
     size = '128M', dsync = false, directio = true
go

disk init
     name = 'audit_log01',
     physname = 'D:\Servers\JT1\devices\audit_log01.dat',
     size = '128M', dsync = false, directio = true
go

create database sybsecurity
     on audit_data01=128,
        audit_data02=128,
        audit_data03=128,
        audit_data04=128
     log on audit_log01=128
go
```

Setting up auditing - correctly (2)

- **Install sybsecurity**
  - $SYBASE/ASE/scripts
  - Shutdown & restart ASE

- **Configure the audit tables**
  - Sp_addaudittable for each table/device above the first device

- **Configure the audit options**
  - Set the queue size fairly big - this is an in-memory queue
  - Should by minimally 10x number of connections

- **Verify setup with sp_helpsegment**
  - Sp_helpsegment ‘aud_seg_0*n*’
  - Each segment should have one table

```
use sybsecurity
go

exec sp_addaudittable 'audit_data02'
exec sp_addaudittable 'audit_data03'
exec sp_addaudittable 'audit_data04'
go

use master
go

exec sp_configure 'audit queue size', 5000
exec sp_configure 'auditing', 1
exec sp_configure 'suspend audit when device full', 0
exec sp_configure 'current audit table', 1, 'with truncate'
go

exec sp_dboption sybsecurity, 'trunc log on chkpt', true
go
```

Setting up auditing - correctly (3)

·Create a log threshold action

- …if not using ‘truncate log on checkpoint’
- By default it will already be sp_thresholdaction on the LCT for the log….so if you have the proc already….

·Create a space threshold action

- ASE doesn’t automatically move from one audit table to the next
- When the current table/segment fills, it needs to fire a threshold
- Threshold action should be to set the ‘current audit table’ to 0 □which means the next audit table in the list
  - ✓Use the ‘with truncate’ option

```
create or replace procedure sp_audit_thresholdaction
             @dbname varchar(30), @segment_name  varchar(30),
             @space_left   int, @status    int
as begin
  declare @devsize int, @dev_pct numeric(5,1)
   select @devsize=size
     from syssegments s, master..sysusages u
     where s.name='aud_seg_02'
       and u.dbid=db_id('sybsecurity')
       and u.segmap & s.segment = s.segment
   select @dev_pct=(@devsize-@space_left)*100.0/@devsize
   print 'segment ''%1!'' in database ''%2!'' crossed space threshold'
             @segment_name, @dbname
   print 'device size=%1!; space left=%2!, percent full=%3!',
             @devsize, @space_left, @dev_pct
   if @dbname='sybsecurity' and @dev_pct > 75.0
   begin
      exec sp_configure 'current audit table', 0, 'with truncate'
      print 'auditing moved to next audit table'
   end
end
go

exec sp_addthreshold 'sybsecurity', 'aud_seg_01', 8192, sp_audit_....
exec sp_addthreshold 'sybsecurity', 'aud_seg_02', 8192, sp_audit_....
exec sp_addthreshold 'sybsecurity', 'aud_seg_03', 8192, sp_audit_....
exec sp_addthreshold 'sybsecurity', 'aud_seg_04', 8192, sp_audit_....
go

exec sp_helpthreshold
go
```

What & Who to audit

·Privileged Users

- DBA’s, SSO’s, etc.
- Probably want just the command text

·Automated processes

- Login, logout
- Critical commands (e.g. dump tran)

·Normal users

- A lot depends on the security regulation
- Failed logins is a common requirement
- Auditing errors could be interesting for app debugging
- Try not to audit <u>normal</u> activity ….or you will rapidly fill the audit trail
  - ✓ ….and blow up proc cache (more on this later)
- If regulation requires login/logout auditing, you will have to really and finally break your bad habits of app servers logging in and out rapidly

·Hint:

- Checkout table 33 in Security Guide Section 10.2.5 (Auditing) for quick preconfigured auditing for common requirements

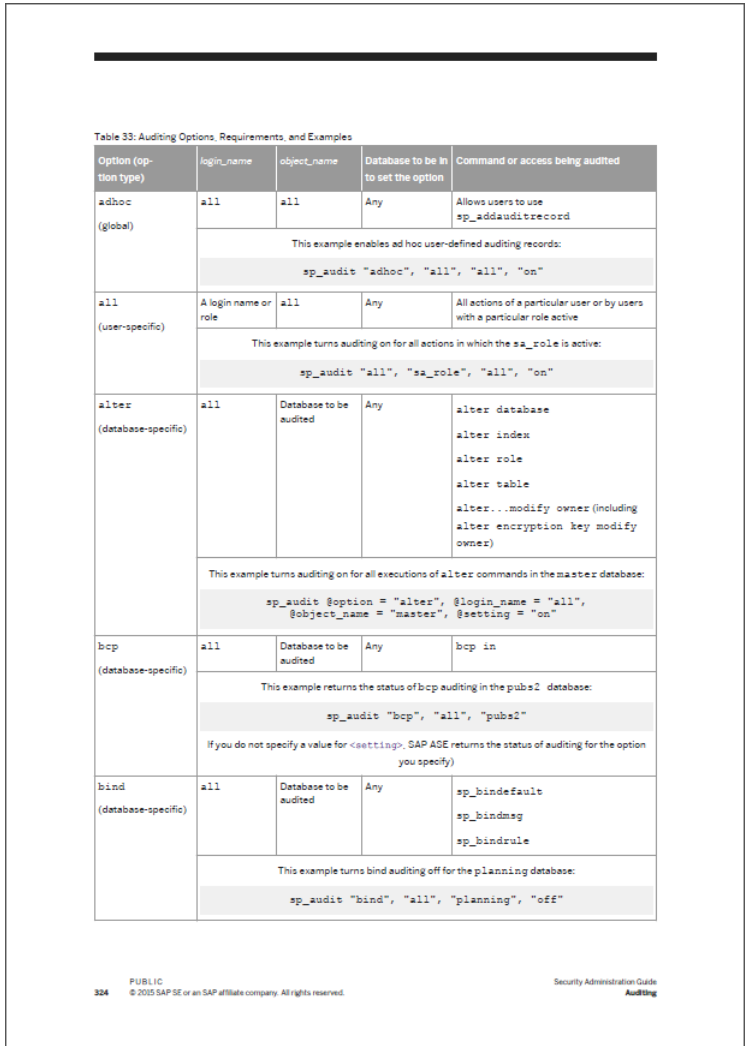


Table 33: Auditing Options, Requirements, and Examples

<table>
<tr><th>Option (option type)</th><th><i>login_name</i></th><th><i>object_name</i></th><th>Database to be in to set the option</th><th>Command or access being audited</th></tr>
<tr><td rowspan="2"><code>adhoc</code>
(global)</td><td><code>all</code></td><td><code>all</code></td><td>Any</td><td>Allows users to use <code>sp_addauditrecord</code></td></tr>
<tr><td colspan="4">This example enables ad hoc user-defined auditing records:
<code>sp_audit "adhoc", "all", "all", "on"</code></td></tr>
<tr><td rowspan="2"><code>all</code>
(user-specific)</td><td>A login name or role</td><td><code>all</code></td><td>Any</td><td>All actions of a particular user or by users with a particular role active</td></tr>
<tr><td colspan="4">This example turns auditing on for all actions in which the <code>sa_role</code> is active:
<code>sp_audit "all", "sa_role", "all", "on"</code></td></tr>
<tr><td rowspan="2"><code>alter</code>
(database-specific)</td><td><code>all</code></td><td>Database to be audited</td><td>Any</td><td><code>alter database</code>
<code>alter index</code>
<code>alter role</code>
<code>alter table</code>
<code>alter...modify owner</code> (including <code>alter encryption key modify owner</code>)</td></tr>
<tr><td colspan="4">This example turns auditing on for all executions of <code>alter</code> commands in the <code>master</code> database:
<code>sp_audit @option = "alter", @login_name = "all", @object_name = "master", @setting = "on"</code></td></tr>
<tr><td rowspan="2"><code>bcp</code>
(database-specific)</td><td><code>all</code></td><td>Database to be audited</td><td>Any</td><td><code>bcp in</code></td></tr>
<tr><td colspan="4">This example returns the status of <code>bcp</code> auditing in the <code>pubs2</code> database:
<code>sp_audit "bcp", "all", "pubs2"</code>
If you do not specify a value for <code>&lt;setting&gt;</code>, SAP ASE returns the status of auditing for the option you specify)</td></tr>
<tr><td rowspan="2"><code>bind</code>
(database-specific)</td><td><code>all</code></td><td>Database to be audited</td><td>Any</td><td><code>sp_bindefault</code>
<code>sp_bindmsg</code>
<code>sp_bindrule</code></td></tr>
<tr><td colspan="4">This example turns bind auditing off for the <code>planning</code> database:
<code>sp_audit "bind", "all", "planning", "off"</code></td></tr>
</table>

324 PUBLIC
© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Security Administration Guide
Auditing

Track Configuration Changes

- **SAP ASE version 16.0 adds ability to track the history of configuration changes made to the server**

- **Configuration changes recorded include**
  - Changes to the configuration File
  - Reading the configuration File
  - Changes to server options
  - Changes to database options
  - Changes to cache configuration
  - Changes to trace flags and switches
  - Changes to number of engines
  - SAP ASE startup and shutdown events
  - Enabling or disabling auditing

- **SAP ASE stores records of configuration changes in the sybsecurity database**

- **Enable configuration history tracking :**

```
sp_audit "config_history", "all", "all", "on"
```

Some other common ones

Failed logins

- sp_audit "login", "all", "all", "fail"

DDL

- sp_audit "create", "all", <dbname>, "on"
- sp_audit “drop", "all", <dbname>, "on"
- sp_audit “alter", "all", <dbname>, "on"

Errors

- sp_audit "errors", "all", "all", "on“

Command Text

- sp_audit “cmdtext”, <loginname>, “all”, “on”

Under penalty of death, do NOT do

·**Audit everything….**

- Sp_audit “all”, <loginname>, “all”, “on”

·**What this says:**

- Audit every command that touches every table/view/whatever….
- …including those commands inside a stored proc (which will be audited as well)
- …including those commands in a loop inside the stored proc
- With full text auditing, we will need to assimilate the full text for
  - ✓Each command in the proc
  - ✓For each command in the loops in the proc….

·**Why did you do this to begin with????**

- You probably meant **Y** sp_audit “cmdtext”, <loginname>

Integrating with 3rd party audit trails

- **sybsecurity is just another database**

- **The process**
  - create a login
  - add login to sybsecurity
  - grant select on any audit table
    - ✓This requires granular permissions enabled

- **What & When to poll**
  - Audit trail has an event datetime
  - Poll all the tables
  - Remember the latest datetime
  - Wait some time
  - Poll again where event datetime > last datetime

```
use master
go

create login audit_trail
     with password SuperSecret123
          exempt inactive lock true
go

create role audit_trail_role
go
grant role audit_trail_role to audit_trail
go
alter login audit_trail
   add auto activated roles audit_trail_role
go

use sybsecurity
go

grant select any audit table to audit_trail_role
go
sp_adduser audit_trail
go
```

Full Text Auditing

- Full-text audit information for *select* (*into*), *insert*, *delete* and *update, stored procs*
- Parameter names along with values are stored in extrainfo column of sysaudits

Audit event	**Audit records before ASE 16**	**Audit records in ASE 16**
insert with constants	sa_role sso_role oper_role sybase_ts_role mon_role; INSERT; ; ; ; ; sa/ase;	sa_role sso_role oper_role sybase_ts_role mon_role; insert mytab values(100, "audit"); ; ; ; ; sa/ase;
update with variables/parameters	sa_role sso_role oper_role sybase_ts_role mon_role; UPDATE; ; ; ; ; sa/ase;	sa_role sso_role oper_role sybase_ts_role mon_role; update mytab set c1 = @var3 where c1 = @var1 and c2 like @var2; ; ; @var3 = 500, @var1 = 100, @var2 = audit; ; sa/ase;
Insert with encrypted column. Encrypted columns are obfuscated in audit records.	sa_role sso_role oper_role sybase_ts_role mon_role; INSERT; ; ; ; ; sa/ase;	sa_role sso_role oper_role sybase_ts_role mon_role; insert mytab1 values(@var1, @var2); ; ; @var1 = ****** , @var2 = audit; ; sa/ase;

Audit Events

- No one location with full list of audit events

- Easiest method is to create your own decode table using logic similar to this

```
create table #tmp
( event_id     int,
  description  varchar(255)
)
go

declare @a int
select @a=1
while (@a<255)
begin
if audit_event_name(@a) is not null
   insert #tmp values (@a, audit_event_name(@a))
select @a=@a + 1
end
select * from #tmp
go

drop table #tmp
go
```

Granular Auditing

Future

- **ASE 16sp02 started granular auditing**
  - Granularity at login or role level for a very few auditing options
    - ✓“cmdtext”, “table_access” and “view_access” audit options support auditing at login level.

- **ASE 16sp03 will provide more full support**
  - All global audit options will have support for auditing at login and role level (see table)
  - “all” audit option will have support for auditing at user defined role level in addition to system defined roles and logins.

Granular Auditing	
adhoc	mount
cluster	network
config_history	password
dbcc	quiesce
disk	role
dump_config	role_locked
errorlog	rpc
errors	security
login	security_profile
login_admin	sproc_auth
login_locked	thread_pool
logout	unmount

Auditing & IQ

·Enable via a SET option

- set option AUDITING = [ON | OFF]
- Need to have SET ANY SECURITY OPTION or MANAGE AUDITING to enable
- Start IQ with -zr & -zo to create SQL log for auditing output

·Records audit events to the transaction log

- Which types of event to audit are set using the sa_enable_auditing_type system procedure
- Procedure sa_audit_string() adds any string to audit trail in transaction log

·Major Differences with ASE

- No separate repository - audit trail is in transaction log and SQL log
- No method to audit access to objects (e.g. tables/procs)
  - ✓ In authors mind, this is not a problem except for failed access as normal access would fill audit trail needlessly
- No method to audit the command text of privileged users

IQ Audit Options & Reviewing Audit Trail

IQ Audit Options	
all	enables all types of auditing.
connect	enables auditing of both successful and failed connection attempts
connectFailed	enables auditing of failed connection attempts
DDL	enables auditing of DDL statements
options	enables auditing of public options
permission	enables auditing of permission checks, user checks, and SETUSER statements.
permissionDenied	enables auditing of failed permission and user checks.
triggers	enables auditing after a trigger event.

- **Reviewing Audit Trail uses DBTRAN utility program**
  - Output from tran log is a .sql file
  - Remember for IQ, the log doesn’t contain modified data - just block info.
  - -g G adds auditing information to the transaction log if the auditing database option is turned on.
  - -d □ Specifies that transactions are written in order from earliest to latest. This feature is intended for auditing database activity: do not apply dbtran output against a database.

- **SQL log (-zr & -zo) contains connection auditing & select auditing**

A word about SAP Replication Server

·SRS also supports auditing

- SRS records all RCL commands when you enable command auditing, except:
  - ✓system information commands (admin who,etc)
  - ✓other cmds that you do not use to configure SRS
- Enabled via config:

```
· configure replication server
· set audit_enable to {on|off}
```

·Audit trail can be sent to errorlog or separate file

- Strongly recommend separate file
- Errorlog could expose audit cmds to 'sa' and sa can change the errorlog location as well

```
· configure replication server
· set audit_dest to ['log'|'<filename>']
```

SRS Config Param
audit_enable
audit_dest

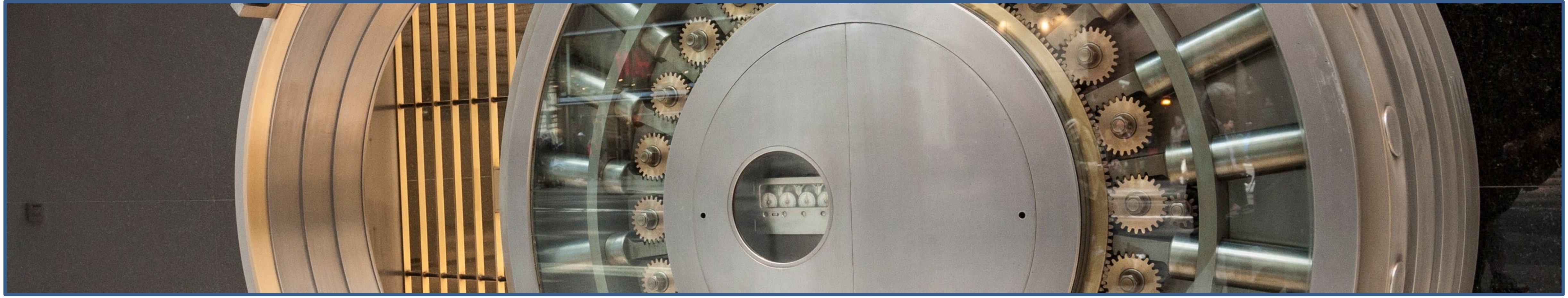
Exploiting SRS for Supporting Forensic Auditing

·Issue with built-in DBMS auditing is that it is hard to analyze over time

- Rapid space explosion requires extracting to a repository
- Problem is that full text auditing would need parsing for forensics on an audit on a single account or similar data element value
- However, it covers the broadest capabilities

·SRS has been used to track changes to a forensic repository

- Great use for replicating to IQ (and platform edition)
- Simple schema
  - ✓Table 1 □Transaction details (user, commit time, source system/database)
  - ✓Table 2 □ Table name & operation (insert/update/delete/exec proc, etc.)
  - ✓Table 3 □Table column information (column name modified, before value, after value)
- Alternative schema
  - ✓Same as normal schema except has tran_id, user, commit time, operation and after images only



Hot fixes, etc.

- Find a list of known vulnerabilities and fixes, etc.

Compliance, Patches and SAP Policy for ASE, IQ, SRS

·**Regulatory Compliance may force keeping system patched**

- Safe harbor provisions may only indemnify the organization if the system has been kept up with all relevant security patches
- In other words, if someone uses a security exploitation that is a year old and your system hasn’t been patched in 2 years, your company may be liable for damages

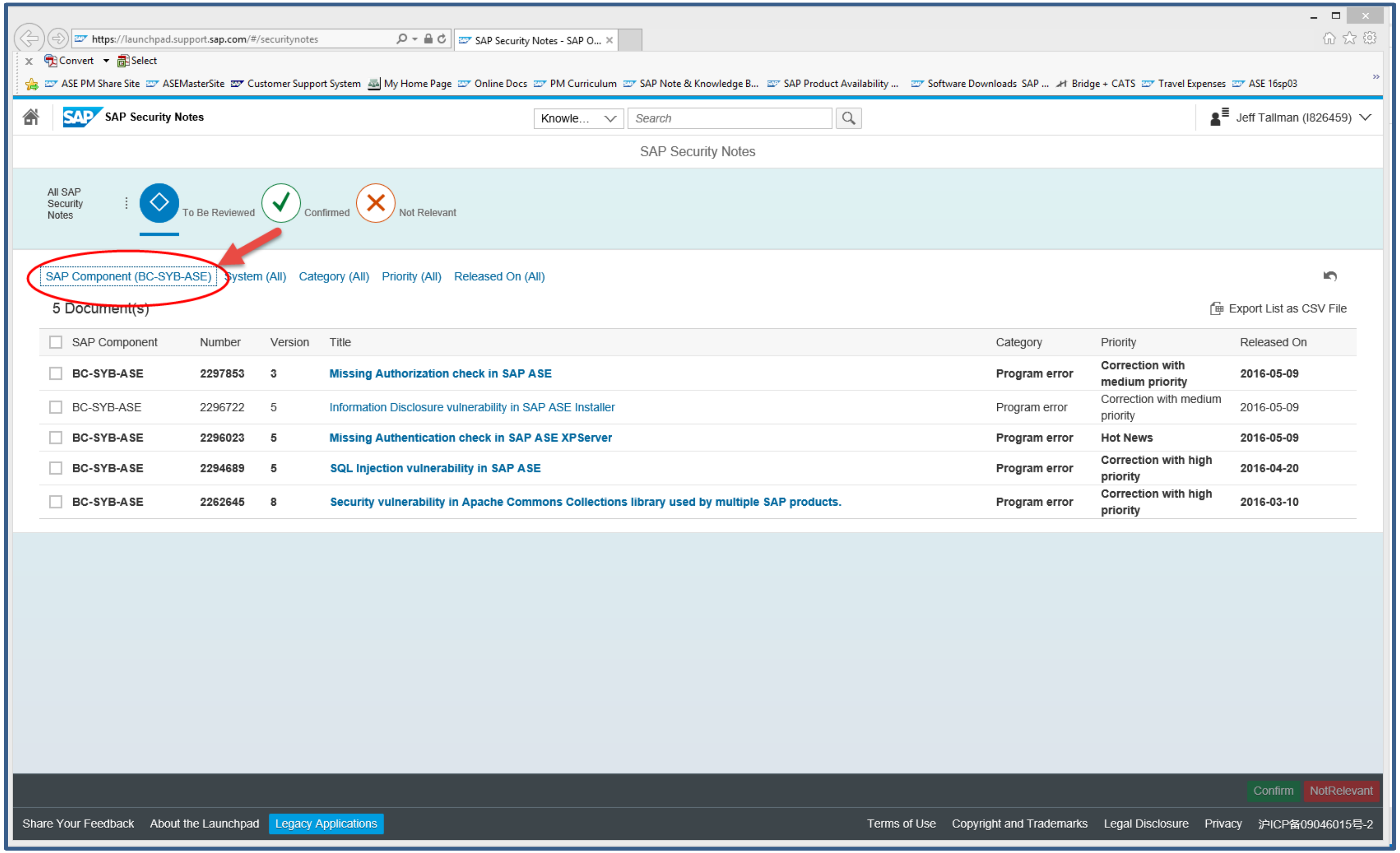
·**ASE Hot Fixes**

- Provides critical late breaking security or stability patches as interim to next SP/PL
- Only provided on the latest supported SP and PL until the next SP or PL is released
  - ✓E.g. if on 15.7, sp138 is latest - there will be a HF for sp138 but not for 137
  - ✓E.g. if on 16.0, sp02 pl04 is latest - there will be a HF for pl04 but not for pl03
- Fixes are rolled into the next SP/PL
  - ✓Which is why there are no HF on previous releases - we’d have to back port a lot of other security fixes

·**ASE is leveraging SAP’s ‘Security Notes’ to disseminate known vulnerabilities**

- Access via the support launchpad

Where to find list of vulnerabilities



SAP Security Notes

SAP Component (BC-SYB-ASE) System (All) Category (All) Priority (All) Released On (All)

5 Document(s)

Export List as CSV File

□ SAP Component	Number	Version	Title	Category	Priority	Released On
□ **BC-SYB-ASE**	**2297853**	**3**	**Missing Authorization check in SAP ASE**	**Program error**	**Correction with medium priority**	**2016-05-09**
□ BC-SYB-ASE	2296722	5	Information Disclosure vulnerability in SAP ASE Installer	Program error	Correction with medium priority	2016-05-09
□ **BC-SYB-ASE**	**2296023**	**5**	**Missing Authentication check in SAP ASE XPServer**	**Program error**	**Hot News**	**2016-05-09**
□ **BC-SYB-ASE**	**2294689**	**5**	**SQL Injection vulnerability in SAP ASE**	**Program error**	**Correction with high priority**	**2016-04-20**
□ **BC-SYB-ASE**	**2262645**	**8**	**Security vulnerability in Apache Commons Collections library used by multiple SAP products.**	**Program error**	**Correction with high priority**	**2016-03-10**

Questions???
Follow-up…
Discussion…
Clarifications…

UKSUG
SAP DATABASE & TECHNOLOGY USER GROUP

THANK YOU

For more information on SAP ASE 16 visit:

www.sap.com/ase

http://help.sap.com/ase1602/

https://ideas.sap.com/SAPASE