

The Case of Gyges, the Invisible Malware

Government-Grade now in the Hands of Cybercriminals

SentinelOne July 2014

By Udi Shamir, Head of Research

Introduction

In March 2014, SentinelOne Research discovered a sophisticated piece of malware dubbed Gyges that is virtually invisible and capable of operating undetected for long periods of time.

We first detected Gyges with our heuristic sensors and then our reverse engineering task force performed an in-depth analysis. It appears to originate from Russia and be designed to target government organizations. It comes as no surprise to us that this type of intelligence agency-grade malware would eventually fall into cybercriminals' hands.

Gyges is an early example of how advanced techniques and code developed by governments for espionage are effectively being repurposed, modularized and coupled with other malware to commit cybercrime.

The following report explains how Gyges escapes detection from traditional security technologies.

The Case of Gyges, the Invisible Malware

Executive Summary

SentinelOne has been intensively researching government-grade malware and rootkits for the past few years. This specific Gyges variant was detected by our on-device heuristic agents and caught our attention due to its sophisticated anti-tampering and anti-detection techniques. It uses less well-known injection techniques and waits for user inactivity, (as opposed to the more common technique of waiting for user activity). This method is clearly designed to bypass sandbox-based security products which emulate user activity to trigger malware execution.

The analysis of Gyges generated hundreds of indicators from our heuristics engine that provided new and intriguing findings.

For example, Gyges uses a hooking bypass technique that exploits a logic bug in Windows 7 and Windows 8 (x86 and x64 versions). It also combines highly advanced anti-debugging and anti-reverse-engineering. Interestingly, the malicious code used for all of these evasion techniques is significantly more sophisticated than the core executable. That led us to believe that it was previously used as a “bus” or “carrier” for much more sophisticated attacks such as government data exfiltration. So we started digging, and eventually recovered government traces inside the “carrier” code, which we later connected to previous targeted attacks that used the same

characteristics. At this point it became clear that the “carrier” code was originally developed as part of an espionage campaign. Some of the use cases we have seen involving this malware, include:

Government:

- Bypass antivirus and sandboxing solutions (breach detection systems)
- Data Exfiltration, eavesdropping on network activities
- Data Exfiltration, key logging
- Data Exfiltration, stealing user identities
- Screen Capturing
- IP Theft

Cyber Crime:

- Bypass antivirus and sandboxing solutions (breach detection systems)
- Money extortion via hard drive encryption (ransomware)
- Online banking fraud

Malware distribution:

- Bypass antivirus and sandboxing solutions (breach detection systems)
- Install rootkits and trojans
- Create botnets and zombie networks
- Critical infrastructure targeting

The following technical details explain how this type of malware is able to remain invisible to most, if not all, common-day security measures.

Technical Findings

- Gyges malware targets Microsoft Windows 7 and 8 platforms and is designed for both the x86 and x64 CPU architectures
- The malware is packed with heavily modified Yoda protector, which provides polymorphic encryption and anti-debugging. It exhibits similarities to Russian espionage malware discovered earlier this year and shares the same crypto engine (first spotted in March).
- The Gyges malware uses API redirection in order to prevent import table rebuilding. The malware API code is redirected to an allocated memory region.
- It rebuilds the import address table (IAT) when the decryptor finishes its task.
- Anti-debugging uses the `NtQueryInformationProcess` Native API with `DebugPort` parameter.
- Anti-debugging, using the `NtSetInformationThread` with `ThreadInformationClass` to `0x11` (`ThreadHideFromDebugger`), the thread will be detached from the debugger.

The Case of Gyges, the Invisible Malware

Dynamic Technical Findings, Hooking Bypass

- The malware verifies that the Windows boot was successful and that the OS is not in safe mode by querying the `GetSystemMetrics(SM_CLEANBOOT)`. If the function returns false, it will turn off Windows by calling `ExitWindowsEx(EWX_REBOOT, 0)`.
- The Malware calls directly to low-level native system API's in order to bypass instrumentation tools or security software that is monitoring higher-level APIs such as the Win32 API libraries. Native system API functions might be changed between service packs and Windows update therefore it is not recommended to use them for legitimate application usage.
- As mentioned above the malware excessively uses the Native Windows API. Since Gyges was compiled as a 32bit application running inside the Windows 64bit OS, it gets loaded inside the Windows-on-Windows (WoW64) subsystem during the switch from 32bit compatibility mode to 64bit mode referred to as "Heaven's Gate." The malware threads executing inside the WoW64 (emulation) environment can execute a FAR CALL instruction. When executing FAR CALL instruction, the processor can perform several calls including one that allows them to gain a higher privilege level.
- The malware is a 32bit executable, but has 64bit payload. The 32bit loader is heavily packed and encrypted using mutated Yoda packer. The payload calls to the 64bit Native Windows API function to bypass usermod hooks. The hooking bypass trick is simple, each translation from 32bit to 64bit will not be detected by hooking what is injected to the Windows-on-Windows subsystem. The FAR CALL instruction translates the malware operations to Native 64bit code, hiding the malicious activity.

Protector

The job of the protector is to obfuscate and hide malicious behavior. It achieves this by converting the original application into sections, which are then extracted by the protector only when the application is running. Sophisticated commercial protectors use virtual machine code that is difficult to decode and analyze. Cybercriminals refer to these protectors as FUD (Fully Undetected). A Fully Undetected Protector service makes malware invisible to anti-virus software.

Three Stages of Infection

1 The malware creates a separate 32-bit process and decrypts itself. Then it switches to the 64-bit segment, executing its payload using the FAR CALL instruction (The file stage, `stage1_upk.i64`, function `InjectHookAndShell`). The payload is pure 64-bit calling only Native System APIs. In this stage, the malware launches its anti-debugging magic using 'PAGE_GUARD' method, allocating memory region and passing it as 'PC_CLIENT' parameter to `NtOpenProcess` function. If a debugger is attached, the call to `NtOpenProcess` will succeed, and the malware will call `ZwTermintaeProcess` function and then exit. Next the malware will open `Explorer.exe` with `NtOpenProcess` and search for the `ntdll.dll` from the PEB (FS:[0x30]).

The initial goal is to copy `explorer.exe`'s `ntdll.dll` to itself, and lastly it patches the end of the second section of `ntdll.dll` with:

```
mov rax, AddressOfShellCode
sub qword [rsp], 5
jmp rax
```

Patching ntdll.dll

It also includes `ZwClose` function with the 'CALL' instruction. The malware's next step is to unmap the original `explorer.exe`'s `ntdll` using `ntdll!NtUnmapViewOfSection` and maps the patched version of `ntdll.dll` at the same address using `ntdll!NtMapViewOfSection` and terminates itself using `ntdll!NtTerminateProcess`.

The Case of Gyges, the Invisible Malware

Three Stages of Infection

2 The malware executes inside Explorer.exe context. It gets control when the patched ZwClose function is called and performs a few indirect jumps/calls mainly to confuse the reverse engineering process and finally restores the original ZwClose saved in the previews stage (Stage 1). The malware then waits for inactivity timeout (we observed a 2 hour wait but it changes from execution to execution). Then it creates desktop name “MyDesktop” using the function CreateDesktopW, performs desktop switching using the function SwitchDesktop and creates new svchost.exe process in suspend mode “%windir%\syswow4\svchost.exe.”

32-bit:

[illegible]

64-bit:

[illegible]

3 The malware executes inside newly-created 32-bit svchost.exe process. This is the most complicated and advanced stage that contains the main payload. The malware monitors system configuration by creating a “watchdog” thread that monitors TaskManager (taskmgr.exe) and terminates it, then switches desktops and performs the same monitoring routine.

- As a measure of true persistence, the malware adds it self to the registry HKCU\Software\Microsoft\Windows\NT\Current\Version\Winlogon, “shell”, value (explorer.exe, C:\Users\M\AppData\Roaming\skype.dat).
- To set up data exfiltration, the malware creates a full screen window with the WebBrowser COM Control, using a WM_CREATE message that it acquires from a web camera picture. To do this, it creates a “WEBCAM” window using capCreateCaptureWindowA and sends a message to it using SendMessageA. The picture is stored in %TMP% directory and then sending data via SSL connection to C&C server located in Russia, which is part of network segment 5.149.208.0.

The Case of Gyges, the Invisible Malware

Conclusion

The Gyges variant not only demonstrates the growing sophistication of malware, but more importantly shows how the lines are blurring between government-grade and mainstream attack code. The fact that “carrier” code can be “bolted on” to any type of malware to carry out invisible attacks is another indication that current approaches to security have reached their end-of-life for detecting advanced threats.

By continuously monitoring activity on the endpoint and on the target device otherwise “invisible” malware cannot hide or evade detection. The “carrier” can no longer evade or bypass a door or a gate on the way to the target device – if an agent, residing on the same device that the malware is targeting, monitors all code executions for malicious activity, without the use of a virtual device or emulation tactics.

We have entered a new era. In addition to antivirus, even advanced protection measures including network monitoring, breach detection systems and sandboxing have become less effective at preventing and detecting advanced threats like Gyges before they can cause extensive damage.

SentinelOne is focused on reinventing endpoint security to protect organizations from advanced threats and nation state malware. The company was formed by an elite team of cyber security and defense experts from Intel, McAfee, Checkpoint, IBM, and the Israel Defense Forces.

2513 E. Charleston Rd
Mountain View, CA 94043

SentinelOne
<http://www.sentinelone.com>

