PriveonLabs Research

Cisco Security Agent Protection Series:

MS Animated Cursor Vulnerability
CVE-2007-0038

Zach Brewer
Security Consultant

Overview

On 03/28/2007, an advisory was published by Determina Research outlining an exploit for the Microsoft Animated Cursor Vulnerability (CVE-2007-0038) which was quickly followed by exploit code released to Milworm.com. With no official patch available and active exploitation taking place over the following weekend, the threat was deemed serious enough by SANS to raise the ISC Infocon level to yellow.

In this document, we will briefly discuss factors surrounding successful exploitation of CVE-2007-0038 as well as the techniques used by Cisco Security Agent to prevent this and other 0-day exploits.

About CVE-2007-0038

Microsoft Animated cursor files (.ANI) are typically loaded in HTML pages in order to change the default Internet Explorer cursor. An animated cursor file structure consists of a standard layout that may include the author's name as well as several icon files used to animate the cursor.

The current ANI vulnerability exists with the LoadAniIcon() function of USER32.dll and its inherent interpretation of ANI files. A similar issue was resolved with update MS05-002 although CVE-2007-0038 was evidently not specifically addressed by that update.

The potential for a successful CVE-2007-0038 exploit is increased due to the fact that Internet Explorer and Outlook Express load ANI files without active user interaction. While testing, Priveon Researchers also observed ANI files crashing explorer.exe through simple access to a directory holding a malformed ANI file. This indicates that the problem might involve other potential target applications. (NOTE: Soon after the initial draft of this document, similar exploit code targeting IFranView 3.99 was released publicly).

The most common form of active exploitation comes in the form of a malicious ANI file referenced on a website. A user's unknowing access of an ANI file causes a target host to download malware of the author's choice from a remote website. Other potential attack vectors include ANI files referenced in HTML email messages. Reports indicate that simply accessing an email message via Outlook Express can trigger the exploit process (NOTE: This has not been independently verified by Priveon researchers).

Vulnerable Products

As of this writing, the following products are reported to be vulnerable:

- Microsoft Windows 2000 Service Pack 4
- Microsoft Windows XP Service Pack 2
- Microsoft Windows XP 64-Bit Edition Version 2003 (Itanium)
- Microsoft Windows XP Professional x64 Edition
- Microsoft Windows Server 2003
- Microsoft Windows Server 2003 for Itanium-based Systems
- Microsoft Windows Server 2003 Service Pack 1
- Microsoft Windows Server 2003 Service Pack 2
- Microsoft Windows Server 2003 with SP1 for Itanium-based Systems and Microsoft Windows Server 2003 with SP2 for Itanium-based Systems
- Microsoft Windows Server 2003 with SP2 for Itanium-based Systems
- Microsoft Windows Server 2003 x64 Edition
- Microsoft Windows Server 2003 x64 Edition Service Pack 2
- Microsoft Windows Vista

PoC Exploit Testing Overview

Testing Environment

The Priveon test lab consisted of one windows XP system that was deemed to be vulnerable to CVE-2007-0038. Additionally, the lab contained one simulated web server hosting exploit code. The vulnerable Windows XP system was placed in the "Desktops – All Types" CSA group. "Desktops – All Types" is a default group containing many out of the box rules for securing standard desktops in a corporate environment.

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

For the purpose of testing CSA default rules against this vulnerability, a simple Calc.exe payload was chosen. Quite simply, browsing to an internal website that referenced our PoC ANI file (Figure 1) resulted in an internet Explorer crash and the invocation of Calc.exe using the Win32Exec API call.

Figure 2: Simple HTML Code Referencing Animated Cursor

```
<HMTL>
    <HEAD>
    <style>
<!--
BODY{ cursor:url("xp.ani"); }
-->
    </style>
    </HEAD>
<BODY>
    <H1>ANI CURSOR VULN TEST</H1>
</BODY>
</HTML>
```

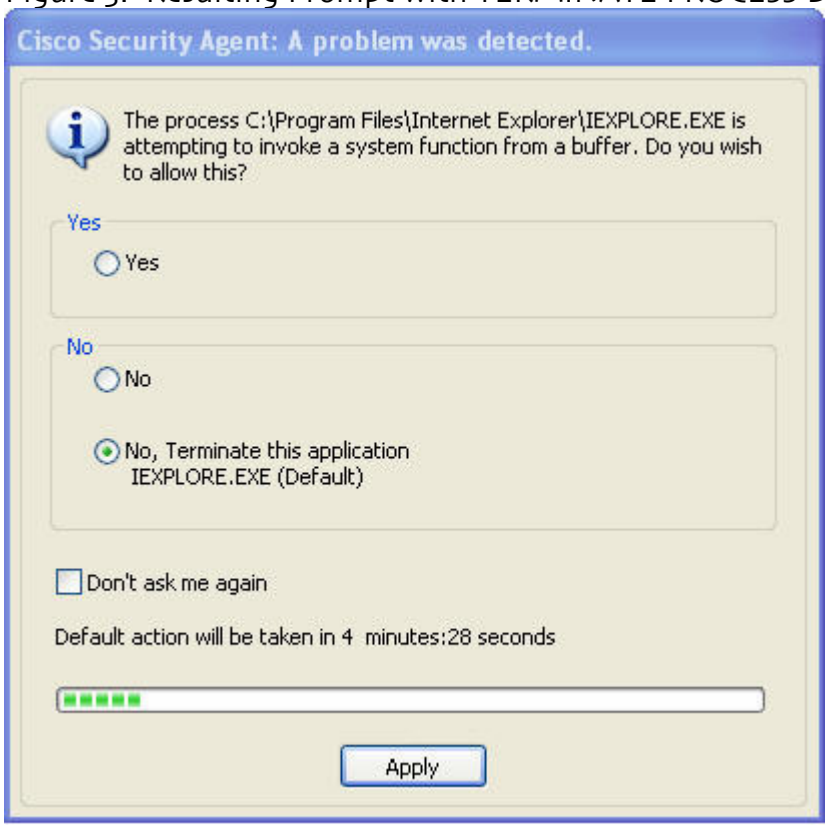
Note: Additional testing results with multi-staged exploits, obfuscation, and other techniques may be added to this document in future revisions.

Once the victim host was deemed to be vulnerable to CVE-2007-0038, the PoC testing continued with the Cisco Security Agent installed.

CSA Protected Exploit PoC Testing Results

With the Cisco Security Agent installed on the simulated target, Internet Explorer was directed to the simulated attacking web server. CSA default System API rules caught the attack and resulted in a DENY user prompt with the default action set to TERMINATE PROCESS (see figure 3).

Figure 3: Resulting Prompt with TERMINATE PROCESS Default Action



Provided that a user chooses the default action, the attack would be successfully prevented and denied before the payload was delivered as demonstrated in Figures 4 and 5. It should also be noted that more advanced payloads would likely be caught despite the users answer to the resulting prompt. Please see other CSA Protection Series documents such as the "Using IFrames for Active Malware Delivery" document at www.priveon.com/research for examples.

Figure 4: Resulting Deny Action Based Upon User Prompt

#	Date	Host	Severity	Event
10	4/2/2007 10:00:47 PM	TST-WKS01	Warning	The process 'C:\Program Files\Internet Explorer\IEXPLORE.EXE' (process id 568) has been dumped into 'C:\Program Files\Cisco Systems\CSAgent\log\iexplore.dmp' on the agent system. This dump file may be useful to technical support to determine whether a previous event was a true positive or a false positive. Details Find Similar
9	4/2/2007 10:00:44 PM	TST-WKS01	Alert	The process 'C:\Program Files\Internet Explorer\IEXPLORE.EXE' (as user TST-WKS01\Administrator) attempted to call the function WinExec("calc") from a buffer (the return address was 0x7c81caa2). The code at this address is '53e83d67 0000e921 eafeff90 90909090 8bff558b ec6aff68 b0f3e877 ff7508e8' 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. Details Rule 182 Wizard Find Similar

Figure 5: Details of DENY Action Including Arguments Passed to WinExec

Policy rule details: Print | Close

Description	Network Applications, Access system functions from a buffer
Module	General Application Permissions - all Security Levels [W, V5.1 r69]
Cisco Security Agent query	*English*: The process @appname is attempting to invoke a system function from a buffer. Do you wish to allow this? *Spanish*: El proceso @appname intenta invocar una función del sistema desde un búfer. ¿Desea permitirlo? *French*: Le processus @appname tente d'invoquer une fonction système à partir d'un tampon. Voulez-vous autoriser cette opération ? *German*: Der Prozess @appname versucht, eine Systemfunktion aus einem Puffer aufzurufen. Soll dieser Vorgang zugelassen werden? *Italian*: Il processo @appname sta tentando di richiamare una funzione di sistema da un buffer. Autorizzare questa operazione? *Japanese*: ???? @appname ????????????? ????? ??????? ?????????? *Korean*: ???? @appname?(?) ???? ??? ??? ???? ???. ?? ???????? *Simplified Chinese*: ?? @appname ???????? ??????? ????????
Event details:	
Event Text	The process 'C:\Program Files\Internet Explorer\IEXPLORE.EXE' (as user TST-WKS01\Administrator) attempted to call the function WinExec("calc") from a buffer (the return address was 0x7c81caa2). The code at this address is '53e83d67 0000e921 eafeff90 90909090 8bff558b ec6aff68 b0f3e877 ff7508e8' 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	4/2/2007 10:00:44 PM
Code	HACL_OVERFLOW_TERMINATE
PInt	182
PString	C:\Program Files\Internet Explorer\IEXPLORE.EXE
time	2993.4 (seconds since boot)
type	APICALL
ProcessId	568
ApiOperation	BufferOverflowDetected
Credentials	os=win32,T=TST-WKS01\Administrator,t=0105000000000005515000000A8F78AD5CA19B4B9639595B6F4010000, G=TST-WKS01\None,g=0105000000000005515000000A8F78AD5CA19B4B9639595B601020000
ApiPInt1	2088880802
ApiPString1	53e83d67 0000e921 eafeff90 90909090 8bff558b ec6aff68 b0f3e877 ff7508e8
ApiPString2	WinExec
ApiPInt2	1302180
ApiPString3	0000807c a2ca817c e7ff3a0b 00000000 6cff3a0b 00000000 0000807c d4de1300 88df1300 08061500 c8f9c07c d01f2500 0b0b0b0b 0b0b0b0b 0b0b0b0b 0b0b0b0b
args(4)	calc
ApiPInt3	2088880786
argi(4)	6
FlattenedForm	(t-1175565643 n-656250000 z--14400 sm-111 sc-13 dm-1 dc-7 cd-558 p*(i-182 w-C:\Program%20Files\Internet%20Explorer\IEXPLORE.EXE r*(type-17 time-29934 pnd-83888335 rapi*(pid-568 op-8 p*(i-2088880802 d-tHEpNbaaPhl6-*pKqcjKqU4*vTi7Q9pAWop639FDiGo a-WinExec i-1302180 d-aaaG8jQYbY35*RZcaaaaaW2*6SaaaaaaaaaG8rT3taaIFpbaiyqfaGC-aZh0FucaISWcISWcISWcISWcISWcla a-calc i-2088880786 i-6) cr-Owin32%00TTST-WKS01\Administrator%00t0105000000000005515000000A8F78AD5CA19B4B9639595B6F4010000% 00GTST-WKS01\None%00g0105000000000005515000000A8F78AD5CA19B4B9639595B601020000%00))))

Note: Should a CSA administrator decide to revoke user access to the CSA user interface (UI), Cisco Security Agent QUERY rules take the default action given in the QUERY rule. In the case of our test, the QUERY rule would take the DENY AND TERMINATE PROCESS action thus resulting in a successful prevention of the exploit.

Similarly, an experienced CSA administrator might write a custom File Access Control rule for any ANI file. The resulting action could be a default DENY or a QUERY with text more relevant to the known ANI vulnerability. Proper testing should take place in order to address the potential impact of custom rules.

Summary of Results

CSA successfully prevented exploitation of the target system and execution of the simulated calc.exe payload through the use of System API rules. It is likely that more advanced payloads would also be caught by additional CSA protective policies. Although limited in testing, the brief 0-day prevention test outlined in this document helps to demonstrate the role of Cisco Security Agent in proactive endpoint security.

References

- http://isc.sans.org/diary.html?storyid=2542
- http://www.milworm.com
- http://www.determina.com/security.research/vulnerabilities/ani-header.html
- http://www.microsoft.com/technet/security/advisory/935423.mspx

Where to Go for More Information:

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

Available @ www.Priveon.com