[Down]loaded by GuLoader Malware | DeepInstinct

By Itay VanzettiThreat Research Team

Published: 2021-12-07 · Archived: 2026-04-05 16:45:57 UTC

GuLoader, also commonly referred to as CloudEyE or vbdropper, was first noticed in the wild around December 2019, and has since been used to distribute malware at scale around the globe. Loader / downloader malware is first-stage malware that is designed to infect a target, and then help execute second-stage malware or malicious payloads. Loaders are typically used to launch Malware-as-a-Service (MaaS) schemes created by cybercriminals to provide paid access to servers and infrastructures to launch distributed malware campaigns. While we observed that GuLoader delivery methods vary, it's most often used via malspam campaigns.

GuLoader malware can be downloaded or created in various ways, but the most common one includes the use of a document attachment with a macro that will execute a malicious payload or an exploit like CVE-2017-11882. GuLoader is unique in that it is written in Visual Basic 6 and contains a Shellcode payload wrapped inside. Yet, like many of today's malware strains, GuLoader also uses a variety of self-defense mechanisms to evade detection and defense.

The First Stage

GuLoader begins with an attachment containing an Office document containing a hidden

OLE embedded object

that is triggered by a macro.

Notice in the image below the tiny black pixel circled in red.

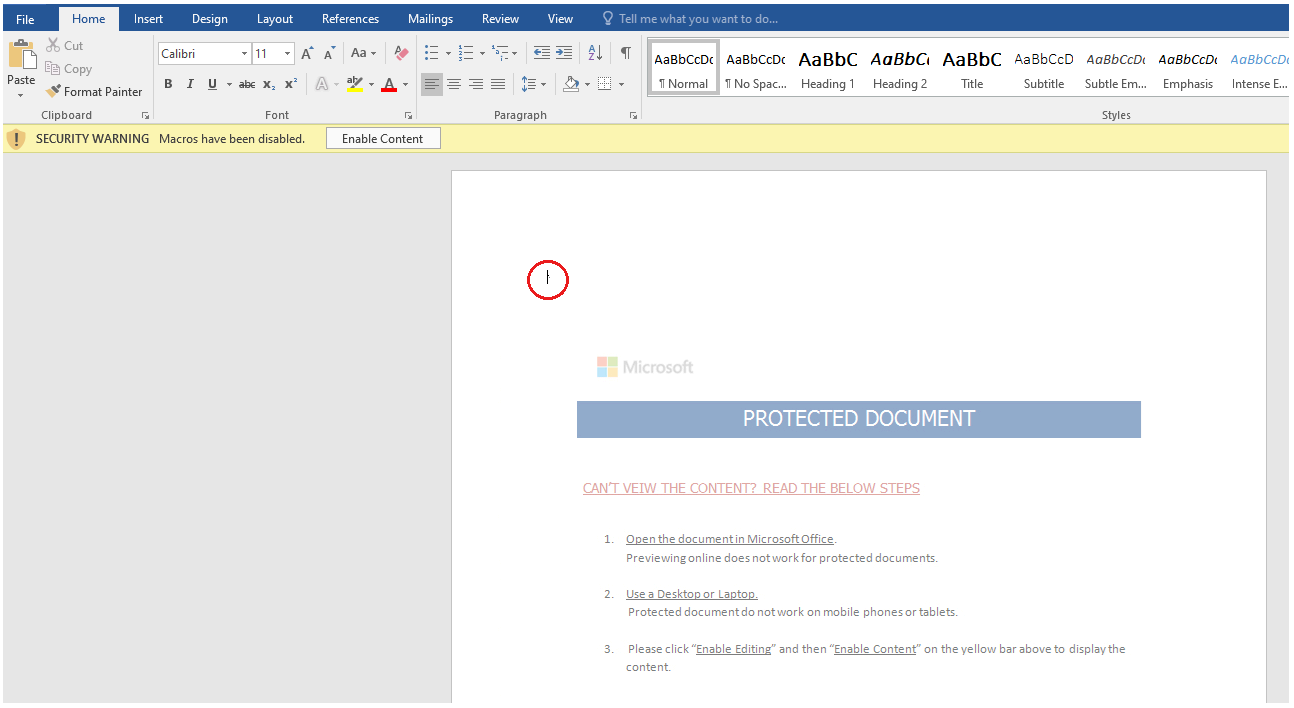

Figure 1: MGuLoader malware

Next, a popup appears asking for permission to run (“WindowsUpdate.exe”); this is the GuLoader malware and this action will enable it to execute.

image.png

Figure 2: A security prompt for GuLoader malware masked as “WindowsUpdate.exe”

After examining the macro, we see that there is an object we likely didn’t notice when we first opened the document. If we reopen the document and go to the small pixel in the upper left corner, we can expand it to the original object size and see the embedded GuLoader malware.

image.png

Figure 3: The VBA macro that triggers the GuLoader malware embedded in the document

image.png

Figure 4: The tiny pixel is now uncovered as a GuLoader executable

The malicious PE executable can also be created by the macros embedded in malicious Office files rather than being embedded inside the Office file, as shown in the example below.

image.png

Figure 5: A VBS code from a script that was created from a different maldoc sample

The script will create the GuLoader malware’s executable file as opposed to embedding it inside the OLE as shown in the previous example.

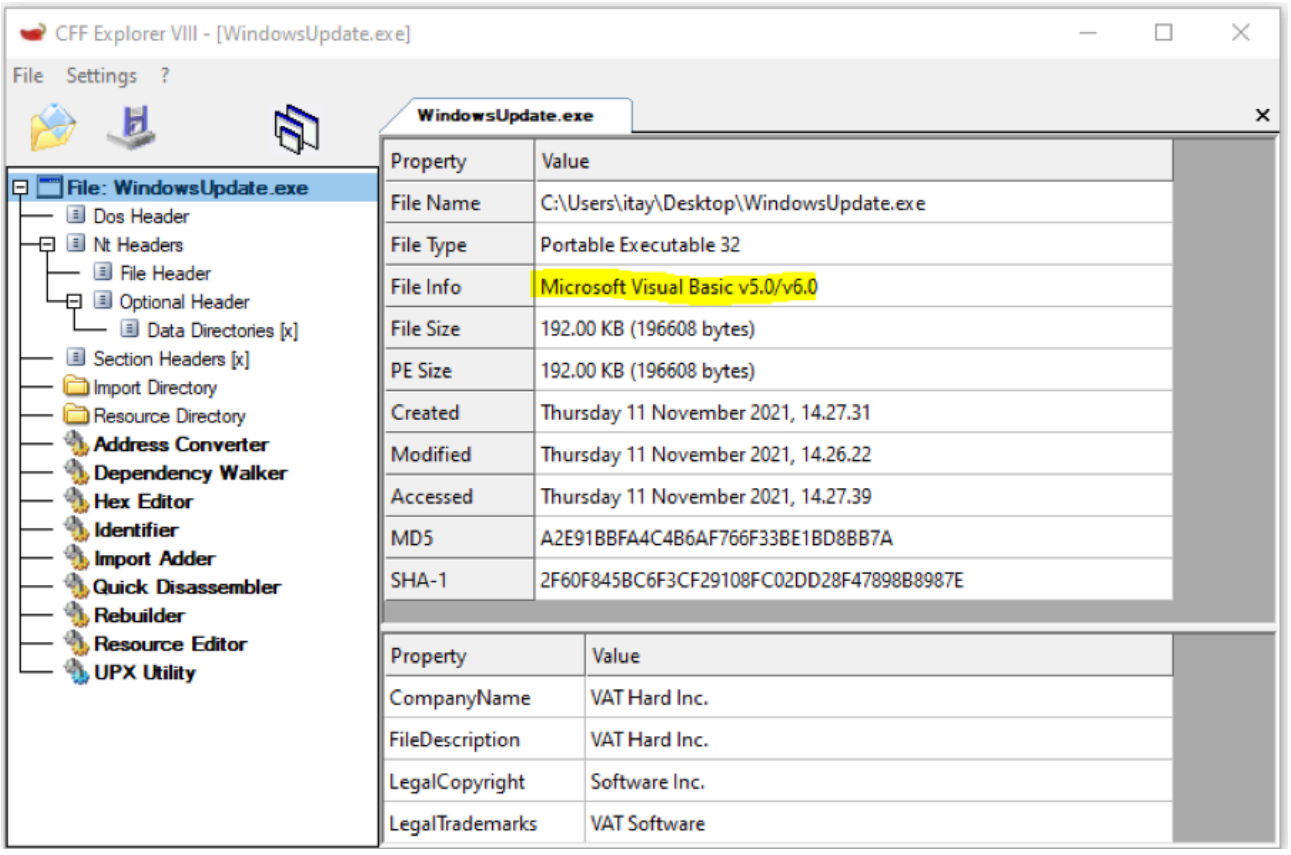

Figure 6: PE header of GuLoader shows that it is written in Visual Basic

The Second Stage

The purpose of GuLoader is to download one or more remote administration tools (RAT), as well as info stealer trojans, including the following:

- Parallax RAT - active since at least 2019
- Remcos RAT - active since at least 2016
- FormBook - active since at least 2016
- Agent Tesla - active since at least 2014
- NanoCore - active since at least 2013
- NETWIRE - active since at least 2012

GuLoader: A Master of Self Defense

GuLoader is notorious for its use of obfuscation and anti-analysis/debugging/sandbox/emulation techniques.

- To bypass security software, GuLoader will search the NTDLL API functions loaded in memory for well-known locations where security products place their hooks. It will then intercept API calls and control the flow detect malicious behavior associated with those API functions. Once found, it will “unhook” those hooks by restoring the original NTDLL’s API functions control flows.
- Use of GetProcAddress. When a program is executed, it determines which Windows API functions calls to use from the Import Address Table (IAT) located in the PE header. This process is called dynamic linking.

Since the imports contain information describing the program's capabilities, it has value for analysis and debugging. For example, if *SetWindowsHookExA()* is imported, this might be an indication that the program is capable of keystroke logging. In order to obfuscate and mask Windows API calls used, GuLoader will import GetProcAddress and use it to resolve any API function's address it wishes to use from user32.dll loaded in memory and achieve the possibility of using it without the need to import it. GuLoader's PE files won't have all the API functions it plans to use declared in IAT, but will include only the generic GetProcAddress() (among others) while covering its true intentions from anti-malware solutions and/or analysts.

- Use of EnumWindows. This sandbox or emulation detection is done by enumerating and counting all top-level windows on the screen. If there is a low number of windows opened, the malware will self-terminate without executing any malicious activity in order to avoid being detected during the analysis process.
- Use of NtSetInformationThread with ThreadHideFromDebugger parameter. This debugging tampering method will cause a crash in the debugged application when a breakpoint is hit in a hidden thread.
- DbgBreakPoint API is called when a debugger attaches to a running process. GuLoader will patch this API by replacing DbgBreakPoint's original assembly instructions with with 0x90 instructions/NOP (no operation). Similarly, to manipulate DbgBreakPoint, it will replace other debugging API calls with dummy calls to cause a crash during the debugging process.
- Adding junk instruction to the shellcode itself, which doesn't have any effect other than delaying the analysis process.

Figure 7: Anti-VM alert message when executing the sample in a VM environment

Upon successful execution, GuLoader will perform a code injection technique called "Process Hollowing" during which malicious code is injected into a legitimate processes address space after it has been put in a suspended state. It then unmaps ("hollows") its memory and injects its malicious code that will then be executed under that process. The purpose of using this technique is to hide the malicious activity among other legitimate processes running in the system, avoiding defenses meant to detect anomalies among the running processes. GuLoader's prime targets for hollowing and injecting the malicious payload typically include the following:

- regAsm.exe - Assembly Registration tool
- MSBuild.exe - Microsoft Build Engine (a platform for building applications)
- RegSvcs.exe – .NET Services Installation tool

The Final Stage

Once GuLoader successfully performs the "process hollowing," it will download and execute the target malware from a remote location. In some occasions, this location is Google Drive or OneDrive to avoid network-based detections as the domains of these vendors are usually classified as benign and trusted.

Figure 8: GuLoader GET request for Remcos RAT

image.png

Figure 9: Remcos RAT downloaded (MZ magic bytes hint of a PE file)

Widespread Reach of GuLoader

GuLoader is very active and is a considerable threat to any environment. In September 2021, a campaign

spreading a Remcos

Remote Access Trojan was identified that utilized GuLoader. Another noticeable campaign that was documented in January 2021 targeted

organizations in Italy

, and spread GuLoader via malspam with a fake RFQ (Request for Quotation) cover story.

image.png

Figure 10: Number of GuLoader samples submitted to MalwareBazaar during January - October 2021. Notice the spikes in January - February and September - October.

Conclusion

A GuLoader infection can have serious consequences on an infected organization, and its effectiveness ensures it will be a significant threat for years to come.

Deep Instinct successfully prevents malware like GuLoader, detecting and stopping it before it can enter a customers' environment. Deep Instinct prevents the malicious payloads regardless of the deployment method exploited. Our deep learning framework allows Deep Instinct to prevent >99% of unknown threats. It is also incredibly precise, ensuring false positives remain <0.1%

If you'd like to learn more about our malware, ransomware, and zero-day prevention capabilities – including our industry best $3M no-ransomware guarantee – we'd be delighted to give you a demo.

IOC's:

Docx: 4797b4b672a93427a126361e058d86c443c4956255624bce2ff2e5d129f88eaa

Vbs: 6394c4e126b8ef4cf8e66d43a54cfd42fd86b3003292f621f0ca427bc12051d8

Pcap taken from: malware-traffic

Read more:

https://www.difesaesicurezza.com/en/defence-and-security/cybercrime-the-guloader-campaign-back-to-italy-via-a-false-quotation/

https://www.gend.co/blog/netskope-cloud-threats-memo-january-28-2021

https://www.apriorit.com/dev-blog/367-anti-reverse-engineering-protection-techniques-to-use-before-releasing-software

---