PriveonLabs Research

Cisco Security Agent Protection Series: The MS Excel hlink.dll Day-Zero Vulnerability

Zach Brewer
Security Consultant

Overview

Once again, the Cisco Security Agent (CSA) product with default policies has proven effective against a day-zero vulnerability. In this document, we will attempt to illustrate how the policies prevent and/or limit the vulnerabilities' exposure for exploitation to the MS Excel hlink.dll vulnerability.

On June 18th, Milworm.com published a PoC (Proof of Concept) PERL script for the hlink.dll vulnerability. The original PERL script was followed with C++ code that claimed to work on both windows 2000 SP4 & Windows XP SP2 with Office 2000. Many other PoC scripts followed in the several days after the initial code was posted. On June 19th, 2006, Microsoft confirmed the existence of a vulnerability within hlink.dll in various versions of Excel. Most recently, a PoC PERL script (milworm exploit 1958), which included both a payload and a confirmed working exploit for Office 2003 SP2, was released.

The Hlink.dll memory corruption vulnerability has been entered as candidate CVE-2006-2086 under the Common Vulnerabilities and Exposures (CVE). CVE-2006-2086 is currently being exploited in the wild although reports vary on the number of incidents attributed to this vulnerability.

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.

Hlink.dll PoC Code Exploit Process Overview

CVE-2006-2086 is a client-side attack that is exploited via a malicious excel file. The actual vulnerability exploits a buffer overflow condition in the Hlink.dll (dll used to process hyperlinks in MS Office programs) and is exploited through a long Unicode hyperlink embedded within an Excel file. The result of the exploit process results in excel crashing and/or hanging and, more importantly, the capability of running arbitrary code.

The most common attack vectors for this exploit include the automated or manual distribution of malicious excel files via e-mail or the web. Social engineering tactics (such as 'phishing') may increase the likelihood of successful exploitation when files are distributed manually.

When a user clicks the embedded link within a compromised Excel file, he or she is prompted with an error message. Once the prompt is accepted, Excel often (but not always) closes unexpectedly and the payload is delivered to the victim host. The payload included with the PoC code obtained from Milworm.com was non-malicious and simply started calc.exe (MS Calculator) from a command shell via standard shellcode obtained from the Metasploit project.

PoC Exploit Testing Overview

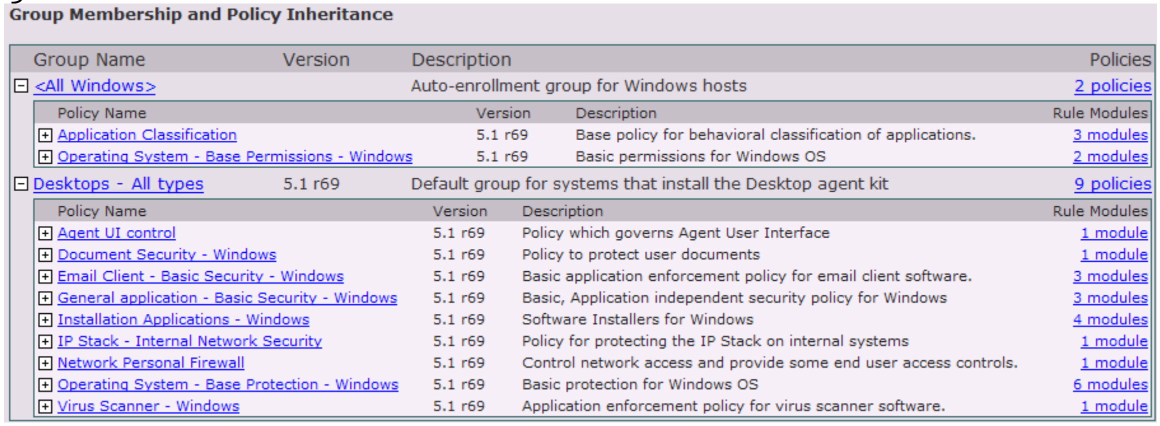
Testing Environment

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 Milworm.com as well as modified code to test various shellcode payloads such as win32_exec, win32_bind, win32_reverse, and win32_dowloadexec which can be obtained via the Metasploit project.

Cisco Security Agent Version Tested:

- CSA 5.1.0.69 (Default CSA Polices as show in Figure 1 below)

Figure 1: Protected Host CSA Policies



Group Membership and Policy Inheritance

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

Group Name	Version	Description	Policies
⊟ 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

The PoC Exploit code's default win32_exec payload simply executes a system command on the victim host. For our example, win32_exec executes the "CALC.EXE" command that is used with the published proof-of-concept code (milworm exploit 1958 published 06/27/2006). This payload is relatively simple but could be modified in order to successfully compromise a target host. In our testing, CSA did not prevent the non-malicious CALC.EXE from executing when using the win32_exec payload. This is to be expected since the PoC code used Calc.exe which is NOT a malicious payload. When testing this PoC code on excel where the compromised file was opened from a remote network share, the buffer overflow was immediately detected as excel was tagged as a network application and was then placed under greater scrutiny. It should be noted, that in either case, similar tests using win32_exec CSA prevented potentially dangerous commands from executing including CMD and REGEDIT.

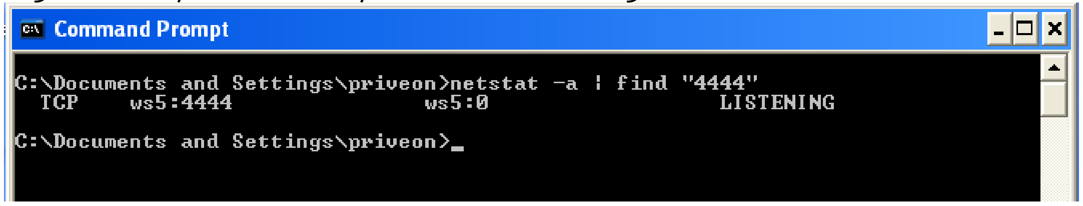
Extended Exploit PoC Testing Results

To further extend our testing and to include a larger sampling of exploits likely to follow the initial posting of exploit code, we decided to continue to test various common shellcode payloads. The first additional payload to be tested was win32_bind.

Results of Win32_Bind Payload with CSA 5.1.0.69 Policies

The result of the win32_bind payload is a command shell listening on a port of the attacker's choosing. They payload can be used for remote exploitation after the local exploit has been successfully delivered. If this payload is delivered, an attacker will have access to a command shell running with System account privileges on the victim host. Figure 2 illustrates a confirmed shell listening on TCP/4444 while Figure 3 illustrates the remote system successfully connecting to this shell.

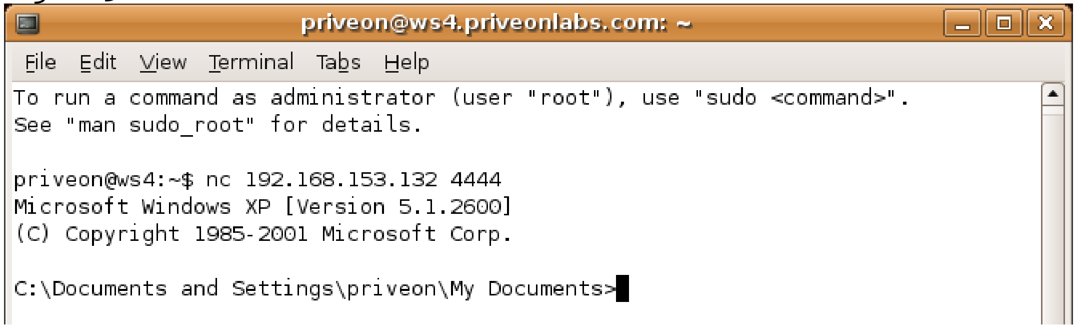
Figure 2: Payload delivery results in listening command shell on victim host



```
C:\Documents and Settings\priveon>netstat -a | find "4444"
  TCP    ws5:4444               ws5:0                  LISTENING

C:\Documents and Settings\priveon>_
```

Figure 3: Remote connections to the victims command shell


priveon@ws4.priveonlabs.com: ~
File Edit View Terminal Tabs Help

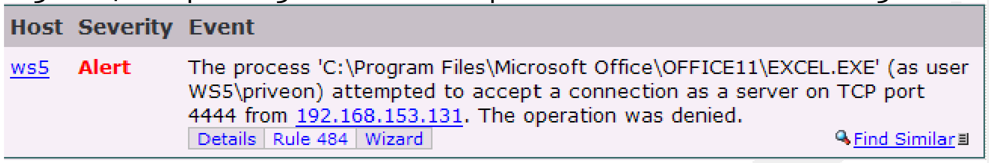
```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

priveon@ws4:~$ nc 192.168.153.132 4444
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\priveon\My Documents>
```

When CSA is active, the win32_bind payload successfully loaded into the memory of the target host; however, the remote attacker was not able to connect to the resulting command shell due to the default Network Access Control rules preventing processes on the system from acting as a Server. The resulting event from attempted access to this shell is displayed in Figure 4.

Figure 4: Reporting denied attempted connection to resulting listener



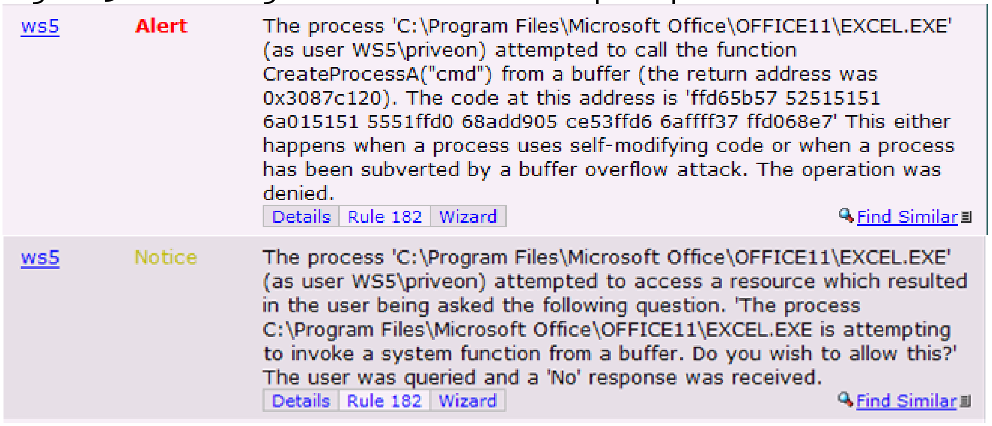
Host	Severity	Event
ws5	Alert	The process 'C:\Program Files\Microsoft Office\OFFICE11\EXCEL.EXE' (as user WS5\priveon) attempted to accept a connection as a server on TCP port 4444 from 192.168.153.131. The operation was denied.

Results of Win32_Reverse Payload with CSA5.1.0.69 Policies

The win32_reverse payload is similar to the win32_bind payload with the exception that the attacker runs the listener and the victim initiates the connection when the payload is delivered. This can be useful to an attacker in situations where firewalls exist between the attacking host and the victim host. Although the compromised system initiates the connection with win32_reverse, the results are the same as the win32_bind payload – an attacker has access to a remote command shell running under the privileges of the System account.

With default CSA 5.1 r69 rules enabled, the win32_reverse payload was not loaded into the victim host memory due to a preventative System API rule. The System API rule that was triggered caused the end user to be prompted as the payload was moved into protected system memory. The resulting events from CSA protection are displayed in Figure 5.

Figure 5: Resulting Events from the user prompt and denied connection



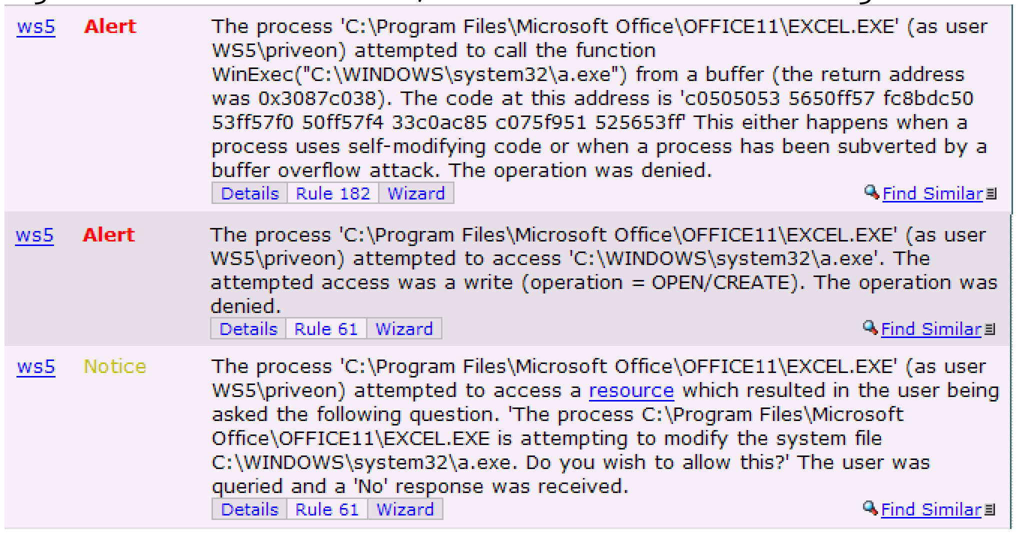
ws5	Alert	The process 'C:\Program Files\Microsoft Office\OFFICE11\EXCEL.EXE' (as user WS5\priveon) attempted to call the function CreateProcessA("cmd") from a buffer (the return address was 0x3087c120). The code at this address is 'ffd65b57 52515151 6a015151 5551ffd0 68add905 ce53ffd6 6affff37 ffd068e7' 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. Details | Rule 182 | Wizard Find Similar
ws5	Notice	The process 'C:\Program Files\Microsoft Office\OFFICE11\EXCEL.EXE' (as user WS5\priveon) attempted to access a resource which resulted in the user being asked the following question. 'The process C:\Program Files\Microsoft Office\OFFICE11\EXCEL.EXE is attempting to invoke a system function from a buffer. Do you wish to allow this?' The user was queried and a 'No' response was received. Details | Rule 182 | Wizard Find Similar

Results of Win32_downloadexec Payload with CSA 5.1.0.69 Rules

The win32_downloadexec is arguably the most relevant of the four payloads selected for our test. Win32_downloadexec downloads a file and executes it on a compromised system. For this test, we chose the infamous sub7 server (server.exe) pre-configured to listen on TCP port 27374. It should be noted that the process spawned by the sub7 executable often changes its name after executing. As a result, the name of the sub7 process running in memory may not match the name of the exe that was downloaded from the remote system. Other potential malware that may be delivered using win32_downloadexec includes spyware, adware, bots, and/or rootkits.

When CSA is active, the payload is downloaded to the %windir%/system32 directory as A.exe. CSA prompts the user prior to excel both writing and executing the Sub7 Trojan. This is displayed on the console event log as seen in Figure 6. It is important to understand that these possible user queries would only prompt the user if the CSA product were operating in interactive mode. If the agent is non-interactive, the Agent would take the default action immediately and the operation would be denied.

Figure 6: Overflow and Sub-7 events as seen in the event log



```
ws5   Alert    The process 'C:\Program Files\Microsoft Office\OFFICE11\EXCEL.EXE' (as user
               WS5\priveon) attempted to call the function
               WinExec("C:\WINDOWS\system32\a.exe") from a buffer (the return address
               was 0x3087c038). The code at this address is 'c0505053 5650ff57 fc8bdc50
               53ff57f0 50ff57f4 33c0ac85 c075f951 525653ff' 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.
               Details | Rule 182 | Wizard                                    Find Similar

ws5   Alert    The process 'C:\Program Files\Microsoft Office\OFFICE11\EXCEL.EXE' (as user
               WS5\priveon) attempted to access 'C:\WINDOWS\system32\a.exe'. The
               attempted access was a write (operation = OPEN/CREATE). The operation was
               denied.
               Details | Rule 61 | Wizard                                     Find Similar

ws5   Notice   The process 'C:\Program Files\Microsoft Office\OFFICE11\EXCEL.EXE' (as user
               WS5\priveon) attempted to access a resource which resulted in the user being
               asked the following question. 'The process C:\Program Files\Microsoft
               Office\OFFICE11\EXCEL.EXE is attempting to modify the system file
               C:\WINDOWS\system32\a.exe. Do you wish to allow this?' The user was
               queried and a 'No' response was received.
               Details | Rule 61 | Wizard                                     Find Similar
```

Summary of Results

CSA was extremely effective against preventing system exploitation as a result of the Excel hlink.dll zero-day vulnerability. The success was proven for the default PoC code as well as additional testing against common payloads to be expected as the exploit gains momentum in the hacking community.

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). If your Excel instance were to open a local file that has the hlink.dll exploit embedded without communicating on the network, the buffer overflow mechanism would not protect the actual code execution as a result of the overflow. However, it is also important to understand that any malicious behavior post-overflow, such as active network servers and reverse shells, would be prevented as a result of other CSA default policies demonstrating the true defense-in-depth nature of the Cisco Security Agent.

All protection mechanisms discussed were provided by utilizing the default policies provided with the CSA product. Beyond default policies, further customization is possible that could monitor and/or prevent specific exploitation attempts of this and other attack vectors. It is this layered security policy approach and granular customization capabilities that makes CSA so powerful and effective against day-zero attacks.

References

- http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-3086
- http://blogs.securiteam.com/index.php/archives/451
- http://www.microsoft.com/technet/security/advisory/921365.mspx
- http://isc.sans.org
- http://www.cisco.com/en/US/products/ps6786/index.html
- http://metasploit.com
- http://www.milworm.com
- http://netcat.sourceforge.net/
- http://www.ollydbg.de/