Web Sever Security

APACHE; MYSQL; NIKTO

BY:
MEGAN CUTLER
AND
RITIKA MOORJANI

Introduction

- Topics Covered
  - Security Checking using Nikto
  - Dangers of Default Installation
  - Ways to Secure Apache
    - Compiling
    - Configuration
  - SSL
  - Damn Vulnerable Web App
  - Dangers of SQL Injection
  - Securing MySQL

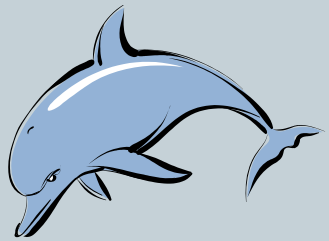
Nikto

- Designed specifically for scanning web servers
- Identifies the type of server running
- Scans for dangerous files, configuration options and dangerous exploits
- lists potential exploits including how an attacker could abuse them
- Released by Cirt.net (http://www.cirt.net/nikto2)

Default Installations

- Apache:
  - Install: yum install httpd
  - Start: service httpd start

- MySQL
  - Install: yum install mysql mysql-server
  - Start: service mysqld start



Nikto Results Against Default Installation

```
[root@localhost nikto]# nikto -h 10.255.32.104
---------------------------------------------------------------------------
- Nikto 1.36/1.37 - www.cirt.net
+ Target IP: 10.255.32.104
+ Target Hostname: 10.255.32.104
+ Target Port: 80
+ Start Time: Thu Nov 12 12:49:11 2009
---------------------------------------------------------------------------
- Scan is dependent on "Server" string which can be faked, use -g to override
+ Server: Apache/2.2.8 (Fedora)
+ Allowed HTTP Methods: GET,HEAD,POST,OPTIONS,TRACE
+ OSVDB-877: HTTP method ('Allow' Header): 'TRACE' is typically only used for debugging and should be disabled. This
    message does not mean it is vulnerable to XST.
+ OSVDB-877: TRACE option appears to allow XSS or credential theft. See http://www.cgisecurity.com/whitehat-
    mirror/WhitePaper_screen.pdf for details (TRACE)
+ OSVDB-3092: GET /manual/ : Web server manual found.
+ OSVDB-3268: GET /icons/ : Directory indexing is enabled: /icons
+ OSVDB-3268: GET /manual/images/ : Directory indexing is enabled: /manual/images
+ OSVDB-3233: GET /icons/README : Apache default file found.
+ 2673 items checked - 7 item(s) reported on remote host
+ End Time: Thu Nov 12 12:49:20 2009 (9 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Ways to Make Apache More Secure

- Compiling
- Configuration
  - Configure Apache to run as its own user under its own group
  - Run in a chrooted environment
  - Configure Virtual Hosts
  - Disable directory indexing
  - Disable server side includes
  - Disable .htaccess and do not allow Override
  - Don’t allow Apache to follow symbolic links
- SSL

Compiling Apache

Reasons to compile:

- Higher level of control over the program configuration
- Admin can compile for speed, reliability or **security**
- Select which modules are enabled and disable all others
- Unused features are not available to be exploited
- Better able to obscure data footprint
- Allows you to use as few default settings as possible

Example Script

```
#!/bin/kish
LD_PRELOAD=/usr/lib/libmtmalloc.so

CC=cc
CFLAGS="-xO3 -mt -fsimple=1 -ftrap=%none -nofstore -xbuiltin=%all -xlibmil -xlibmopt -xregs=no%frameptr"
CPPFLAGS="-I/usr/local/include/ -I/usr/local/include/openssl -I/opt/SUNWspro/include -I/opt/sfw/include -I/usr/sfw/include"

CXX=CC
CXXFLAGS="-xO3 -mt -fsimple=1 -ftrap=%none -nofstore -xbulitin=%all -xlibmil -xlibmopt -xtarget=native -xarch-native -
    xregs=no%frameptr"
LDFLAGS="-L/usr/local/lib -L/opt/sfw/lib -L/usr/ucblib -L/usr/sfw/lib -R/usr/local/lib -R/opt/sfw/lib -R/usr/ucblib -R/usr/sfw/lib"

export CC CFLAGS CXX CXXFLAGS CPPFLAGS LDFLAGS

./configure     \
--enable-layout="int525"     \
--enable-modules='auth-digest dav dav-fs dav-lock deflate info mime-magic rewrite so speling ssl unique-id usertrack vhost-
    alias' \
--with-ssl=/usr/local/ \
--with-ldap-lib=/opt/sfw/lib \
--with-ldap-include=/opt/sfw/include \
--with-mpm=perfork
```

Virtual Hosts

- Allow hosting of multiple websites on a single server
- Only requires securing a single server
- Virtual hosts can be run under unique user accounts; ensures each host can only access its own files
- IP based; each host has a different IP address
  - Non-designated IP addresses rejected
- Name based; single IP address, different hostnames
  - Requires an add-in to apache to configure SSL
- Good for internal trusted servers such as intranets

Example Virtual Host

```
<VirtualHost 10.0.0.10:443>
DocumentRoot "/export/srv/www/vhosts/mom.shop/htdocs/"

<Directory "/export/srv/www/vhosts/mom.shop/htdocs">
Options Indexes FollowSymLinks
AllowOverride none
Order Deny,Allow
Deny from all
Allow from All
</Directory>

ServerName www.mom.shop
ServerAlias mom.shop

SSLEngine ON
SSLCertificateFile /export/srv/www/vhosts/mom.shop/ssl/ssl.crt/mom.shop_cert.pem
SSLCertificateKeyFile /export/srv/www/vhosts/mom.shop/ssl/ssl.key/mom.shop_key.pem

Alias /cgi-bin/ "export/srv/www/vhosts/mom.shop/cgi/"
<Directory "/export/srv/www/vhosts/mom.shop/cgi/">
    SSLOptions +StdEnvVars
    Order Allow,Deny
    Allow from All
    Options ExecCGI
    AddHandler cgi-script .cgi
</Directory>

</VirtualHost>
```

Secure Sockets Layer - SSL

- Encrypts segments of network connections
- Used to ensure authenticity and confidentiality
- Uses a certificate to verify the Server's identity
- Typically used for sensitive transactions; ie: when credit card info is entered during online shopping
- Each Virtual Host can be configured with is own SSL Certificate
- As we have seen, SSL is not perfect

Creating an SSL Certificate

- Automated process; no people are involved
- Generate a Private Key
  - Openssl genrsa –des3 –out server.key 1024
- Generate Certificate Signing Request
  - Openssl req –new –key server.key –out server.csr
- Remove Passphrase from Key
  - Openssl rsa –in server.key –out servername.key
- Generate Self-Signed Certificate
  - Openssl x509 –req –days 365 –in server.csr –signkey servername.key –out server.crt

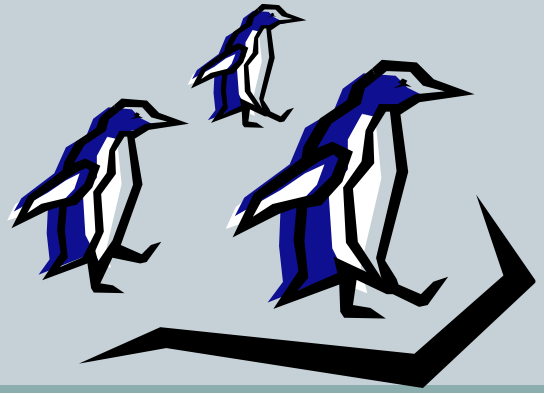
SSL Certificate

Nikto Results Against Compiled Installation

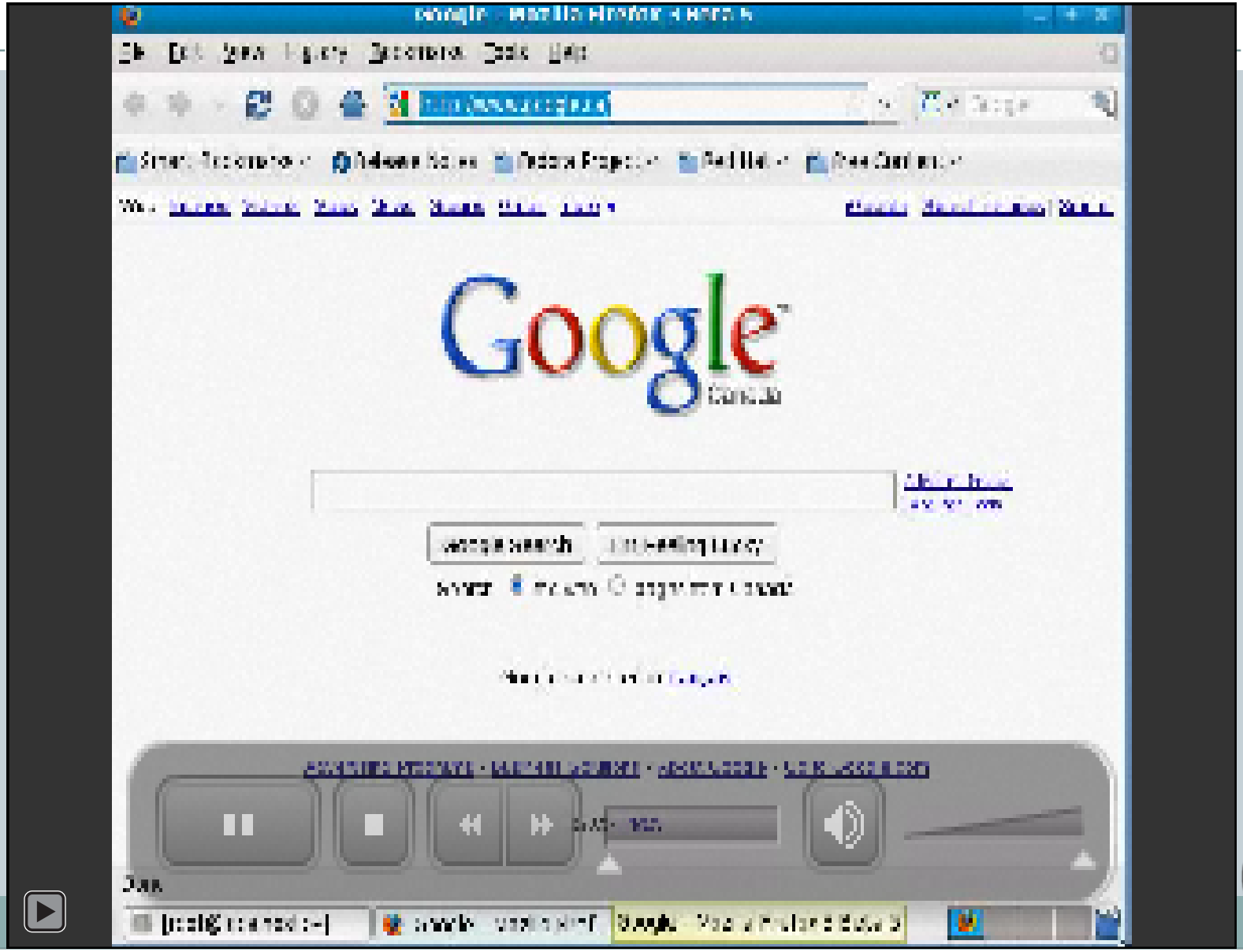
```
[root@localhost nikto]# nikto -h 142.204.16.10 -p 443
---------------------------------------------------------------------------
- Nikto 1.36/1.37 - www.cirt.net
+ Target IP: 142.204.16.10
+ Target Hostname: net1.senecac.on.ca
+ Target Port: 443
+ Start Time: Thu Nov 12 13:16:18 2009
---------------------------------------------------------------------------
- Scan is dependent on "Server" string which can be faked, use -g to override
+ Server: Apache
- Server did not understand HTTP 1.1, switching to HTTP 1.0
+ Server does not respond with '404' for error messages (uses '400').
+ This may increase false-positives.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ 2673 items checked - 1 item(s) reported on remote host
+ End Time: Thu Nov 12 13:19:23 2009 (13 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Damn Vulnerable Web App

- It is a web application that is damn vulnerable.
- It is light weight, easy to use and full of vulnerabilities to exploit.
- It has been developed for the use of information security professionals and students to test the tools in a legal environment.



Damn Vulnerable Web App



SQL Injection

What is SQL Injection?

- A code injection technique that exploits a security vulnerability occurring in the database layer of an Application

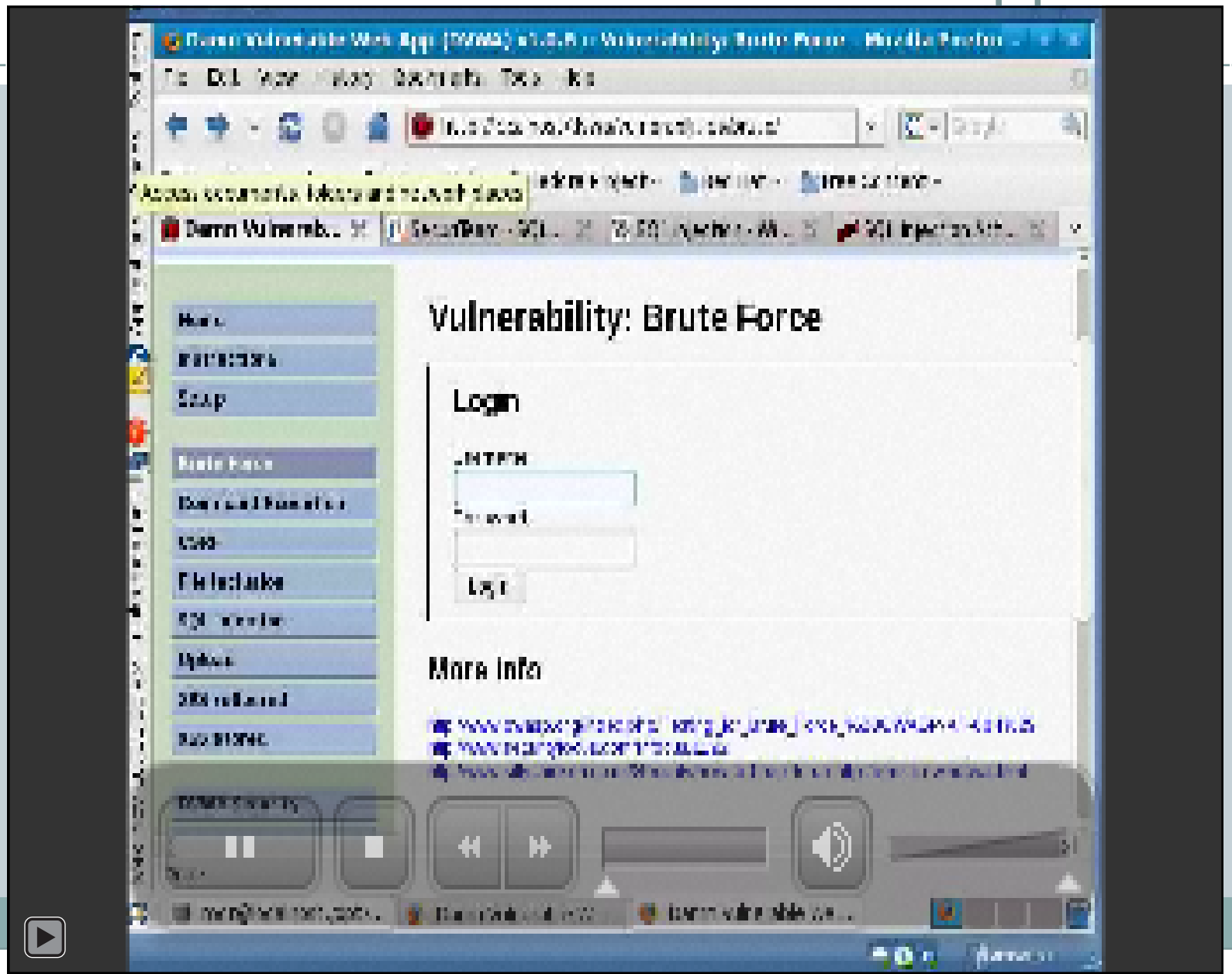
When does SQL Injection Occur?

- When user input is incorrectly filtered for string literal escape characters embedded in SQL statements

This causes unexpected execution of unauthorized code

Damn Vulnerable Web App

Damn Vulnerable Web App



SQL Injection Examples

- **Incorrectly filtered escape characters**
  - statement = "SELECT * FROM users WHERE name = '" + userName + "';"
- **Incorrect type handling**
  - statement := "SELECT * FROM data WHERE id = " + a_variable + ";"
- **Conditional responses**
  - SELECT booktitle FROM booklist WHERE bookId = 'OOk14cd' AND 1=1;

Protecting MySQL

- Every piece of data supplied by a user must be validated to be sure it does not contain information that is not expected.
- User can supply data through the following:
  - Web form
  - Through HTTP Post
  - CGI parameters

Conclusion

- Default Installations are dangerous and should be avoided
- Compile and configure web servers with security in mind
- Limit permissions of server so that a take over does not result in root privileges
- Use tools such as Nikto and Damn Vulnerable Web App to test vulnerability of your severs
- Questions?