Red Team Operating in a Modern Environment

Learning to Live Off the Land

But what does that even mean?

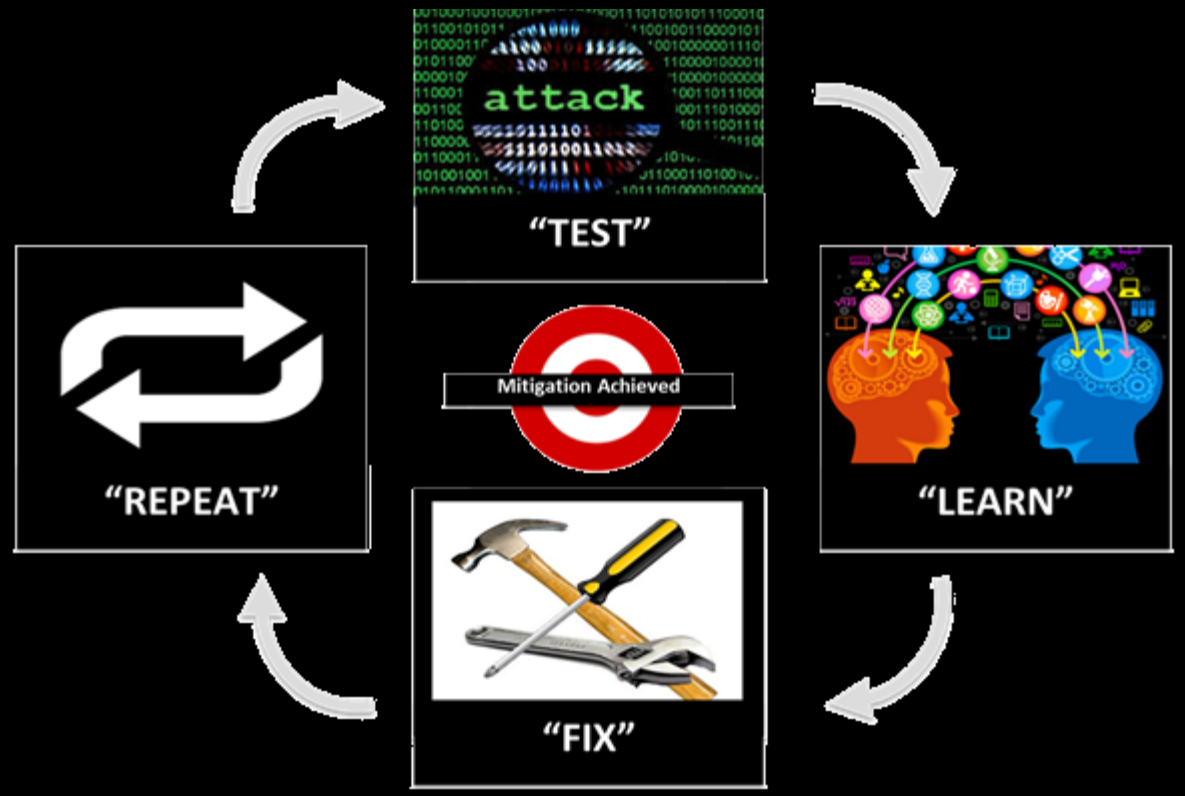
- This presentation will cover:
  - My interpretation of modern red teaming (not the “purple side”)
  - Common security mechanisms encountered in red team engagements
  - Methods to bypass these protections using a bit of critical thinking
  - The importance of “living off the land” (and how to reduce your carbon emissions by 25%)
  - Heavily focused on Windows Environments, but concepts apply to everything

whoami

- Red Team Operator @ ReliaQuest
  - Anything I say is not representative of my employer
- Passionate about security
  - Specific focus on offensive and stealthy operations
  - I don’t like to get caught
- Twitter/Github/IRC/Carrier Pigeon: Und3rf10w

Red Teaming

- Essentially:
  - Long term engagements
  - Expanded scope
  - Large focus on adversary emulation
  - Usually a blackbox engagement
  - Lower focus on exploitation,
    - Configuration abuse
    - Detection enhancement



- Why bother?
  - Enhances all facets of an organization
  - Capable of exposing vulnerabilities a pentest simply can not
  - Greater overall value to security teams

A Look at Enterprise Security Tools

- What types of security defense are commonly in play in a large enterprise environment?
  - User training
  - SIEM
  - IDS
  - Host based protections
  - Interceptive inspections
    - SSL Decryption Tools
    - Sandbox inspection tools

Initial Compromise Techniques

Channeling Your Inner Nigerian Prince

First rule of thumb

- Stop thinking like a penetration tester
  - Every action should be methodical
  - Every action should be performed with stealth in mind
- Think like an attacker
  - Channel your inner stalker
    - Get really comfortable with your OSINT
      - Plenty of Facebook/LinkedIn trawling
      - I can easily spend a month doing but this
      - Makes for easier execution of future campaigns
  - Operate quietly
    - One wrong defense tripped could be losing everything
    - Get in the mindset that your livelihood is at risk
    - Obfuscate everything

Two Ways I've Gained Initial Compromise

Physical Reconnaissance

- Enumerate any infrastructure exposed to the public
- It'd be a shame if something were to happen

Konboot & local console attacks

- If we can USB boot, try to Konboot
- If we're at a publicly accessible terminal, try to get a payload

Social Recon

- Social network trawling
- Generate org lists; gather accessible company documents, website, partners, and make templates

Enumerate technical protections

- Who is hosting the email infrastructure?
- What kind of attachments are allow?
- Is there a sandbox in place?

Execute phishing campaign

- Consider how to get people to execute your payload
- Consider how to get your payload past protections
  - Sandboxing
  - Anti-virus

Phishing for Wins

- Easily the most effective route of entry
- Step your phish game up!
  - I can not stress how important in-depth recon is
- Gather template documents
  - Google Dork: $COMPANY_NAME filetype:doc
  - Search for email signatures
    - Mailing lists are great for this
- Build a very small target list
  - Each campaign should be limited to a small list of specifically chosen targets
  - You should gather as much as possible about those targets
    - OSINT EVERYTHING

Phishing Infrastructure

Phishing Payloads

- Two types of Payloads
- Browser based attacks
  - Browser Exploit
  - Login Portal Clone
- Attachment based attacks
  - Office Document w/Macro
  - Zip file with JS script attached (common ransomware vector)

Type of Payload	Chance of Success	Profit	Time Investment
Browser Based	Higher	Lower	Lower
Attachment Based	Lower	Higher	Higher

Enumeration for Phishing

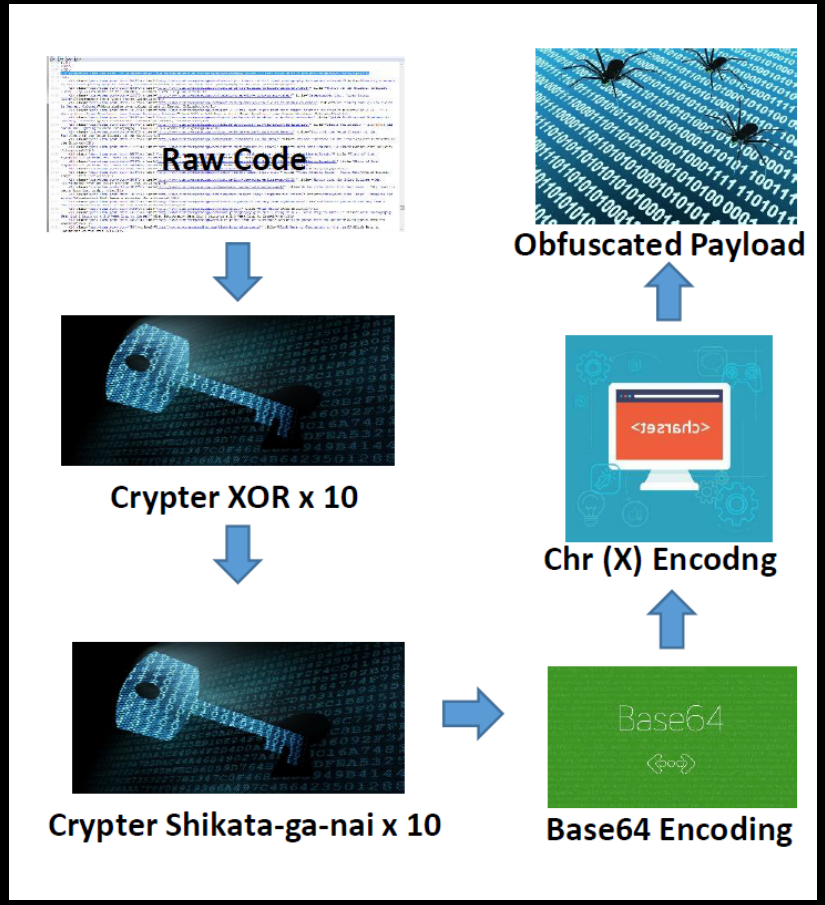
- As an operator, you need to enumerate possible protections in place
  - Send a document with a begin payload that informs you when it has been activated
    - (Small pixel in the email and/or possibly a macro)
    - Based on the amount of time it took to activate from the time it got sent, and how many activations were made (as in, you saw multiple activations from different sources within a short window of time), determine if you're dealing with a sandbox
  - Look at MX records:

Protection Type	MX Record
Barracuda	.barracudanetworks.com
Proofpoint	.secureserver.net
Office 365	.mail.protection.outlook.com
Google	.googlemail.com

Enhancing Your Macros

- Obfuscation of VBA code is critical
- Put encryption in your code, even if it's XOR
- Make it polymorphic
  - Store the decryption key as a document variable, modify the encryption key upon execution, resave the document
- Look for more novel ways of executing things
  - Executing "`powershell.exe`" VS "`rundll32 SHELL32.DLL,ShellExec_RunDLL powershell.exe`"
- Page skip / content revelation upon macro execution
- Resources:
  - https://github.com/Pepitoh/VBad
  - https://github.com/khr0x40sh/MacroShop
  - https://www.youtube.com/watch?v=BGWy1R7NyOk
  - https://blogs.mcafee.com/mcafee-labs/macro-malware-employs-advanced-sandbox-evasion-techniques/?utm_source=twitter&utm_campaign=Labs#sf37458688

Example of an ‘enhanced’ macro



Teach a Man To Phish

- Have access to a network, but can no longer pivot through technical exploitation?
- Internal Phish!
  - Imagine trust as a weighted system:
  - **-5** for having an attachment
  - **-10** for having a macro
  - **+20** if sender is an internal user
  - Send phishing emails to higher-privileged users from compromised user's accounts
    - Don't expect to win style points with this.
    - Also fairly risky from an attacking perspective, but may be worth it

Endpoint Operations

This is not the end

Internal Security Controls

- There are a number of different protections to consider when operating internally within an environment:
  - What privileges do we have as a user?
  - What type of monitoring & detection is in place on the endpoint?
  - What type of monitoring & detection is in place on the network?
  - What type of monitoring & detection is in place on the logs?
- Proper reconnaissance is key
  - Enables you to make correct operating decisions
- Asynchronous payloads
  - This is used by Cobalt Strike & Empire
  - Combine this with kill-dates and operation times and you become much more effective

Be Like Ninja

- Observe user actions
  - What services do they connect to normally?
  - What servers does their workstations communicate with normally?
  - What services are running on the workstation?
  - What privileges does this user have?
- Stop dropping binaries
  - Execute as much in memory as possible
  - PowerShell is the greatest post-exploitation tool ever
    - Execute commands in memory
    - Import and execute DLLs in memory

Host Based Protections

- Generally categorized in 2 groups:
  - Antivirus solutions
    - Generally doing basic signature and heuristics checks
      - SEP
      - Windows Defender
  - Advanced Endpoint Solutions
    - Much more advanced checks such as kernel hooking, memory scraping, whitelisting
      - Cylance
      - Carbon Black

Damn HIDS, you scary!

- Bypass techniques I've observed:
  - Do nothing with your backdoor but inject a thread to operate out of into the analysis process
    - Most endpoint solutions whitelist themselves by default
  - Abuse Image File Execution Options
    - You can attach any program to any executable using the registry key:
    - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options**{name of the executable}**, and adding the value: **"Debugger"="{full path to the debugger}"**
    - A malformed debugger path will result in the program not executing at all
    - Carbon Black in process of deploying patch for this
      - I haven't tested this technique with Cylance
  - Code sign your application

Getting Credentials

- Obligatory Mimikatz plug
  - Execute this in memory (hi PowerShell)
  - Modify it: "`sed –i 's/mimikatz/mimipuppies/`", you'll be surprised at the effectiveness
  - Do this periodically over a 1-month period on your compromised hosts
- Obligatory Responder Plug
  - This will not always work in every environment
  - But when it does, it's basically magic
  - See Inveigh for in memory execution via PowerShell
- Local credential theft
  - Many PowerShell scripts exist for this (KeeTheif, tons of PowerSploit scripts)
- Keyloggers, obviously

Persistent Payload Design

Impregnate yourself

An Ideal Backdoor

- Ideally, a backdoor should be execute upon login or system startup
- It should be lightweight
- It should only perform any malicious operations deliberately at the instruction of the operator
  - Ideally, the initial backdoor wouldn't ever do anything malicious except inject threads into other running processes
- Shouldn't get caught by standard signature checks or heuristics
- Capability of in place upgrade/modification with little effort
- Should appear fairly begin upon cursory inspection
  - Generic Print Driver.exe

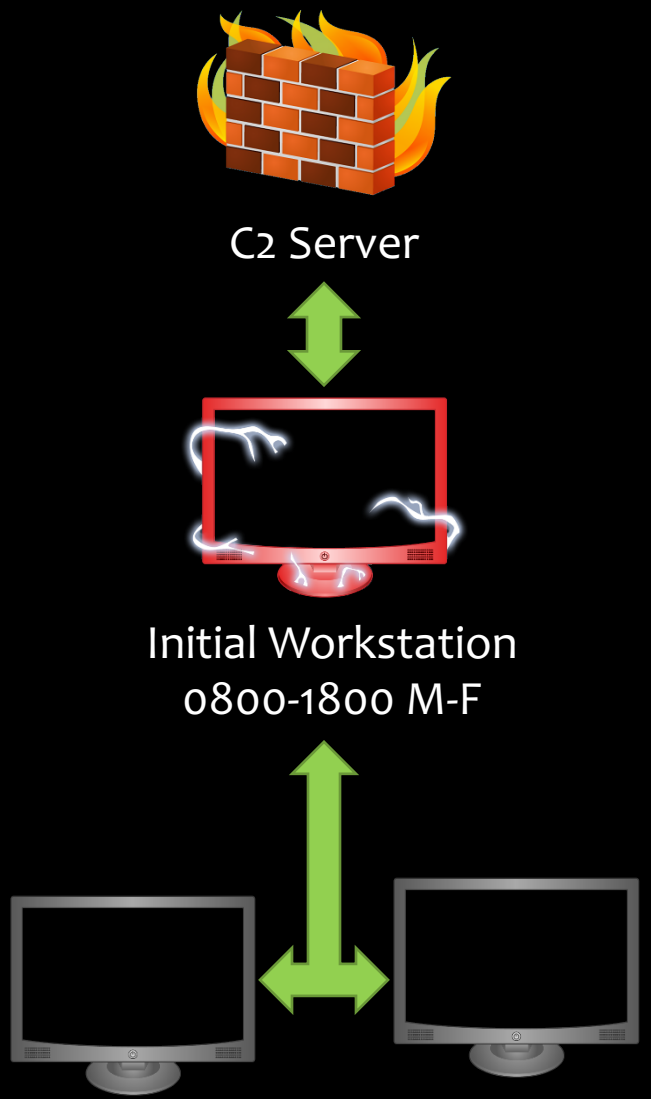
It’s my first time, where does it go?

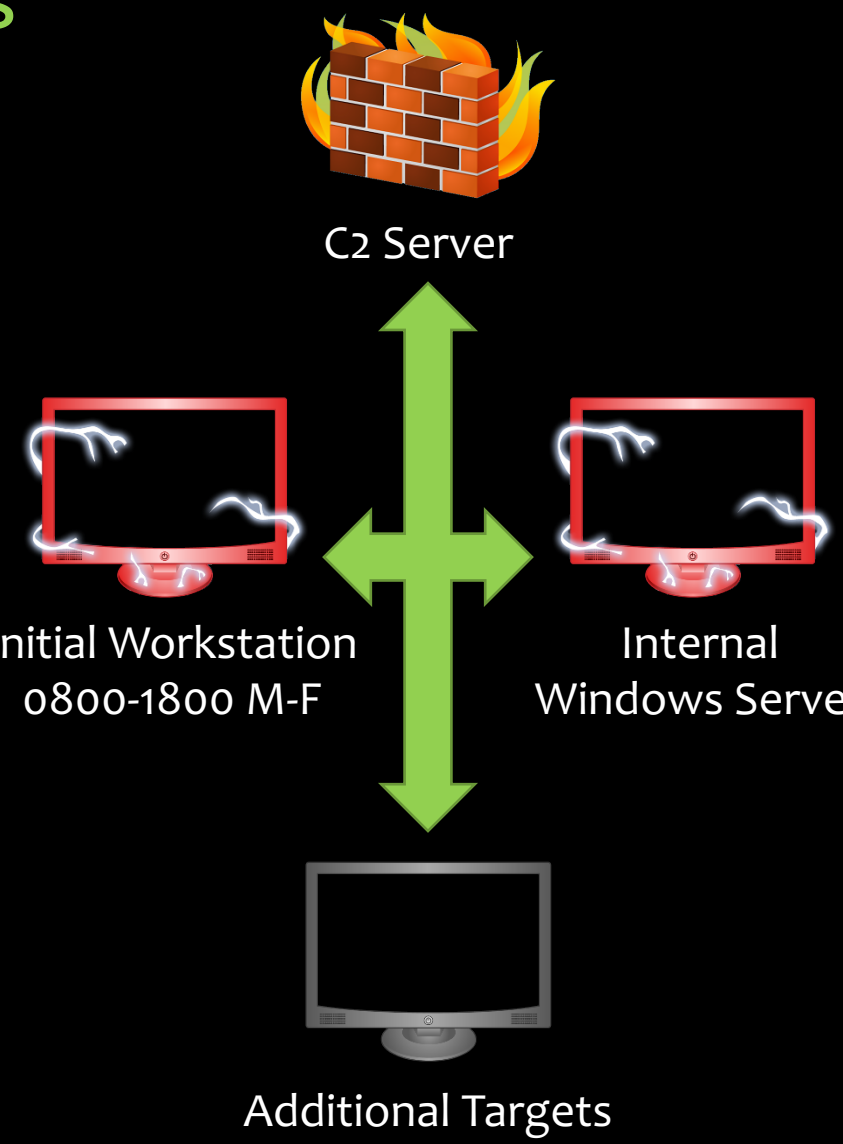
- Are you an admin?
  - C:\Windows\System32
  - C:\Windows\SysWOW64
  - Any (within reason) existing application directory
  - Registry startup entry
- Not admin? That’s okay
  - %TEMP%\..\..\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
    - At the very least, put a shortcut to where you hide your actual binary
  - Create a malformed outlook rule
    - https://silentbreaksecurity.com/malicious-outlook-rules/

Internal Operations

Alternatively: “I can’t believe that worked…”

Internal Infrastructure Design for Attackers





Pivoting Mechanisms

- PSExec
  - Used by system administrators and hackers everywhere
  - Avoid using the default PSExec binary
    - May tools have a signature for it
    - Microsoft ATA will not catch custom versions
  - Possible to PSExec powershell commands if you want to avoid touching disk
- Winrm
  - Use Windows Remote Management
  - Not usually enabled
- WMI
  - Use Windows Management Instrumentation
  - Worth a shot if PSExec is not working

Think Like a SysAdmin

- Internal networks are a default password art exhibition
- Built-in system administrator tools are of great use to an attacker
  - Net.exe
  - PowerShell
  - PSExec
- Third party tools are useful as well
  - PowerView
  - Bloodhound

Command	Description
Net group “domain comptuers” Net view /domain:$DOMAIN	Enumerate hosts in a domain
Net group “Domain Admins” /Domain	Enumerate DAs in the domain
netsh wlan show profiles name=*	Show WLAN profiles
netsh wlan show profiles name=$networ key=clear	Dump WLAN key for $network
Nltest /domain_trusts	Show current domain trusts
Nltest /dcname:$domain	Identify PDC for domain
Dir \\$SERVER\c$	Test if you’re an admin on $SERVER

Access Token Abuse

- Access tokens are associated with every process and thread
- Core Windows mechanism that handles authentication for Single Sign on
- Managed by lsass
- Steal tokens by spawning a new thread as your desired user, grab the token from that thread, kill the process and move on
- If you have the krbtgt hash, you can abuse the Golden Ticket
  - Generally this hash is pretty much never changed. Your ticket is essentially as good as the length of time the hash isn't changed
  - There is very little in the way of logging of golden tickets

Targets of Opportunity

A low blow to the crown jewels

Targets to Focus On

- Automation Servers
  - Puppet, Anisble, SCCM
  - Used by SysAdmins everywhere
  - These servers usually have the literal keys to the kingdom. Monitoring typically minimal as they are oft treated as “automate & forget”
- CA Servers
  - Hello code signing!
    - Steal internal code signing certificate, go update all of your backdoors
- SIEM
  - Mostly for the hilarity of it all. These servers also usually contain valuable amounts of sensitive data
- Network Engineers and System Administrators
  - Once you have the ability to pivot and pwn their workstations, you’re pretty much guaranteed infinite keys to the kingdom
    - Looks for SSH keys, KeePass databases, network architecture diagrams

Questions Please!

No seriously, please ask some

There's a lot I didn't wanted to cover that I didn't get the chance to

Contact Info

Jabber: Und3rf10w@fuckd.at
Jabber: Und3rf10w@jabber.de
IRC: Freenode/Und3rf10w
Twitter: @Und3rf10w
Github: Und3rf10w

Email: *Please attempt to contact me via other means first*