AV Evasion With the Veil Framework

#avlol

@HarmJ0y
@ChrisTruncer
@TheMightyShiv
@VeilFramework

VERIS GROUP

@VeilFramework

- **Will Schroeder @HarmJ0y**
  - Former national research lab keyboard monkey
- **Chris Truncer @ChrisTruncer**
  - Florida State Graduate - Go Noles!
- **Michael Wright @TheMightyShiv**
  - Pulled away on assessment : (

- Veris Group pentesters by day, antivirus evasion researchers by night

Overview

- The Problem
- Public Reaction and Ethical Considerations
- The Veil Framework
- Payload Releases
- Veil-Evasion Demo
- Payload Delivery
- Veil-Catapult Demo
- How to stop us

The Problem

Antivirus can’t catch malware but does catch pentesters

File name: meterpreter.exe

Detection ratio: 35 / 48

Our Solution

- A way to get around antivirus as easily as professional malware
- Don’t want to roll our own backdoor each time
- Find a way to execute existing shellcode in an av-evading way

Our Solution

```
=========================================================================
 Veil-Evasion | [Version]: 2.4.0
=========================================================================
 [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=========================================================================

 Main Menu

        24 payloads loaded

 Available commands:

        use             use a specific payload
        info            information on a specific payload
        list            list available payloads
        update          update Veil to the latest version
        clean           clean out payload folders
        checkvt         check payload hashes vs. VirusTotal
        exit            exit Veil

 [>] Please enter a command:
```

Veil-Evasion's Approach

- Aggregation of various shellcode injection techniques across multiple languages
  - These have been known and documented in other tools
- Focused on automation, usability, and developing a true framework
- Some shellcodeless Meterpreter stagers as well

Ethical Considerations

- The disclosure debate is not new...
- Pentesters are 5+ years behind the professional malware community
- This is already a problem the bad guys have solved

HD Moore's Take

"The strongest case for information disclosure is when the benefit of releasing the information outweighs the possible risks. In this case, like many others, the bad guys already won."

https://community.rapid7.com/community/metasploit/blog/2009/02/23/the-best-defense-is-information

Our Take

- We chose the path of full public disclosure
- We want to help the security industry better emulate threats
- AV vendors can see our code!

Public Reaction

- “surely this will just result in 21 new signatures for all major AVs and then we're back to square one?”

- “Isn't our entire field meant to be working towards increasing security, rather than handing out fully functioning weapons?”

- “The other point here is that anything that helps to **expose how in-effective AV really is at stopping even a minimally sophisticated attacker** is a good thing.”

  http://www.reddit.com/r/netsec/comments/1fc2xp/veil_a_metasploit_payload_generator_for_bypassing/

The Veil Framework

Veil-Evasion

Veil-Evasion Features

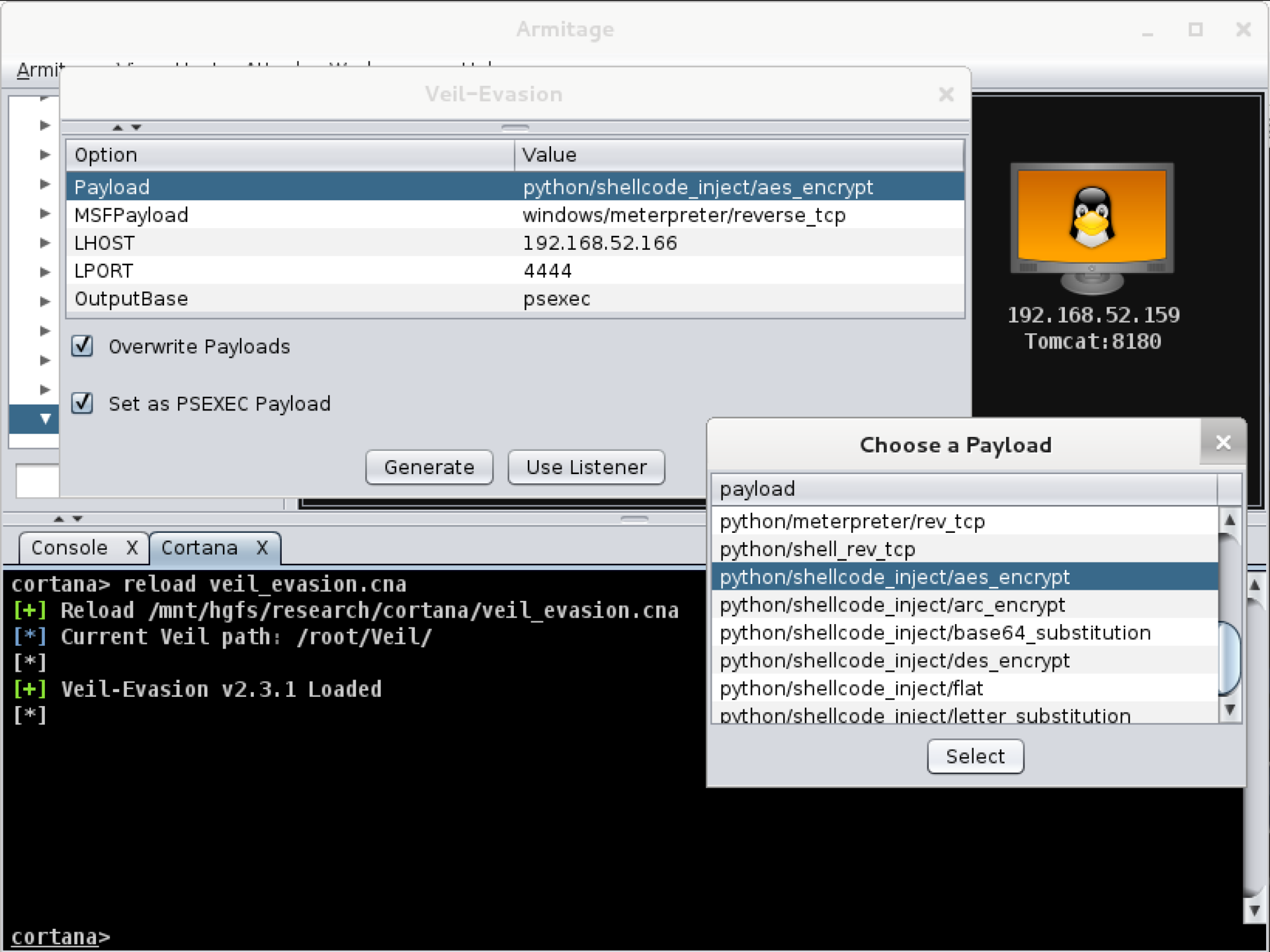
- Can use Metasploit-generated or custom shellcode
  - MSF payloads/options dynamically loaded
- Third party tools can be easily integrated
  - Hyperion, PEScrambler, BackDoor Factory, etc.
- Command line switches to allow scriptability

Armitage Integration

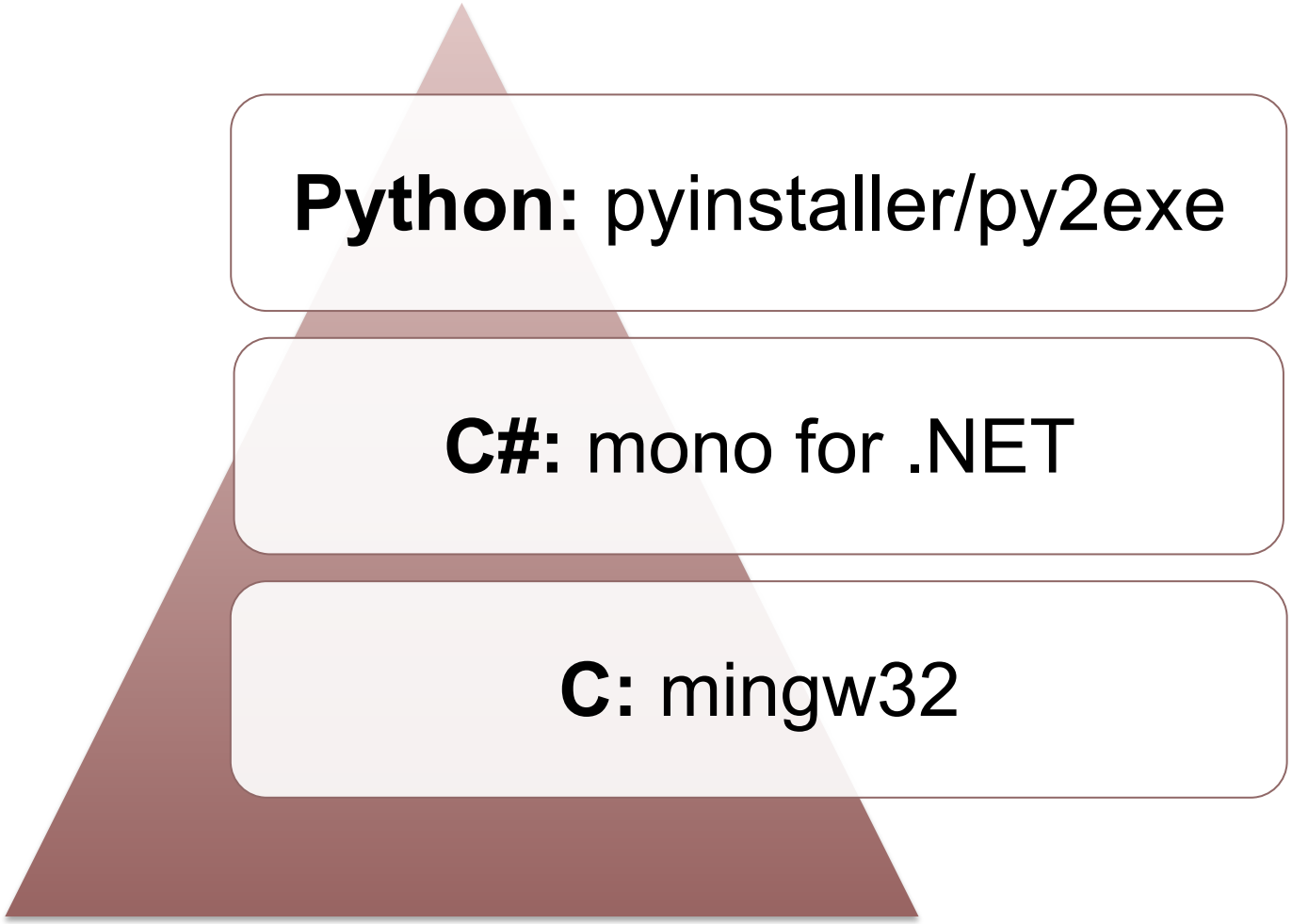
- The **veil_evasion.cna** script allows for the graphical integration of Veil-Evasion into Armitage/Cobalt Strike

- Payloads can be generated and optionally substituted into all psexec calls seamlessly

http://blog.strategiccyber.com/2012/08/03/cortana-real-time-collaborative-hacking-with-bots/


Armitage
Veil-Evasion
Option
Value
Payload
python/shellcode_inject/aes_encrypt
MSFPayload
windows/meterpreter/reverse_tcp
LHOST
192.168.52.166
LPORT
4444
OutputBase
psexec
Overwrite Payloads
Set as PSEXEC Payload
Generate
Use Listener
192.168.52.159
Tomcat:8180
Choose a Payload
payload
python/meterpreter/rev_tcp
python/shell_rev_tcp
python/shellcode_inject/aes_encrypt
python/shellcode_inject/arc_encrypt
python/shellcode_inject/base64_substitution
python/shellcode_inject/des_encrypt
python/shellcode_inject/flat
Select
Console X
Cortana X
cortana> reload veil_evasion.cna
[+] Reload /mnt/hgfs/research/cortana/veil_evasion.cna
[*] Current Veil path: /root/Veil/
[*]
[+] Veil-Evasion v2.3.1 Loaded
[*]
cortana>

Native Compilation



Module Development

- Implement your own obfuscation methods
- Lots of reusable functionality
  - Shellcode generation is abstracted and can be invoked as needed
- https://www.veil-framework.com/tutorial-veil-payload-development/

Am I Getting Caught?

- A running hash list of every payload generated is kept in ~/veil-output/hashes.txt
- Mubix's vt-notify script* can alert us if a customer submits a Veil payload to virustotal.com

*https://github.com/mubix/vt-notify

checkvt

```
Available commands:

        use             use a specific payload
        info            information on a specific payload
        list            list available payloads
        update          update Veil to the latest version
        clean           clean out payload folders
        checkvt         check payload hashes vs. VirusTotal
        exit            exit Veil

[>] Please enter a command: checkvt

[*] Checking Virus Total for payload hashes...

[!] File payload14 with hash f330c03f9f0ec14cfd5e4d387b91195633340130 found!

[>] Hit enter to continue...
```

Shellcode Injection

- Void pointer casting
  - no guarantee the memory region is executable
- VirtualAlloc
  - allocate memory as RWX, copy code in and create a thread
- HeapAlloc
  - create a heap object and manually allocate memory

DEP and Pyinstaller

- Pyinstaller produced .exe's are DEP enabled by default
  - this ruins some shellcode injection methods
- Luckily Pyinstaller is open source
  - we can recompile to turn off DEP opt-in
- https://www.veil-evasion.com/dep-pyinstaller/

Payload Releases

#VDay

V-Day

- We release **at least** one new payload on the 15th of every month
- 24 currently published payloads
- 20+ additional payloads have been developed so far
  - we’re going to be releasing for a while :)

Shellcodeless Stagers

- Stage 1 Meterpreter loaders don’t have to be implemented in shellcode
- Meterpreter stagers can be written in higher-level languages
- https://github.com/rsmudge/metasploit-loader

Veil Stagers

- The following are the stagers currently available in the framework:

Language	Stager
C	meterpreter/rev_tcp
C	meterpreter/rev_tcp_service
C#	meterpreter/rev_tcp
python	meterpreter/rev_tcp
python	meterpreter/rev_http
python	meterpreter/rev_https

Stager Basics

How a Meterpreter stager works:

1) a tcp connection is opened to the handler

2) the handler sends back 4 bytes indicating the .dll size, and then transfers the .dll

3) the socket number for this tcp connection is pushed into the edi register

4) execution is passed to the .dll just like regular shellcode (void * or VirtualAlloc)

DEMO #1

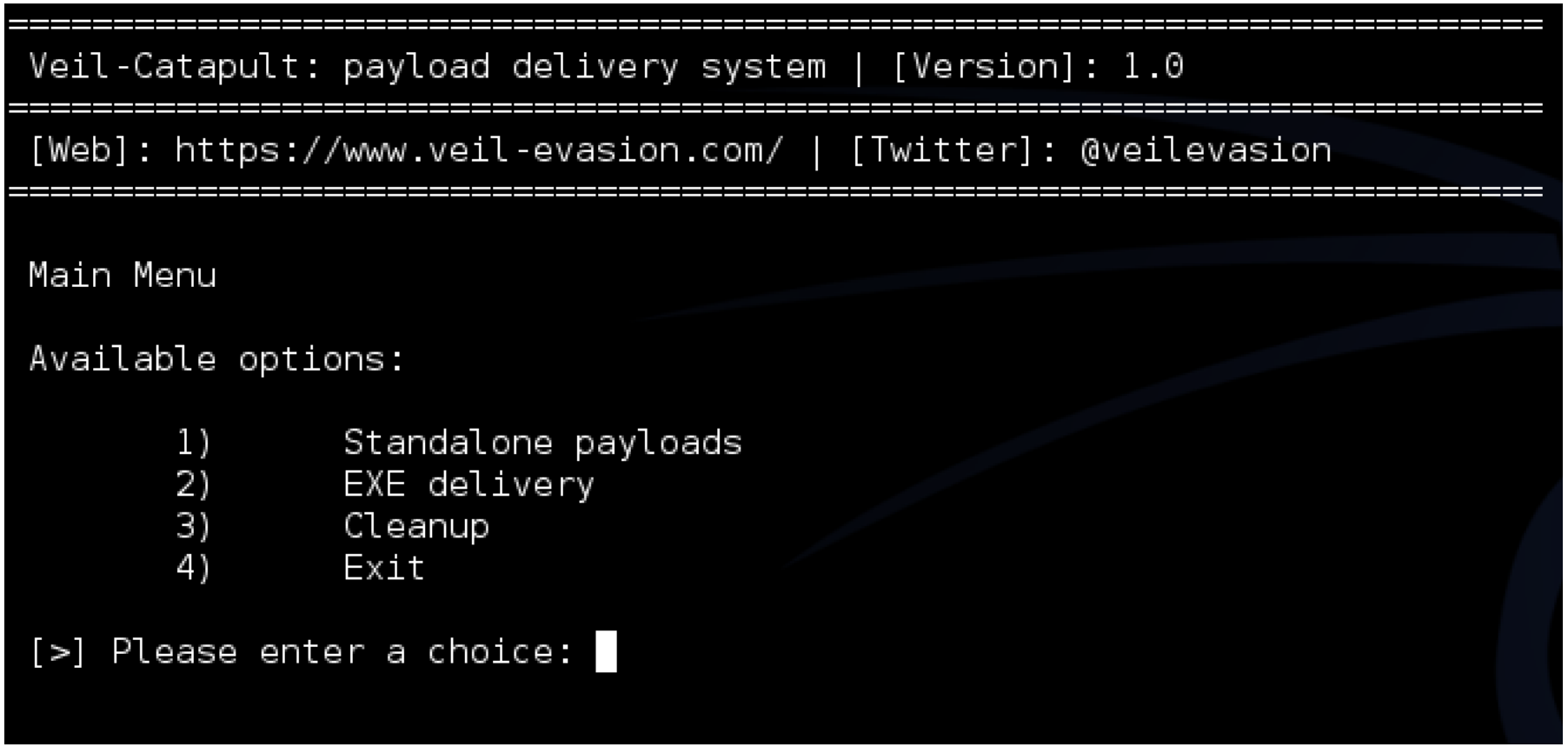
Veil Framework

Veil-Catapult

Veil-Catapult

- Our payload delivery system
- Features nice integration with Veil-Evasion for on-the-fly payload generation
- Cleanup scripts generated for payload killing and deletion
- Command line flags for every option

Veil-Catapult



.EXE Delivery

- Users can invoke Veil-Evasion to generate a payload, or specify an existing .exe
- Payloads are delivered in one of two ways:
  - upload/execute using Impacket and pth-toolkit
  - host/execute \\UNC path to the attacker's box
- UNC invocation gets otherwise detectable .EXEs right by some AVs (lol @MSE)

Standalone Payloads

- **Powershell:** shellcode injector, bye bye disk writes
  - http://www.exploit-monday.com/2011/10/exploiting-powershells-features-not.html

- **Barebones python:** uploads a minimal python installation to invoke shellcode (see: next slide)

- **Sethc backdoor:** issues a registry command to set up the sticky-keys RDP backdoor

Barebones Python

- Uploads a minimal python .zip installation and 7zip binary
- Python environment unzipped, shellcode invoked using “-c …”
- The only files that touch disk are trusted python libraries and a python interpreter
- Gets right by reputation filters and antivirus!

DEMO #2

How to Stop Us

#avlol

Predictable Behavior

- A lot of malware and Veil-Evasion payload behaviors are fairly predictable:
  - Immediate reverse connection to a target
  - RWX memory page allocation, binary code copying, thread creation, etc.
- A small set of APIs are usually used in a very specific and non-standard way

Ambush IPS

- An intrusion prevention system that allows for flexible rules to be written for API calls
- Rules can be written to stop Meterpreter stagers without affecting normal execution
- http://ambuships.com/

EMET

- Microsoft's Enhanced Mitigation Experience Toolkit
- Has some mechanisms that stop the ability for an executable to inject shellcode
- Ruins powershell shellcode injection
- http://technet.microsoft.com/en-us/security/jj653751

Where to Find Veil

- **Web**: https://www.veil-framework.com
- Now in Kali! **apt-get install veil**
- **Github**: https://github.com/Veil-Framework/Veil/

Questions?

- harmj0y@veil-framework.com
  - @harmj0y
- chris@veil-framework.com
  - @ChrisTruncer
- shiv@veil-framework.com
  - @TheMightyShiv