The Veil-Framework

Will (@harmJ0y)

Veris Group – Adaptive Threat Division

The Veil-Framework

- A toolset aiming to bridge the gap between pentesting and red teaming capabilities
  - **Veil-Evasion**: flagship tool, generates AV-evading executables
  - **Veil-Catapult**: initial payload delivery tool
  - **Veil-PowerView**: situational awareness with Powershell
  - **Veil-Pillage**: fully-fledged post-exploitation framework

Veil-Evasion

#avlol

The Initial Problem

- Antivirus doesn't catch malware but (sometimes) catches pentesters

File name: meterpreter.exe

Detection ratio: 35 / 48

Our Initial 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/our stagers in an AV-evading way

Twitter Reaction

Chris
@obscuresec

Following

The main thing that bothers me about @veilframework is that new pentesters will never know what it was like to do this all manually. :)

scriptjunkie
@scriptjunkie1

Following

@obscuresec @veilframework Back in my day, we had to obfuscate bits by hand uphill both ways!

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 and "auxiliary" modules as well

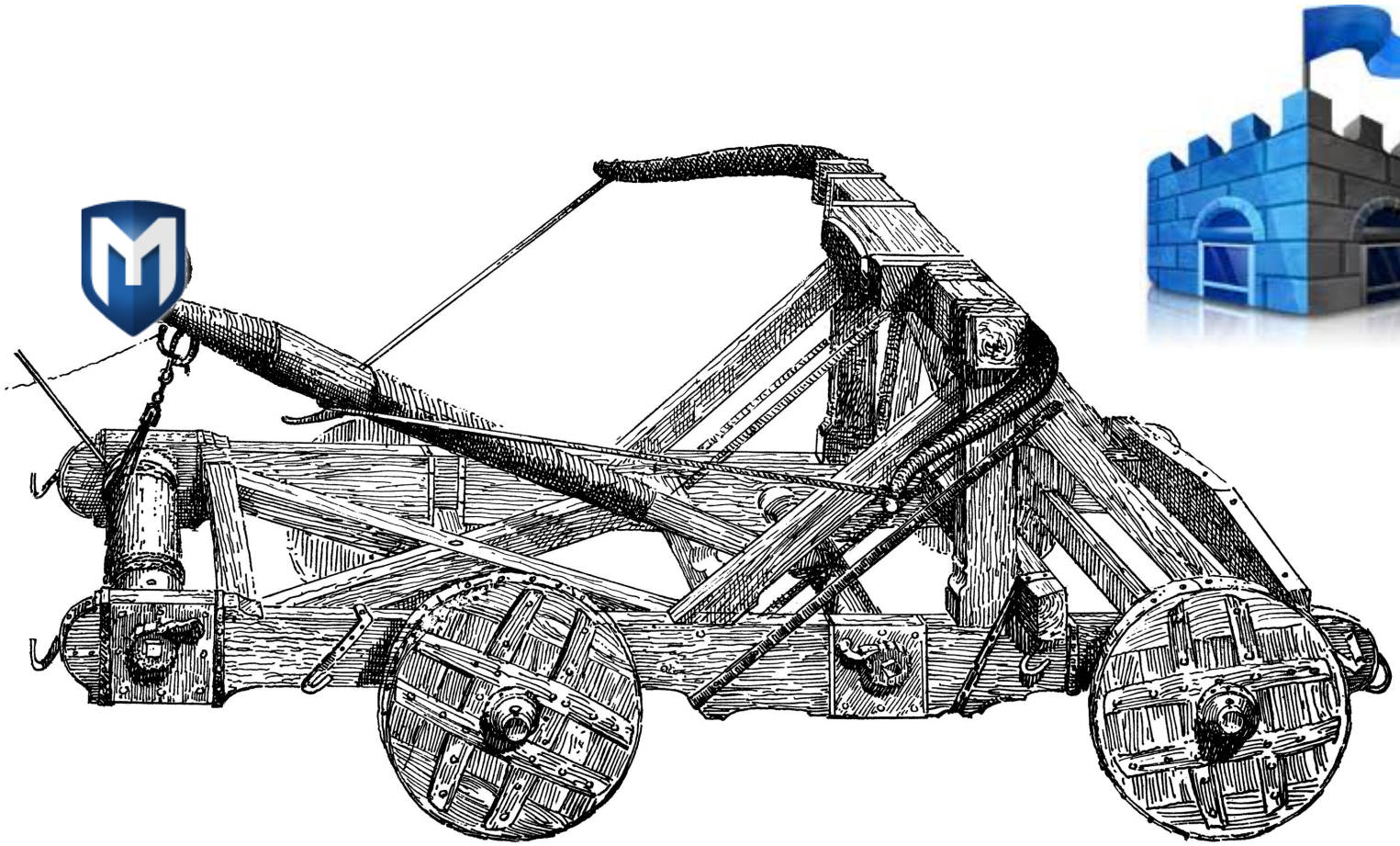
V-Day

- Since 9/15/2013, we've release at least one new payload on the 15th of every month
- 30+ currently published payload modules
- 20+ additional payloads have been developed so far
- we're going to be releasing for a while :)

Veil-Catapult

Payload Delivery

Veil-Catapult



Veil-Catapult

- Our basic payload delivery tool, released at Shmoocon '14
- Tight integration with Veil-Evasion for on-the-fly payload generation, can upload/execute or host/execute
- Cleanup scripts generated for payload killing and deletion
- Now obsoleted with the release of Veil-Pillage

Veil-PowerView

Situational Awareness with Powershell

Veil-PowerView

- A pure Powershell situational awareness tool
- Arose partially because a client banned “net” commands on domain machines
- Otherwise initially inspired by Rob Fuller’s netview.exe tool
  - Wanted something a bit more flexible that also didn’t drop a binary to disk
- Started to explore and expand functionality

Get-Net*

- Full-featured replacements for almost all "net *" commands, utilizing Powershell AD hooks and various API calls
  - Get-NetUsers, Get-NetGroup, Get-NetServers, Get-NetSessions, Get-NetLoggedon, etc.
- See README.md for complete list, and function descriptions for usage options

Meta-Functions

- **Invoke-Netview**: netview.exe replacement
- **Invoke-ShareFinder**: finds open shares on the network and checks if you have read access
- **Invoke-FindLocalAdminAccess**: port of local_admin_search_enum.rb Metaspoit module
- **Invoke-FindVulnSystems**: queries AD for machines likely vulnerable to MS08-067

User Hunting

- **Goal:** find which machines specific users are logged into
- **Invoke-UserHunter**: finds where target users or group members are logged into on the network
- **Invoke-StealthUserHunter**: extracts user.HomeDirectories from AD, and runs **Get-NetSessions** on file servers to hunt for targets
  - Significantly less traffic than **Invoke-UserHunter**

Domain Trusts

- PowerView can now enumerate and exploit existing domain trusts:
  - **Get-NetForestDomains**: get all domains in the forest
  - **Get-NetDomainTrusts**: enumerates all existing domain trusts, à la nltest
- Most PowerView functions now accept a "**-Domain <name>**" flag, allowing them to operate across trusts
  - e.g. **Get-NetUsers –Domain sub.test.local** will enumerate all the users from the sub.test.local domain if an implicit trust exists

Veil-Pillage

Post-exploitation 2.0

Veil-Pillage

- A post-exploitation framework being released at Defcon
- Multiple trigger options (wmis, psexec, etc.)
- Completely modular, making it easy to implement additional post-exploitation actions
- Comprehensive logging and cleanup capabilities

exe_delivery

- Catapult functionality ported to Pillage
- Executables can be specified, or generated with seemless Veil-Evasion integration
- .EXEs are then uploaded/triggered, or hosted/ triggered with a \\UNC path
  - This gets some otherwise disk-detectable .EXEs right by some AVs

powersploit/*

- Several PowerSploit modules are included in Pillage
- A web server is stood up in the background
  - the 'IEX (New-Object Net.WebClient).DownloadString(...)' cradle is transparently triggered
- Makes it easy to run PowerSploit across multiple machines

Hashdumping

- Different approaches work in different situations
- Dependent on architecture, Powershell installation, AV-installation, etc.
- Some involve dropping well-known, close-sourced tools to disk
  - sometimes this is needed, but we want to stay off disk as much as possible

Hashdumping: Pillage Style

- Let's aggregate some of the best techniques and build some logic in:

  if (powershell_installed) { Powerdump/PowerSploit}

  else { determine_arch {
  host/execute appropriate binaries } }

- Expose these techniques to the user for situation-dependent decisions

Questions?

- harmj0y@veil-framework.com
  - @harmj0y
- harmj0y in #veil/#armitage on freenode
- https://www.veil-framework.com
- Get the Veil-Framework:
  - **Github:** https://github.com/Veil-Framework/
  - **Read more**: https://www.veil-framework.com