

# Hack-proofing Oracle 9iAS

Aaron Newman

[anewman@appsecinc.com](mailto:anewman@appsecinc.com)

Application Security, Inc.

[www.appsecinc.com](http://www.appsecinc.com)

Download updated version of presentation from  
<http://www.appsecinc.com/news/briefing.html>



**APPLICATION  
SECURITY, INC.**

[www.AppSecInc.com](http://www.AppSecInc.com)

Presentation #104

Hack-proofing Oracle 9iAS

# Agenda

- Understanding the hacker
- Buffer overflows
- Firewall configuration
- Application-level attacks
  - SQL Injection Demo
- 9iAS Security Tips
- Resources, Conclusion, and Wrap Up

# Understanding the hacker



**APPLICATION  
SECURITY, INC.**

[www.AppSecInc.com](http://www.AppSecInc.com)

Presentation #104

Hack-proofing Oracle 9iAS

# Security is a process

- Security of application depend on administrator more than on software itself
- Security is a process, not a product
- OpenHack – 4<sup>th</sup> annual eWeek competition
  - Cross-site scripting vulnerability in Oracle app
  - Not specific to Oracle
  - Result of how app is written

<http://www.eweek.com/category2/1,3960,600431,00.asp>

# Administrator's mindset

- In order to protect yourself, you should establish a paranoid mindset
  - Should not trust anything passed to the app
  - Should not trust that single layer will be secure
  - Should not trust developers, vendors, etc...
- Security is only as strong as the weakest link
  - Usually easiest to attack a webserver indirectly
  - Attack those aspects that have not neglected



# How Hack is Planned

- Several ways hacks can be undertaken
  - Hacker attacking based on a hole
  - Hacker looking at a specific target
- Hack is carried out much like a special operations mission
  - Create a map
  - Recon work
  - Start probing
  - See how far hacker can get



# Create the map

- Try to locate the target
- Perform IP scan of target
  - Ping main website – get IP Address
  - Scan IPs in range
  - Look at whois.net
- Perform port scan of each found server
  - Locate all applications externally addressable
  - Look for common ports (80, 7777, 443, 4443)



# Google as an attack tool

- Constantly inventorying web sites
- Knows about files that are no longer linked
  - Even if you removed pages, info may still be floating around on Google
- Take you to other sites that know about
  - <http://uptime.netcraft.com/up/graph?site=...>
- Also can be used to find sites using 9iAS
- <http://www.wired.com/news/infostructure/0,1377,57897,00.html>



# Your site is recon'ed

- Second step is for the hacker to start collecting information about your site
- Might create an account on your application
  - Use that to figure out information about other users
- Looks for software running and versions
  - Allows a hacker to pick a specific attack
  - Should try to hide this type of information
- Crawls the web site
  - Getting full picture of your web site
- Checks default/easy-to-guess passwords



# Holes in your security

- Oracle Website – Alerts Web page  
<http://otn.oracle.com/deploy/security/index2.htm?Info&alerts.htm>
- Vulnerabilities on SecurityFocus.com
- Finds existing or builds an exploit
- Long-term brute-force of accounts
  - Never use the default users or passwords



# Access gained to non-critical applications

- Most times a hacker is able to penetrate non-critical systems
- Once control is gained, used as a springboard for further activity
- Network sniffer installed
- Scanning and attacks mounted from there



# Look at [www.oracle.com](http://www.oracle.com)

- Over 200 other publicly addressable Oracle servers
- [www.oracle.com](http://www.oracle.com) is locked down tightly
- Other applications may not be as secure

# Understanding buffer overflows



**APPLICATION  
SECURITY, INC.**

[www.AppSecInc.com](http://www.AppSecInc.com)

Presentation #104

Hack-proofing Oracle 9iAS

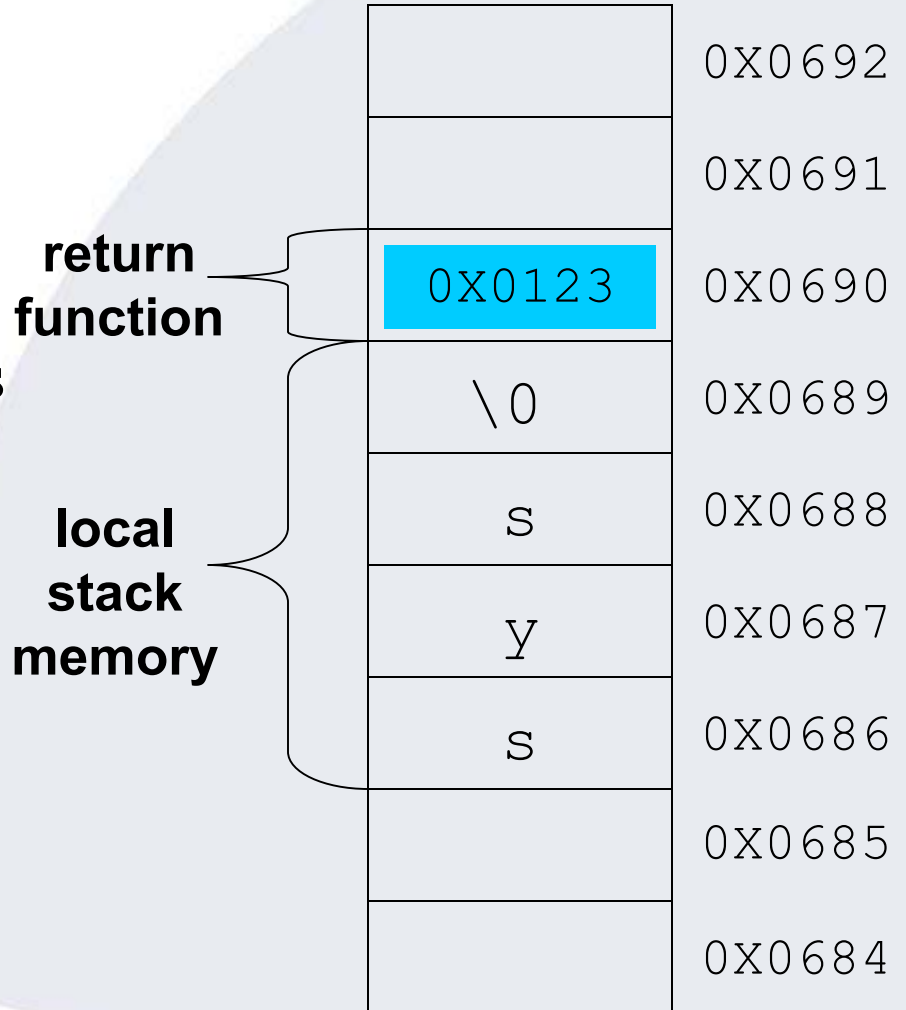
# What is a buffer overflow

- When a program attempts to write more data into buffer than that buffer can hold...
  - ...Starts overwriting area of stack memory
    - That can be used maliciously to cause a program to execute code of attackers choose
    - Overwrites stack point



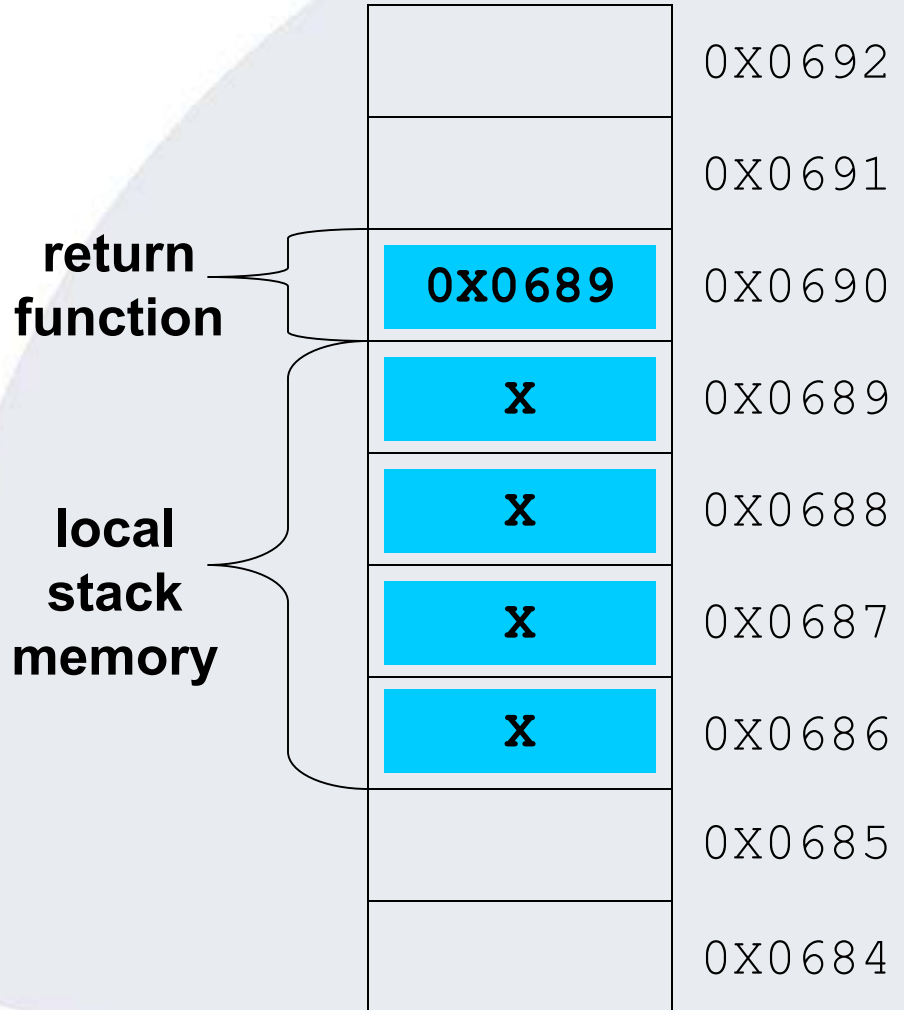
# Mechanics of stack-based buffer overflow

- Stack is like a pile of plates
- When a function is called, the return address is pushed on the stack
- In a function, local variables are written on the stack
- Memory is written on stack
  - char username[4] reserved 4 bytes of space on stack



# Mechanics of stack-based buffer overflow

- When function copies too much on the stack
- The return pointer is overwritten
- Execution path of function changed when function ends
- Local stack memory has malicious code



# Example 9iAS buffer overflows

- As in all complex software
  - Many buffer overflows exist!
- Below are examples in the PL/SQL Module:  
`http://website/pls/dad/admin_/help/AAAA..`  
`http://website/pls/admin_/help/AAAAA..`
- Presenting credentials to the web server using the “Authorization” HTTP header
- Apply the fix from <http://metalink.oracle.com>
  - 9iAS release 1 must be patched

# More recent PL/SQL buffer overflows

- Below are examples of PL/SQL overflows:
- **TO\_TIMESTAMP\_TZ** buffer overflow
  - SELECT TO\_TIMESTAMP\_TZ('1999-12-01 11:00:00 -8:00',  
'YYYY-MM-DD HH:MI:SS TZH:TZMXXXX[230 additional Xs]')  
FROM DUAL;
- **TZ\_OFFSET** buffer overflow
  - SELECT TZ\_OFFSET('US/EasternXXXX[74 additional  
Xs]') FROM DUAL;



# More recent protocol buffer overflows

- Username buffer overflow

```
C:\oracle\bin> loadpsp -name -user XXX[1150+  
characters]/test@iasdb test
```

- Authentication mechanism chokes on long username
- This could be the Slammer Worm for Oracle
  - Actually much worse
  - Worm will not surface and servers will stay unpatched



# Discovered last week

- CREATE DATABASE LINK buffer overflow
- Occurs for connection string > 1000 chars

```
CREATE DATABASE LINK [linkname] CONNECT TO
[username]
IDENTIFIED BY [password]
USING '[connection string]'
SELECT * FROM TEST@[linkname]
```

- SCOTT has CONNECT ROLE
- CONNECT role has CREATE DATABASE LINK

# 9iAS buffer overflows

- ORADAV Format string buffer overflow
- Distributed Authoring and Versioning
- Enabled in 9iAS by default
- Changed from standard Apache code
  - Default to log certain bad responses
- Presentation and exploit code at BlackHat
- Done by same guy that did Slammer worm presentation last year at BlackHat

# Firewall Configuration



**APPLICATION  
SECURITY, INC.**

[www.AppSecInc.com](http://www.AppSecInc.com)

Presentation #104

Hack-proofing Oracle 9iAS

# Firewalls

- What does a firewall do?
  - Blocks traffic
- Why is it important?
  - Reduces the types of attacks
  - Reduces the number of targets
- Firewall is the “*hard-crunchy outside*”
  - Internal network is the “*soft chewy inside*”



# Myth – behind a firewall you are secure

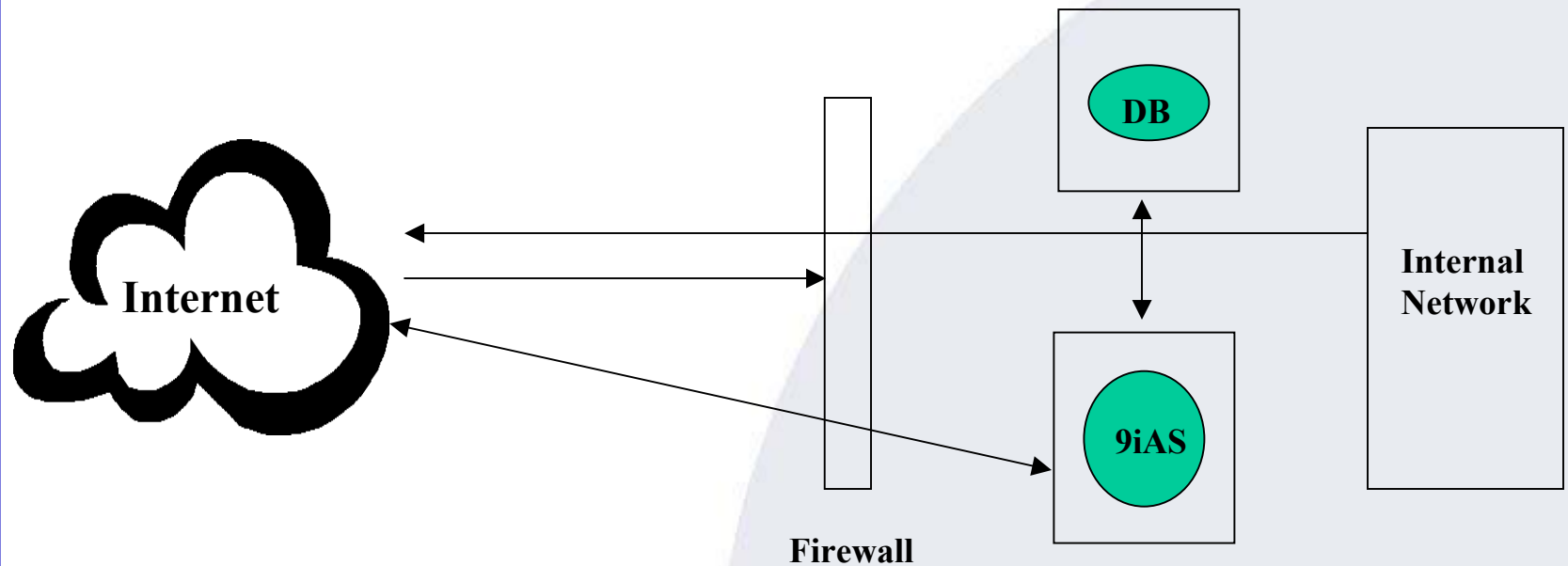
- Does setting up a firewall block all attacks on your application server?
  - NO!!!
- Firewalls only filter out non-HTTP attacks
- Don't stop holes in application logic
- Do not stop attacks on Apache or its mods

# Configuring the firewall

- Filter any packets not on port 80/443
- Filter any packets not going to web server
- Keeping the firewall patched
  - Well known attacks on firewalls
- Could use `valid_node_checking` in `listener.ora`



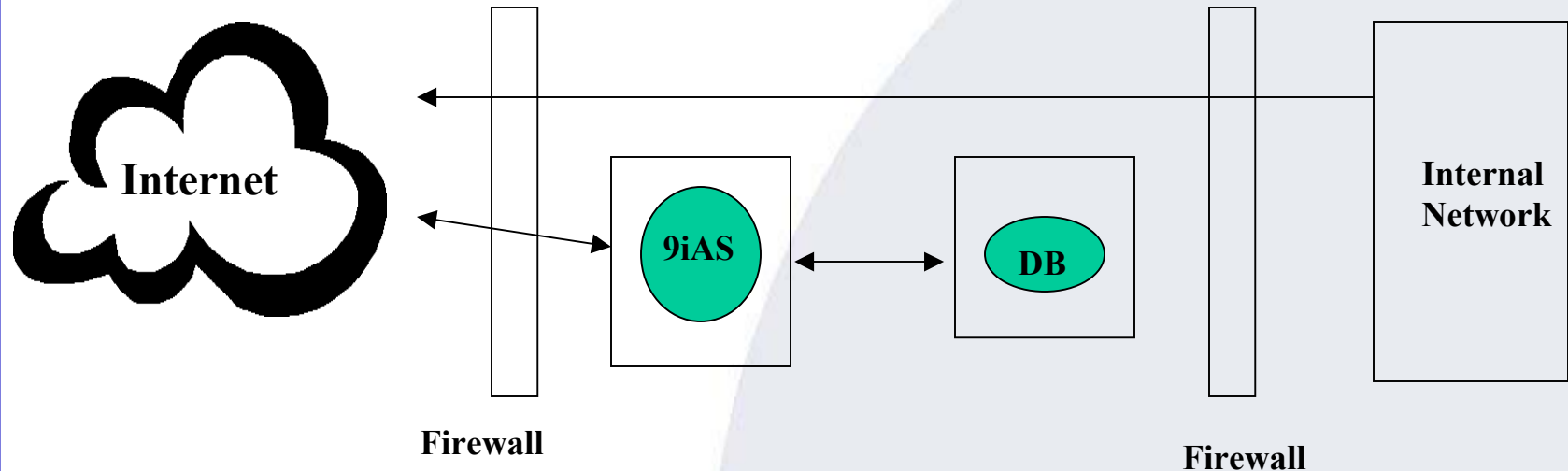
# Single firewall



- Least secure model
- Single point of protection - No DMZ



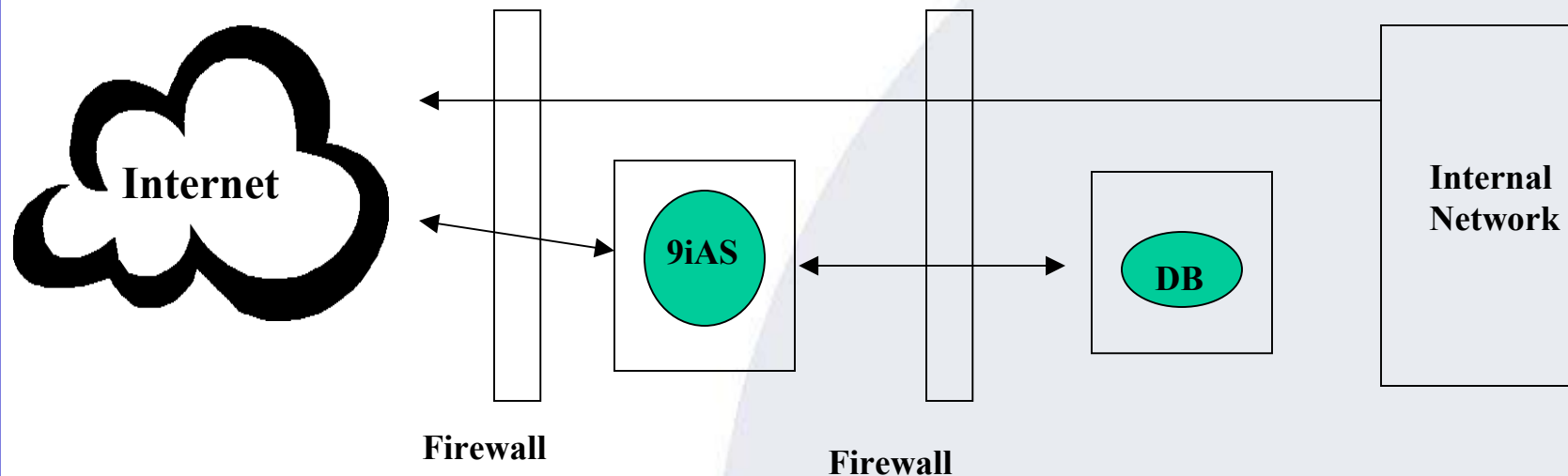
## 2 Firewalls, Database in DMZ



- More expensive
- More protection for internal network



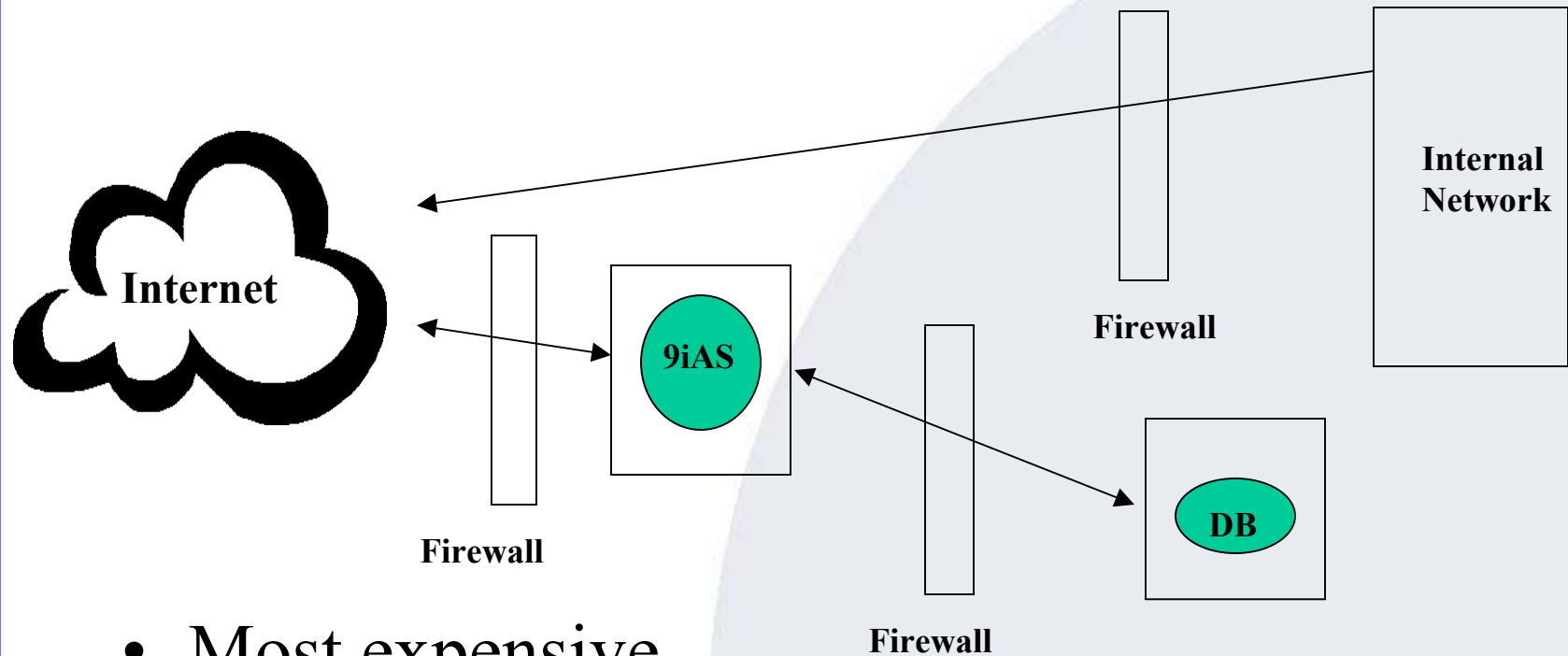
## 2 Firewalls, Database not in DMZ



- Not recommended
- Internal net attackable thru database



# 3 Firewalls



- Most expensive
- Most secure
- Internal network and web apps segregated



# Understanding SSL

- Use of SSL is great
- Does little to protect you from attacks
  - Only protects the information from being sniffed
- More effort should be concentrated on securing the end point
- **SSL SHOULD BE IMPLEMENTED INTERNALLY IN YOUR ORG!**

# Application-level Attacks



**APPLICATION  
SECURITY, INC.**

[www.AppSecInc.com](http://www.AppSecInc.com)

Presentation #104

Hack-proofing Oracle 9iAS

# Can attacks go through a firewall?

- **YES!!!**
  - Parameter manipulation
  - Cookie Poisoning
  - Hidden fields
  - Cross-site scripting
  - SQL Injection



# Parameter manipulation

- How does it work?
  - Modify the parameters in URL
- Change:  
`http://mywebsite/myapp.cgi?acctnum=2`
- To:  
`http://mywebsite/myapp.cgi?acctnum=3`



# Cookie Poisoning

- How does it work?
  - Modify cookie used by a site to maintain state
- Issues when a cookie is improperly used for authentication
  - Many apps encode (not encrypt) cookie
- Cookies should be implemented properly

# Hidden fields

- How does it work?
  - Changing hidden fields used for pricing
- *E-shop lifting*
- Changing a field used to stored the price of a product being purchased on-line
- Once again, trusting client-side code

# Cross-site scripting

- How does it work?
  - Submitting HTML/Script code to be executed later by other site visitor
- Message boards allowing users to submit text
- User input not validated
- Text may contain HTML or JavaScript tag
- Causes page views to perform malicious tasks

# SQL Injection

- How does it work?
  - Modify the query
  - Change:

```
Select * from my_table where column_x = '1'
```

- To:

```
Select * from my_table where column_x = '1'  
UNION select password from DBA_USERS where  
'q' = 'q'
```



# Example JSP page

```
String sql = new String("SELECT *  
FROM WebUsers WHERE Username=' " +  
request.getParameter("username") +  
' AND Password=' " +  
request.getParameter("password") +  
' "  
  
stmt = Conn.prepareStatement(sql)  
Rs = stmt.executeQuery()
```



# Valid Input

- If I set the username and password to:  
**Username: Bob**  
**Password: Hardtoguesspassword**

- The sql statement is:

```
SELECT * FROM WebUsers WHERE  
Username='Bob' AND  
Password='Hardtoguess'
```

# Hacker Input

- Instead enter the password:

**Aa' OR 'A' = 'A**

- The sql statement now becomes:

```
SELECT * FROM WebUsers WHERE  
Username='Bob' AND Password='Aa'  
OR 'A' = 'A'
```

- The attacker is now in the database!



# Selecting from other Tables

- To select data other than the rows from the table being selected from.
- UNION the SQL Statement with the DBA\_USERS view.



# Example JSP Page

```
String sql = new String("SELECT
* FROM PRODUCT WHERE
ProductName=' " +
request.getParameter("product
_name") + "'")
stmt =
Conn.prepareStatement(sql)
Rs = stmt.executeQuery()
< return the rows to the browser >
```

# Valid Input

- Set the **product\_name** to :  
**DVD Player**

- The SQL Statement is now:

```
SELECT * FROM PRODUCT WHERE  
ProductName='DVD Player'
```



# Hacker Input

- Set the product\_name to :

```
test' UNION select username,  
password from dba_users where  
'a' = 'a
```

- The SQL Statement is now:

```
SELECT * FROM PRODUCT WHERE  
ProductName='test' UNION  
select username, password from  
dba_users where 'a'='a'
```



# Preventing SQL Injection

- Validate user input
  - Parse field to escape single quotes to double quotes
- Use the object parameters to set parameters
  - Bind variables



# SQL Injection demo

JSP page, Oracle HTTP Server, Jserv,  
Oracle database



**APPLICATION  
SECURITY, INC.**

[www.AppSecInc.com](http://www.AppSecInc.com)

Presentation #104

Hack-proofing Oracle 9iAS

# Listener Attack Demo



**APPLICATION  
SECURITY, INC.**

[www.AppSecInc.com](http://www.AppSecInc.com)

Presentation #104

Hack-proofing Oracle 9iAS

# 9iAS Security Tips



**APPLICATION  
SECURITY, INC.**

[www.AppSecInc.com](http://www.AppSecInc.com)

Presentation #104

Hack-proofing Oracle 9iAS

# Remove informational pages

- <http://hostname/cgi-bin/echo>
- <http://hostname/cgi-bin/printenv>
- <http://hostname/perl/printenv>
- <http://hostname/j2ee/examples/jsp/>
- <http://hostname/ojspdemos>
- **<http://hostname/server-info>**
- <http://hostname/server-status>
- <http://hostname/jserv/status?module=127.0.0.1>
- <http://hostname/perl-status>



# Why do we remove informational pages

- This information can be used in ways administrator might miss
- `http://hostname/server-info`
  - Contains a list of the people currently querying pages, including the source IP address
  - Clients are easy targets
  - Piggy back from client into application using clients privileges



# Block using directives

- Block server-status page
  - <Location /server-status
  - SetHandler server-status
  - Order deny, allow
  - Allow from localhost
  - </Location>
- Block files with security and directory info
  - <Files ~ “^\.ht”>
  - Order deny, allow
  - Deny from all
  - </Files>



# Block “Gateway Configuration Menu”

- [http://hostname/pls/admin\\_/gateway.htm](http://hostname/pls/admin_/gateway.htm)
- [http://hostname/pls/admin\\_/globalsettings.htm](http://hostname/pls/admin_/globalsettings.htm)
  - Used to create DADs
  - Can be used to create malicious access to database
  - Versions prior to 1.0.2.2 very vulnerable



# Disabling features

- Directory indexing
  - Gives too much information
- .htaccess files
  - Contains authorization information
  - Parsed every time loaded
  - Stay away from these
- Disable all sample applications
- If you don't use a feature - disable it

# Resources, Conclusion, and Wrap Up



**APPLICATION  
SECURITY, INC.**

[www.AppSecInc.com](http://www.AppSecInc.com)

Presentation #104

Hack-proofing Oracle 9iAS

# How to Combat Hackers

- Stay patched –
  - <http://metalink.oracle.com>
- Security alerts:
  - [www.appsecinc.com/resources/maillinglist.html](http://www.appsecinc.com/resources/maillinglist.html)
- Security Discussion Board
  - [www.appsecinc.com/cgi-bin/ubb/ultimatebb.cgi](http://www.appsecinc.com/cgi-bin/ubb/ultimatebb.cgi)
- Check out security solutions at:
  - [www.appsecinc.com/products](http://www.appsecinc.com/products)

# Defense in depth

- All complex software has security holes
- Some are known by the bad guys
- Assume someone out there knows how to break a single layer of defense
- Set up multiple layers of defense
- Reduce the likelihood of being “owned” because fewer people can break multiple layers

# How to Combat Hackers

- Multiple levels of security
  - Perform audits and pen tests on your applications on a regular basis
  - Encryption of data-in-motion
  - Encryption of data-at-rest
  - Monitor your log files
  - Implement intrusion detection



# Questions?

- About
  - Oracle security features
  - Vulnerabilities
  - Protecting your applications
- Download free evaluation software at:
  - [www.appsecinc.com](http://www.appsecinc.com)
- Email me at:

**[anewman@appsecinc.com](mailto:anewman@appsecinc.com)**

**[www.appsecinc.com](http://www.appsecinc.com)**



**APPLICATION  
SECURITY, INC.**

[www.AppSecInc.com](http://www.AppSecInc.com)

Presentation #104

Hack-proofing Oracle 9iAS