How To Build an Auditable Security Framework

Spokey Wheeler, MBCS CITP
IBM

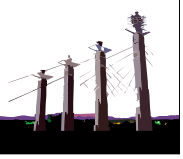
C03
Monday, April 28, 2008 • 1:00 P.M. – 2:00 P.M.

2008 IIUG Informix Conference

www.iiug.org

Agenda

- Before we start
- Some Thoughts on “Auditability”
- A List of Things to Consider
  - Reasonably but not completely comprehensive
- Presentation of the Audit Requirements
- Presentation of the Results
- Conclusion

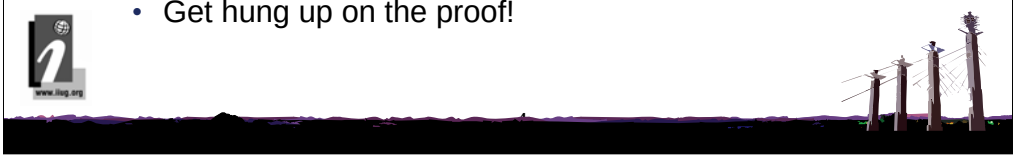


Before We Start

- This is NOT a technical presentation
- For the ins and outs of securing your databases,
  - Attend one of Jonathan Leffler's presentations:
    - Colloquially – The Paranoid DBA

Some Thoughts on “Auditability”

- Being “auditable” is different from being secure.
  - It means you have a quantifiable, measurable yardstick
  - Of how well your actual security
  - Matches the security you say you should have.
- Auditable security is not *necessarily* good security!
  - But auditable security is more useful to your business☺
- YOU define how secure you need to be
  - You need to do this before you try to audit it
- Don’t get hung up on the presentation
  - Get hung up on the proof!



Some Thoughts on “Auditability” (cont)

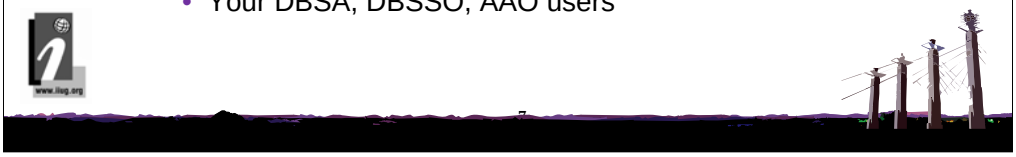
- **Define your goals carefully.**
  - **Be very specific.**
  - **Ensure they are measurable.**

A List of Things to Consider

- Platform
- Files
- User
- Network
- Application
- Database Access
- Role Separation
- Auditing
- Scripting
- Incident Detection
- Fixing it!
- Other Issues

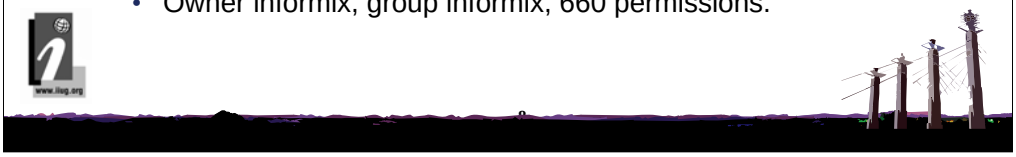
Sample: Auditable Platform Requirements

- Is physical access to the machine limited?
  - Who could pick it up and walk out with it?
- Define, very clearly, your password requirements,
  - Including length
  - Character sets required
  - Aging – or frequency of changes
- Consider implementing keystroke logging
  - For highly privileged users:
    - root
    - informix
    - Your DBSA, DBSSO, AAO users



Sample: Auditable File Requirements (cont.)

- Do your file permissions match the original ones?
  - Is any file or directory in $INFORMIXDIR publicly writeable?
- Have you removed files you do not need?
  - SNMP, ON-Perf, ISM, …
- Can anybody read shared memory dumps?
  - Do not use /tmp (but /tmp/informix might be OK)
- Have you removed permissions from the block special equivalents of your character special devices?
  - If /dev/dsk/c0t0d0s1 is readable or writable,
  - It doesn't matter that /dev/rdsk/c0t0d0s1 is not!
- Are the permissions on your chunk files correct?
  - Owner informix, group informix, 660 permissions.



Sample: Auditable File Requirements (cont.)

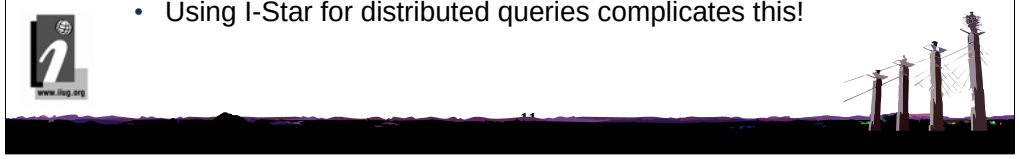
- If you back up to disk, who can read the backups?
- Who can read the CDR spool files, etc.?
- Are the key configuration files properly protected?
  - sqlhosts
  - onconfig
  - adtcfg
  - seccfg
- Who can read the audit configuration directory?
- Did your installation media come from a trusted site?

Sample: Auditable File Requirements (cont.)

- Who has access to user root?
  - Does the root password meet security requirements?
- Ditto for user informix.
- Who is a member of the DBSA group?
- Who is a member of the DBSSO group?
- Who is a member of the AAO group?
- Who is a member of the bargroup group?
- Have you implemented PAM?
- Are all inactive logins removed?

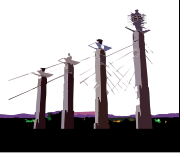
Sample: Auditable File Requirements (cont.)

- Does Enterprise Replication (ER) use encryption?
  - No unencrypted data over long-haul networks
- Is High-Availability Data Replication (HDR) safe?
  - Is it implemented on a separate subnet with no user access?
- Are client-server communications encrypted?
  - Have you allowed DES encryption? (Don't)
  - Have you allowed ECB mode? (Don't!)
- Does the server trust other hosts?
  - Avoid /etc/hosts.equiv
  - Suppress ~/.rhosts files
  - s=0 set in server sqlhosts file?
  - Using I-Star for distributed queries complicates this!



Sample: Auditable File Requirements (cont.)

- Has MAX_PDQPRIORITY been restricted?
- Is ODBC access to the data possible ONLY through approved stored procedures?
- Do authorized applications set roles?
- Are you exploiting default roles?



A Sample Definition of Auditable Database Access Requirements

- Have you modified seccfg?
  - To restrict the groups that can connect to the server.
- Is NODEFDAC set?
  - In $INFORMIXDIR/etc/informix.rc?
- Has DBCREATE_PERMISSION been set?
  - In the $ONCONFIG file
- Are all permissions granted without GRANT OPTION?
- Are any permissions granted to PUBLIC?
- Do you follow the Principle of Least Privilege?
  - Are the permissions granted to users and roles the absolute minimum needed for people to fulfill their jobs?

Sample: Auditable Role Separation Requirements

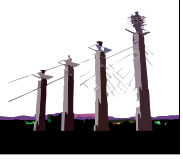
- Is auditing enabled?
- Has role separation been implemented?
  - Permissions on etc, aaodir, dbssodir.
- Are the permissions on the various directories correct?

Auditing

- Auditing has a performance cost.
  - Some things are easy to audit, others aren't.
  - However, some checks are only possible with auditing.
- Define the minimum audit requirements for all users
- Define rigorous audit requirements for key users:
  - root
  - informix
  - Your DBSSOs, DBSAs and AAOs
  - What about the DBAs?
- Specifically exclude high-cost actions unless vital!
  - Some (such as ACTB) are costly
  - Row-level auditing (RDRW, INRW, DLRW, UPRW) is worse!

Scripting

- You don't need to script if you don't want to!
- Many checks are easily scripted
  - Especially under UNIX.
- Some things are easier to inspect manually.
- Scripting under Windows is different from UNIX.
  - Cygwin helps hide Windows from the UNIX user
  - But be aware of the Windows registry and permissions
- There are publicly available scripts on the Internet
  - They will give you a head start,
  - But they might NOT match your requirements
    - They may also be difficult to tailor
- For your own absolute security, write your own scripts!



Incident Detection

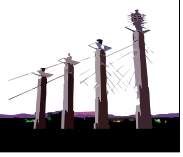
- There is no point in taking all these steps
  - Unless you regularly review the audit output
  - And ensure that no unacceptable incidents have occurred!
- Audit reviewers need to be trusted
  - Audit reviewers should be outside the DBA team if possible
- Review schema and configuration file changes too
  - Do you version control your configuration files?

Fixing It!

- There is no point identifying potential issues
  - If you don't fix them!
- Audit requirements should specify a 'time-to-fix'
  - So issues do not remain unaddressed for too long.
- When you have fixed something
  - Re-run the audit checking to prove it's fixed!

Other Issues

- How do you validate your security?
  - As opposed to your auditability?
  - Ethical hacking or penetration testing can be expensive
  - And can be very risky
- How can you validate security if you have outsourced?
- What happens to data on the wrong side of a firewall?
  - Your data is behind a firewall, isn't it?

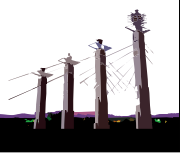


Presentation of the Audit Requirements

- Whatever works for YOU!
  - Can be as detailed or as simple as you want
- Describe why you chose the specific criteria
- Describe why issues have been ignored or excluded
  - Be careful – the auditors may feel differently! ☺
- The auditors may not know as much as you:
  - They may have less product-specific technical knowledge,
  - But they have probably seen more secure (and insecure) systems than you.

Presentation of the Audit Results

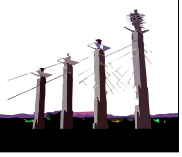
- Whatever works for YOU!
- Could be as flashy or as simple as you want
- Excel could be useful
  - Especially for graphing how close you are to meeting requirements
- Remember to explain why you can't meet an objective
  - And how you intend to fix it!



Sample Presentation of the Results

Physical Access

	Sales_db	Dw_db	Rep_db
Is it impossible to physically take the machine without being challenged?	▪	▪	▪
Have password requirements, including length, composition and aging been defined?	▪	▪	▪
Has keystroke logging been implemented for high-access users	▪	▪	▪
Is it impossible for someone to power down the server and gain high access on rebooting?	▪	▪	▪



Thank You!

- Any questions?

Session ####

How to Build an Auditable Security Framework

Spokey Wheeler
IBM
spokey.wheeler@uk.ibm.com