Bobbing and Weaving to Avoid Prying Eyes

securityintelligence.com/gootkit-bobbing-and-weaving-to-avoid-prying-eyes/

July 8, 2016

Banking & Finance July 8, 2016

By Limor Kessem 7 min read

GootKit research was performed by Tomer Agayev and Gadi Ostrovsky.

Discovered in the wild in the summer of 2014, GootKit is believed to be a privately held cybercrime tool that is not sold to other criminals in underground forums and is operated by a closed gang. Considering its stealth, data theft and browser manipulation capabilities, GootKit is one of the most advanced banking Trojans active nowadays. It is used in online banking fraud attacks that target consumer and business bank accounts primarily located in Europe.

In online banking fraud attacks witnessed throughout 2016, GootKit's masters leverage this malware's capabilities to infiltrate the endpoints of retail and business banking customers, steal their personal authentication credentials and manipulate their online banking sessions with social engineering. They eventually take over those accounts and transfer cash to mule accounts they control.

Beyond its overall modus operandi, GootKit is a malware project that implements stealth and persistency alongside real-time, web-based activities like dynamic webinjections, which modify the banking website as rendered in the infected machine's browser. Since it is operated by one gang, GootKit is believed to have its own in-house developers focused on evolving its stealth mechanisms, security evasion techniques and fraud capabilities.

IBM X-Force Research analysis of the GootKit Trojan showed some interesting changes were made to the code in June 2016, including:

GootKit's lighter video-grabbing module;

Enhanced VM detection capability; and

Installation flow modifications designed for continued evasion.

Lightening the Video Grabber's Load

X-Force researchers who analyzed GootKit believe that in an effort to shrink the size of videos captured on infected victim machines, GootKit's developers made some cosmetic changes to their video-grabbing module.

Up until its recent variants, GootKit recorded videos of the victim's desktop and exfiltrated them to its command-and-control (C&C) server in MP4 files. These files are typically hefty in size.

Related: GootKit Launches Redirection Attacks in the UK
A new format enforced by GootKit's developer is .ivf files, which are encoded by using the Indeo codec from Ligos Corporation. This is a peculiar move on GootKit's part because .ivf files are quite an old format. The codec was popular in the 1990s since it was the first to allow full-speed video playback without using hardware acceleration. This standard was

superseded by newer formats and became rare. It’s possible that .ivf files were selected for smaller size, and also for being a format that most modern-day security software may not inspect or block upon exfiltration to the criminal’s servers.

Once GootKit has a video in this format, it compresses the file to .lzma for a lighter file that can preserve the video’s quality. LZMA files are similar to other compression algorithms, such as ZIP, that compress data to save disk space. However, LZMA compression is known to provide faster decompression times than other algorithms. These types of files are mostly used in Unix-based operating systems.

GootKit’s Strong Suit: Security Evasion

It appears that keeping itself as elusive as possible is a top concern for GootKit. One of GootKit’s most advanced malware capabilities lies in its security and research tool evasion techniques.

It is evident to IBM X-Force researchers that GootKit’s developers have built and continue to build high walls around the malware to keep it away from researchers’ prying eyes. In recent GootKit builds, we can clearly observe these capabilities.

Serving Up VM Detection

Phase One: VM Checks by GootKit’s Dropper

The first check for virtual machine (VM) values takes place at the dropper phase before GootKit’s payload is deployed. The dropper verifies the system’s processor value inside the Windows Registry, searching for a specific name used in servers.

The dropper further checks for VM resources on disk and for additional specific values in the Registry. For example, it checks the target device’s basic input/output system (BIOS) to find values that may indicate a virtual machine client installed on it, including:

> AMI;
>
> BOCHS;
>
> VBOX;
>
> QEMU;
>
> SMCI;
>
> INTEL – 6040000;

- FTNT-1; and
- SONI.

The next check goes over the machine’s MAC address, looking for a list of values that usually appear when the endpoint is a virtual machine.

If any of the suspect values are found, GootKit raises a VM flag and quits its activity. The info is sent to the C&C to allow the botmaster to take action on the issue — like blacklist the endpoint.

At this point, if all checks are clear, the dropper unpacks the GootKit payload and deploys its modules on the infected endpoint.

Phase Two: VM Checks by the GootKit Trojan

After the GootKit payload is deployed, some VM checks are repeated and others are added. For example, the malware scans for telling BIOS Registry values. It then checks for a white list of names acceptable for the central processing unit (CPU); any other name would raise a VM flag. This check is performed because VMs typically attribute a specific name to the CPU that would differ from names used in physical endpoints.

This verification of CPU-related information is not unique. In mid-2015, the Dyre malware checked for processor cores on the target devices before fully installing, aiming to detect VMs that typically only run on one core to save resources. That sets VMs apart from most user machines that would run on at least two cores.

In the next check, GootKit scans for IDE/SCSI hard drives on target machines. GootKit’s developers are looking to scan for VM values (specifically VMWare, VBOX and SONI) that are more deeply embedded in the target system and cannot easily be modified by crafty researchers who would set up a VM to appear identical to a potential victim’s device. For GootKit, this means reliability: It can better detect whether the device it is being run on is a physical machine or a VM iteration.

Take the Road Less Traveled to Look Less Familiar

To ensure its successful deployment to as many machines as possible, the malware is constantly changing things up, keeping static and signature-based security software guessing.

The twists implemented into GootKit’s most recent build include changes to the file type and the process into which the malware is injected. For example:

- The malware loader is injected to SVCHOST.EXE.

- Malware is written in the form of a DLL file to:

LUA Rights: %APPDATA%\Microsoft\Internet Explorer\
ADMIN Rights: %WINDIR%\System32

Most modern-day banking malware — GootKit included — are executable files that get deployed on the infected machines by a dropper. But recent changes to GootKit modified the essence of deployment. Instead of executing a .exe file, GootKit only loads a DLL into a poisoned process of its choice.

This is where the next change comes into effect. Instead of the common method — used by almost all banking malware — of injecting its malicious code into the explorer.exe process, GootKit injects into a service host (svchost) process.

While both processes are commonly run by the system at all times, and both are able to be injected into, perhaps loading GootKit's DLL through a process that runs multiple different instances at the same time can be confusing to detect. Explorer.exe, in contrast, is a process that only runs as one instance at a time.

Switching Up the Persistence Mechanism

Another refreshed function for this advanced malware is the persistence mechanism. All malware aims to survive reboots and attempted deletion. In the past, GootKit used to persist by writing a user shell registry key that executed when the user logged in to the machine. This has been switched up to resemble a method previously used by Dyre: setting up the malware as a scheduled task of the operating system.

In its new build, GootKit presents two persistence options.

First, when deploying the malware with least-privilege user account (LUA) rights, GootKit uses a scheduled task written under a randomized name. The task is triggered to run every minute, acting as a watchdog. It also runs after every boot to ensure GootKit's presence on the endpoint after virus scans or system updates.

In cases where GootKit is deployed with Windows admin rights, it can write its payload as a Windows service. The biggest advantage to running an application as a service is that it will start running before a user logs on and continue running even after the current user logs off. Thus, in the case of malware like GootKit, the Trojan is more persistent overall.

The service’s name here is randomized, allowing GootKit to appear vague enough to a user opening the running processes list and benign to any security software that may be running a scan.

The malicious file’s service properties window appears below:

Conclusion

GootKit is a constantly evolving malware. Its developers have been upgrading its evasion and persistence mechanisms often and with sophistication to maintain this Trojan’s deployment rates and stealth. The malware’s dropper is typically delivered into user endpoints via well-known exploit kits such as Neutrino and Angler, making it relatively effective in compromising users who may not have fully patched their operating systems.

In terms of its targets, GootKit configuration file analysis reveals that it mostly targets French and U.K. banks. Some of this malware’s configurations also target Italian and Spanish banks, according to IBM X-Force Research. X-Force data indicated the malware was further detected in infections on Japanese endpoints. However, those seemed opportunistic rather than deliberate since Japanese banks did not appear in GootKit configurations and the gang continues to focus on Europe.

Overall, GootKit is known to be deployed in limited regions. As a result, it only accounts for 4 percent of the global attack volume by financial malware.

X-Force Research expects to see GootKit continue to evolve. IBM Security can help banks and targeted organizations learn more about this high-risk threat.

To help stop threats such as GootKit, banks and service providers can use adaptive malware detection solutions. They can also protect customer endpoints with malware intelligence that provides real-time insight into fraudster techniques and capabilities, designed to address the relentless evolution of the threat landscape.

Users looking to prevent advanced malware infections on their endpoints must keep their operating system up to date at all times, update frequently used programs and delete those they no longer use. The best browsing hygiene for the prevention of Trojan infection includes disabling ads and avoiding susceptible sites typically used as infection hubs.

Since GootKit and banking malware like it are usually delivered as email attachments, it is critically important to never click on links or attachments in unsolicited email.

Relevant Sample MD5

ebe4babd813271e3c93906c4244d2140;

15e43c836e25f91990ccad5779f21960;

ca18863391c96a6f1afec35a108f4257;

59d0ae7370cdff2c672a09c4aeba39ca;

e7689bf5b51c34a8b03242da4d50bf4c;

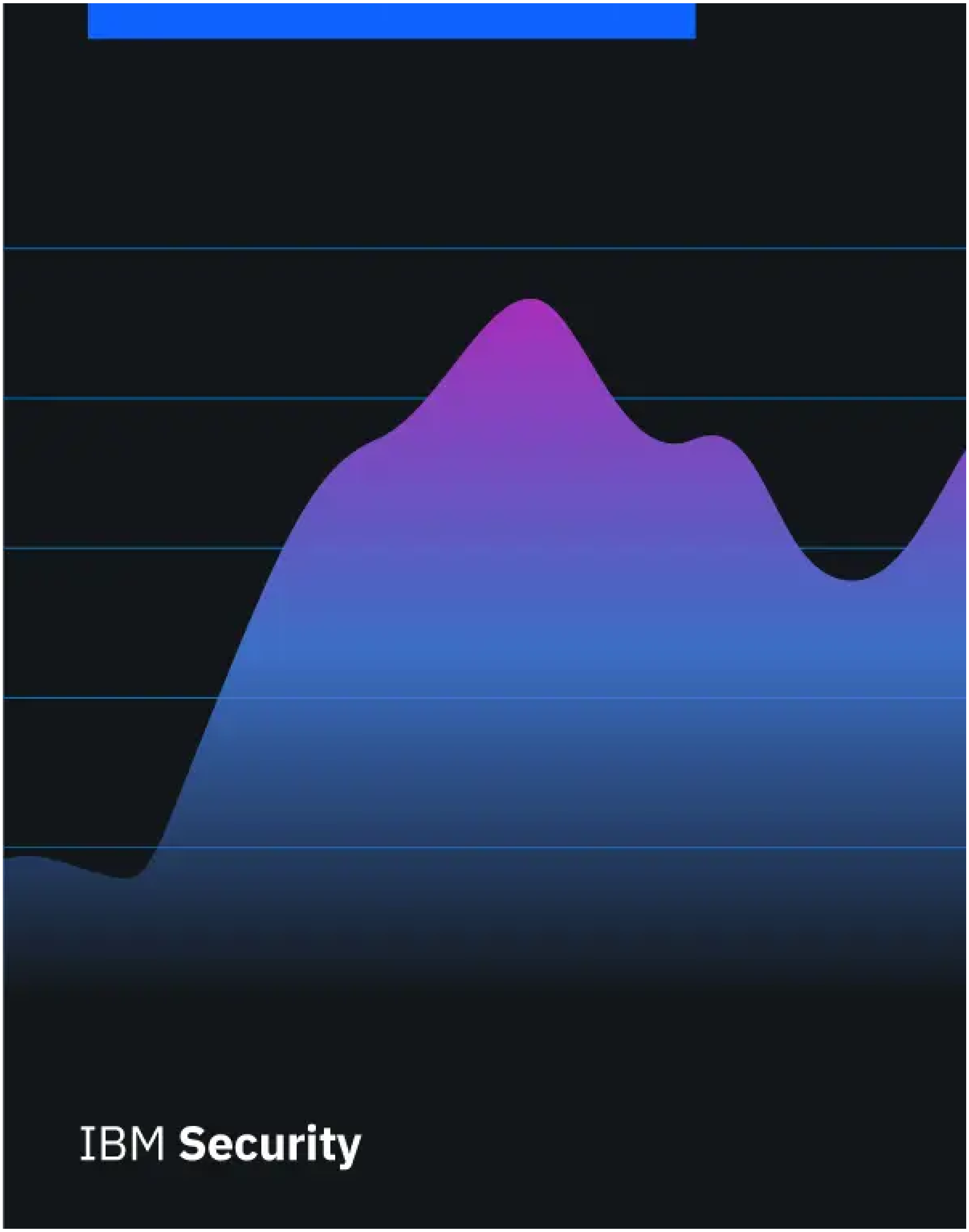
f7125a3c9e780a460644bbbe181786d5; and

bbab0810bfb2d1aa14d06d41fbd1f08e.

IBM X-Force Research will be updating information and indicators of compromise on GootKit via the X-Force Exchange platform. Join XFE today to keep up to date about this threat and other findings from our cybercrime labs.

Limor Kessem
Executive Security Advisor, IBM

Limor Kessem is an Executive Security Advisor at IBM Security. She is a widely sought-after security expert, speaker and author and a strong advocate for wom...


IBM Security