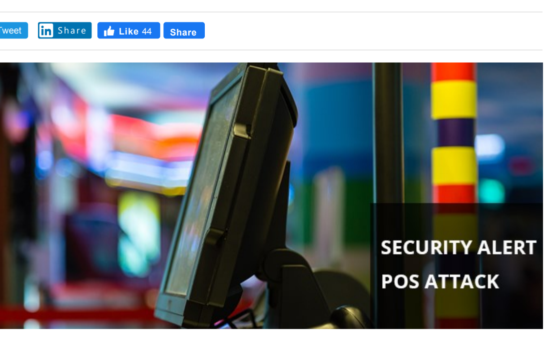
MOVING TARGET DEFENSE BLOG

CYBERSECURITY TRENDS, EXPLORING MOVING TARGET DEFENSE AND PUTTING ENDPOINT THREAT PREVENTION FIRST

FIN8 IS BACK IN BUSINESS, TARGETING THE HOSPITALITY INDUSTRY

Posted by **MICHAEL GORELIK** on June 10, 2019

Find me on:



During the period of March to May 2019, Morphisec Labs observed a new, highly sophisticated variant of the ShellTea / PunchBuggy backdoor malware that attempted to infiltrate a number of machines within the network of a customer in the hotel-entertainment industry. It is believed that the malware was deployed as a result of several phishing attempts.

The last documented version of ShellTea was in 2017, in a **POS malware attack**. Given the nature of the industry targeted in the attack uncovered by Morphisec, we assume that this was also an attempted POS attack. As the attack was prevented by the Morphisec solution, the POS malware could not be downloaded to the machines.

This is the first attack observed during 2019 that can be attributed to *FIN8* with high probability, although there are a few indicators that overlap with known FIN7 attacks (URLs and infrastructure).

In this report, we investigate this latest variant of ShellTea, together with the artifacts it downloaded after the Morphisec Labs team detonated a sample in a safe environment.

FIN8 - TECHNICAL DETAILS

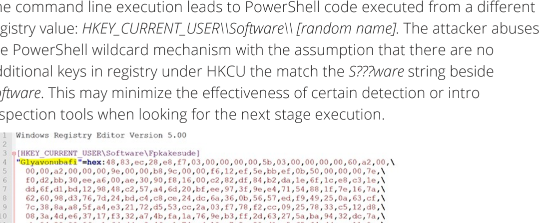
We begin by examining the different stages of the fileless dropper in detail.

FILELESS EXECUTION

Following successful infiltration, the malware persists through registry: HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

```
"C:\\Windows\\System32\\cmd.exe '/c' 'powershell.exe' '-w' '1' '-nop' '-c'
         'start-process powershell.exe -windowstyle hidden -arg '-nop -c $a=
                (Get-ItemProperty registry::HKCU\\S???ware\\Fpkakesude);IEX $a.Xshuzugewogazi''"
```

The command line execution leads to PowerShell code executed from a different registry value: *HKEY_CURRENT_USER\\Software\\ [random name]*. The attacker abuses the PowerShell wildcard mechanism with the assumption that there are no additional keys in registry under HKCU the match the *S???ware* string beside *Software*. This may minimize the effectiveness of certain detection or intro inspection tools when looking for the next stage execution.

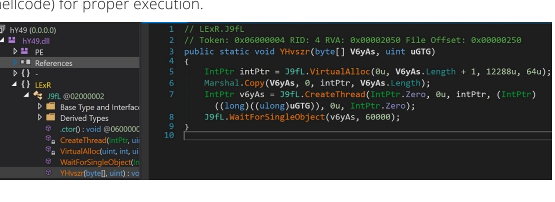


The code from the [*random name*] key executes an additional PowerShell command that decodes base64 assembly and invokes it within the memory while passing it as parameters an additional [*random name*] registry value – the ShellTea shellcode:

"Xshuzugewogazi"="$null=[System.Reflection.Assembly]::Load([System.Convert]::FromBase64String("TVq…."));[LExR.J9fL]::YHvszr((Get-ItemProperty -Path HKCU:Software\\Fpkakesude -Name Glyavonubafi).Glyavonubafi, 0);"

EXECUTING .NET ASSEMBLY FOR SHELLCODE EXECUTION

The base64 encoded assembly that we saw in the previous stage is a .NET stager that self-injects a shellcode (the parameter from previous stage) by creating a new thread with the shellcode entry. The thread also needs a parameter (the same shellcode) for proper execution.



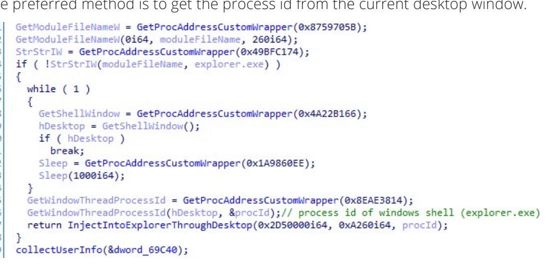
SHELLCODE

CUSTOM FUNCTION RESOLUTION

To operate and evade standard analysis tools, most of the functions are hashed. The hashing algorithm has a high degree of similarity to the previous ShellTea version, with a slight modification of the seeds and constants. In this version the attacker also utilizes functions from ole32 for stream processing.

INJECT INTO EXPLORER

As in the previous version, ShellTea continues the operation after persisting through the explorer.exe process. While it includes multiple ways to find Explorer, the preferred method is to get the process id from the current desktop window.



After finding the process id, the shellcode uses standard functions to allocate and write memory within Explorer and then uses low-level API *RtlCreateUserThread* for thread injection.

VIRTUAL ENVIRONMENT AND SANDBOX BYPASS

ShellTea utilizes a number of techniques to identify if it is running within a virtual environment or is being monitored.

Virtual environment by firmware

ShellTea extracts the firmware string information using *NtQuerySystenInformation* with the *SystemFirmwareTableInformation* flag. Then it searches for a set of known strings as described in this article by Checkpoint (additional strings have been added in this case, e.g. Visual studio).

Looking for monitoring processes

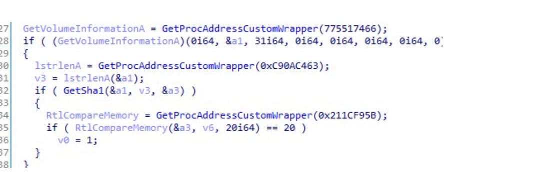
As part of the anti-debugging or anti-monitoring techniques, ShellTea iterates over all the running processes, applies CRC32 on each process name (after converting the string to capital letters), and then compares the value against a predefined set of CRCs. Note that a bug in the previous version has been fixed and more CRCs were added.

We wrote a python script which is based on a set of known processes and identified the list of the processes that are being searched for. These are:

WINDBG.EXE, WIRESHARK.EXE, PROCEXP.EXE, PROCMON.EXE, TCPVIEW.EXE, OLLYDBG.EXE, IDAG.EXE, IDAG64.EXE, DUMPCAP.EXE, FILEMON.EXE, IDAQ64.EXE, IDAQ.EXE, IMMUNITYDEBUGGER.EXE, PETOOLS.EXE, REGMON.EXE, SYSER.EXE, TCPDUMP.EXE, WINDUMP.EXE, APIMONITOR.EXE, APISPY32.EXE, IRIS.EXE, NETSNIFFER.EXE, WINAPIOVERRIDE32.EXE, WINSPY.EXE

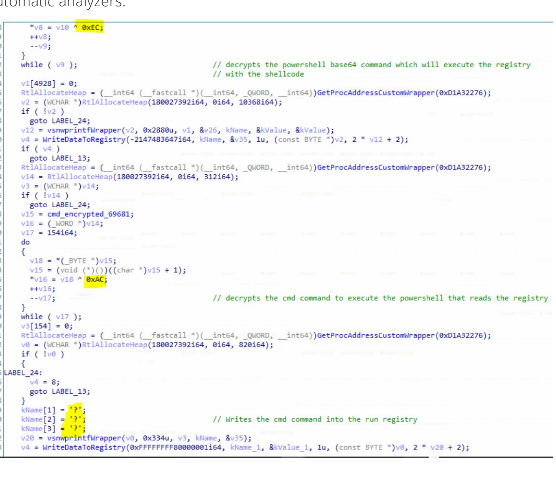
Validate Hard Disk Volume

The hard disk volume name is hashed with SHA1 and compared to a preconfigured SHA1.



PERSISTENCY

Upon successful bypassing of sandboxes, the shellcode executes a persistency module. If the attack is yet to be persistent (validates beforehand), it decrypts the PowerShell base64 command, then decrypts the CMD command for persistency and writes those into the registry as described in the first step. Note that every string is decrypted with different XOR parameter which may fail some of the automatic analyzers.



C2 PROTOCOL

COMMANDS

The ShellTea backdoor communicates on top of HTTPS and supports a number of commands based on what is returned from the C2 server.

- It may write data/shellcode it received from the C2 into registry
- It may reflectively load the delivered executable into the process (and then execute it)
- It may create a file and execute it as a process, then mark it as deleted (after restart)
- It may execute the shellcode as is by creating additional thread
- It may execute any PowerShell command using downloaded native Empire ReflectivePicker (will be described later).

COMMUNICATION

ShellTea is proxy aware malware – if direct communication fails it will try to execute the proxy aware API. Most of the API are standard and are mapped from wininet. Following a decryption of the embedded domains we get the following list:

telemerty-cdn-cloud[.]host, reservecdn[.]pro, wsuswin10[.]us, telemetry[.]host

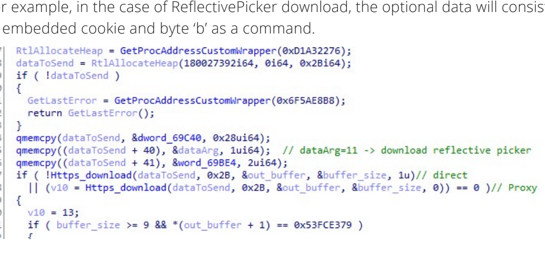
At the time of the investigation, telemerty-cdn-cloud[.]host was used and mapped to 104.193.252[.]162.

Note that cdn substrings are also used in recent Fin7 attacks and it is a convenient method to bypass some of the network filters and DLP solutions.

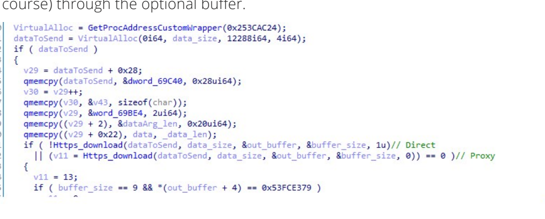
ShellTea also uses ole32 Stream object functions (e.g. CMemStm::Write) to manipulate the downloaded memory stream (downloaded by InternetReadFile directly into the stream).

The C2 identifies the post request using the additional optional data that is sent immediately after the headers request using *HttpSendRequestA*.

For example, in the case of ReflectivePicker download, the optional data will consist of embedded cookie and byte 'b' as a command.



In case a buffer needs to be sent back (in case of recon data collected on the endpoint), a magic header cookie is attached to the data and sent as is (encrypted of course) through the optional buffer.

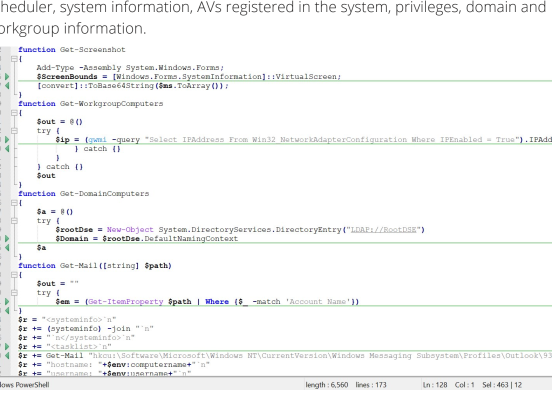


RECON STAGE

One of the first artifacts the shellcode downloads is a PowerShell code and a .NET native ReflectivePicker. Because PowerShell was executed outside PowerShell (from within the Explorer process) it will bypass many of the blacklisting defenses.

POWERSHELL DOWNLOAD

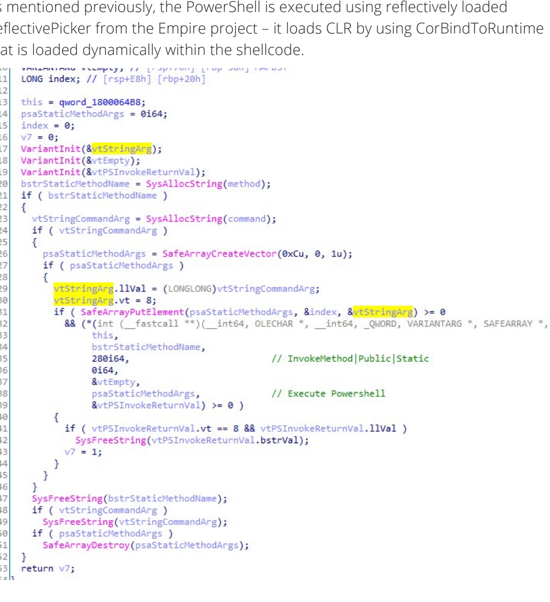
The PowerShell script collects all possible information on the user and the network, including snapshots, computer and user names, emails from registry, tasks in task scheduler, system information, AVs registered in the system, privileges, domain and workgroup information.



The results are Gzipped and saved under random file in the temp folder. Following successful collection of information, the data is send back to the C2 and the file is deleted.

EMPIRE REFLECTIVEPICKER

As mentioned previously, the PowerShell is executed using reflectively loaded ReflectivePicker from the Empire project – it loads CLR by using CorBindToRuntime that is loaded dynamically within the shellcode.



CONCLUSION

The hospitality industry, and particularly their POS networks, continues to be one of the industries most targeted by cybercrime groups. In addition to this attack by FIN8,we've seen multiple attacks by FIN6, FIN7 and others.

Many POS networks are running on the POS version of Window 7, making them more susceptible to vulnerabilities. What's more, attackers know that many POS systems run with only rudimentary security as traditional antivirus is too heavy and requires constant updating that can interfere with system availability.

As we see here, attack syndicates are constantly innovating and learn from their mistakes – the numerous improvements and bug fixes from the previous version of ShellTea are evident. The techniques implemented can easily evade standard POS defenses.

Morphisec immediately prevented this attack from ever getting to the point where it could access POS endpoints. It is lightweight, with no need for updates and does not need to be online to provide full protection. Moreover, it serves as a compensating control for Windows 7 systems, providing a virtual patch that protects vulnerabilities.

ARTIFACTS

SHELLTEA BACKDOOR:

6353D7B18EE795969659C2372CD57C3D

4B9EFD882C49EF7525370FFB5197AD86

REFLECTIVEPICKER:

DC162908E580762F17175BE8CCA25CF3

PowerShell recon script:

4BEB10043D5A1FBD089AA53BC35C58CA

DOMAINS:

telemerty-cdn-cloud[.]host

cdn-amaznet.club

reservecdn[.]pro

wsuswin10[.]us

telemetry[.]host

IPs:

104.193.252[.]162:443

37.1.204[.]87:443

SUBSCRIBE TO OUR BLOG

Stay in the loop with industry insight, cyber security trends, and cyber attack information and company updates.

FIRST NAME
LAST NAME
EMAIL
COMPANY
SUBSCRIBE

SEARCH OUR SITE

Keyword...

RECENT POSTS

Parallax: The New RAT on the Block
Remote Employees Offer Different Security Challenges
Why Client-Grade Technology Doesn't Cut It for Cloud Workload Protection
Trickbot Delivery Method Gets a New Upgrade Focusing on Windows 10
Introducing the Morphisec Unified Threat Prevention Platform -- Version 4
Endpoint Security Is Harder than Ever
Trickbot Trojan Leveraging a New Windows 10 UAC Bypass
Morphisec Protects Customers Against Internet Explorer Scripting
Endpoint Detection and Response Is Not the Next Step
Are Guests Safe From a Hotel Data Breach?

POSTS BY TAG

Cyber Security (94)
Endpoint Security (74)
Attack Analysis (45)
Cyber Attacks (45)
Company News (38)
See all