MineBridge RAT Is on the Rise, With a Sophisticated Delivery Mechanism

blog.morphisec.com/minebridge-on-the-rise-sophisticated-delivery-mechanism

Posted by Alon Groisman on March 9, 2021
Find me on:
LinkedIn

- Tweet
-

The MineBridge RAT was first identified in January 2020 by security researchers at FireEye, who observed the backdoor attacking financial institutions in the United States with some targets located in South Korea as well. MineBridge was initially classified as a C++ backdoor that was delivered via phishing campaigns.

The updated RAT began using a macro embedded in Word document in late February. According to attribution models from several other cybersecurity firms, it appears to be the work of TA505, who have also used a Get2 loader and Clop/Cryptomix.

This blog post will cover a new delivery chain for the *MineBridge RAT*, including technical details and binary indicators.

Technical Introduction:

Morphisec Labs recently identified an updated delivery chain for the latest MineBridge RAT delivery.

Though we haven’t currently established the entry point of how the first scheduled task was created, we assume that as before it follows the impersonation of Google-related application updates. Both scheduled tasks within the attack chain have Google in their names and previously-known delivery methods of MineBridge also involved Google updates.

c

MineBridge Attack Flow

Below are short descriptions of the attack stages:

- The total execution length of the attack was about two hours from the first task executed until the final stage of connection to the C2
- At first, a scheduled task executed a very short encoded PowerShell that executes a remote PowerShell obfuscated command. The URL is hidden behind a cutt.ly URL shortener service.
- The remote obfuscated PowerShell command downloads a NetSupport client from a freshly registered Bitbucket storage through the same cutt.ly service.
- After downloading all the required legitimate NetSupport executables, the same obfuscated remote PowerShell continues to download a custom client configuration file that points to an intermediate C2. This configuration is downloaded from a different domain also masked by cutt.ly.

- Next, the NetSupport executes and immediately connects to the intermediate C2. Following the connection, a PowerShell command is executed and persisted into a new scheduled task.
- The new PowerShell encoded command again executes a remote obfuscated Powershell like a previous stage.
- The remote obfuscated PowerShell downloads an old TeamViewer from a different Bitbucket repository that was also just created for this attack. Like the previous stage, the repository stores legitimate artifacts of the TeamViewer while the malicious “msi.dll” is downloaded from a different malicious domain. If the msi.dll is removed by AV, it will be re-downloaded when the next task executes.
- Msi.dll sideloading is not a new technique, it has been long used by MineBridge RAT and some others to execute malicious code within the context of TeamViewer. This time the DLL mimics a Themida packer though the encryption is done by VMprotect.

First PowerShell

First PowerShell

The first stage of PowerShell execution uses a standard Invoke-WebRequest to download the next stage PowerShell from a cutt.ly *URL Link shortener* service (hxxps://simplename[.]website/upd/?t=psns). Using a URL shortener as well as using public content sharing services has become very popular among the different actors as it's almost impossible to block based on reputation.

“Google Photo Sync” is the registered scheduled task that is responsible for the execution of this **encoded** PowerShell command;

encoded PowerShell command

Second PowerShell

The PowerShell in the next stage is slightly obfuscated. The obfuscation is done by separating string characters and assigning them to long random parameter names, later concatenating them at runtime. The long variable parameter names confuse some of the string similarities in AI algorithms that search for suspicious obfuscated strings. Many actors also apply this technique to evade detection by AI algorithms.

Second PowerShell

The following script appears when we decode the PowerShell content.

All cutt.ly links lead to a freshly created public Bitbucket project that contains all files necessary to run NetSupport clients. The only file missing is the Client32.ini configuration file that is downloaded from a different domain *simpledomen[.]club/support.php* (request header is validated by the attacker). As soon as all the artifacts are downloaded, a NetSupport backdoor is added to the autoruns for persistence.

The repository has no Commits, Branches, Comments, or any discussion aside from the NetSupport files. The only Commit is for the README.md file that contains a generic Bitbucket content possibly to make it look like a new legitimate project.

commit for the readme file

Client32.ini file:

Before running the NetSupport client, the PowerShell script tests if the file Client32.ini exists. If so, the configuration file is removed and downloaded again. This way the attack chain will not be interrupted by the intermediate NetSupport C2 change or a remediation attempt by the local AV.

Client32.ini contains NetSupport settings. The actor sets NetSupport to be hidden while disabling all app notifications and any pop-ups. At the moment we wrote this post, the NetSupport backdoor was configured to communicate with update-system[.]cn:443 as the main gateway and updatesystem[.]website:443 as the secondary gateway.

main gateway and updatesystem

Third PowerShell:

As soon as the NetSupport client succeeds in connecting to the intermediate C2, it executes a PowerShell command that is similar in format to the first PowerShell command with a slight modification of the cutt.ly link. The cutt.ly link then expands to the same URL with a slight change in the parameter name to hxxps://simplename[.]website/upd/?t=pstv. **Note that the last 2 letters represent the backdoor name acronyms (ns for NetSupport, tv for TeamViewer)**.

NetSupport also adds a persistent scheduled task for the next stage.

The “Google Disk Sync” scheduled task is the encoded representation of the above command:

Fourth PowerShell:

As in the previous stages, the remote obfuscated command is responsible for the download of a new backdoor component: an old TeamViewer that is vulnerable to a side-loading (DLL hijack) vulnerability. As before, all the artifacts needed for the execution of TeamViewer are downloaded from a newly created Bitbucket repository (different from the first one) while the malicious msi.dll component (the sideloaded DLL) is downloaded from the same “*client32”* domain simpledomen[.]club/watchdog.php. The attacker chooses to rename the TeamViewer application name to QTConnect.exe.

minebridge trojan

“client32” domain

While most DLL hijacks are usually executed after the application is loaded, in the case of TeamViewer the msi.dll is directly in the import table of the executable. Therefore it will execute before the application officially starts its execution; this is a very lucrative target for attackers to avoid runtime monitoring. This is also why some vendors decided to flag the vulnerable TeamViewer even though it is signed by a legitimate certificate.

MineBridge rat

In the scope of this blog, we will only touch upon some of the basic characteristics of the delivered RAT. MineBridge Remote-Access Trojan actors frequently change their packing techniques to hide their final payload. In this latest campaign, we identified the impersonation of a Themida packer while masking VMprotect.

String Encryption

After unpacking has completed, we quickly identified the main string decryptors (Unicode and Ascii). We uploaded a basic script that should decrypt most of the strings.

main string decryptor

Our first analysis showed a nonsignificant change in commands, no change in mutex and persistence, different C2 domains, and a slight modification in the communication pattern.

Commands

- runexe_command
- runexe_URL
- rundll_command
- rundll_URL
- update_command
- update_URL
- restart_command4
- terminate_command
- kill_command

- Shutdown_command
- Reboot_command

An example for one of the commands:

Minebridge command

Communication

Minebridge communication

Minebridge

Persistence

As before the persistence is automated through IshellLink functionality, adding “Windows Defender.lnk” to the startup folder.

Windows Defender.lnk

Conclusion

The delivery of MineBridge has evolved since it was first identified in January. This new variant can evade detection solutions through obfuscation and other means, thus allowing it to continue its attack in infected systems. Morphisec customers are secured against

MineBridge’s evasion tactics through the power of our prevention-first, zero-trust runtime endpoint and server protection technologies powered by moving target defense.

IOCs:

NetSupport client and downloaders	simplename[.]website/upd/?t=psns simpledomen[.]club/support.php bitbucket[.]org/Net-Support/pub/downloads/
NetSupport client	update-system[.]cn:443 updatesystem[.]website:443
TeamViewer / MineBridge downloaders	simplename[.]website/upd/?t=pstv simpledomen[.]club/watchdog.php bitbucket[.]org/TVSoft/public/downloads/
MineBridge C2	ninjakick[.]club polarrsearch[.]xyz rogaikopyta[.]xyz utkailipa[.]xyz 5tvstar[.]cn sweepchance[.]xyz sweepchance[.]club goldendragon888[.]cn sub-url: tf346765jh67/indexes_data.php
msi.dll	7c6b9051654cbbcd02985c16fcb95ce1a78c58ba54bdc0605f0c4ee669c67ff9
Mutex	DynGateInstanceMutex
Scheduled tasks	“Google Photo Sync” “Google Disk Sync”

Older domains	billionaireshore[.]top
	vikingsofnorth[.]top
	realityarchitector[.]top
	gentlebouncer[.]top
	brainassault[.]top
	greatersky[.]top
	unicornhub[.]top
	corporatelover[.]top
	bloggersglobbers[.]top
	billionairesho[.]top
	sub-url: munhgy8fw6egydubh/9gh3yrubhdkgfby43.php

Contact SalesInquire via Azure