PriveonLabs Research

Cisco Security Agent Protection Series:

Web Server Protection with CSA

HTTP Explorer Directory Traversal

Zach Brewer
Security Consultant

Overview

Variations of web server directory traversal attacks have plagued both Windows and UNIX web servers. In its most basic form, a directory traversal attack is an attempt to access files on a system that should not be accessible via a web server component. Directory Traversal attacks usually result from improper sanitization of user input strings by the web server software. A successful directory traversal attack can result in disclosure of sensitive system information, privilege escalation, and/or system compromise.

An example of full system compromise via directory traversal is CVE-2000-0884 (IIS Unicode Directory Traversal). CVE 2000-0884, arguably the most famous directory traversal attack to date, affected both Microsoft IIS versions 4 and 5. When successfully exploited, CVE-2000-0884 can result in an attacker's ability to run arbitrary commands under system account privileges – an example of the potential severity of directory traversal vulnerabilities.

In this document, we will outline CSA's ability to defend against directory traversal attacks by utilizing HTTPExplorer, a basic open source web server for the Windows XP platform. On 12/21/2006, exploit code was published to milworm.com outlining a basic directory traversal attack in HTTP Explorer 1.02. Although the aforementioned exploit has been fixed in version 1.03, this vulnerability is an excellent example of directory traversal attacks that are all too common in web server software.

Exploit Process Overview

The HTTPExplorer 1.02 directory traversal attack is a remotely exploitable vulnerability that uses the familiar ../../ (dot dot slash) method of directory traversal. Exploit code that is currently available simply discloses the contents of the boot.ini file (C:\boot.ini) although the traversal method could be modified to disclose any sensitive file on the system. Similarly, changing the code to download malware to the system via TFTP/FTP or other built-in windows XP file transfer method is also theoretically possible.

PoC Exploit Testing Overview

Testing Environment

The Priveon test lab consisted of one windows XP system running the HTTPExplorer web server application and one simulated "attack system" running windows XP and Perl. The Windows XP system running HTTPExplorer was placed in the Cisco Security Agent (CSA) "Servers – All Types" and "IIS Web Servers" groups. Additionally, we added HTTPExplorer to the CSA "IIS Web Server" Application class.

While HTTPExplorer is obviously not an IIS web server, the IIS Web Server application class and group enforces data access control rules that prevent common string patterns used in windows directory traversal attacks. Ideally in a production CSA environment, we would add a new application class for HTTPExplorer (or other custom web server software) and add the HTTPExplorer application class to the appropriate web server protection rules.

Cisco Security Agent Version Tested:

- CSA 5.1.0.69 (Default CSA Policies as shown in Figure 1)

Figure 1: Protected Host CSA Policies from v5.1.0.69

Group Name	Version	Description	Policies
<All Windows>		Auto-enrollment group for Windows hosts	2 policies
Policy Name	**Version**	**Description**	**Rule Modules**
Application Classification	5.1 r69	Base policy for behavioral classification of applications.	3 modules
Operating System - Base Permissions - Windows	5.1 r69	Basic permissions for Windows OS	2 modules
Servers - All types	5.1 r69	Default group for systems that install the Server agent kit	5 policies
Policy Name	**Version**	**Description**	**Rule Modules**
Agent UI control	5.1 r69	Policy which governs Agent User Interface	1 module
General application - Basic Security - Windows	5.1 r69	Basic, Application independent security policy for Windows	3 modules
Installation Applications - Windows	5.1 r69	Software Installers for Windows	4 modules
Operating System - Base Protection - Windows	5.1 r69	Basic protection for Windows OS	6 modules
Virus Scanner - Windows	5.1 r69	Application enforcement policy for virus scanner software.	1 module
Servers - IIS Web Servers	5.1 r69	Systems running Microsoft IIS web server	1 policy
Policy Name	**Version**	**Description**	**Rule Modules**
Web Server - Microsoft IIS - Windows	5.1 r69	Application enforcement policy for IIS web server software.	2 modules
Systems - Test Mode [test]	5.1 r69	Systems operating in test mode	0 policies

Exploit PoC Testing Results – Unprotected System

The testing process involved Proof of Concept code obtained from Milworm.com which discloses the contents of boot.ini (figure 2). Additional variations of the HTTPExplorer traversal vulnerability were also tested as seen in figure 3. By utilizing the traversal pattern, attackers could potentially access system resources including the SAM and SYSTEM files. Once the SAM file has been compromised, attackers can compromise local system accounts using only the offline SAM file and a related cracker such as Abel, 1ophtcrack, or John the Ripper.

Figure 2: Milworm.com code disclosing the contents of boot.ini

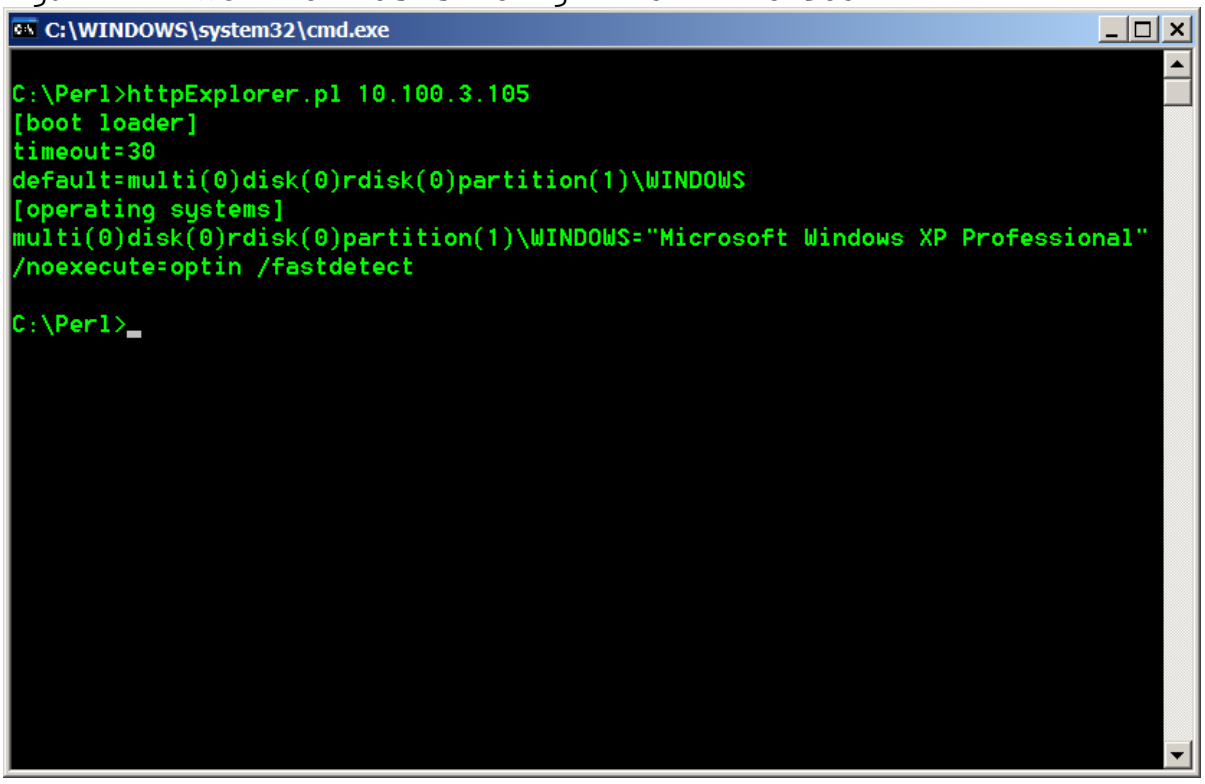
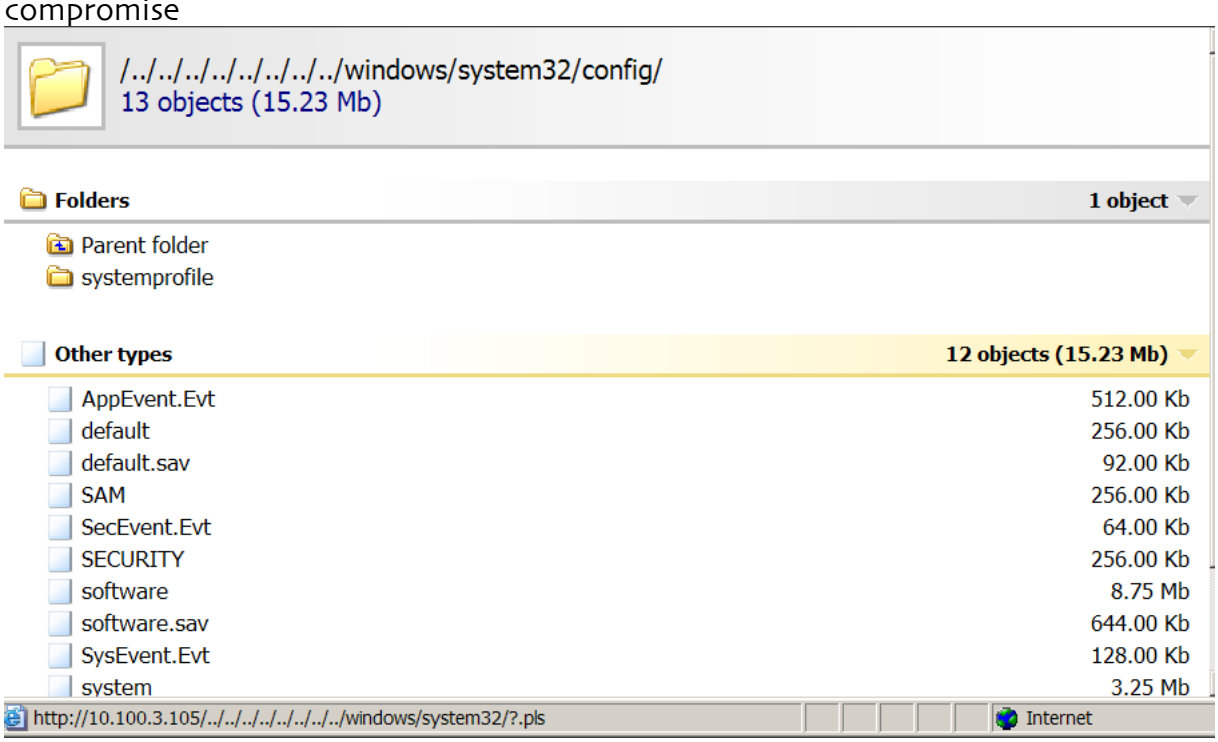


Figure 3: Variation of content disclosure leading to SAM (and local account) compromise



CSA Protected Exploit PoC Testing Results

After testing the Milworm.com PoC code, the attacks were attempted with the Cisco Security Agent installed on the lab virtual hosts.

CSA successfully prevented access from the server component of HTTPExplorer to the System API as seen in figures 4 and 5. Specifically, System API rules prevented HTTPExplorer from accessing boot.ini and other sensitive files via system calls.

NOTE: You may notice that some events reference CSA user query messages. CSA rules that utilize the Query rule action are dependant upon the level of interactivity provided by the CSA administrator. If certain elements of the CSA User Interface (UI) are disabled by the administrator, then the rules take the default action as specified in the given rule (most often DENY).

Figure 4: CSA events resulting from HTTPExplorer Directory Traversal attack

Figure 5: Detailed CSA event resulting from CSA system API rules

Event Text	The process 'C:\Documents and Settings\Administrator\Desktop\httpExplorer\hexplorer.exe' (as user TST-WKS01\Administrator) attempted to call the function CreateFileW("C:\Documents and Settings\Administrator\Desktop\httpExplorer\my shared folder\..\..\..\..\..\..\..\..\..\boot.ini") from a buffer (the return address was 0x4a2f66). The code at this address is '036a006a 01680000 008050e8 d2b4f8ff 8bf0ff15 b0104000 8bc65ec2 04009090' This either happens when a process uses self-modifying code or when a process has been subverted by a buffer overflow attack. The operation was denied and process terminated.
Event Time	12/27/2006 5:27:17 PM
Code	HACL_OVERFLOW_TERMINATE
PInt	182
PString	C:\Documents and Settings\Administrator\Desktop\httpExplorer\hexplorer.exe
time	1579.2 (seconds since boot)
type	APICALL
ProcessId	1548
ApiOperation	BufferOverflowDetected
Credentials	os=win32,T=TST-WKS01 \Administrator,t=0105000000000000515000000A8F78AD5CA19B4B9639595B6F4010000,G= TST-WKS01 \None,g=0105000000000000515000000A8F78AD5CA19B4B9639595B601020000
ApiPInt1	4861798
ApiPString1	036a006a 01680000 008050e8 d2b4f8ff 8bf0ff15 b0104000 8bc65ec2 04009090
ApiPString2	CreateFileW
ApiPInt2	1237716
ApiPString3	90e31200 662f4a00 bc6b1700 00000080 01000000 00000000 03000000 00000002 00000000 00000000 09ed4800 34ed1200 dced1200 01000000 b19d4e00 0a000000
args(4)	C:\Documents and Settings\Administrator\Desktop\httpExplorer\my shared folder\..\..\..\..\..\..\..\..\..\boot.ini
ApiPInt3	4861782
argi(4)	3
FlattenedForm	(t-1167258436 n-609375000 z--18000 sm-110 sc-13 dm-1 dc-7 cd-558 p*(i-182 w-C:\Documents%20and%20Settings\Administrator\Desktop\httpExplorer\hexplorer.exe r*(type-17 time-15792 pnd-83889459 rapi*(pid-1548 op-8 p*(i-4861798 d-dOgaQfaAaaaaacf6stl-*Vi8*xbSqaealASxctaaqcj a-CreateFileW i-1237716 d-qoUeay2lkbaVRDbaaaaaagaaaaaaaaaadaaaaaaaaiaaaaaaaaaaaKq7ibanTlbaC3Ueaeaaaaq SD6eakaaaaaw-C:\Documents%20and%20Settings\Administrator\Desktop\httpExplorer\my%20shared%20folder\..\..\..\..\..\..\..\..\..\boot.ini i-4861782 i-3) cr-Owin32%00TTST-WKS01\Administrator%00t0105000000000000515000000A8F78AD5CA19B4B9639595B6F4010000% 00GTST-WKS01\None%00g0105000000000000515000000A8F78AD5CA19B4B9639595B601020000%00))))
Disassembly	**Address** **Code** **Instruction** 004a2f56 036a00 add ebp,dword[edx+0x0] 004a2f59 6a01 push dword(0x1) 004a2f5b 6800000080 push dword(0x80000000) 004a2f60 50 push eax 004a2f61 e8d2b4f8ff call 0x42e438 004a2f66** 8bf0 mov esi,eax 004a2f68 ff15b0104000 call dword[0x4010b0] 004a2f6e 8bc6 mov eax,esi 004a2f70 5e pop esi 004a2f71 c20400 ret 0x4 004a2f74 90 nop 004a2f75 90 nop

Summary of Results

CSA successfully prevented exploitation of the HTTPExplorer Web Server Traversal. System API control rules successfully prevented disclosure of the Boot.ini and other sensitive system information. A layered approach to security including HIPS is especially important when protecting web servers and other external facing systems.

References

- http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884
- http://www.milworm.com

Where to Go for More Information:

Custom Research Documents
Exploit Reverse Engineering/Forensics
Security Implementation
Real-World Training
Managed Services

Available @ www.Priveon.com