Threat Actor TA505 Targets Financial Enterprises Using LOLBins and a New Backdoor Malware

cybereason.com/blog/threat-actor-ta505-targets-financial-enterprises-using-lolbins-and-a-new-backdoor-malware

Written By
Cybereason Nocturnus

April 25, 2019 | 11 minute read

Research By: Eli Salem

The cybersecurity community has long known that any information technology tool that is used for legitimate purposes can also be manipulated by attackers to enhance their malware. Recently, however, many native Windows OS processes are being used for malicious purposes as well.

In this research, we introduce a meticulously planned, malicious operation against a financial institution in April of 2019. This advanced operation combines a targeted phishing attack with advanced tools that gather intel on the environment. The operation chooses whether or not to create persistence and installs a sophisticated backdoor called ServHelper used to take over the network.

Key Aspects of TA505's Operation

- Highly targeted phishing campaign to a small number of specific accounts within the company.
- Signed and verified malicious code. This is an extra precaution taken to avoid detection.
- A deliberate timeline, indicated by the timing of the phishing attack and signing of the malicious code.
- A selective persistence mechanism and self destruct commands based on autonomous reconnaissance.
- Large emphasis on removal of evidence using self destruct commands and deleting scripts.
- Multiple C2 domains, in the event of blacklisting or inability to connect for another reason.

- The operation integrates four different LOLBins, which indicates the attackers continued, advanced attempts to avoid detection.

The attack was carried out by TA505, a threat actor that is behind infamous campaigns like the infostealer malware Dridex, the Locky ransomware, and more. More recently, TA505 carries out targeted attacks on multiple continents, including North America, Asia, Africa, and South America. Primarily focusing on large financial institutions, this group carries out well-planned, advanced attacks in order to extract valuable data it can later leverage.

As 2019 begins, it is clear that the most widely-used and effective attack vector for malware is still email attacks. The initial phishing attack focused on a number of accounts in a specific financial institution at a single time and date. This enterprise was explicitly targeted with a small number of emails to a very small number of accounts within the company. This hints at the possibility of reconnaissance done at an earlier stage of the operation in order to select the best targets.

The malware uses a signed and verified certification from Sectigo RSA Code Signing CA to spread. This is an extra precaution taken to avoid detection. It gives the malware the appearance of legitimacy when dealing with various defense mechanisms. Furthermore, the malware was signed mere hours prior to the attack- an indication of the operation's deliberate timeline and nature.

Particularly interesting and unusual is the selective persistence mechanism used by some of the tools in this operation. Whereas most malware will attempt to gain persistence whenever possible, the tools being used in this operation create persistence based on their environment. The tools will collect information about the target machine and send this information to a remote C2 server, which will decide whether to set up persistence. This is a technique commonly used in advanced operations.

The attackers clearly placed a large emphasis on covering their tracks. The malware gathered information specifically to decide whether to remove evidence from an infected computer. This demonstrates the level of care taken in this operation, and indicates at the threat actor's potential goal- to plant a backdoor that would remain undetected for as long as possible in order to more effectively exfiltrate data.

The malware makes extensive and varied use of LOLbins and legitimate, native Windows OS processes to perform malicious activities, including the delivery of the payload and the implementation of the ServHelper backdoor. The ServHelper backdoor used in this operation is a relatively new malware family that was discovered at the end of 2018. This continues the trend Cybereason researchers have seen over the past several months towards the wider adoption of LOLbins for attacks.

Cybereason detected the new ServHelper backdoor and analyzed the campaign in order to identify the techniques and tools being used.

A Breakdown of the ServHelper Backdoor Spear Phishing Campaign

A breakdown of the attack from the Cybereason Platform.

Phase One: Gaining Access

This attack begins with a spear phishing attack through a targeted email campaign. Over 80 files were sent to 40 email accounts within the organization, within the span of about an hour. The email contains Microsoft Excel attachments with malicious macros. When the file is opened, it loads in Microsoft Excel and urges the user to enable macros.

The malicious .xls files that contains the macros

After the victim clicks the *Enable Content* button, the macro commands are executed and invoke the Windows OS process msiexec.exe. This process is the Windows Installer, a software component and application programming interface of Microsoft Windows used for the installation, maintenance, and removal of software.

The macro commands use msiexec.exe to connect to a remote C2 server and download the first stage payload. A second msiexec.exe process is created to execute the payload and take part in the second stage of the attack chain.

msiexec manipulated to communicate with the C2 server.

After a TCP connection is established with the C2 server, a payload Alg is downloaded to the infected machine. This payload is a dropper for several files the malware uses in the second stage of the attack with msiexec.exe.

The malware being downloaded from the C2 server.

The full attack tree of the first phase is visible in the Cybereason Platform. This includes all initial activity, including the attempt of the Windows Installer (msiexec.exe) to communicate with the C2 server.

The infiltration of the malware as seen in the Cybereason Platform.

Phase Two: Deploying the Backdoor

After Alg is downloaded, it is loaded as a binary with a .tmp extension to msiexec.exe and begins to execute its sequence of malicious activity. This temporary file acts as the main dropper of the malware and the deployer of the malware across the target machine.

Within the .tmp file are three folders, which contain two modules pegas.dll and nsExec.dll as well as a .nsi script.

According to threat intel, NSIS (Nullsoft Scriptable Install System) is a legitimate tool to create installers for Windows. This indicates the script is used to install something pertinent to the malware. It also allows the attack to evade detection, as it is a legitimate tool.

The content inside the temporary file.

The script contains commands for the .tmp to execute, and instructs the .tmp file to manipulate and execute pegas.dll from a function *kest()*using the nsExec.dll module. This activity is considered legitimate because NSIS scripts are legitimate by their nature.

The NSIS script

After being loaded and executed by msiexec.exe, the temporary file creates several additional files. It creates another temporary file that will be loaded into the memory map of the creator temporary file, but, more importantly, it creates two modules pegas.dll, and nsExec.dll.

The malware creates the file nsExec.dll.

The malware creates the file pegas.dll.

In addition, the temporary file also create a rundll32.exe process using the Windows command line in order to load the dropped module pegas.dll and execute it from a function *kest()*.

The execution command of rundll32.exe that will load pegas.dll.

The rundll32.exe process executes pegas.dll. These events were detected by the Cybereason Defense Platform.

The malware deployment in the Cybereason Platform.

Functionality of Modules

nsExec.dll

The nsExec.dll module was created by Nullsoft, and is related to the NSIS installers mentioned above. nsExec.dll will execute command-line-based programs and capture the output without opening a DOSBox. This gives the attacker the ability to execute the command line and run the rundll32.exe process without appearing on the target machines desktop. This increases the stealthiness of the malware when executing commands from the Windows command line.

pegas.dll

pegas.dll is the main module responsible for executing the full capabilities of the backdoor. This includes the creation of malicious activity in the target machine, including reconnaissance, information stealing, and backdoor capabilities. In addition, this module is also responsible for communicating with another C2 server that determines the next steps for the malware. These steps are executed by the pegas.dll module.

Interestingly, pegas.dll is actually a signed and verified module by certification company Sectigo RSA Code Signing CA.This is very unusual, and is the mark of a sophisticated threat actor. This certificate company is also known for being used in the recent famous Norsk Hydro LockerGoga Ransomware Attack from last month. This attack used their Sectigo certificate to propagate.

The use of this certificate gives the malware an advantage that most modern malware does not have: legitimacy. Malware that is “confirmed” and “verified” will appear harmless, and may lower the guard of security and IT specialists in an investigation. This shows that the threat actors that developed this malware are more advanced than most malware authors.

The abused certificate.

pegas.dll makes use of several defense mechanisms. It is packed twice in order to ensure that it is difficult to reverse-engineer. In addition, the module is compiled merely a few hours before the spear phishing attack occurs, which ensures it is quite new. Despite this, the Cybereason Defense Platform was able to collect all the data associated with the module.

The malware packed twice.

rundll32.exe executes the module pegas.dll from the function *kest()*. This function is one of several functions from the malware export table that is responsible for the initial execution of the malicious code.

The other functions from the export table are *loer()*and *tempora()*.

The three export functions.

These functions share similar functionality in terms of code and functions. All the variables *kest()* contains also appear in *tempora()*. In addition, *kest()*and *loer()* share function *FUN_12345f08()*, which is essentially the only functionality *loer()* has.

The export functions pseudo-code in Ghidra.

FUN_12345f08() is one of the most important functions for the malware. The malware authors execute pegas.dll with *kest()* as soon as possible to make use of *FUN_12345f08()*. Within *FUN_12345f08()* there is a new indicator of compromise, a domain joisff333.icu that acts as the second C2 for the malware. This domain is reached from the rundll32.exe process. In addition, a string enu.ps1 appears, which indicates that this malware will use PowerShell. Lastly, a string asfasga33fafafaaf is also visible, which appears to be related to the creation of the mutex BaseNamedObjects\Global\asfasga33fafafaaf.

FUN_13246F08 pseudo-code in Ghidra.

Using static investigation, additional indicative strings become visible that show some of the malware capabilities including network ability, C2 commands, additional domains, and PowerShell execution activity.

Additional indicative strings from the unpacked malware.

Reconnaissance and Information Collection

After the execution of rundll32.exe, the PowerShell script enu.ps1 is executed. This script is encoded with Base64 in order to avoid detection by antivirus products.

enu.ps1 obfuscated using Base64.

Upon decoding the script, it is clear that the script is responsible for gathering reconnaissance on the target machine. This includes collecting information with WMI queries to identify if the user is an administrator.

Internal reconnaissance by WMI queries.

The ServHelper backdoor gathers additional intel on the target machine including the users SID*.* An SID of *S-1-5-32-544* is an identifier for the built-in Administrators group. This includes the local administrator and all local and domain administrators user groups. Gathering this information indicates two things: the malware authors are targeting organizations as opposed to regular home computer users, and within the organization they are targeting the highest priority user machines.

The malware searching for administrators users.

Once the malware confirms the target machine is an administrator group user, it collects information from the method WindowsIdentity *GetCurrent()*.

The malware continuing to search for administrators users and collecting information on them.

After the malware is able to verify this user is an administrator, it collects additional information about the target machine and retrieves data on all file system drives, including the virtual drives

The malware collecting data about the system drives.

The last bit of information the script attempts to aggregate is the name of the server and the names of the local groups on the computer. It uses net.exe, a legitimate Windows OS process, to collect this information.

Local group administrators reconnaissance.

The complete process tree of the deployment of the malware shows four different LOLBins that took part in the attack cycle. Between the extensive use of LOLBins and the use of a signed and verified module pegas.dll, it is clear the malware authors went to great lengths to evade detection.

Full process tree by process hacker.

Phase Three: The Dynamic C2 Server

After deployment, the malware connects to a second C2 domain from rundll32.exe. The malware makes use of multiple C2 domains to ensure there is at least one C2 server available to attack from. It communicates with pegas.dll to decide the malware execution's next steps.

The second C2 server responds dependent on the information gathered on the target machine. This is an indication of the level of sophistication of the malware. Whereas most malware collects and sends data to a fixed C2 server, which responds the same regardless of the information, this takes on a more dynamic approach.

The malware communicates with its second C2 server.

Interaction with the Second C2 Server

In the process of communicating with the C2 domain, there are several key commands and activities that the malware performs. Some of these activities are known from previous variants seen in the wild.

Abusing Certifications

The following network activity observed from msiexec.exe illustrates how the malware leveraged a signed and verified certification from Sectigo RSA Code Signing CA to propagate.

Abusing certification traffic.

Executing Reconnaissance

The shell command is responsible for executing the net user /domain command on the target machine. This command is a remote control command that allows the attacker to execute additional reconnaissance activities.

Executing the net user /domain command.

The Persistence Mechanism

The persist command is responsible for the malware persistence mechanism on the target machine. The C2 server decides whether to create persistence. This is another major indication of the sophistication of the malware, as the malware does not create persistence on every target machine, but only does so on certain computers.

The C2 server informs the malware to create persistence.

After the persist command is sent by the C2 server, the malware creates persistence using the registry. It creates a registry key Intel Protect under Run and store pegas.dll as the value.

Persistent mechanism.

Internal Reconnaissance

The malware changes its behavior based on the information it gathers from the infected machine. This information is stored in the *HTTP Payload, /URL: /jquery/jquery.php* packet that the infected machine delivers to the C2 domain.

The information is divided into four hardcoded parameters:

- *Key*: the hardcoded parameter that stays the same in every iteration of the malware.
- *Sysid*: the information about the operating system, service pack, and computer name.
- *Resp*: the information about whether the user has administrator group privileges in the active directory.
- *Misc*: additional information regarding the PowerShell file. This is the newest parameter, and was observed only in this version of ServHelper. This includes additional reconnaissance to help decide persistence.

Four hardcoded strings that will be used to deliver information to the C2 server.

The behavioral change in the malware's response to the C2 domain is dependent on whether the infected machine is a high priority target for the malware authors. When the machine is not a valuable target, the *Resp* parameter will fill with, "the specified domain either does not exist or could not be contacted".

The malware informs the C2 server about the findings (No AD admin).

If the malware identifies that the target machine is valuable, it will fill the *Resp* parameter with, "*User accounts for testing.com (The domain controller)*" with the names the groups and accounts the machine has, such as, "Administrator", along with the name of the user.

The malware informs the C2 server about the findings (With AD admin).

Similar to many kinds of backdoor malware, ServHelper is able to lurk quietly in a sleep state. This is a known backdoor characteristic that lets the malware stay under the radar and strike at the exact time the attacker wants to activate additional malicious activities. ServHelper gets information about the target machine date and time using the function GetSystemTime().

The malware staying under the radar using the sleep function.

The malware contains several other commands known to be associated with ServHelper that appear in the strings mentioned above. This includes:

- *loaddl*: a command responsible for downloading and executing additional modules using the rundll32.exe process.
- *selfkill:* a command that is responsible for self-terminating and deleting the malware from the machine.

The events that indicate the attempts to connect to the C2 server are identified by the Cybereason Defense Platform.

The data exfiltration attempt within the Cybereason Platform.

Conclusion

Through the evaluation of this malware, we discovered an evasive infection technique used to deploy the newest variant of the ServHelper Backdoor. We were able to detect and evaluate a targeted spear phishing attack conducted by the malware authors. This signals a continuation of an existing trend towards phishing attacks as an initial attack vector.

In this discovery, we highlighted the use of legitimate, native Windows OS processes used to perform malicious activities to deliver a payload without being detected, as well as how the ServHelper Backdoor operates and deploys itself without being noticed. We also showed how sophisticated this backdoor is, and how it specifically targets valuable machines and users in order to maximize potential damage.

The analysis of the tools and techniques used in this ServHelper spear phishing attack show how truly effective LOLBins are at evading antivirus products and how malware authors can maximize the use of them. This furthers the trends we have seen since 2018 with regards to LOLBins more common use. This research is an example of widespread LOLbin adoption, and shows that the use of LOLBins will only grow as we will continue through 2019.

The affected customer was able to contain the attack before any damage was done. The ServHelper Backdoor was controlled, msiexec.exe and rundll32.exe were terminated, and all the downloaded files were deleted. Furthermore, the connections to the malicious C2 domains were blocked and the attack was halted in its tracks.

Part of the difficulty in identifying this attack is in how it evades detection. It is difficult to catch, even for security teams aware of the complications in ensuring a secure system, as with our customer. LOLBins are deceptive because their execution seems benign at first, or even sometimes safe. In addition, the use of a signed and verified file with certification increases the likelihood that the malware will stay under the radar.

As the use of LOLBins becomes more commonplace, we suspect this complex method of attack will become more widely used as well. The potential for damage will grow as attackers look to other, more destructive payloads.

Last month we discovered a new variant of the prolific Ursnif trojan.
Check out our live webinar on the discovery.

Indicators of Compromise

SHA1 880b383532534e32f3fa49692d676d9488aabac1	Hash	Work report 011042019.xls
SHA1 63aeb16b5d001cbd94b636e9f557fe97b8467c8d	Hash	Alg
SHA1 ad35fa0b3799562931b4bfa3abd057214b8721ff	Hash	msie988.tmp
SHA1 06f232210e507f09f01155e7d0cb5389b8a31042	Hash	pegas.dll
79.141.171[.]160	IP	First C2
aasdkkkdsa3442[.]icu	Domain	Second C2
joisff333[.]icu	Domain	Second C2
zxskjkkjsk3232[.]pw	Domain	Second C2

About the Author

Cybereason Nocturnus

The Cybereason Nocturnus Team has brought the world’s brightest minds from the military, government intelligence, and enterprise security to uncover emerging threats across the globe. They specialize in analyzing new attack methodologies, reverse-engineering malware, and exposing unknown system vulnerabilities. The Cybereason Nocturnus Team was the first to release a vaccination for the 2017 NotPetya and Bad Rabbit cyberattacks.

All Posts by Cybereason Nocturnus