Yuan Fan
Arcsight

BLACK HAT BRIEFINGS

Advance SQL Injection Detection by Join Force of Database Auditing and Anomaly Intrusion Detection

This topic will present the proposal/idea/work from the author's master graduate project about effective detection of SQL Injection exploits while lowering the number of false positives. It gives detail analysis example of how database auditing could help this case, and also presents the challenge with anomaly detection for this attack and how the author tried to solve them. Finally a correlation between the two will be presented.

Yuan Fan, *CISSP, has worked in the network security area for more than 7 years. He currently works for ArcSight as a Software Engineer. He holds a Master of Computer Engineering degree from San Jose State University. The tool he is writing for master graduate research project related to this topic is a Java-based, multilayer anomaly intrusion detection system.*

Advance SQL Injection detection by join force of Database Auditing and Anomaly intrusion detection

Yuan Fan, CISSP

BlackHat 2005

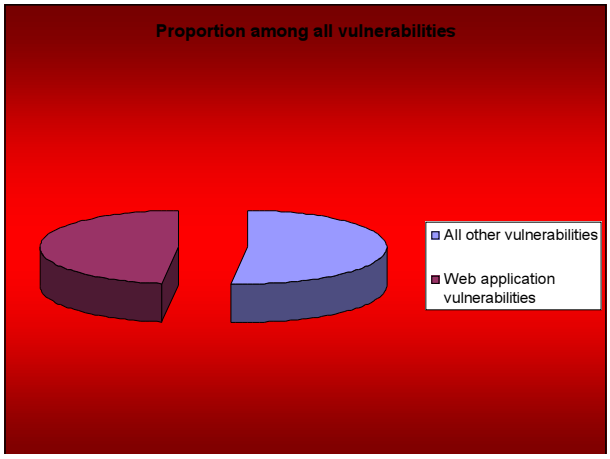
Related Topics

- "Detection of Web Application Attacks" [BlackHat 2004]

- This topic:

1. Presents an idea/work and a detailed example for database auditing to help SQL Injection detection.
2. Presents a master program project about anomaly intrusion detection for SQL Injection and correlation ideas between database auditing and anomaly intrusion detection.

Web Application Security Vulnerabilities Trends



Source: Symantec Threat Report

What portion of production web applications do not use a database?

What is SQL Injection?

- An attack method that exploits the vulnerability of web applications by using deliberate input data to generate a SQL query against the backend database to achieve goals such as unsolicited data exploitation and manipulation.

- Examples:
  - SELECT Username FROM Users WHERE Username = '' OR ''='' AND Password = '' OR ''='‘
  - ….union select username from all_users where rownum<2;

Existing mechanisms to protect against SQL Injection

- Input validation from client side to server side
- Web Application IDS/Firewall
- Homegrown methods using raw regular expressions (A simplified signature-based detection plus a little fuzziness?)

  /\w*((\%27)|(\'))((\%6F)|o|(\%4F))((\%72)|r|(\%52))/ix

  Quote: "Either leads to many false positives or leads to too many signatures…"

Limitations of Existing Signature-based Solutions

- Many false-positives

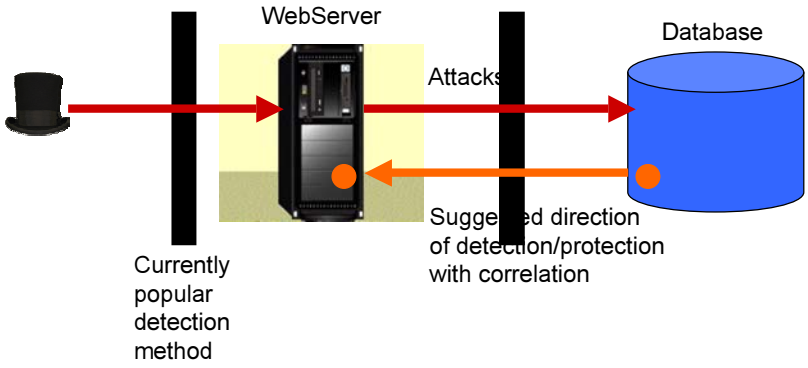
  No one looks at flood alerts anymore…

- Vulnerable to attack evasion

- Unnecessary restrictions for the end user

  "O'Connor" is no longer allowed as username or password?

Reverse Direction



One Thing Never Forget for Web Applications

- Database

  No matter how powerful fragmentation, encoding and other advanced evasion techniques are…

  After all, what does this attack turn to, and what does it target?

Database Auditing

- Auditing what?
  What we will suffer from SQL Injection attack!
  (Privilege escalation, unexpected actions, etc.)

- W W W W
  Who is doing what on which object at when

Database Independent – Oracle, MS SQL, etc.

Start as a Process

1. Grant
   Set RIGHT privilege for the db account
   that is connecting from web application.

2. Define triggers
3. Audit!

Auditing Examples

- Audit DROP ANY TABLE BY WEB BY ACCESS;
- Audit NOT EXISTS;

  Auditing all SQL statements that fail because the target object does not exist. (Isn't this very nice for the SQL Injection case?)
- Audit ALTER, DELETE, UPDATE ON SALARYTABLE;
- Audit SELECT ANY TABLE WHENEVER NOT SUCCESSFUL;
- …

Fine-grained audit using triggers

- Database Triggers can detect the number of rows returned; general audit cannot
- Triggers can expose the full SQL used by the user action (ora_sql_txt function in 9i or dbms_fga in 10g)
- MS SQL Server: Trigger is one of very few choices available for fine-grained auditing.

Abnormal Web application attack Intrusion Detection

- Method

1. Modeling
2. Neural network training
3. Real time abnormal detection and scoring

Abnormal Web Intrusion Detection Modeling

- Challenges of modeling

How to score/digitalize items?

UN/**/ION/**/ SE/**/LECT/**/

How to lower the number of false alarms while trying not to miss the real abnormal alert?

Abnormal Web Intrusion Detection Continued

- Value length
- Parameter abnormal score (see next slide)
- Status Code
- InTraffic
- OutTraffic

Note: Every item is a number!

Abnormal Web Intrusion Detection Continued

- Training
  - Web Log parsing, normalization, scoring (we also have TCPDUMP parsing module)
  - Each URL has its own pattern
  - How is the Parameter abnormal score generated?
  - Fuzzy but half awareness (eg: status code 500)
- Detection

  Neural network space distance

Abnormal Web Intrusion Detection Continued

- Sample web log:

2004-10-04 00:16:06 192.168.20.231 - 192.168.1.80 80 GET /nt80_21/nt80no/iGetCodeJS.asp 62166.39 200 0 448 0 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) http://192.168.1.80/nt80no/list.asp?lname=ComputerNetwork'%20and%20%201=(select%20id%20from%20admin%20where%20len(name)>2)%20and%20'1'='1

Abnormal Web Intrusion Detection Continued

- Snippet of training results of HTTP request URL parameters

```
<?xml version="1.0" encoding="UTF-8" ?>
<WebCGIData version="0.1">
  <entry URL="/nt80_21/nt80no/login.asp" />
  <entry URL="/nt80_21/nt80no/iGetCodeJS.asp"
    paralength="7" paraname="NONAME"     parascore="18" />
  <entry URL="/nt80_21/nt80no/shop.asp"   paralength="2"
    paraname="id" parascore="2" />
  <entry URL="/nt80_21/nt80no/bbs3.asp"   paralength="14"
    paraname="bbs2" parascore="3" />
  <entry URL="/nt80_21/nt80no/cong.asp"   paralength="5"
    paraname="name" parascore="3" /> …
```

Correlation

- What is common between attacks?

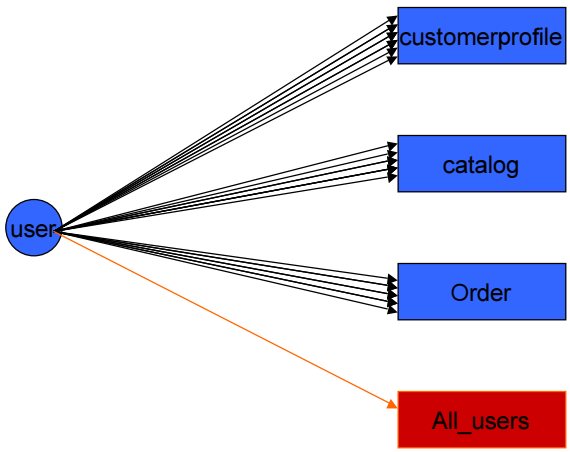
  An SQL Injection is not exceptional

- Timestamp (short range) is still the key, but you can do more.

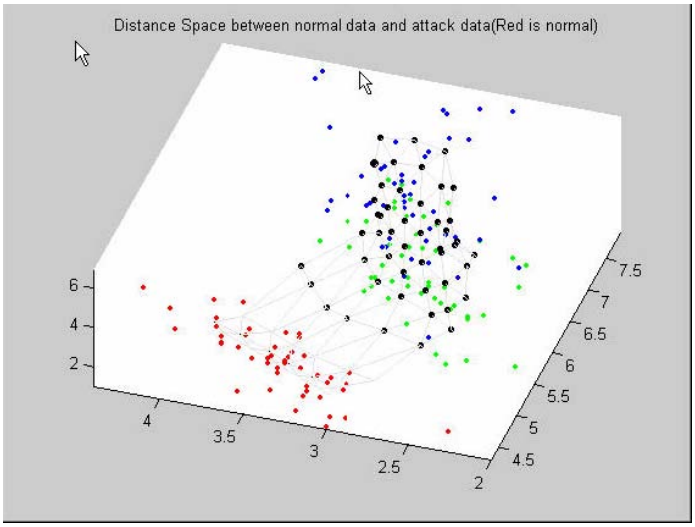
Web side:	‘ union select password from users where ‘’=‘
DB Audit log side:	USERNAME | OBJ_NAME | ACTION_NAME | RETURNCODE TESTUSER | USERS | SELECT | 942

No Success

Normal and Abnormal Events



Visualization – Normal vs. Abnormal Data



Further Research/Work

- HTTP Post
  - Internal module for web server to intercept
  - Use TCPDump + ngrep + perl
  - Use Snort + HomeMadeHTTPPostRule
- More abnormal pattern detection on Database audit side (similar to the mechanism we used for web log!)
- More Real Attack Env Testing

Some Commercial Products

- ArcSight (SIM/SEM)
  - Offers audit agents for Oracle, MSSQL, and other databases
  - Aggregate events from devices, system and apps
  - Correlate events between Web apps and db audits
  - Visualization, Reports, Data-mining, Incident-mgmt
- Netcontinuum, Imperva, etc. (Web IDS/firewall)
- Lumigent, apexSql (Database auditing)
- And many more…

References

- Timo Honkela: Self-Organizing Maps in Natural Language Processing http://www.cis.hut.fi/~tho/thesis
- "Anomaly Detection of Web-based Attacks" Christopher Kruegel, Giovanni Vigna
- http://www.securityfocus.com/infocus/1768
- http://www.securityfocus.com/infocus/1714
- "Detection of Web Application Attacks.ppt" Blackhat 2004
- http://Asktom.oracle.com
- www.imperva.com/application_defense_center/white_papers/sql_injection_signatures_evasion.html
- Symantec Internet Threat Report Volume VII, published March 2005

Many Thanks To:

My professor Xiao,
BlackHat & Everyone here
and all my friends!

Q&A