HTTP Fingerprinting

The next generation

Eldar Marcussen
Stratsec

eldar.marcussen@stratsec.net
Agenda

• Why
• HTTP
• Fingerprinting theory
• The next generation
• Demo
• Conclusion
@wireghoul

• Pentester
• Blogger
• Husband
• Father
• Geek
Why

- Understanding of remote environment
- Load balancer vulnerabilities
  - HAProxy DoS (SA44083)
  - Pound Format String vuln (SA11528)
  - Pound Buffer overflow (CVE-2005-1391)
  - Varnish DoS (SA33852)
  - mod_proxy Integer overflow (CVE-2010-0010)
Prior work

• HMAP: A Technique and Tool For Remote Identification of HTTP Servers - Dustin Lee

• Detecting and Defending against Web-Server Fingerprinting - Dustin Lee, Jeff Rowe, Calvin Ko, Karl Levitt

• HTTPrint; An Introduction to HTTP Fingerprinting - Saumil Shah

• Identifying web servers – Jeremiah Grossman

• More
Existing tools

- HTTPrint
- Hmap
- Waffit/wafw00f
- Lbd
- Halberd
- More
HTTP Basics

HTTP 0.9 - http://www.w3.org/Protocols/HTTP/AsImplemented.html
HTTP 1.0 - RFC1945
HTTP 1.1 - RFC2616
HTTP 0.9 Request

GET /CRLF
HTTP 0.9 Response

<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
</body></html>
HTTP 1.0 Request

GET / HTTP/1.0 CRLF
User-Agent: Mozilla/4.0 CRLF
CRLF
HTTP/1.0 200 OK


Server: Apache/2.2.14 (Ubuntu)

ETag: "a711f-b1-4a2e722183700"

Content-Length: 177

Connection: close

Content-Type: text/html

<html><body><h1>It works!</h1><p>This is the default web page for this server.</p></body></html>
HTTP 1.1 Request

GET/ HTTP/1.1CRLF
Host: localhostCRLF
User-Agent: Mozilla/4.0CRLF
CRLF
HTTP 1.1 Response

HTTP/1.1 200 OK


Server: Apache/2.2.14 (Ubuntu)

ETag: "a711f-b1-4a2e722183700"

Content-Length: 177

Connection: close

Content-Type: text/html

<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
METHOD Example

HEAD / HTTP/1.0 CRLF
CRLF

POST / HTTP/1.0 CRLF
Content-Type: application/x-www-form-urlencoded CRLF
CRLF
id=1&name=test
Fingerprinting

Analysis of responses

- Semantic
- Lexical
- Syntactical
Semantic analysis

How the agent interprets a request.

• Range: 1-, 2-, 3-,
• HEAD SHOULDERS KNEES AND TOES
Lexical analysis

Specific words, phrases and punctuation in responses.

• HTTP/1.1 501 Unknown or unimplemented http action
• HTTP/1.1 501 Method Not Implemented
• HTTP/1.0 501 Not Implemented
• HTTP/1.0 501 Unsupported method ('POST')
Syntactical analysis

Ordering and context of words, phrases, header, etc.

• ‘Server’ header occurs after ‘Date’ header
• ETag format
Detecting Load balancer

Common indicators

- Rejects unusual HTTP requests
- HTTP1.0 responses to HTTP/0.9 requests
- HTTP/1.0 400 error responses
- Adds identifying headers
Detecting WAF

Common indicators

• Rejects unusual HTTP requests
• Accepts unusual HTTP requests
• Rejects valid HTTP requests with “suspicious” characters (./, ../)
Detecting web servers

Common indicators

• Server headers
• Gracefully handles HTTP/0.9
• Defaults to HTTP/1.1 responses
• Syntactical evidence (ETag header)
Enumeration

Detecting back-ends / server pools

• DNS
• Handle debugging headers
• Compare responses from large number of requests
BUT WAIT THERE’S MORE
Profiling configuration

Easy

- Timeout
- Application headers

Also easy?

- Configured modules
- Script bindings
Apache handlers

Allows module to handle request METHOD

• Many modules don’t enforce strict verb checks
• Can be used to remotely detect modules and script bindings
• Can bypass authentication
• Don’t always work
Demo
Demo

```bash
root@bt:~:/lbmap# ./lbmap2 http://www.<redacted>.com
lbmap - http fingerprinting tool
Eldar "Wireghoul" Marcussen - Scanning http://www.<redacted>.com

$VAR1 = 'signaturematch';
$VAR2 = {
    'F5 WAF' => 1
};
$VAR3 = 'signature';
$VAR4 = '01BCBC--A0--99A0BCA0BCA0BCA0BCBBCBCBCCBCBCBCBCBCBCBCBCBCBCBCBCBCBCBCBCBCB0A0--';
$VAR5 = 'webserver';
$VAR6 = {
    'F5' => 7,
    'Apache' => 18
};
```
Demo

```bash
root@bt:~/lbmap# ./lbmap2 http://exampleproject.org
lbmap - http fingerprinting tool
Eldar "Wireghoul" Marcussen - Scanning http://exampleproject.org

$VAR1 = 'signature';
$VAR2 = '01BCBC--99--99BCBC--BCA0BCA0BCBCBCBCBCBCBCBCBCBCBCBCBCBCBCBCBCBCBCB099--';
$VAR3 = 'proxyserver';
$VAR4 = {
  'proxy03.exampleproject.org' => 1,
  'proxy01.phx2.exampleproject.org' => 1
};
$VAR5 = 'webserver';
$VAR6 = {
  'Apache/2.2.15 (Red Hat)' => 4,
  'Apache' => 18
};
```
Demo

```plaintext
root@bt:/~/lbmap# ./lbmap2 http://www.gluedeomap.com
lbmap - http fingerprinting tool
Eldar "Wireghoul" Marcussen - Scanning http://www.gluedeomap.com
$VAR1 = 'loadbalancer';
$VAR2 = {
    'BIGIP' => 4
};
$VAR3 = 'signature';
$VAR4 = '01BCBC-----99BCBC--BCA0BCA0BCBCBCBCCBCBCCBCBCCBCBCCBCBCCBCBCCBAC0----';
$VAR5 = 'backend';
$VAR6 = {
    '10.220.2.22:81' => 1,
    '10.220.2.23:81' => 1,
    '10.220.2.24:81' => 2
};
$VAR7 = 'reverseproxy';
$VAR8 = {
    '1.1 varnish' => 4
};
$VAR9 = 'webserver';
$VAR10 = {
    'Apache/2.2.14 (Ubuntu) Resin/3.1.8' => 2,
    'Apache' => 20
};
```
Summary & Conclusion
Conclusion

• Current fingerprinting does not give complete picture
• Fingerprinting can do more than just identify web agents
• Fingerprinting can be unreliable
• Better tools needed
Tools

Source code and download from

- [https://github.com/wireghoul/lbmap](https://github.com/wireghoul/lbmap)
- Please fork and contribute
Thanks

@stratsec
@owasp
@net__ninja
@tecR0c
@dieinafire23
@smokingjohnson

@csearle
@ivanristic
Shodan HQ
And others...
Questions