LokiBot Campaign Targets Microsoft Office Document Using Vulnerabilities and Macros

fortinet.com/blog/threat-research/lokibot-targets-microsoft-office-document-using-vulnerabilities-and-macros

July 12, 2023

Affected platforms: Microsoft Windows
Impacted parties: Windows users
Impact: Control and collect sensitive information from a victim's device
Severity level: Critical

In a recent FortiGuard Labs investigation, we came across several malicious Microsoft Office documents designed to exploit known vulnerabilities. Specifically, CVE-2021-40444 and CVE-2022-30190 are remote code execution vulnerabilities. Exploiting these vulnerabilities allowed the attackers to embed malicious macros within Microsoft documents that, when executed, dropped the LokiBot malware onto the victim's system. LokiBot, also known as Loki PWS, has been a well-known information-stealing Trojan active since 2015. It primarily targets Windows systems and aims to gather sensitive information from infected machines.

In this article, we will delve into the specifics of the identified documents, explore the payload they delivered, and outline the behavioral patterns exhibited by LokiBot. Our analysis aims to shed light on the intricacies of this threat and increase awareness regarding its operational methods.

1st Stage

During May 2023, we obtained two types of Word documents for analysis. The first type featured an external link embedded within an XML file, “word/_rels/document.xml.rels,” while the second type included a VBA script that executed a macro immediately upon opening the document. Notably, both files contained a strikingly similar bait image, depicted in Figure 1.

Figure 1: The lure picture from the Word document

The Word document that targets CVE-2021-40444 contained a file “document.xml.rels”, shown in Figure 2, with an external link using MHTML (MIME encapsulation of aggregate HTML documents). This web archive file format combines a website's HTML code and companion resources into a single file. This link also uses Cuttly, a URL shortener and link management platform, to redirect users to the cloud file-sharing website, “GoFile.” Further analysis revealed that a file named “defrt.html” was downloaded upon accessing the link. This file exploits the second vulnerability, CVE-2022-30190. The content of this file and the decoded data is displayed in Figure 3.

Upon executing the payload, it initiates the download of an injector file named “oehrjd.exe” from the following URL: http[:]//pcwizard[.]net/yz/ftp/. Detailed information regarding the execution file can be found in the subsequent section.

Figure 2: The Document.xml.rels contains a malicious external link in oleObject

Figure 3: Malicious content from “defrt.html” and decoded data

The second document was discovered towards the end of May. Upon analyzing the VBA script embedded within the Word document, as illustrated in Figure 4, the code is automatically executed due to its use of the “Auto_Open” and “Document_Open” functions. Various arrays are decoded within the script and saved to a temporary folder under the name “DD.inf” (Figure 5). It includes a command to create an “ema.tmp” file to store data after line 29 in the “DD.inf” file. The data is then encoded using the “ecodehex” function and saved as “des.jpg”. The script then uses rundll32 to load a DLL file with the function “maintst.” Finally, it deletes all temporary, JPG, and INF files created throughout this process.

Figure 4: The VBA macro from the Word document

Figure 5: The content in “DD.inf”

The Compromised Website

As mentioned, the VBA script creates an INF file to load a DLL. The purpose of this DLL file, named “des.jpg,” is to download an injector from the URL “https[:]//vertebromed[.]md/temp/dhssdf[.]exe” for use in a later stage. It's worth noting that the download link doesn't belong to a typical file-sharing cloud platform or the attacker's command-and-control (C2) server. Instead, it leverages the website “vertebromed.md,” which has been active since 2018. The injector file, “dhssdf.exe,” was created on May 29, 2023, as shown in Figure 6. Additionally, within the same folder, we discovered another MSIL loader named “IMG_3360_103pdf.exe,” created on May 30, 2023. Although this file isn't directly involved in the Word document attack chain, it also loads LokiBot and connects to the same C2 IP.

Figure 6: Web page and the compromised folder

2nd Stage – Injector

In this section, we analyze the injector obtained from Follina (SHA256: 9eaf7231579ab0cb65794043affb10ae8e4ad8f79ec108b5302da2f363b77c93). The injector is written in Visual Basic (VB), and we provide an overview of its basic information in Figure 7.

Figure 7: The information on the VB injector

Initially, the code extracts individual letters from predetermined strings. These letters are then combined to form an API string, subsequently mapped to the corresponding functions illustrated in Figure 8.

Figure 8: API functions

The injector utilizes a hardcoded key to decrypt the payload, as shown in Figure 9. The decryption process is outlined in pseudo-code in Figure 10. The decrypted data is decompressed using the “RtlDecompressBufferEx” API and the parameter “COMPRESSION_FORMAT_LZNT1”. The complete procedure through Python code and the partial payload is illustrated in Figure 11.

Figure 9: The key and encrypted data

Figure 10: The pseudo-code for decryption

Figure 11: The Python code and the final payload

The injector incorporates various evasion techniques, including:

- Checking the “BeingDebugged” flag of PEB (Process Environment Block)
- Utilizing the “NtGlobalFlag” to determine if the process was created by a debugger

- Verifying the existence of virtual machine paths, such as “\VMWare” and “\Oracle\virtualbox guest additions”
- Employing two calls to the “GetTickCount” API and using Sleep() to check if the time has been accelerated
- Using the “FindWindowW” function to identify the presence of specific debuggers, such as “OllyDbg,” “x64dbg”, “x32dbg”, “WindDbg,” “WinDbgFrameClass,” “ObsidianGUI,” “Soft Ice,” “ImmDbg,” “Zeta Debugger,” and “Rock Debugger”
- Checking the “ProcessDebugObjectHandle” (0x1E)

After obtaining the payload and verifying the overall environment, the injector utilizes the “VirtualAllocEx” function to allocate memory for the subsequent execution of LokiBot.

Figure 12: Assembly code for allocating memory

3rd Stage – LokiBot

LokiBot is specifically designed to gather sensitive information from various sources, including web browsers, FTP, email, and numerous software tools installed on the compromised system. Analyzing the C2 traffic to “95[.]164[.]23[.]2/swe/h/pin[.]php” in Figure 13, we determined that the version is 0x0012 and the notable Binary ID is “ckav[.]ru”. As this version of LokiBot has remained unchanged since March, we will only highlight its major components and features.

Figure 13: C2 traffic caputre of LokiBot

First, the MD5 hash derived from the MachineGuid is in the end pcap, “D0BECCE5760947DD9FFD80DB”. This hash serves as a mutex to ensure that multiple instances of LokiBot are not running simultaneously. It employs the “MoveFileExW” API to create a folder named “%APPDATA%\Roaming\576094” and a file named “47DD9F.exe” using a substring of the MD5 from MachineGuid. The file is marked as hidden by the “SetFileAttributes” function and setting the attribute to FILE_ATTRIBUTE_HIDDEN (0x2). The corresponding registry settings associated with LokiBot are depicted in Figure 14.

Figure 14: Registry setting

The list of targeted software names is stored in an array, and a partial list is provided in Figure 15.

Figure 15: Partial data of targeted software

Conclusion

LokiBot is a long-standing and widespread malware active for many years. Its functionalities have matured over time, making it easy for cybercriminals to use it to steal sensitive data from victims. The attackers behind LokiBot continually update their initial access methods, allowing their malware campaign to find more efficient ways to spread and infect systems.

LokiBot exploits various vulnerabilities and employs VBA macros to launch its attacks. It also leverages a VB injector to employ several techniques to evade detection or analysis. As a result, it can bypass certain security measures and pose a significant threat to users.

To protect themselves, users should exercise caution when dealing with any Office documents or unknown files, especially those that contain links to external websites. It is essential to be vigilant and avoid clicking on suspicious links or opening attachments from untrusted sources. Additionally, keeping the software and operating systems up to date with the latest security patches can help mitigate the risk of exploitation by malware.

Figure 16: LokiBot attack chain

Fortinet Protections

This malware is detected and blocked by FortiGuard Antivirus as:

- W32/LokiBot.DYST!tr
- W32/Injector.SBX!tr
- W32/Injector.XX!tr
- MSIL/Kryptik.AIVP!tr
- JS/Follina.N!tr
- VBA/Agent.0F29!tr
- MSOffice/Agent.9C55!tr

The FortiGuard AntiVirus service is supported by FortiGate, FortiMail, FortiClient, and FortiEDR, and the Fortinet AntiVirus engine is a part of each of those solutions. Customers running current AntiVirus updates are protected.

The FortiGuard Web Filtering Service blocks the C2 server.

FortiGuard IP Reputation and Anti-Botnet Security Service proactively block these attacks by aggregating malicious source IP data from the Fortinet distributed network of threat sensors, CERTs, MITRE, cooperative competitors, and other global sources that collaborate to provide up-to-date threat intelligence about hostile sources.

If you think this or any other cybersecurity threat has impacted your organization, contact our Global FortiGuard Incident Response Team.

IOCs

C2:

95[.]164[.]23[.]2

Files:

17d95ec93678b0a73e984354f55312dda9e6ae4b57a54e6d57eb59bcbbe3c382
23982d2d2501cfe1eb931aa83a4d8dfe922bce06e9c327a9936a54a2c6d409ae
9eaf7231579ab0cb65794043affb10ae8e4ad8f79ec108b5302da2f363b77c93
da18e6dcefe5e3dac076517ac2ba3fd449b6a768d9ce120fe5fc8d6050e09c55
2e3e5642106ffbde1596a2335eda84e1c48de0bf4a5872f94ae5ee4f7bffda39
80f4803c1ae286005a64ad790ae2d9f7e8294c6e436b7c686bd91257efbaa1e5
21675edce1fdabfee96407ac2683bcad0064c3117ef14a4333e564be6adf0539
4a23054c2241e20aec97c9b0937a37f63c30e321be01398977e13228fa980f29