Protecting Against SQLi in Real-Time

Stuart Hancock

stuart.hancock@dbnetworks.com
bob.dewolfe@dbnetworks.com
AGENDA

• SQL injection attacks
  • primary database security focus
• SQL injection detection/prevention
  • current technologies don’t work
• SQL threat assessment technology
  • a new approach
Overview

Web-based attacks: vast majority of data loss

Verizon Business report: 900+ breaches, >900M records lost

Threat is primarily from the outside

PCI compliance is no panacea

Attacks follow money – Financial, Hospitality, Retail
Overview

- Majority of losses to Web-based attacks
- 2004-2009: 900+ breaches, >900M records lost
Outside Threats are Primary

- 70% of attacks are external
- 21% of victims were PCI-compliant
- Attacks follow money:
  - Financial – 33%
  - Hospitality – 23%
  - Retail – 15%
  - The rest – vast majority unreported
High Profile Targets Get Attacked

- High profile targets become hacking trophies
  - Stratfor
  - NSA
  - Oklahoma DOC
  - Symantec
  - US Census Bureau
  - United Nations
SQL Injection Attacks

Database attacks result in:

- Leakage of sensitive information
- Destruction of important information
- Defacement of websites
- Distribution of malicious code

SQL Injection remains the preferred method of attack
SQL Injection Scenario

**Web Threat**

1. Enter bad input into a form (or cookie, url...)

2. Application layer creates new statement
   
   ```
   Select SSN from personnel where 
   userName = 'bob' or '1'='1' -- ...
   ```

3. Malicious SQL
Can we get valuable data?

Username injected with NO password
Entire Database Dumped
State of the Industry

Current offerings (multi-tiered model):
- Code Review/Scanning
- Pattern Recognition (WL/BL)
- SQL Statement by Statement Training

Problems:
- Good luck with writing perfect code
- Chasing the horse that left the barn
- Very long learning cycle, high false positives
CODE REVIEW

“Software will always have bugs and by extension, security vulnerabilities. A practical goal for a secure software development lifecycle (SDLC) should be to reduce, not necessarily eliminate, the number of vulnerabilities introduced and the severity of those that remain.”

- Michael Howard, Microsoft, Senior Security Program Manager
Whitelist/Blacklist, Statement by Statement Learning

• Not effective – horse already left the barn
• Not possible to predict new/unique attacks
• Application and Web Servers are SQL generators
  - Not possible to learn all generated SQL
  - High false positive rate as a result
What’s needed:

**Adaptive Database Firewall**

- Requires full understanding – application/database tiers
- Profiles Web-based applications
- Deep semantic/parametric analysis, all SQL statements
- High sensitivity but low false positive rate
- Lexically new statements assessed for structural attacks
- Short training period
- Continuously refined profiles adapt to application changes
Time to safety

Compared to non-adaptive white-list/black-list technology

- Applications are protected sooner – much sooner
- Less resources consumed to achieve protection
- Application changes less likely during learning cycle
- Protects against new/unique attacks not previously seen
- Capable of monitoring future attack vectors
Rapid Learning

Long tail arrival of new statements

Short learning cycles
• Reduce deployment costs
• Keep pace with rapid application changes
Classic SQL Unique Statement Trend
When an Attack is Detected, one should...

- **Alert**
  - Create alerts via email, syslog (SIEM), and SNMP
  - Audit logs identify breach - secure, signed logs

- **Inform**
  - Analytics provide nature and scope of attack
  - Analytics provided to provide efficient review
  - Integration with third party audit/compliance facilities

- **Block if desired**
  - Database session kill capability terminates attack
  - Web tier integration provides session blocking
  - Blocking presents challenges
Evolution of Database Security

Time

SQL Injection Protection

Black Lists

White Lists

Enhanced White Lists

Threat Assessment Technology
Current Approaches:

- Continue as is, no DB protection
- Development phase – code analysis/improvement
  - Applicable to new code development when possible
  - Too many vulnerabilities, expensive, time-consuming
  - Extensive required testing expensive, delays releases
- Post-breach: Forensics assess scope of damage
  applicable to determine liability, accountability

.....or.....
Real-Time Production Phase Protection

- Genuine Protection in Real-Time
- Short learning cycle
- Multi-environment adaptable
- Profiling of each application’s database activity
- Drop-in, transparent install
- Passive monitoring
- Multiple heterogeneous databases simultaneously
- Web-tier attack vector correlation
THANK YOU!

Questions?

Stuart Hancock
stuart.hancock@dbnetworks.com
301-788-3192

Bob DeWolfe
bob.dewolfe@dbnetworks.com
978-317-8197