We Must Secure Data Not Software

Start Security in the Right Place
Quick Quiz!

• How many people here know “where” their important data is held?

• How many people here understand exactly “who” can see or “modify” important data?

• How many people here understand the true “privilege model” employed to protect “important data”? 
Agenda

• The problem space
• Solution design space
• Data security solutions from Oracle
• Data security solutions from third parties
• Wrapping up
Mind Control – (How You Must Think)

• We must secure the **data** not the Oracle software
  • Oracle is generic software
  • You build your own database structure/layout/design
  • You build your own applications (web, forms, Apex, Java...)
  • You must build your own security – oooops.!!!
  • Most often Security is not done well or forgotten in the rush to SLA’s, performance, Features....

• The old days were grant DBA, no audit, make it work
• Focus for years has been on hardening not securing
  • I was complicit in the creation of check lists of course
  • **These lists are still very valuable but not on their own**
Current State Of Affairs (of Data Security)

• In my experience not as bad as ten years ago
• But still not brilliant
• Most sites I perform audits at exhibit:
  • Weak passwords; I mean really weak – username=password
  • Password management not implemented – often passwords not changed for years
  • No database audit enabled
  • No granularity of privileges – often all users have the same privilege profile
  • Excessive DBA access – multiple methods to connect
  • Developer access...
The Focus in the Hacking Press is Wrong

- Exploits focus on bugs in software (rarer in the C code) mostly in the PL/SQL packages
- Exploits often do things like “GRANT DBA TO PUBLIC”
- Often security products also include rules based on these types of exploits
- [http://www.exploit-db.com/exploits/10268/](http://www.exploit-db.com/exploits/10268/) is a good example
- The focus is on the “software” and abusing the software
  - This is valid BUT exploits almost never cover stealing production data!
Example- Hardening Approach

• Not many checklists exist for Oracle databases
• Most are from same initial source or are very similar
• Some structure there but not good enough
  • “tip based rather than method based”
• Lists don’t focus on securing the data
• Difficult to implement for a large number of databases
• CIS for instance has 158 pages
Example – Access to Important Data

• Multiple routes to the same data exist in Oracle
  • If you know “Perl” Oracle is similar; if not intended!
• Normal wisdom says access is controlled by privileges on the table itself – “GRANT SELECT ON USER.TAB”
• View the permissions on the table should reveal who can view the data – err, no... Well yes but not quite!
  • There are “other privileges” - sweeping
  • Oracles shared memory leaks data
  • Oracles trace/dumps leak data
  • People leak data to files
  • Data gets copied...
Securing Data In an Oracle Database

• **Security Patches.**
  - Applying patches is difficult in terms of time, testing and more but usually there is no workaround instead of installing the patch. But at the end of the day its a “patch” / “no-patch” issue

• **Hardening**
  - Hardening is a worthwhile process to bring the basic level of security “up” for your databases. All databases must be hardened; not all measures work in all cases

• **Data security**
  - This is the design work that should be done before the database is built. Design data access privileges, least privilege, design user accounts [users, batch, reports...], design controls, audit
Companies do Spend on Desktop Security

• Companies usually have general security policies
• Not many have Database/data Security policies already
• Often general policies don’t apply at database level
  • Passwords/management for instance
• Spend is often on AV, Firewalls (not data)
• IDS / IPS often purchased but not for database
• Even data security products not often seen in existing sites -IDS/IPS/DAM – I don’t see often; some huge sites
• Usually none or weak data level security exists
Application Level Security

- Often see application level security in sites
- Customers add security controls at the application layer and allow “free for all” access to the data layer
  - They expect the application to protect the data
- Often audit is enabled at Application level:
  - Often triggers for “before and after” – horrendous performance
  - None or very little database audit, often not focused, usually no management and usually no reporting/alerting
  - Still this issue of performance exists – its not real if audit is designed
- Usually no operating system audit – “layers of access!”
Data Security has to be part of the Design

• This should be obvious
• Data security has to be part of the design of every application and deployment
• Lots of sites (big, very big and small) think they have done this BUT THEY HAVE NOT!
• Education and knowledge of what exists is most important
• Adding data level security later is harder to do!
Design a Policy

• Creating a data security policy has to be the first step
  • A document to wave around in the air
  • To chastise people with
  • To measure against
  • To build new systems against
  • To test compliance with
• Without a policy you do not know what secure data should look like for your company
• Before you create a policy you must understand what you have already in terms of security of data and where your data is!
Implement the Policy

• Once you have a data security policy you must implement it
• **Strategic solutions** - Start with solutions that work across most databases
• **Tactical solutions** - Also start with technical solutions needed on some/all databases to fix gaping holes
• Audit controls could be used first
• Security controls where the risk is greater
• Test for compliance
So What is The Solution?

• Where do you spend your money?
• Start with a policy – “a template”
• Without a policy there is no “start”, no “end”, no “measurements”
• You have to know what you want, when you have it and how efficient it is
• With databases – all data must be secured!!!
• Where do you spend your money?
  • products?
  • Free solutions – in terms of license?
  • All solutions cost money to manage and deploy
Fundamentally Strategic Solutions

• I will cover just one example – here it is:
• **Stop people connecting to the database. Period!**
• Why? – in my experience too many people have access
• Remove accounts – procedural – that policy again
• Block access - IP restrictions, firewalls...
• Harden all passwords
• Ideally make sure the application is not broken
Must know where data is first!

• Locate the data
  • Start with known facts – CREDIT_CARD table
  • Who can access it
  • Are there copies
  • Who can access the copies
  • What sweeping access exists
  • Where is the data leaked by the database and who can see it
  • Where is data leaked outside of the database – files, exports...
Strategic Solutions First

• When you know where the data is protect access to it
• Decide on possible solutions
  • Firewall between users/data
  • IDS/IPS/DAM products to protect against un-authorised access
  • Monitor data access with an audit solution
  • Decide on free internal / free third party / commercial solutions
Free Oracle Security Solutions

- What is amazing is that free solutions are not used often
- **Free; in terms of no additional license**
- Not free in terms of implementation
- Some really nice things can be done
  - Audit
  - Encryption
  - Fine Grained Audit
  - Row Level Security
  - Secure Application Roles
  - Proxy connections
Example - Proxy DBA Access

- I have been designing this for years for clients
- It solves a privilege problem
- Create one powerful account (not with DBA)
- Lock the powerful account
- Create a number of DBA accounts – little, no privilege
- Grant “proxy” through the powerful account
- Audit proxy access for accountability
Example – Secure Application Roles

- Don’t enable all privileges by default
- Enable privileges based on context
- Create a function to decide if a privilege should be enabled
- Never see these used
- Never see even password protected roles
Example – Use Core Audit

- Oracle has rich audit features
  - Core audit
  - Fine Grained Audit
  - Triggers – System / DML
  - Who/when
- All features can be correlated
- Most people never use; those that do use only limited parts – people complain about performance
- Layered audit is necessary
- Audit can be sent to database/operating system/SYSLOG/XML
Third Party Solutions

- There are a lot of Oracle security products available
- IDS / IPS / DAM – i.e. Sentrigo
- Commercial vulnerability scanning – i.e. PFCLScan
- Free Scanners – i.e. Scuba
- Data masking – net2000
- Forensics – V3rity
- Encryption – Protegrity
- Data location – Braintree
- More...
Commercial Solutions

• What are the core products?
  • Intrusion Detection
  • Intrusion Prevention
  • Data Activity Monitoring
  • Audit
  • Application Firewalls
  • Virtual Patching
• Hardware / software
• Network based / host based
• Log based / Traffic Based
The Market is Legitimised

- Guardium - $220M bought by IBM
- Secerno - $??M bought by Oracle
- Sentrigo - $??M bought by McAfee
- Imperva - $75M IPO filed for
- That leaves only one of the top 5 not acquired / IPO’d
- Other “players” include
  - Lumigent – rising from the ashes with BeyondTrust
  - Oracle Audit Vault – not really in the same “place”
  - Many more...
- Market size - $130M, Noel Yuhama says 21% per year rise through 2012
- Vulnerability scanning market > $1BN by 2014
Vulnerability / Compliance Scanning

- Vulnerability scanners look for potential risk issues
  - Not real time
  - Some focus on bugs
  - Less so on data and compliance
- Not as many products available
- All IDS/IPS/DAM/Audit vendors have Vulnerability scanning built-in/Attached or licensed
- Most IDS/IPS/DAM products also now have data discovery built-in/Attached
Win Battles Not Wars

• Understand the problem first – where is data/who can?
• Develop a proper policy
• “Recommended” / “compulsory” => “non compliance”
• Strategic solutions – firewalls, stop connections
• Audit first – know what’s going on – bolster the policy
• Add security controls
• Add security features – secure code, encryption...
• Reduce – reuse – recycle – “Bob The Builder”
Spend Wisely

- This is the hardest thing to do
- Spend as little money as possible to reduce the risk as much as possible
- The problem is we need to know the risk
  - Where is the data
  - Who can see it
  - What can we do to prevent that access?
- 100db * 200issues * 40 people *?? = lots of “dosh”
Conclusions

• Understand that you must secure data not software
• It is not “Oracle Security” it is “Data Security”
• Remember 3 components (patch, harden, data)
• Understand what you have now
• Develop a strategy / policy
• Implement
• Spend wisely
Questions?

Any Final Questions?
We Must Secure Data Not Software

Start Security in the Right Place