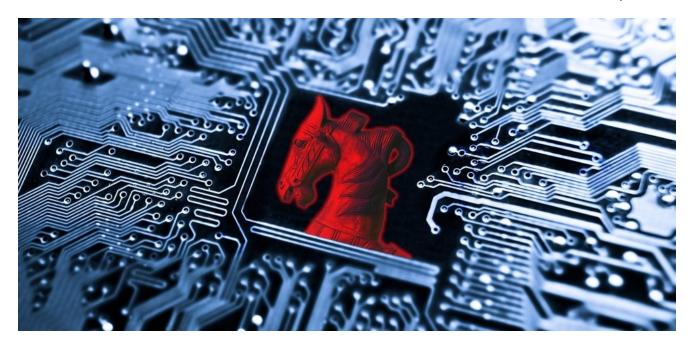
Mitigating Emotet, The Most Common Banking Trojan

protection witigating-emotet-the-most-common-banking-trojan/

July 26, 2018



Get Free Account

Join Now

Recently, Proofpoint released a fairly surprising <u>report</u>, stating that Banking Trojans have surpassed Ransomware as the top malware threat found in email. This is not too surprising, due to the rising difficulty of cashing out cyber-ransom operations, and the increasing awareness of enterprises for these kinds of threats. In addition, Emotet created recent headlines with the <u>US-CERT alert</u> about it.

Banking Trojans usually work by logging keystrokes of unsuspected users while they log into their online bank accounts, thus stealing their credentials to their online account. Infamous Banking Trojans include Ursnif, Dridex, Zeus and more.

Most noticable, is a malware named Emotet, which according to Proofpoint's recent report, consists of 57% of all Banking Trojans. At Intezer, we have also noticed a large amount of Emotet samples uploaded to our platform daily, both by our customers and our free community users.

To help organizations and individuals cope with this threat, I decided to post an informative blog post on how we can assist with our technology, in the battle against Emotet.

Emotet's Components and Similarities

As you might know, at Intezer, we like to approach the malware problem <u>by identifying code</u> <u>reuse and similarities</u>. In the case of Emotet (and other Banking Trojans), this approach proves itself to be very effective.

Emotet is usually composed from several different components: The initial dropper/loader, and the actual payload (which in many cases are several different ones). The initial dropper is designed to be very stealthy by constantly mutating. It's rare to see two different Emotet loaders that are exactly the same, due to obfuscation or other code mutation techniques. This is extremely effective in order to deal with signature-based defense.

The below report shows a code similarity analysis on the Emotet dropper. It is clear to see that almost 95% of the code is unique, which illustrates the mutation and evasiveness of this module. However, we can also see 3 "genes", or 5% of the file's code that was already seen in other Emotet variants. We can also observe that the malware was detected only by 12 anti-malware engines at the time of the first analysis in our system, which demonstrates the effectiveness of such evasion techniques.

Also, notice the file's metadata: it tries to mimic the Microsoft Windows product details, so that the file would seem legitimate to the untrained user — although there's absolutely no Microsoft code in this binary.

INTEZER Analyze TM I Examples	Enterprise Edition Plans Support	Mistory ↑ New File 0 0 0 0 Ital Tevet
だけ、 Emotet 8717b12462951 Malicious Family: Emotet	bfa615e3a03ffb336aa840e5d5046d4ad635b2ddb437faa928f Mailcious This file contains code from malicious software, therefore it's very likely that it's malicious.	SHA256: 8/177b12462951bfa615e3a03ffb336aa840e5d5046d4ad635b2ddb437 virustotal Report (12 / 68 Detections)
Original File 124 KB Malicious 8717b12462951bfa615e3a03ffb336aa840e5d5	8717b12462951bfa615e3a03ffb336aa840e5d5046d4ad635b2ddb437 Code Reuse (56 Genes)	ffaa928f 🕺 Malicious Probably Packed 🗘 🕁
Dynamic Execution	Emotet Malware 3 Genes 5.36%	
Memory (C:\Users\WMJNDesktop\geZOttm5N 56 KB Malicious Emotet (295 Genes) Memory (C:\Windows\SysWOW64bysdhcp.exe)	Unique Unknown	O 53 Genes 94,64%
56 KB Malicious Emotet (293 Genes) Memory (C:\Users\WMJI:Desktop\geZCktmSN	File Metadata Size 124 KB SHA256 8717b12462951b5a615e2a03ffb336aa840e5d5046d4ad635b2d	
56 KB Malicious Emotet (150 Genes) Memory (C:Windows/SysWOW64/Sysdhcp.exe)		
56 KB	ssdeep 1536:KTIH1Ag5rfx5bBJHYC+i17Pe3zH4bUUwnhiGj/bIDHK1Gt6NC	savorAgsinSjn2+LovwortAsv

Digging into the 3 gene connection, we can see exactly which other Emotet variants share some of these genes:

辥 INTEZER An	alyze TM Examp	les Enterprise	e Edition Plans Support			History	New File	Itai Tevet 🜘
	新 Emotet Malware						\$	4 37
	References https://feodotr							
	Date Discovered ~Su							
	Name	Label	SHA256	virustotal	Reused Genes			
					2 Genes			
					1 Genes			
					O 1 Genes			
			4b9743e52f7528b8ff712b3et		1 Genes			
					1 Genes			
					1 Genes			

In almost all Emotet cases, we noticed that although the dropper tries to mutate and evade signature-based defenses, we find a small amount of similarity (usually 2-5 "Genes"). It's a very small connection, but it's always there.

The dropper's goal is to execute the final payload in memory, where the attacker has much more "freedom" to stay hidden. Indeed, after running the dropper, we see modules in memory that contain much more code (see 56 genes of the dropper vs 296 genes in one of the payloads), and also have a much stronger code similarity to other Emotet variants.

INTEZER Analyze TM I Examples		1 History 1 New File
Xi 8717b124629 Emotet Maliciou Family: Emot		SHA256: B717b12462951bfa615e3a03ffb336aa840e5d5046d4ad635b2ddb437 virustotal Report (1/2 / 68 Detections)
Original File		
	Memory (C:\Users\WMJI\Desktop\geZcXtm5NM exe) 🧮 Maliciou	
	Code Reuse (296 Genes)	
Dynamic Execution 556 KB 556 KB Malicious Emotet (295 Genes) Memory (C:\Users\WMgI\Desktop\geZcKtm5N	Emotet Malware	295 Genes 99.66%
56 KB Malicious Emotet (295 Genes) Memory (C:Windows/sysW0W64lsysdhcp.exe)	Quasar RAT Malware 4 Genes 1.35%	
56 KB Malicious Emotet (293 Genes) Memory (C:\Users\WMJ\Desktop\ge2cktm5N	AlmaLocker Malware 3 Genes 1.01%	
56 KB Malicious Emotet (150 Genes) Memory (C-Windows/SysWOW64(sysdhcp.exe)	Princess Locker Malware 3 Genes 1.01%	
56 KB	APT29	

This is a pattern we see time and time again with Emotet: the initial payload is mutated/obfuscated and contains a small connection to Emotet, whereas the final payloads in memory are "unpacked" and contain code that correlates clearly to other variants.

Another interesting example of code similarities within Emotet, is the sample mentioned in <u>Check Point's blog post</u> where Emotet was embedding the free software Nirsoft in order to steal email credentials. Indeed after execution of the initial loader, we can see a module in memory that shares code with Nirsoft Mail PassView:

INTEZER Analyze TM Examples	Enterprise Edition Plans Support Blog About Intezer	0	1 New File Sign In Sign Up	
Emotet 621c0a11ee01 Malicious Family: Emote		SHA256: 621c0a11ee0 virustotal		
Original File	Memory (C:\Windows\SysWOW64\loaddll32.exe)	Suspicious Administratio	on Tool NirSoft	
	Code Reuse (688 Genes)			
Dynamic Execution				
200 KB Malicious Emotet (60 Genes) C\Users\WMJI\Desktop\PrsT8\WAsi.dll	Admin Tool		683 Genes 99.27%	
112 KB Suspicious Administration Tool NirSoft (Memory (C\Windows/SysWOW64Noaddill32.e	Unique Unknown — 4 Genes 0.58%			
112 KB Suspicious Administration Tool NirSoft (File Metadata			
	Size 112 KB SHA256 d5a4074f908b38fcab5eb401d62fe5f22e1468a12			
124 KB Suspicious Administration Tool NirSoft (Memory (C:Windows/SysWOW644loaddil32.e.,				

🍿 INTEZER Analyze™ 🗉	Examples Enterprise Edition Plans Support	Blog About Intezer	New File Sign In Sign Up
En (A) NirSo Admin			× €
6x40d0c9 6x40d0ce 6x40d0d4 6x40d0d4 6x40d0d5 6x40d0d5	push 0x1400 lea eax, dword ptr [ebp - 0x1dd8] push ebx push eax call 0x12030		
Original File 0x40d0db 0x40d0de 0x40d0e3 Malicious 0x40d0e5	add esp, 0x18 push 0x4145f0 ; lea edi, dword ptr [ebp - 0x58] ; call 0x11c8a ;		
0x40d0eb 621c0a11ee0 0x40d0f0 0x40d0f2	push 0x414694 ; mov esi, eax call 0x11c8a		
0x40d0f7 0x40d0fc 0x40d0ff 0x40d0ff 0x40d104	<pre>mov dword ptr [ebp - 0xc], eax call 0x11c8a push 0x4146bc ;</pre>	"encryptedPassword" "usernameField"	
0x40d109 Malicious 0x40d10c 0x40d10c 0x40d10c 0x40d111 0x40d111 0x40d114 0x40d110	mov dword ptr [ebp + 0xc], eax call 0x11c8a push 0x4146cc mov dword ptr [ebp - 8], eax call 0x11c8a		
6x40d11e 9x40d123 9x40d126 0x40d126 0x40d126 0x40d127 0x40d127 0x40d127 0x40d127			
0x40d137 Suspiciou: 0x40d138 0x40d139 0x40d139 Memory (C 0x40d139	push esi push eax call 0x12036 pop ecx		
Suspicious			
Memory (C:Windows\SysWOW64\loaddll32			

https://analyze.intezer.com/#/analyses/22c91372-2663-4d5c-940f-7090f222cef0/sub/7fe6c86f-cc1e-4d36-80e2-6fca66f0e4cd

Mitigating Emotet

(A) I have a suspicious file that I suspect is Emotet

You can analyze the suspicious file with <u>Intezer's community edition</u>. It's free \bigcirc You'll have the same insights as the above examples, and know instantly if the file is indeed malicious, and if it's really Emotet or a different kind of threat. <u>Search for other malicious Emotet files</u> <u>across your network</u> using instructions in (C).

(B) I suspect that one of our machines is infected by Emotet

1. Check for scheduled tasks using <u>Autoruns</u> to spot binaries that are a part of Emotet's persistence mechanism. Analyze these files using <u>VirusTotal</u> or <u>Intezer</u>.

2. Since Emotet's payload resides in memory, memory analysis is your best choice to triage and analyze a suspicious endpoint. Use <u>winpmem</u> to obtain a memory image of the computer, and then use <u>Volatility</u> to extract all loaded executables from memory. You can do that by using <u>Volatility's procdump and dlldump plugins</u>. Once you obtain all binaries from memory, send them to <u>Intezer</u> for analysis to identify Emotet payloads. Code similarity detection works just fine also with memory-extracted items.

3. Submit the detected Emotet payload to your anti virus vendor and update signatures. As initial response, you can kill the process and remove the persistence binary using Autoruns.

(C) I suspect, or I don't know, if machines across my network are infected with Emotet

As mentioned before, due to the fact that Emotet resides in memory, a good approach would be to search for a <u>YARA</u> signature across all the network. For that, you'll need a relevant YARA rule that detects Emotet payloads, as well as a YARA memory scanner.

I recommend checking out Loki, an open-source YARA scanner by Florian Roth (@cyb3rops) that can help you scan endpoints across your network. For YARA rules to detect Emotet, there's one that GoDaddy's security team has written: <u>https://github.com/godaddy/yara-rules/blob/master/emotet.yara</u>, and the one within the CAPE project: <u>https://github.com/ctxis/CAPE/blob/master/data/yara/CAPE/Emotet.yara</u>.

Once a file is identified using a YARA rule, we recommend to dump the suspicious process and analyze it with <u>Intezer Analyze</u>, to confirm the initial detection. Signature-based detection such as YARA may result sometimes in false-positives, especially in memory, so it's always a good practice to reconfirm.

Summary

Emotet and other Banking Trojans cause a huge pain for enterprise organizations and endusers alike. Usually, these types of malware are highly evasive and polymorphic, which poses a challenge for many security solutions to protect against them. However, time and time again we observed that when utilizing code similarity detection techniques, these threats can be better identified and mitigated.

We hope that the information provided in this post, together with our free malware analysis platform, will help to decrease the number of Emotet victims worldwide.



Itai Tevet

Once led a government CERT. Now, CEO at Intezer, changing the way we detect, analyze and respond to malware.