Topics in Web Application Security

BlackHat Seattle, 2003
Introductions

Jeremiah Grossman
Founder and CEO, WhiteHat Security

Bill Pennington
Senior Engineer, WhiteHat Security
Agenda

- Web Server Fingerprinting
- Cross Site Tracing - XST
- Web Application Forensics
Why Fingerprint?

- Determine the specific version of the target web server.
- Determine the configuration settings.
- Develop countermeasures to fingerprinting.
- Make patch delivery easier.
Send the same HTTP request and get different responses

Perform a single or standard set of HTTP request towards a web server.
The varied differences in the responses will allow for accurate fingerprinting.
The Common Web Servers
January 2003

- Apache
- Microsoft
- Zeus
- SunOne

Source: Netcraft
bash-2.05a$ nc www.ebay.com 80
OPTIONS * HTTP/1.1
Host: www.ebay.com

HTTP/1.1 200 OK
Server: Microsoft-IIS/4.0
Date: Thu, 06 Feb 2003 05:29:08 GMT
Connection: close
Public: OPTIONS, TRACE, GET, HEAD, POST, PUT, DELETE
Content-Length: 0

bash-2.05a$ _
Servers with no banner

"...paranoid so you don't have to be"

rain forest puppy | rfpolicy v2 | cipherwar
Servers with no banner

bash-2.05a$ nc www.wiretrip.net 80
HEAD / HTTP/1.0
HTTP/1.1 200 OK
Date: Thu, 06 Feb 2003 04:46:28 GMT
Pragma: no-cache
Cache-control: no-cache
Connection: close
Content-Type: text/html

bash-2.05a$ _
bash-2.05a$ nc www.ebay.com 80
OPTIONS * HTTP/1.1
Host:www.ebay.com

HTTP/1.1 200 OK
Server: Microsoft-IIS/4.0
Date: Thu, 06 Feb 2003 05:29:08 GMT
Connection: close
Public: OPTIONS, TRACE, GET, HEAD, POST, PUT, DELETE
Content-Length: 0

bash-2.05a$ _
bash-2.05a$ nc www.ebay.com 80
OPTIONS * HTTP/1.1
Host:www.ebay.com

HTTP/1.1 200 OK
Server: Microsoft-IIS/4.0
Date: Thu, 06 Feb 2003 05:29:08 GMT
Connection: close
Public: OPTIONS, TRACE, GET, HEAD, POST, PUT, DELETE
Content-Length: 0

bash-2.05a$  _
IIS 5.0

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Thu, 06 Feb 2003 05:18:32 GMT
P3P: CP="BUS CAO CNT COM CUR DEV DSP INT NAV OUR PSA PSD SAM STA TAI UNI"
PICS-Label: (pics-1.1 "http://www.icra.org/ratingsv02.html" l r (cb 1 lz 1 nz 1
oz 1 vz 1) "http://www.rsac.org/ratingsv01.html" l r (n 0 s 0 v 0 l 0))
P INT NAV OUR PSA PSD SAM STA TAI UNI"
Expires: Thu, 01 Dec 1994 8:00:00 GMT
Set-Cookie: SITESERVER=ID=b8966420548a464193b1d96cf648efad; domain=.dell.com; pa
th=/; expires=Wed, 06-Feb-2008 05:18:32 GMT;
Set-Cookie: SITESESSION=ID=b8966420548a464193b1d96cf648efad; domain=.dell.
.com; path=/;
Content-Length: 0
Accept-Ranges: bytes
DAV: <DAV:sql>
DAV: 1, 2
Public: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIN
D, PROPPATCH, LOCK, UNLOCK, SEARCH
Allow: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND,
, PROPPATCH, LOCK, UNLOCK, SEARCH
Cache-Control: private

bash-2.05a$ _
Quick Check

IIS 4.0 - Public: OPTIONS, TRACE, GET, HEAD, POST, PUT, DELETE

IIS 5.0 - Public: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH

We can now differentiate between IIS 4.0 and IIS 5.0 and between Apache and IIS!
Apache 1.3.x

bash-2.05a$ nc www.netcraft.com 80
OPTIONS * HTTP/1.1
Host: www.netcraft.com

HTTP/1.1 200 OK
Date: Thu, 06 Feb 2003 04:56:27 GMT
Server: Apache/1.3.26 (Unix) mod_perl/1.27
Content-Length: 0
Allow: GET, HEAD, OPTIONS, TRACE
Connection: close

bash-2.05a$ _
bash-2.05a$ nc www.apache.org 80
OPTIONS * HTTP/1.1
Host:www.apache.org

HTTP/1.1 200 OK
Date: Thu, 06 Feb 2003 05:01:59 GMT
Server: Apache/2.0.44 (Unix)
Cache-Control: max-age=86400
Expires: Fri, 07 Feb 2003 05:01:59 GMT
Allow: GET,HEAD,POST,OPTIONS,TRACE
Content-Length: 0
Content-Type: text/plain
Quick Check

Apache 1.3.x - Allow: GET, HEAD, OPTIONS, TRACE

Apache 2.0.x - Allow: GET, HEAD, POST, OPTIONS TRACE

We can now differentiate 1.3.x and 2.0.x because of the added POST OPTION.
bash-2.05a$ nc www.hammerofgod.com 80
OPTIONS * HTTP/1.1
Host: www.hammerofgod.com

HTTP/1.1 200 OK
Date: Thu, 06 Feb 2003 05:35:18 GMT
Content-Length: 0
Accept-Ranges: bytes
DAV: <DAV:sql>
DAV: 1, 2
Public: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH
Allow: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH
Cache-Control: private
Server: <img src="http://www.hammerofgod.com/images/skull.gif">
bash-2.05a$ nc www.telewest.co.uk 80
OPTIONS * HTTP/1.1
Host: www.telewest.co.uk

HTTP/1.1 200 OK
Server: Netscape-Enterprise/3.6 SP2
Date: Thu, 06 Feb 2003 16:36:37 GMT
Content-length: 0
Public: HEAD, GET, PUT, POST
Netscape 4.1

bash-2.05a$ nc central.sun.net 80
OPTIONS * HTTP/1.1
Host:central.sun.net

HTTP/1.1 200 OK
Server: Netscape-Enterprise/4.1
Date: Thu, 06 Feb 2003 16:25:08 GMT
Allow: HEAD, GET, PUT, POST, DELETE, TRACE, OPTIONS, MOVE, INDEX, MKDIR, RMDIR
Content-length: 0

bash-2.05a$
bash-2.05a$ nc www.iplanet.com 80
OPTIONS * HTTP/1.1
Host:www.iplanet.com

HTTP/1.1 200 OK
Server: Netscape-Enterprise/6.0
Date: Thu, 06 Feb 2003 16:32:32 GMT
Allow: HEAD, GET, PUT, POST, DELETE, TRACE, OPTIONS, MOVE, INDEX, MKDIR, RMDIR
Content-length: 0
Adequate Entropy

The results from the sampling of HTTP output using only “OPTIONS *” provided enough data to start fingerprinting.
## Server Responses

<table>
<thead>
<tr>
<th>Server</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Microsoft-IIS/4.0</strong></td>
<td>Public: OPTIONS, TRACE, GET, HEAD, POST, PUT, DELETE</td>
</tr>
</tbody>
</table>
| **Microsoft-IIS/5.0**                            | Public: OPTIONS,TRACE,GET,HEAD,DELETE,PUT,POST,COPY,MOVE,MKCOL,PROPFIND,PROPPATCH,LOCK,UNLOCK,SEARCH  
Allow: OPTIONS,TRACE,GET,HEAD,DELETE,PUT,POST,COPY,MOVE,MKCOL,PROPFIND,PROPPATCH,LOCK,UNLOCK,SEARCH |
| **Apache/1.3.26 (Unix)**                         | Allow: GET, HEAD, OPTIONS, TRACE                  |
| **Apache/2.0.41-dev (Unix)**                      | Allow: GET,HEAD,POST,OPTIONS,TRACE                |
| **Oracle9iAS/9.0.2 Oracle HTTP Server**          | Allow: GET, HEAD, OPTIONS, TRACE                  |
| **Oracle9iAS-Web-Cache/9.0.2.0.0 (N)**           | Allow: GET, HEAD, OPTIONS, TRACE                  |
| **Netscape-Enterprise/3.6 SP2**                   | Public: HEAD, GET, PUT, POST                      |
| **Netscape-Enterprise/4.0**                       | Allow: HEAD, GET, PUT, POST                       |
| **Netscape-Enterprise/4.1**                       | Allow: HEAD, GET, PUT, POST, DELETE, TRACE, OPTIONS, MOVE, INDEX, MKDIR, RMDIR |
| **Netscape-Enterprise/6.0**                       | Allow: HEAD, GET, PUT, POST, DELETE, TRACE, OPTIONS, MOVE, INDEX, MKDIR, RMDIR |
If the server allows and supports the “OPTIONS” HTTP Request Method, then with a reasonable level of certainty, we can conclude what the major version number is for a popular web server. The “Server” response header is no longer necessary to determine what a web server is running.
Other Request Methods

- Server Specific Methods
  - Track - IIS only method
- Various HTTP response codes
  - ///<dir> will return 400 status code on some Apache versions
- Various HTTP Status messages
  - Alternating capitalization
Research is not complete!

Larger pool of HTTP Requests

More requests allow closer and more detailed accuracy of web server fingerprinting.
Fingerprinting Countermeasures

- Microsoft IIS
  - URL Scan
  - Secure IIS
  - Server Mask
- Apache
  - mod_rewrite
  - httpd.conf changes
  - source code modifications
Add the following line to your URLScan.ini file:

```
[options]
UseAllowVerbs=0

[Deny Verbs]
OPTIONS
```

Caution! Can cause some applications to break. (Frontpage, OWA)
Add the following to your httpd.conf

```
RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^OPTIONS
RewriteRule .* - [F]
```
Questions?
Agenda

- Web Server Fingerprinting
- Cross Site Tracing - XST
- Web Application Forensics
Cross-Site Tracing

A variation of cross-site scripting that increases the threat exposure.

What can XST do that XSS cannot?

Bypass HTTPOnly Restrictions
Access to Basic Authentication Credentials
Access to NTLM Credentials

A web application is no longer required to cross-site script a user if the web server supports the TRACE request method.

Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Exploit Requirements

Cross Site Scripting
A vulnerable web application
A user that clicks on a link or views malicious content.

Cross Site Tracing
Web server that supports the trace request
A place to host the XST code
Cross domain bypass bug (if cross domain is required)
Steps of Cross-Site Scripting

Attacker inserts code into a site or sends a malicious HTML link to a user.

User views the malicious content or clicks on the malicious link.

Malicious code is executed with the hosting domain context, granting access to the cookie data.

Cookie data is passed off domain to a third-party.
Steps of Cross-Site Tracing

Attacker inserts code into a target site or hosts the code on a controlled web page.

User views the web pages and malicious code executes within the browser.

Code directs the browser to send a TRACE request to a target domain.

Cookie, Basic Authentication, and NTLM credentials are sent back to the browser within the HTML Body.

Authentication information is sent to a third-party.
XST Points to Remember

This is a multi-platform multi-technology issue.

Not restricted to ActiveX.

Flash, Java, etc.
General Remedies

1. Sufficiently patch all web browsers against known domain restriction bypass flaws. This is more important part of security policy now more than ever.
2. Disable or disallow the TRACE Request method on production and development (unless needed) web servers.
3. Web server vendors should update their web server packages to disable TRACE by default.
4. Web server vendors should inform their users on how to disable or disallow TRACE on existing web servers.
5. ActiveX controls supporting arbitrary HTTP request should be marked unsafe for scripting by default. Other such technology vendors (Flash, Java, Shockwave, VBScript, etc.) should attempt to implement greater security mechanisms regarding disallowing unauthorized HTTP requests.

Users have the ability to disable all active scripting and increase the safety of their credentials. However, this may negatively impact the functionality of many web sites.
Server Specific

Server Specific (Resolutions should be confirmed by appropriate vendor)

IIS
- URL Scan

Apache
- Source Code Modification
- Mod_Rewrite Module

RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* [F]

(Thank you to Rain Forest Puppy)

** The Limit or LimitExcept directive in the httpd.conf file does not appear to be able to restrict TRACE. **
Microsoft IIS URLScan

Add the following line to your URLScan.ini file:

```
[options]
UseAllowVerbs=0

[Deny Verbs]
TRACE
```

Caution! Can cause some applications to break.
(Frontpage, OWA)
Apache mod_rewrite

Add the following to your httpd.conf

```plaintext
RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
```
XST Demo
Agenda

- Web Server Fingerprinting
- Cross Site Tracing - XST
- Web Application Forensics
Forensics

**forensics** (f -r nsks, -z ks)

n. (used with a sing. verb)

1. The art or study of formal debate; argumentation.
2. The use of science and technology to investigate and establish facts in criminal or civil courts of law.
Why?

I could not find any good tools
I only found one document and it was a marketing document.
I was bored...
What are Web Attacks?

- SQL Injection
- Cross Site Scripting
- Parameter Tampering
- Directory Traversal
- Various Web Server Flaws
  - Unicode
  - Double Decode
  - SSL Overflows
Avenues of Attack

Port 80 - Clear text, easy to watch with an IDS system

Port 443 - SSL encrypted can be watched with an IDS but advanced configuration is required, often not done
HTTP Request

Get Request
Easy everything is logged

POST Request
Only the Path is logged
bummer...
Yes they will generally detect Nimda/CodeRed (Unicode/double decode) attacks.

You could write rules to detect some basic attacks http://www.cgisecurity.com/web-attacks.rules

It is almost impossible to detect certain attacks with a NIDS
Log Files

IIS Log Files
Stored in %winnt%/system32/logs/<servicename>

Typically - C:/winnt/system32/logs/w3svc/*.log
IIS Log File Formats

IIS log file format

UserIP, UserName, Date, Time, Service, Computer Name, ServerIP, Time Taken, Bytes Sent, Bytes Received, Status Code, Windows Status, Request Type, Target, Parameters

IIS can log to IIS, W3C Extended and NCSA common file format
File name determines type of log

IIS format log files begin with in

W3C extended log files begin with ex

NCSA log files begin with nc
Apache Log File Locations

%apache_home% /log/access.log
Apache Log File Formats

By default Apache logs to NCSA common format or the combined log file format:

clientip,ident,username,date/time,request,status,bytes sent
Performance Tip

Separate log files on a separate physical disk from content
Problems with web server logs

They are generally very large
Contain lots of non-security related entries

POST data is rarely logged

Many attacks can occur via POST request

Some attacks can simply not be determined by log files
Log File Sizes

www.whitehatsec.com
From Jan 22nd - Feb 19th
466,829 lines

eCom/Online Gaming
Feb 1 - Feb 7
1,198,140 lines
# Analysis of log content

<table>
<thead>
<tr>
<th>File Extension</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Odd Request&quot;</td>
<td>4,183</td>
</tr>
<tr>
<td>Odd URL</td>
<td>107</td>
</tr>
<tr>
<td>.css</td>
<td>15,596</td>
</tr>
<tr>
<td>.cgi</td>
<td>786</td>
</tr>
<tr>
<td>.Html</td>
<td>235,670</td>
</tr>
<tr>
<td>.pdf</td>
<td>4,606</td>
</tr>
<tr>
<td>Images</td>
<td>401,174</td>
</tr>
</tbody>
</table>

© 2002 by WhiteHat Security, Inc.
What defines a bad request?

401 Response codes - Authentication required

500 Response codes - Server error, SQL injection

200 Response code - could be the worst of all, success
Weird Characters

Some things should generally not be in a URL

‘ < > * .. etc...
Odd Request Methods

99% of applications use only GET and/or POST

Why is someone HEADing me and should I let them?
## Odd Request Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Head</strong></td>
<td>Just returns server header no data. Used to probe for the existence of files</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td>Used to determine the capabilities of a web server and fingerprinting</td>
</tr>
<tr>
<td><strong>Trace</strong></td>
<td>Used for diagnostics. A possible attack vector XST</td>
</tr>
<tr>
<td><strong>Any WebDAV method (PropFind...)</strong></td>
<td>Used for managed web content. (Frontpage) and in some more robust web applications (OWA)</td>
</tr>
</tbody>
</table>
Introducing the HillBilly

Not really an analysis tool, more of a data reduction tool

Searches for odd URLs
500 errors
Strange request methods
HillBilly Syntax

./hillbilly.pl -t <common,iis4,iis5> -l <logfile> -f <outputfile> -g (Look for odd GET request) -p (Look for 500 errors) -o (Look for odd request methods)
Odd URL search

```bash
./hillbilly.pl -t common -l access_log -g
```

Regex = `/[^A-Za-z0-9\./\\?(%20)=_]&-]/`

Looks for request that contain characters other than these

Will find Unicode, Double Decode, Cross Site Scripting, SQL Injection, Command Execution, Directory Traversal in a GET request.
Weird Character Output
Weird Character Reduction

```
bash-2.05a$ ./hillbilly.pl -l access_log -t common -g|wc -l
  107
bash-2.05a$ wc -l <access_log
  466845
bash-2.05a$ _
```
Ecom log file reduction

Log file reduced from 1,198,140 to 285,314
500 Errors

./hillbilly.pl -l access_log -t common -p

Looks for any request method that generates a 500 error

500 errors can indicate a SQL injection attack

Large numbers of 500 errors from a single user over a short period can indicate an attack

Check application server and SQL server logs

Your time is synced right?
Ecom odd request types

Log file reduced from 1,198,140 to 0
Odd Request types

./hillbilly.pl -l access_log -t common -o

Looks for any request type other than GET or POST
Can point out probing request or fingerprinting attempts
Ecom odd request types

Log file reduced from 1,198,140 to 2269
Prepare for the worst

Configuring web server log files
Know where they are!

Additional utilities
URLScan (IIS)
mod_protect (Apache)
Code Seeker (Cross platform)
Other logs

SQL server logs
Make sure they are on and at least logging errors
Listen to your DBA whine about performance!

Application Server Logs
Make sure they are on
Make sure you understand them
If you can’t sync it at least try to get it close

You should really try to sync it, really
Using HillBilly as an IDS

Danger this is untested!!!

Danger this is probably insecure!!!

Apache

CustomLog "|/usr/bin/hillbilly.pl -t common -l -g >> /var/log/hillbilly.log" common
Future Plans

Write code others can read and use

Recognize and automatically parse web server logs

Add option to only look for successful request

Use some sort of magic to profile the log file to look for truly deviant request

Pretty output
Questions