PriveonLabs Research

Cisco Security Agent Protection Series:

Microsoft XML Core Services XMLHTTP ActiveX Control Code Vulnerability CVE-2006-5745

Security Consultants:
Larry Boggis
Zach Brewer

Overview

On November 3rd, 2006 Microsoft published a security advisory addressing vulnerability in the Microsoft XML 4.0 Core Services. The advisory specifically deals with issues in the XMLHTTP 4.0 ActiveX control.

CVE: *2006-5745*
IMPACT: *Local/Remote Code Execution*
AFFECTED SOFTWARE:
XML Core Services 4.0 when installed on Windows 2000 SP4
XML Core Services 4.0 when installed on Windows XP SP2
XML Core Services 4.0 when installed on Windows Server 2003 SP0-1

Specifically, this vulnerability exists because Microsoft XML Core Services setRequestHeader() cannot handle HTTP requests correctly. As of the date of this writing (11.10.2006), there is no vendor update available to address this issue.

A successful exploit of the MS XML Core services vulnerability could allow a remote attacker to gain full control of the remote system. The remote attacker can, after exploitation, have total control including permissions alteration, software installation, and file manipulation. SANS Internet Storm Center® (ISC) and other security websites have reported that CVE-2006-5745 is actively being exploited in the wild.

The purpose of this document is to explain and expand upon the PoC exploits available as well as how the Cisco Security Agent (CSA) product can protect systems from this day-zero vulnerability.

Exploit PoC Code Exploit Process Overview

CVE-2006-5745 is a remotely exploitable attack with various malware variants currently propagating in the wild. This attack can occur as part of a directed attack using the Internet or email as a delivery mechanism. A more likely scenario might include the inclusion of the CVE-2006-5745 into web content or automated attack tools such as the Web Attack Toolkit. The Web Attack Toolkit has been used to deliver malware in "browse-by" attacks seen first-hand by Priveon Incident Response handlers. The addition of CVE-2006-5745 would provide yet another attack vector in the arsenal of malware writers. In addition, persuading a user with Internet Explorer to view specially crafted HTML content, a remote, unauthenticated attacker may be able to execute arbitrary code on a vulnerable system.

PoC Exploit Testing Overview

Testing Environment

The Priveon test lab consisted of several Operating Systems and application combinations including various patch revision levels running on virtual hosts. The purpose of the testing process is to confirm prevention of payload delivery through the Cisco Security Agent. The testing involved concept code obtained from milw0rm.com using several different payload types (examples: win32 bind-shell; win32 downloadexec)

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

Desktops - All types	5.1 r69	Default group for systems that install the Desktop agent kit	9 policies

Policy Name	Version	Description	Rule Modules
Agent UI control	5.1 r69	Policy which governs Agent User Interface	1 module
Document Security - Windows	5.1 r69	Policy to protect user documents	1 module
Email Client - Basic Security - Windows	5.1 r69	Basic application enforcement policy for email client software.	3 modules
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
IP Stack - Internal Network Security	5.1 r69	Policy for protecting the IP Stack on internal systems	1 module
Network Personal Firewall	5.1 r69	Control network access and provide some end user access controls.	1 module
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

Exploit PoC Testing Results – Unprotected System

Testing of the CVE-2006-5745 vulnerability was completed using the code published on milw0rm.com. The code initially published on November 8th, however, utilized a harmless calc.exe payload. In order to provide a more realistic test environment, a win32 bind-shell payload was used in our testing in place of the calc.exe payload.

Figure 2: Test 1 - Payload delivery resulted in listening command shell on target

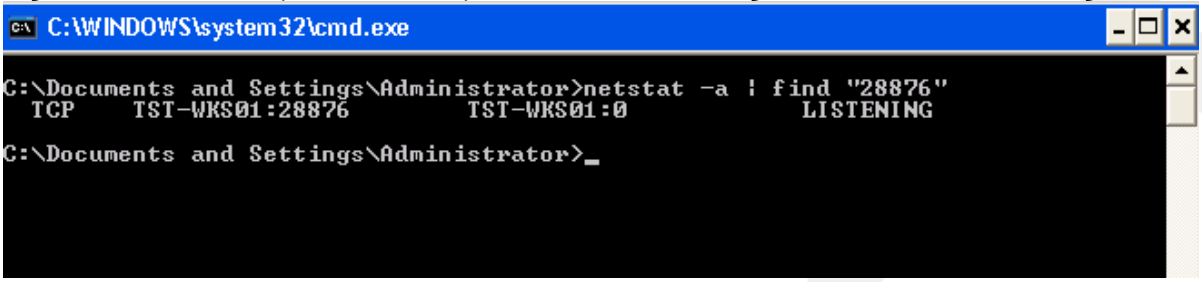
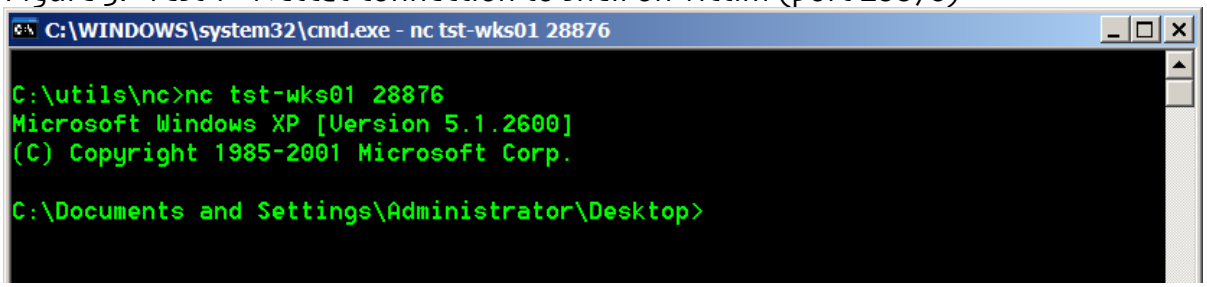


Figure 3: Test 1 - Netcat connection to shell on victim (port 28876)



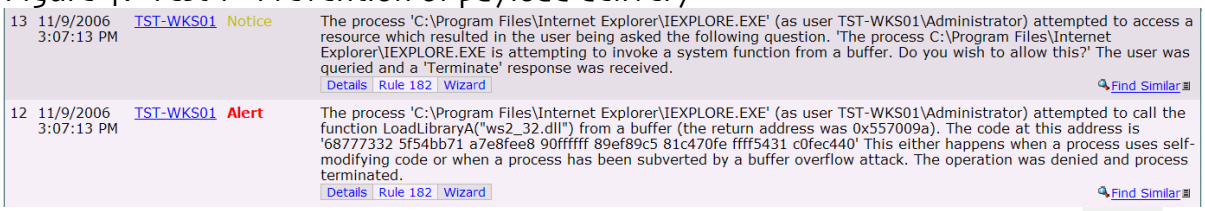
A second test was also completed using code published to milworm.com on November 10th. This exploit code labeled "MS Internet Explorer 6/7 (XML Core Services) Remote Code Exec Exploit 3" leveraged a win32_downloadexec payload.

CSA Protected Exploit PoC Testing Results

After proving that the win32 bind-shell and win32_downloadexec payloads were successful and that our systems were vulnerable to the exploits available on the Internet (milw0rm.com), we proceeded by installing the Cisco Security Agent Product on the target hosts to illustrate the successful prevention of the exploit on unpatched systems.

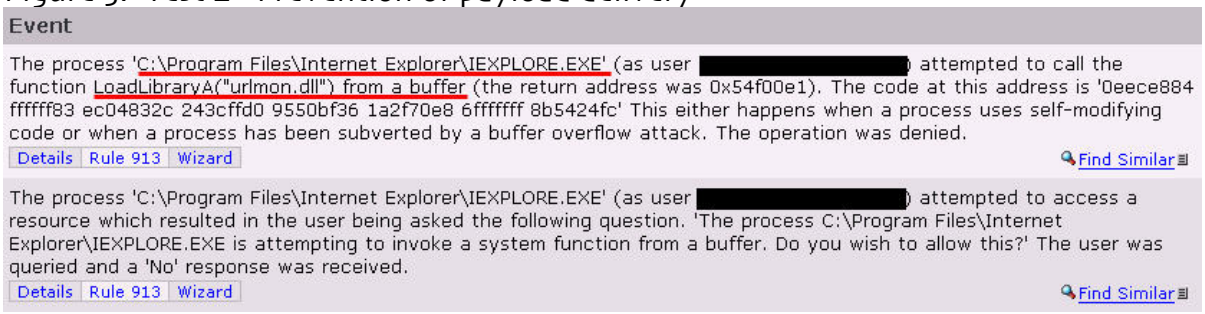
In our lab environment, we attempted to recreate the exploit process with the Cisco Security Agent installed on our target host. In the first scenario, IEXPLORE.EXE [tagged as a network application by CSA] communicated on the network to access the malicious HTML code on a remote web server. This resulted in the prevention of the payload execution and resulting system call through CSA's buffer overflow protection.

Figure 4: Test 1 - Prevention of payload delivery



The second test also simulated malicious HTML content posted to a remote web server. This time the payload attempted to download and execute content [of the attacker's choice]. CSA also successfully prevented this payload execution and resulting system call as seen in Figure 5.

Figure 5: Test 2 - Prevention of payload delivery



It is also important to note that had the payload deliveries somehow evaded detection or were for some reason allowed, CSA would have prevented the attacks through other Application Control, File Access Control, Registry Access and Network Access Control mechanisms providing true defense in-depth protection. In the sample data below (Figure 6.), the buffer overflow and initial payload were purposely allowed to show CSA protection beyond the first system call:

Figure 6: Test 2 – Additional CSA Protection Mechanisms (events from bottom up)

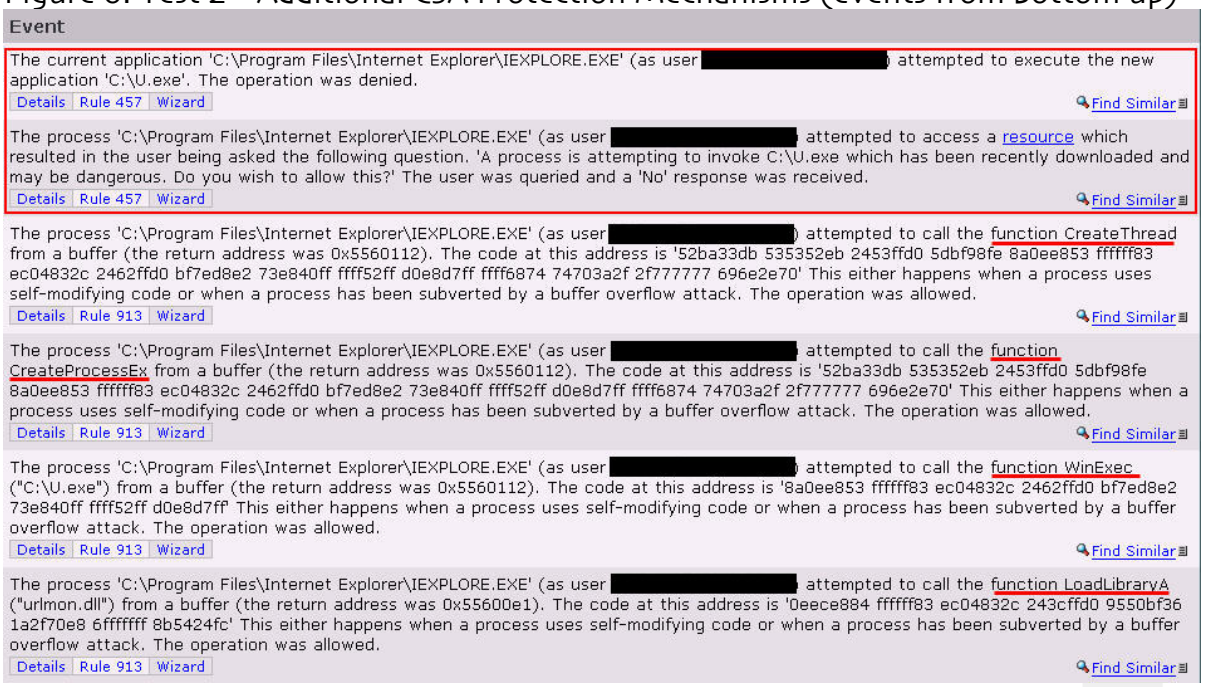


Figure 7: Test 1 - Buffer Overflow Details as illustrated in the CSA MC Event Log

Event Time	11/9/2006 3:07:13 PM
Code	HACL_OVERFLOW_TERMINATE
PInt	182
PString	C:\Program Files\Internet Explorer\IEXPLORE.EXE
time	12594.0 (seconds since boot)
type	APICALL
ProcessId	2768
ApiOperation	BufferOverflowDetected
Credentials	os=win32,T=TST-WKS01 \Administrator,t=0105000000000000515000000A8F78AD5CA19B4B9639595B6F4010000,G= TST-WKS01\None,g=0105000000000000515000000A8F78AD5CA19B4B9639595B601020000
ApiPInt1	89587866
ApiPString1	68777332 5f54bb71 a7e8fee8 90ffffff 89ef89c5 81c470fe ffff5431 c0fec440
ApiPString2	LoadLibraryA
ApiPInt2	1290448
ApiPString3	0000807c 9a005705 dcb01300 7773325f 33322e64 6c6c0000 29d9bb69 4c952300 9cf81800 a4b31300 0037b269 6cb11300 b02dc169 62fb1800 f4b01300 3cb11300
args(4)	ws2_32.dll
ApiPInt3	89587850
argi(4)	1
FlattenedForm	(t-1163102833 n-406250000 z--18000 sm-110 sc-13 dm-1 dc-7 cd-558 p*(i-182 w-C:\Program% 20Files\Internet%20Explorer\IEXPLORE.EXE r*(type-17 time-125940 pnd-83914601 rapi*(pid-2768 op-8 p*(i-89587866 d-OD3CY8fv7g3PO7p6q-***N47jwCGedN***pvXaS*ede a-LoadLibraryA i-1290448 d-aaaG8Pjaxva3Woba3nNmFnJmUqgBSbaaPK9UPXuLJaaN4JbaKo7eaaWnYMgBXobaW2sWPj2-yaa9Woba8e7eaa a-ws2_32.dll i-89587850 i-1) cr-Owin32%00TTST-WKS01\Administrator% 00t0105000000000000515000000A8F78AD5CA19B4B9639595B6F4010000% 00GTST-WKS01 \None%00g0105000000000000515000000A8F78AD5CA19B4B9639595B601020000%00))))
Disassembly	**Address** **Code** **Instruction** 0557008c 7332 jae 0x55700c0 0557008e 5f pop edi 0557008f 54 push esp 05570090 bb71a7e8fe mov ebx,0xfee8a771 05570095 e890ffffff call 0x557002a 0557009a** 89ef mov edi,ebp 0557009c 89c5 mov ebp,eax 0557009e 81c470effff add esp,0xffffe70 055700a4 54 push esp 055700a5 31c0 xor eax,eax 055700a7 fec4 inc ah 055700a9 40 inc eax

Summary of Results

CSA prevented exploitation of systems vulnerable to the CVE-2006-5745 vulnerability using default CSA desktop policies in all testing scenarios and utilizing various payload types. The success was proven with the default PoC code which is readily available via milw0rm.com.

It is important that anyone managing a CSA deployment thoroughly understand the policy they have chosen to deploy on their protected systems. By default, only certain applications and programs that communicate over the network are protected by buffer overflow mechanisms (System API Rules). Since the attack vector of the CVE-2006-5745 vulnerability is most effective via remote HTML content, the application accessing the malicious content [in this case IEXPLORE.EXE] is protected. The true defense-in-depth nature of the Cisco Security Agent includes additional protection mechanisms also demonstrated in this test.

References

- http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-5745
- http://www.microsoft.com/technet/security/advisory/927892.mspx
- http://metasploit.com
- http://netcat.sourceforge.net/

Where to Go for More Information:

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

Available @ www.Priveon.com