Black Lotus Labs Blog:

A look inside the TrickBot botnet

Oct. 12, 2020

A look inside the TrickBot botnet

Bolstered by contributions from Black Lotus Labs and others, Microsoft and FS-ISAC recently took court action to disrupt a particularly insidious botnet called TrickBot – a significant source of ransomware and banking credential theft.

As the threat intelligence arm of Lumen Technologies, Black Lotus Labs exists to defend a clean internet, and we collaborate with other threat intelligence and security companies to do just that. In the Microsoft TrickBot case, we provided intelligence and a legal declaration that was filed with the court in support of Microsoft's application for a temporary restraining order to halt TrickBot's operations

This type of industry collaboration is key to disrupting bad actors' activities and increasing their cost of doing business.

A (Brief) History of TrickBot

TrickBot (a.k.a. The Trick, Trickbot, or TrickLoader) is a dangerous and widespread botnet that has operated with relative impunity since it emerged in 2016. It is believed to have evolved from an earlier botnet known as Dyre or Dyreza. From its onset, TrickBot had the ability to steal credentials when a user accessed banking websites from an infected machine; hence, it was initially categorized as a banking trojan.

The criminal gang behind TrickBot has regularly updated its malicious software, adding modules with new functionality to increase its effectiveness and potential to cause harm. They have incorporated tools such as Mimikatz and Cobalt Strike – often used by penetration testers and criminal attackers – to map victim networks, steal operating system credentials, and spread inside organizations. By 2018, the one-time banking trojan had clearly evolved to become a loader for other malware, including Ryuk ransomware.

Botnets are often built with a single or small number of fixed controllers, referred to as command-and-control servers, or C2s. If TrickBot had a single server providing C2 capabilities, the groups monitoring TrickBot – including Black Lotus Labs – would have successfully neutered it by now. Instead, TrickBot has evolved to use a complex infrastructure that compromises third-party servers and uses them to host malware. It also infects consumer appliances such as DSL routers, and its criminal operators constantly rotate their IP addresses and infected hosts to make disruption of their crime as difficult as possible. To add to the complexity, different C2s may serve different purposes. For example, one group of servers ("core C2s") might communicate directly with the bots, and another group ("plugin servers") might serve up plugin modules after the initial infection.

An Earlier Disruption Attempt

Our collaboration with Microsoft was not the industry's first attempt to disrupt TrickBot. A few weeks prior to our collaborative effort, we learned that bots were receiving TrickBot configurations to replace the list of active C2 server IPs with reserved addresses (127.0.0.1, 0.0.0.0). These configurations prevented communication to the C2s, in effect blocking the infected bot from participating in the botnet. The unusual configurations were revealed publicly in early October, but the identity of the responsible parties remains a mystery.

While this disruption did remove some bots from the botnet, it did not prevent new infections from occurring because the botnet infrastructure remained intact. In fact, during the week of October 6, the TrickBot infections continued, with one of our partners even witnessing new TrickBot installations being delivered from another nefarious botnet – Emotet.

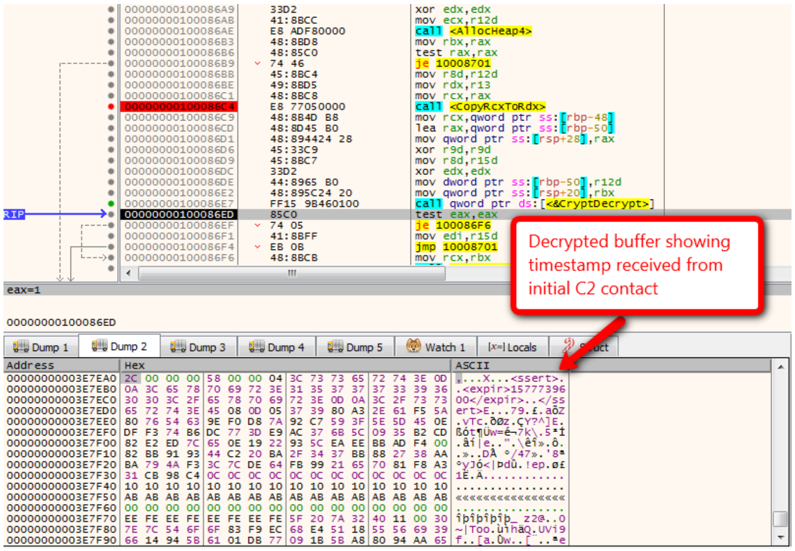
Black Lotus Labs vs. TrickBot

For several years, Black Lotus Labs has used the data from our global network visibility to target numerous malware families to better understand them, track their activity, and disrupt or mitigate their malicious activity. In 2019, we focused on TrickBot by reverse-engineering a TrickBot malware sample and writing a validator that allowed us to add detection of TrickBot C2s to our monitoring systems.

Reverse Engineering

With TrickBot, an infected host contacts the C2, which sends back commands to execute or software to download. The goal of reverse engineering is to understand these exact communication protocols and encryption techniques so we can validate C2s.

Because we don't have access to the original source code, a reverse engineer on the Black Lotus Labs team must dissect the contents of the binary executable itself. Developers of malware actively seek to make such analyses more difficult by encoding, packing, and even encrypting sections of the file. By hiding the presence of character strings that might reveal useful information such as C2 IP addresses, commands, and encryption keys. Our team's goal in reverse engineering is to understand more about how the malware operates in a network environment in order to detect and monitor it across our global network.



```
000000010086A9   33D2            xor edx,edx
000000010086AB   41:8BCC         mov ecx,r12d
000000010086AE   E8 ADF80000     call <AllocHeap4>
000000010086B3   48:8BD8         mov rbx,rax
000000010086B6   48:85C0         test rax,rax
000000010086B9   74 46           je 10008701
000000010086BB   45:8BC4         mov r8d,r12d
000000010086BE   49:8BD5         mov rdx,r13
000000010086C1   48:8BC8         mov rcx,rax
000000010086C4   E8 77050000     call <CopyRcxToRdx>
000000010086C9   48:8B4D B8      mov rcx,qword ptr ss:[rbp-48]
000000010086CD   48:8D45 B0      lea rax,qword ptr ss:[rbp-50]
000000010086D1   48:894424 28    mov qword ptr ss:[rsp+28],rax
000000010086D6   45:33C9         xor r9d,r9d
000000010086D9   45:8BC7         mov r8d,r15d
000000010086DC   33D2            xor edx,edx
000000010086DE   44:8965 B0      mov dword ptr ss:[rbp-50],r12d
000000010086E2   48:895C24 20    mov qword ptr ss:[rsp+20],rbx
000000010086E7   FF15 9B460100   call qword ptr ds:[<&CryptDecrypt>]
000000010086ED   85C0            test eax,eax
000000010086EF   74 05           je 100086F6
000000010086F1   41:8BFF         mov edi,r15d
000000010086F4   EB 0B           jmp 10008701
000000010086F6   48:8BCB         mov rcx,rbx

eax=1

00000000100086ED

Address           Hex                                              ASCII
000000003E7EA0    2C 00 00 00 58 00 00 04 3C 73 73 65 72 74 3E 0D  ,...X...<ssert>.
000000003E7EB0    0A 3C 65 78 70 69 72 3E 31 35 37 37 37 33 39 36  .<expir>15777396
000000003E7EC0    30 30 3C 2F 65 78 70 69 72 3E 0D 0A 3C 2F 73 73  00</expir>..</ss
000000003E7ED0    65 72 74 3E 45 08 0D 05 37 39 80 A3 2E 61 F5 5A  ert>E...79.£.aõZ
000000003E7EE0    80 76 54 63 9E F0 D8 7A 92 C7 59 3F 5E 5D 45 0E  .vTc.ðØz.ÇY?^]E.
000000003E7EF0    DF F3 74 B6 DC 77 3D E9 AC 37 6B 5C 09 35 B2 CD  ßót¶Üw=é¬7k\.5²Í
000000003E7F00    82 E2 ED 7C 65 0E 19 22 93 5C EA EE BB AD F4 00  .âí|e..".\êî».ô.
000000003E7F10    82 BB 91 93 44 C2 20 BA 2F 34 37 BB 88 27 38 AA  .»..DÂ º/47».'8ª
000000003E7F20    BA 79 4A F3 3C 7C DE 64 FB 99 21 65 70 81 F8 A3  ºyJó<|Þdû.!ep.ø£
000000003E7F30    31 CB 98 C4 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C  1Ë.Ä............
000000003E7F40    10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10  ................
000000003E7F50    AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB  ««««««««««««««««
000000003E7F60    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000003E7F70    EE FE EE FE EE FE EE FE 5F 20 7A 32 40 11 00 30  îþîþîþîþ_ z2@..0
000000003E7F80    7E 7C 54 6F 6F 83 F9 EC 68 E4 51 18 55 56 69 39  ~|Too.ùìhäQ.UVi9
000000003E7F90    66 14 94 5B 61 01 DB 77 09 1B 5B A8 80 94 AA 65  f..[a.Ûw..[¨..ªe
```

Decrypted buffer showing timestamp received from initial C2 contact

1 Example of decrypting communication between a bot and C2

When TrickBot infects a machine, the malware generates a unique bot identifier (ID), and communications to the C2 include both the bot ID and a short group tag (referred to in the TrickBot configuration files as "gtag") which indicate the campaign leading to the infection. The gtag used in TrickBot campaigns typically has three or four letters and one to four digits, such as "tmt2'" "ono76," "chil45" or "mor124." The TrickBot communication protocol itself implements one- to two-digit commands to update the bot software, download additional modules and their configurations, or even download a different malware entirely (such as ransomware). Communications from the C2 to the bot are encrypted (making it more difficult for network monitoring to discern the contents of the communications) and include a digital signature (protecting the integrity of messages).

Validating

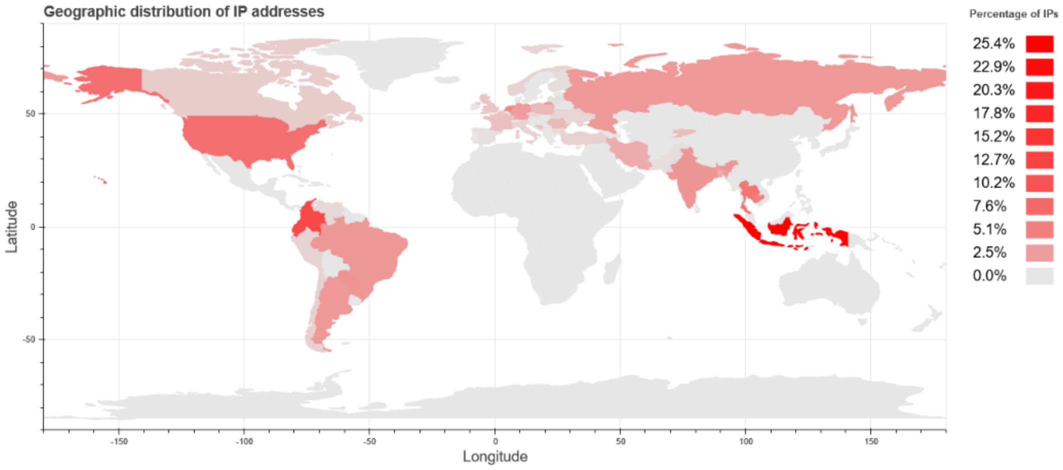
After we finish reverse-engineering the TrickBot sample, our team writes a validator, which simulates bot communication to the C2s. This enables us to confirm that a suspected C2 server was responding in the manner unique to TrickBot. We regularly monitor C2s to track changes in the TrickBot infrastructure so we know when a C2 has gone offline. In addition to core C2s, we validate plugin servers, which together represent an important subset of the overall TrickBot infrastructure.

Leveraging our network visibility, we also identify infected bots by examining communication with confirmed C2s. We use machine learning algorithms that monitor our network data to identify other suspected C2s by training a classifier on network characteristics from C2 traffic. Additionally, we review third-party reports of suspicious behavior, and subsequently test those suspected C2s using our validator.

Once we confirm active C2s, we submit abuse notifications to the relevant ISPs and hosting providers to have the C2s taken down. In the past year, Black Lotus Labs has seen more than 500 IP addresses serving as active TrickBot C2s.

Based on our data, below are the top 15 countries where we saw hosting TrickBot C2s, weighted by the number of days a C2 was active:

Country Name	C2s * Days Active
Indonesia	1,362
Colombia	652
Ecuador	637
United States	362
Thailand	330
Cambodia	285
Netherlands	232
Paraguay	174
India	165
Brazil	159
Russia	156
Argentina	154
Germany	135
Iran	86
Bangladesh	86



What's Next

Black Lotus Labs will continue to closely monitor to see what impact the collaboration with Microsoft has had on TrickBot's infrastructure and activities, and we hope to learn more for future disruption efforts. While our work might not remove the threat posed by TrickBot, it will raise the cost of doing business for the criminal gang behind the botnet because they will be forced to divert resources away from exploitation activities in order to rebuild the parts of their infrastructure that we disrupted. Meanwhile, we will continue our work to notify victims, null route traffic for malicious servers, disrupt botnets and other malicious activity, and share intel with partners in our efforts to make the internet a safer place for all.

Indicators of Compromise:

Sample hash:
4691a862842f286924357f074927b4e9f276c6073458e3dd4c66218cf4918ea1

C2 IPs:

5.152.210[.]188
5.182.210[.]224
5.182.211[.]124
5.182.211[.]138
27.147.173[.]227
36.66.218[.]117
36.89.182[.]225
36.89.243[.]241
36.91.45[.]10
36.91.87[.]227
36.94.33[.]102
45.127[.]222.8
45.138.158[.]33
45.148.10[.]174
45.66.10[.]22
45.89.125[.]148
45.89.127[.]27
51.77.112[.]252
51.83.196[.]234
51.89.215[.]186
62.108[.]35.9
80.210.32[.]67
83.220.171[.]175
85.204.116[.]117
89.249.65[.]53
91.200.100[.]71
91.200.103[.]236
92.38.135[.]61
92.62.65[.]163
93.189.42[.]225
96.9.73[.]73
96.9.77[.]142
96.9.77[.]56
103.111.83[.]246
103.12.161[.]194
103.196.211[.]120
103.221.254[.]102
103.36.48[.]103
103.76.169[.]213
104.161.32[.]108
104.161.32[.]118
107.155.137[.]15
110.93.15[.]98
112.109.19[.]178
117.252.214[.]138
121.100.19[.]18
121.101.185[.]130
122.50.6[.]122
129.232.133[.]39
131.161.253[.]190
139.60.163[.]45
156.96.46[.]27
158.181.155[.]153
176.31.28[.]85
177.190.69[.]162
179.127.88[.]41
180.211.170[.]214
181.112.157[.]42
181.129.104[.]139
181.129.134[.]18
181.143.186[.]42
182.253.113[.]67
185.14.30[.]247
185.142.99[.]94
185.172.129[.]100
185.234.72[.]114
185.234.72[.]35
185.236.202[.]249
185.25.51[.]139
185.99.2[.]106
185.99.2[.]115
186.159.8[.]218
190.136.178[.]52
190.145.83[.]98
190.152.182[.]150
190.214.28[.]74
190.99.97[.]42
192.3.246[.]216
194.5.249[.]214
194.5.249[.]215
194.87.236[.]171
195.123.238[.]83
195.123.239[.]193
195.123.240[.]18
195.123.240[.]93
195.123.241[.]224
195.123.241[.]229
195.161.62[.]25
200.116.159[.]183
200.116.232[.]186
200.171.101[.]169
200.29.119[.]71
201.231.85[.]50
212.22.70[.]59
220.247.174[.]12