

rain forest puppy / hivercon 2002

A close-up, high-contrast photograph of a dog's eye, likely a Weimaraner, with a striking blue iris. The eye is the central focus, looking slightly to the right. The surrounding fur is dark and textured, with some highlights from the lighting. The overall mood is intense and focused.

Web server fingerprinting

rain forest puppy  
hivercon 2002



Note: updated slides are available at:

<http://www.wiretrip.net/rfp/talks/hivercon-2002/>

# What is application fingerprinting?

- Looking for a unique set of characteristics, or 'fingerprint'
- Similar to TCP/IP stack fingerprinting (a la nmap)
- Goes beyond version banners

# How can we use app fingerprinting?

- Identify an application, and even it's version
- Can pierce anonymity (removal of banners, etc)
- Can detect real vs fake applications (emulated services and honeypots)
- May indicate vulnerability existence without exploitation
- Tests may also be used to measure RFC-conformance

# Who can use app fingerprinting?

- Consultants: pen-tests, assessments
- Attackers

More importantly, admins need to be aware that obscuring version banners doesn't provide any measure of security *or* obscurity...



# What can be fingerprinted?

- Anything that interacts with the user (i.e. most network services)
- More interaction typically yields a better fingerprint
- Version identification depends on how code changes between versions (bug fixes may alter the program in a barely noticeable way)
- HTTP is perfect because the protocol has many various aspects that affect processing

# HTTP fingerprinting

- Some web application assessment tools rely on the HTTP banner
- Admins are removing the banner (Urlscan, Apache config, source tweak, etc)
- HTTP protection devices are removing banners (web app firewalls, security proxies, load balancers, etc)
- Some HTTP servers have same banner for multiple versions (IIS)



# HTTP fingerprinting—lots to fingerprint

- rfp.labs web server scanning project, a la Netcraft
- Scanned 3 class A networks looking for web servers
- Found 150,000+ web servers, and many dozens of web server software
- Automated tool can take samples from randomly found servers

# HTTP fingerprinting—the request

GET /default.asp HTTP/1.0

# HTTP fingerprinting—the request

leading whitespace

different valid/invalid methods

filename representations

GET /default.asp HTTP/1.0

type/amount of whitespace separators

different HTTP versions

# HTTP fingerprinting—other stuff

- Headers: special and invalid encodings, plus the return order
- Page responses: returned HTML on 404, 302, etc
- Abnormalities: characteristics due to implementation or other weirdness
- HTTP 0.9 requests: mixed bag of support
- Filename encodings: unicode, double-encode, etc
- Cookies: can reveal what's in the processing stream

# HTTP fingerprinting—header example

GET / HTTP/1.0

Apache

HTTP/1.1 200 OK  
Date: Sat, 23 Nov 2002 21:34:22 GMT  
Server: Microsoft-IIS/5.0  
Connection: close  
Content-Type: text/html

HTTP/1.1 200 OK  
Date: Sat, 23 Nov 2002 21:36:40 GMT  
Server: Apache/1.3.26 (Unix)  
Connection: close  
Content-Type: text/html

IIS

HTTP/1.1 200 OK  
Server: Microsoft-IIS/5.0  
Date: Sat, 23 Nov 2002 21:37:59 GMT  
Connection: Keep-Alive  
Content-Length: 1270  
Content-Type: text/html

HTTP/1.1 200 OK  
Date: Sat, 23 Nov 2002 21:32:50 GMT  
Connection: Keep-Alive  
Content-Length: 1270  
Content-Type: text/html  
Server: Apache/1.3.26 (Unix)

# HTTP fingerprinting for identification

- No banner? Use a fingerprint to determine what it is
- Provides a banner? Use a fingerprint to see if it's truthful or lying
- File extension identified as ASP/PHP? Verify the file handler
- File extension .html? Verify it's not dynamic



# HTTP fingerprinting for versioning

- Remotely identify which service packs/SRPs on an IIS system

```
deb5647b3bf685996436ecc13de08564 IIS5 sp0, sp1  
e1360eacceef8559d403d84a24b4cd209 IIS5 sp2, srp1  
979b3d197cf71be7f98c9d9e9acb61c0 IIS5 srp2, sp3
```

Be able to determine patch/vulnerability level  
*without running an exploit*

# HTTP fingerprinting—what's on the horizon

- Emulated honeypots and services are not good enough
- Vulnerability testing/assessment without triggering the vuln
- HTTP obscurity techniques will be pierced
- Patch level determination through port 80 (for Windows/IIS in particular)
- Potential identification of inline HTTP devices



# Questions?

<http://www.wiretrip.net/rfp/talks/hivercon-2002/>



Bonus tool updates!

- Latest libwhisker version
- Features various bugfixes beyond version 1.5



- Latest whisker version
- Updated scan test database
- Documentation!
- HTML output, logging
- 'Newbie' guided walkthrough (poor man's GUI)
- Incorporates some of the identification techniques discussed







Available for download at:

<http://www.wiretrip.net/rfp/talks/hivercon-2002/>