The Enemy Within: Organizational Insight Through the Eyes of a Webserver

By
Ramece Cave
Research Analyst / Application Developer

Who Am I

Started working in IT Security in 1999 as a Fraud and Abuse Investigator for UUNET. Held various forensic focused roles. Transitioned into Research and Development in 2009 focusing on the areas of malware analysis, reverse engineering and host and network intrusion detection.

Research Interests

- Analysis automation and correlation
- Covert channel analysis and identification
- Threat Intelligence
- Protocol Anomalies
- Mobile Malware

Presentation Overview

This presentation will cover some of the information provided by webservers and how it can reflect on a organizations current security posture regarding its web services. We will also be looking at supporting information from malicious campaigns and information collected on various malware domains, how they all intertwine and evolve into other nefarious practices.

Presentation Outline

- Evolution of the web server role
- Attack roles of web servers
- Summer of Mischief
- Web Server Cause and Effect (The Naughtiness Factor)
- Remote vs. Local Hosting: The Good, The Bad and The Ugly
- A Deeper Look Into Ugly: Proof of Concept
- Closing
- Questions

Evolution of the Web Server Role

- Late 90s – hosting basic web pages, the idea of a web presence not common.
- 2000 to 2006 – Web presence is a must, businesses and organizations highly dependent upon web service availability. Key business component.
- 2007 to Today – Web presence is critical and mandated for survival, down time is unacceptable, can result in a loss of millions of dollars per hour. E-Commerce is now a way of life. We live by the mantra: Make everything faster, better, keep it online, keep it fresh, no matter what.

Web Server Attack Roles

- Usually the victim
- Compromises result in defacements or theft of data from attached databases.
- Highly coveted and guarded, if this were a game of chess the webserver is the King, and everyone protects the king.

The Summer of Mischief

Summer of Mischief: Threat Campaigns Targeted at the US and Abroad

Staring in the beginning of May, 2013 a string of operations were initiated by various threat actor groups. Their objective was to disrupt the US banking and economic systems. Three of the campaigns were:

- OpUSA
- OpPetrol
- Op911

Summer of Mischief Stats

- 1,002 IP Addresses
- 59 Countries
- 595 Providers/Business’
- 2,151 Domains
- 112 Targeted Server Versions
- Majority of target domains were located on web hosting providers

Top Targeted Web Server Platforms

- 763 Apache
- 106 Nginx (Pronounced: Engine-Ex)
- 1435 Microsoft Internet Information Services (IIS)

Why is this important?

- Based on previous events malicious campaigns resulted in large-scale distributed denial of service (DDoS) attacks. During the campaigns, notably OpUSA, no attacks were reported.

- The campaigns were largely reported as a failure

- DOS or DDoS attacks by design serve two purposes:
  - Cause chaos and pandemonium
  - Divert attention to the attack target.

- When preparing for a preemptive attack, typically bandwidth and resource usage are monitored for spikes beyond normal usage. (what about intrusion?)
- History has a tendency to repeat itself.

Summer of Mischief: Apache Fall Out

Apache/2.2.24 (Unix) mod_ssl/2.2.24 OpenSSL/1.0.0-fips mod_auth_passthrough/2.1 mod_bwlimited/1.4

- Single Web Host/VPS Provider
- 70 Compromised Domains
- Affecting four IP Addresses

FrontPage/5.0.2.2635

- 37 IP Addresses
- 22 Providers and Businesses
- 8 Countries
- 58 Domains

Internet Information Services 6.0

- 398 IP Addresses
- 814 Domains
- 49 Countries
- 293 Various business' and service providers

Summer of Mischief:
IIS Fall Out

Microsoft Internet Information Server IIS 5.x

- 161 Domains
- 65 IP addresses
- 19 Countries
- 55 Providers and Businesses

Microsoft Internet Information Services IIS 7.x

- 195 Domains
- 48 IP Addresses
- 36 Providers and Businesses
- 14 Countries

Summer of Mischief: Nginx Fall Out

Nginx

- 59 IP Addresses
- 14 Versions
- 43 Providers and Businesses
- 16 Countries

How the malicious actor might interpret your web server.

Internet Information Services 5.0 (IIS 5.0)

Microsoft-IIS/5.0

- Organizational Reasoning: Default Installation (2000), cutting edge at the time, still works, complacent with results and functionality.
- Malicious Interpretation: “Hello hackme, Unicode exploit anyone?”

Microsoft-IIS/5.0 PHP/5.2.17

- Organizational Reasoning: Intergraded PHP into IIS in 2009 but have done nothing since. Still applicable for our day-to-day operations and needs.
- Malicious Interpretation: Vulnerable PHP and server, were to begin.

Internet Information Services 5.0 (cont)

Microsoft-IIS/5.0 mod_ruid2/0.9

- Organizational Reasoning: Trying to be secure, forward thinking, Unicode vulnerability neutralized (only affects local non-admin user). The main concern of compromise has been resolved.
- Malicious Interpretation: Nice thinking, but your still using an outdated and no longer supported web server with multiple vulnerabilities.

Internet Information Services 6.0

Microsoft-IIS/6.0'

- Organizational Reasoning: Still supported, heavily integrated into our infrastructure. No need to upgrade at the moment, much more secure and robust then 5.0. It's been 11 years, still going strong.
- Malicious Interpretation: Still thinking IIS is secure, probably not patched for overflow and bypass vulnerabilities. Will probably use this server until forced to upgrade. Still heavily used despite security concerns.

Microsoft-IIS/6.0 PHP/5.2.5

- Organizational Reasoning: In 2003 or there about implemented PHP to push web content to the next level. Since 2007 we have been happy with the results and productivity of the server in this configuration. No foreseeable plans to upgrade until needed.
- Malicious Interpretation: Another example of IIS insecurity, this time with added benefit of an outdated and vulnerable version of PHP that further extends my exploit potential. Not as prevalent as the others, but a welcomed addition.

Nginx

Nginx/x.x.x

Organizational Reasoning: Robust, highly configurable and can be used as a reverse –proxy server. Trendy, its not IIS or Apache.

Malicious Interpretation: So many versions, have you read the advisories? Are you using as standalone server or proxy? Are you using uWSGI and Python? Did you secure your host? Your proxy has just become mine, thanks for hosting my malware distribution domain and C&C domain.

Apache

Apache/2.x.x (Unix) mod_ssl/2.2.17 OpenSSL/0.x.x-fips-rhel5 mod_auth_passthrough/2.1 mod_bwlimited/1.4 FrontPage/5.0.2.2635

- Organizational Reasoning: We must provide quick and proven effective means for our customers to deliver their content in an expedient, repeatable, productive manner.

- Malicious Interpretation: You are a webhost provider that implemented a one-solution fits all scenario for your customers sometime in or around 2005. Nothing has been upgraded or patched since, either in fear of service interruption, or lack of oversight.

The Naughtiness Factor: When Webservers Attack

How Webservers Break Bad

Web sites are often compromised due to:

- Vulnerable HTTP Server
- Unpatched or Secured Server

Typically after a website is compromised one of the first actions is to restore any defaced content, and check for deleted or accessed data, patch the security hole, or nuke and rebuild the server. This often addresses the immediate threat, but is not a long term solution.

What if the defacement or compromise is never detected, what else can happen… The answer, the server could be used for hosting malicious content or recruited for other nefarious purposes.

Compromised Servers Distributing Malware

During the past 19 months in my research thousands of web servers have been identified as possible unwilling distributers of malware and other malicious content.

One of those servers was identified as being defaced during the Summer of Mischief campaign outlined earlier. Based on the vital statistics provided by the server, the initial compromised occurred one month prior to the campaign's official start. The following slides are the current stats as of 03/11/2014 regarding the aforementioned servers. Many of the servers identified also preformed rudimentary scans of numerous ports including:

1433 – MSSQL
445 – Microsoft Directory Services
3389 – Remote Desktop Protocol
3306 - MySQL

IIS 5.0 and 6.0 Compromised (Possibly) Server Stats

5.0

- 179 Domains
- 44 IP Addresses
- 452 Windows PE32 Binaries

6.0

- 1,598 Domains
- 26 IP Addresses
- 11,492 Windows PE32 Binaries

Apache FrontPage and Nginx Compromised (Possibly) Server Stats

Apache w/FrontPage Extension

- 656 Domains
- 297 IP Addresses
- 452 Windows PE32 Binaries

Nginx

- 3,892 Domains
- 625 IP Addresses
- 48,924 Windows PE32 Binaries

Remote vs. Local Hosting:
The Good, Bad and the Ugly

The Good, The Bad, The Ugly

The Good

- Less administration costs and time
- Lower internal IT overhead

The Bad

- Not protected by IT security protocol
- Shared hosting

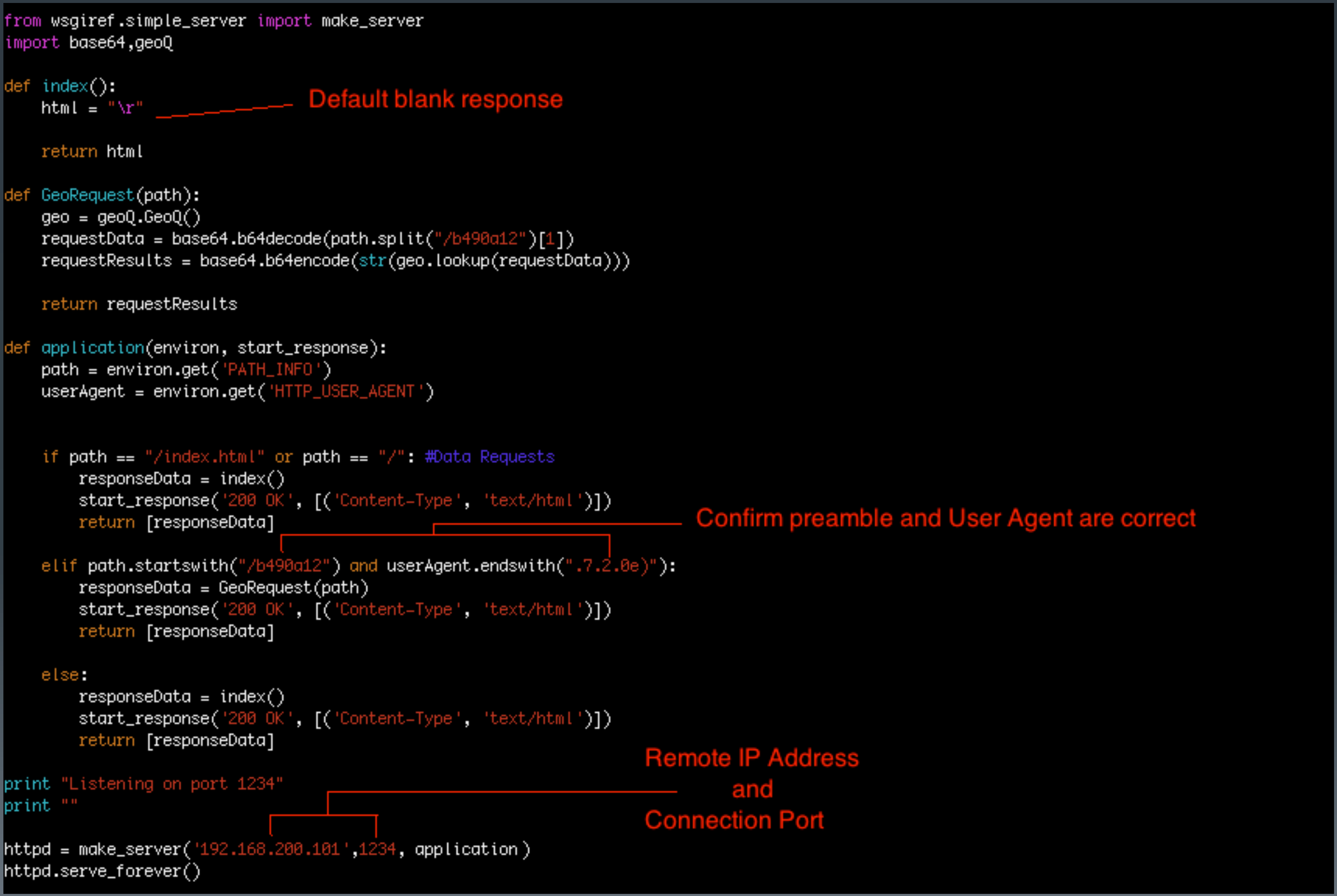
The Ugly

- If one site is compromised, all are at risk
- Security patches may not applied in a timely manner, if ever

A Deeper Look Into Ugly: Proof of Concept

Once the remote server are compromised they take on an entirely new threat. Since most providers only filtered limited incoming and outgoing traffic, usually based on the acceptable use policy (AUP). This leaves the potential for new possibly unwanted services being introduced on top of the webserver. The following slide demonstrates a non-malicious web application running on another webserver to process incoming requests for geolocation information adhering to a specific format. **Many C&C servers operate in this manner**.

- No administrator or root permissions are required for ports above 1024
- The application can be written in any number of already installed web capable languages for example: Python, Ruby, and Node.js
- The application can be written to exploit a local vulnerability for even greater access
- Incoming and outgoing traffic permitted due flexible or no firewall implementation.



```
from wsgiref.simple_server import make_server
import base64,geoQ

def index():
    html = "\r"

    return html

def GeoRequest(path):
    geo = geoQ.GeoQ()
    requestData = base64.b64decode(path.split("/b490a12")[1])
    requestResults = base64.b64encode(str(geo.lookup(requestData)))

    return requestResults

def application(environ, start_response):
    path = environ.get('PATH_INFO')
    userAgent = environ.get('HTTP_USER_AGENT')


    if path == "/index.html" or path == "/": #Data Requests
        responseData = index()
        start_response('200 OK', [('Content-Type', 'text/html')])
        return [responseData]

    elif path.startswith("/b490a12") and userAgent.endswith(".7.2.0e)"):
        responseData = GeoRequest(path)
        start_response('200 OK', [('Content-Type', 'text/html')])
        return [responseData]

    else:
        responseData = index()
        start_response('200 OK', [('Content-Type', 'text/html')])
        return [responseData]

print "Listening on port 1234"
print ""

httpd = make_server('192.168.200.101',1234, application)
httpd.serve_forever()
```

Default blank response

Confirm preamble and User Agent are correct

Remote IP Address
and
Connection Port

Closing

In this presentation we covered the types of insight a malicious actor can obtain from identifying header responses. This information is given with every request to a site, unless specifically modified. We also explored how this information is still actively being exploited after nearly 15 years since first discovered. In both coordinated malicious campaigns and identified malware distribution servers.

Moving forward there is no 100% solution for hosting web content, there are pros and cons for local vs. remote, which server to use and which language to implement.

Suggestions and Recommendations

- Keep servers patched and updated
- Upgrade when needed
- Monitor for suspicious applications and connections
- Limit access rights to non-privileged users
- When considering hosting providers, ask about firewall, patch management and update polices.
- Be diligent in your security practices

Q & A

Contact Information
http://www.n00dle.org
rrcave@n00dle.org
@feedbrain