Trickbot – An analysis of data collected from the botnet

GovCERT.ch

September 20, 2019

1 Introduction

We are monitoring various threats and in that context we have collected quite some data about the Trickbot botnet in the past few years. This paper is based on an analysis of selected aspects of our Trickbot data collection. Some of our analysis is rather straightforward, yet, we also take the freedom to make some speculative statements, which might turn out to be debatable or plain wrong. In that spirit we are open for discussions and are happy to receive comments by the readers of this article.

Our analysis consists of two main parts. In the first part we consider the PE timestamps of Trickbot droppers (i.e., the binaries being distributed by the Trickbot operators) and of the respective payloads (i.e., the PE binaries which are unpacked and then executed once a dropper is executed). The analysis is based on a collection of approximately 2100 droppers and corresponding payloads which were collected between July 2016 and February 2019. The main insights from this analysis are:

- The PE timestamp of many trickbot droppers is backdated, while the PE timestamp of the payloads is unmodified and thus reflects the actual production time of samples.
- The same payload is re-packed over and over again into different droppers. We have observed up to 69-fold repacking.
- The working times of the operators is consistent with working hours in the Moscow time zone.
- The production of Trickbot binaries is likely operated by humans, and thus not fully automated.

In the second second part we analyse a collection of Trickbot config files which we have collected by emulating the protocol over a period of 4-5 months end of 2018 beginning of 2019. The config files contain information on the Trickbot infrastructure such as exfiltration sites used by different stealer modules, the first level C2 infrastructure, etc., as well as lists of targeted financial institutions.

The main insights from this analysis are:

- There is a sequence visible in two configuration types (static injects and mailconf) that shows that the attackers are regularly exchanging these infrastructure elements.
- The sequence is less clear in the main configuration file where we can observe some temporal overlapping of the C2 servers.

- The lifetime of how long a C2 server remains in service varies. The C2 servers in the main config are used only for a short time (with some exceptions) and the C2 servers from the static inject and mailconf file are used for a longer period.
- This leads to the conclusion that the attackers are actively managing their infrastructure by exchanging the C2 servers on a regular base.
- We also extracted the targets from the configuration files and observed that the main targets are banks in the US, Great Britain, Ireland and Germany. Interestingly, German targets were added during our analysis period in the month of November.

2 An Analysis of Dropper and Payload Timestamps

As many malware families, Trickbot is delivering its samples ("droppers") in packed form. The effective "payload" is contained within the dropper and unpacked upon execution. The payloads are also in PE format and can be easily recovered using simple memory dumping and PE restoration techniques.

Our subsequent analysis is based on a collection of approximately 2100 droppers and corresponding payloads which were collected between July 2016 and February 2019. For each dropper we consider three different timestamps: The *dropper's PE timestamp*, the corresponding *payload's PE timestamp*, and the *first-seen timestamp* of the dropper as reported by VirusTotal [1] and / or Abuse.CH [2] (if both services report a first-seen timestamp for a sample, we choose the earlier of the two).

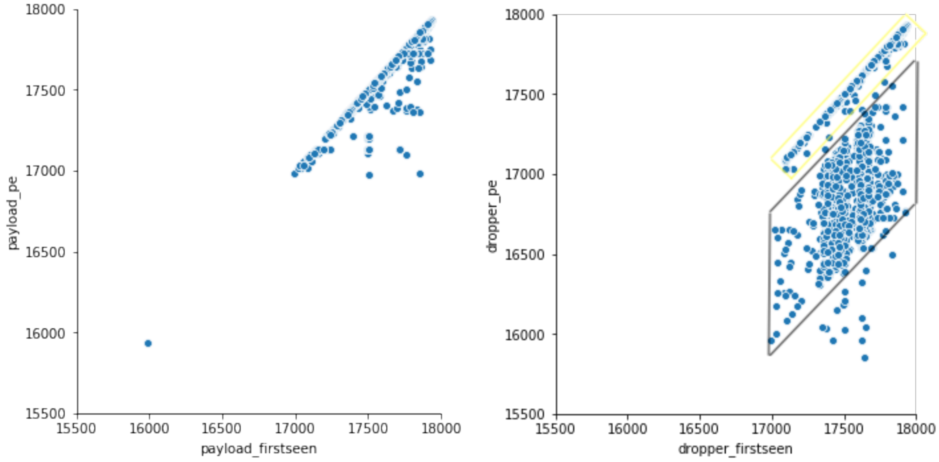
2.1 Backdated Droppers and Unmodified Payload Timestamps

For non-targeted malware that is distributed through spam waves (such as the Trickbot family), we would expect that the first-seen time of a sample is a reasonable estimate for the time when a sample was released into the wild. Further assuming that samples are produced shortly before being released into the wild, we can expect the first-seen times to approximate the production times (i.e., the PE timestamps) of samples.

In the following we use the first seen timestamps to analyze whether, and if so, to what extent Trickbot payload and dropper PE timestamps are forged. Figure 1 compares the first seen times with the PE timestamps of droppers and payloads.

First we look at the relation between *payload PE timestamps* vs. first-seen timestamps in Figure 1(a). Our interpretation of this figure is that the *payload PE timestamps are not backdated*, i.e., that the PE payload timestamps correspond to the actual compilation times of the payloads. The reason is that the distribution in the plot corresponds to what one would expect from a random process such as the collection of malware samples using honeypots. In fact, we see that most samples are caught relatively soon (first seen timestamp is roughly equal to the PE timestamp) and the number of samples that survive longer in the wild is falling off quickly.

Next we consider the *dropper PE timestamps* in Figure 1(b). The figure suggests that there are two type of droppers: those that are not backdated (the "yellow samples" in the figure) and those that are backdated by roughly 300 - 1000 days (the "black samples" in the figure). One could argue that the "black samples" are not backdated samples but rather just samples that go undetected in the wild for a longer time. We do not think so because there



(a) Payload PE timestamp vs. first-seen times from threat feeds.

(b) Dropper PE timestamp vs. first-seen times from threat feeds.

Figure 1: PE timestamps vs. first-seen dates for droppers and payloads (measured in days since 1970).

is a time gap between the yellow and black samples. As mentioned earlier, catching samples in the wild is a random process probably following a Poisson distribution. The existence of the gap is not consistent with such a random process. A much more plausible explanation is that gap is caused by backdating the black droppers.

Further evidence for dropper backdating. There is another observation that strengthens the backdating hypothesis for droppers and the "non-modification hypothesis" for payloads. The earliest published research (we have found) mentioning the Trickbot family dates back to fall 2016 [3, 4]. This research suggests that the inception of the Trickbot family likely dates back to summer or fall 2016.

We have looked at the timestamps of the samples mentioned in the research reports, and they support our observations: In fact, the Malwarebytes [3, 4] article contains hashes of a dropper[1] and payload [2] pair. The respective timestamps of dropper and payload are `09.03.15 00:49` and `11.10.16 19:04`. The payload timestamp is consistent with the conjectured inception date of the Trickbot family and thus seems not to be backdated. On the other hand, the dropper timestamp dates back to spring 2015 way before the family's conjectured inception date and is therefore likely backdated.

In a nutshell, we believe that payload PE timestamps reflect the actual production time of the payloads. Concerning droppers, it seems that there are roughly two categories of droppers. Namely those that are backdated by several hundred of days and those that are not backdated.

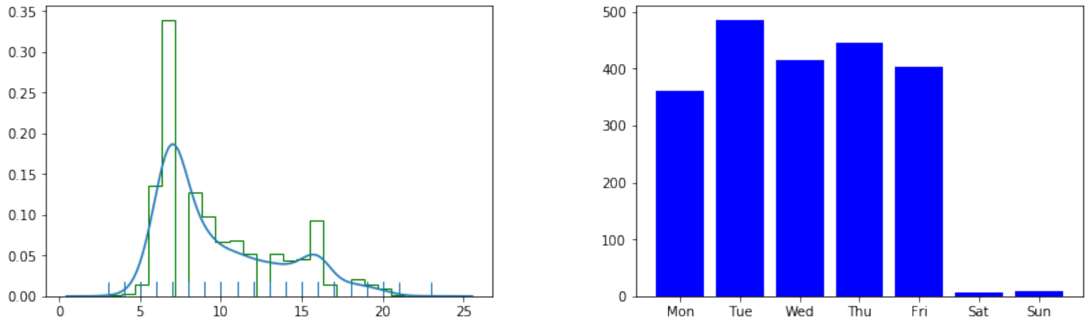
[1] `f26649fc31ede7594b18f8cd7cdbbc15`

[2] `f24384228fb49f9271762253b0733123`

2.2 Working Days and Hours of Trickbot Operators

Under the assumption that the payload PE timestamps reflect the actual production dates we can try to establish the hours of activity of the operators producing Trickbot samples. To this end we have plotted the distribution of the hours found in the payload PE headers in Figure 3(a). The plot clearly shows periods of activity and periods of rest. The lifetime of these periods matches rather well with a human's activity and rest periods. We thus conclude that the production of new samples is not entirely automated but rather performed by humans. Figure 3(b) shows the number of samples produced on different days of the week. This is again highly consistent with human working habits: most weekends are off, slight under-productivity on Mondays etc.

Timestamps have been used to "determine" the timezone of malware operators in the past [7]. There is inherent uncertainty of a couple of hours in such attributions, due to the fact that the malware operators can be early birds, late risers etc. (assuming that cybercrime operations allow for flexible working hours). Moreover, PE timestamps can be modified at will. Yet, Trickbot has been attributed to Eastern actors in several publications [5, 6]. We believe that the working hours in our plot seem to be compatible with this attributions. For instance, the period of rest which is 22h - 3h in UTC time, translates into a period of rest from 1h - 6h in UTC+3 which e.g., corresponds to Moscow's timezone.



(a) PE timestamps of payloads (in hours UTC) on x-axis, relative frequency on y-axis.

(b) PE timestamps of payloads grouped by weekdays on x-axis, absolut frequencies on y-axis.

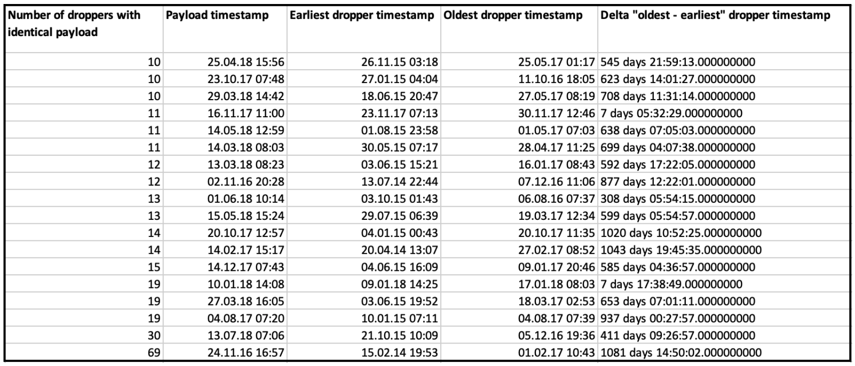
Figure 2: PE timestamps vs. first-seen dates for droppers and payloads.

2.3 Repackaging of Payloads

A widely known technique to avoid AV detection is to pack the same malware sample using different variants of packing algorithms resulting in different binaries which are deployed in the wild. We were wondering whether we could find signs of payload packing in our Trickbot data set. To this end we have clustered droppers that contain the same[3] payload. An excerpt

[3] We consider payloads to be equal when they have same PE timestamp. Since we unpack payloads from memory the resulting payloads are not identical, i.e., they do not have the same hash value. For a selected few samples we have verified manually using binary diffing techniques that payloads with the same PE timestamps contain essentially equivalent code, and that the code of payloads with different PE timestamps has lower similarity.

from the results is shown in Figure 5. The results clearly confirm that the Trickbot operators are practicing repacking.



Number of droppers with identical payload	Payload timestamp	Earliest dropper timestamp	Oldest dropper timestamp	Delta "oldest - earliest" dropper timestamp
10	25.04.18 15:56	26.11.15 03:18	25.05.17 01:17	545 days 21:59:13.000000000
10	23.10.17 07:48	27.01.15 04:04	11.10.16 18:05	623 days 14:01:27.000000000
10	29.03.18 14:42	18.06.15 20:47	27.05.17 08:19	708 days 11:31:14.000000000
11	16.11.17 11:00	23.11.17 07:13	30.11.17 12:46	7 days 05:32:29.000000000
11	14.05.18 12:59	01.08.15 23:58	01.05.17 07:03	638 days 07:05:03.000000000
11	14.03.18 08:03	30.05.15 07:17	28.04.17 11:25	699 days 04:07:38.000000000
12	13.03.18 08:23	03.06.15 15:21	16.01.17 08:43	592 days 17:22:05.000000000
12	02.11.16 20:28	13.07.14 22:44	07.12.16 11:06	877 days 12:22:01.000000000
13	01.06.18 10:14	03.10.15 01:43	06.08.16 07:37	308 days 05:54:15.000000000
13	15.05.18 15:24	29.07.15 06:39	19.03.17 12:34	599 days 05:54:57.000000000
14	20.10.17 12:57	04.01.15 00:43	20.10.17 11:35	1020 days 10:52:25.000000000
14	14.02.17 15:17	20.04.14 13:07	27.02.17 08:52	1043 days 19:45:35.000000000
15	14.12.17 07:43	04.06.15 16:09	09.01.17 20:46	585 days 04:36:57.000000000
19	10.01.18 14:08	09.01.18 14:25	17.01.18 08:03	7 days 17:38:49.000000000
19	27.03.18 16:05	03.06.15 19:52	18.03.17 02:53	653 days 07:01:11.000000000
19	04.08.17 07:20	10.01.15 07:11	04.08.17 07:39	937 days 00:27:57.000000000
30	13.07.18 07:06	21.10.15 10:09	05.12.16 19:36	411 days 09:26:57.000000000
69	24.11.16 16:57	15.02.14 19:53	01.02.17 10:43	1081 days 14:50:02.000000000

Figure 3: Repacking of payloads. Table shows clusters of droppers (which are different) but which contain the same payload once unpacked.

We have also included the timestamps of the payload as well as of the earliest and oldest dropper containing the payload. The table further confirms our previous analysis: it clearly shows that in many but not all cases the same payload is packed into droppers whose timestamps vary considerably due to backdating.

2.4 Trickbot Production Cycles?

In this last and possibly most speculative part of our PE analysis we are comparing dropper and payload PE timestamps. Naively, we would expect that payloads are produced / compiled first, and then packed, resulting in the dropper containing the payload. As a consequence we would expect that dropper PE timestamps are somewhat older than the payload PE timestamps, and that the difference in timestamps reflects the *production time* of a Trickbot sample.

Figure 4 compares the PE timestamps of droppers and payloads. The plot reveals roughly two groups of samples. Those that fall into the "green region" and those that fall into the "red region". The red region consists of the samples whose droppers are backdated (see our discussion above). This region is useless for our analysis of production times. The samples in the "green region" are those whose payload and dropper are roughly produced around the same time. These are thus the samples that are fit for a production time analysis.

The table in Figure 5 shows the distribution of production times of the "green samples". For a total of 838 samples (which corresponds to $\sim$ 39% of our sample set) we found a production time in the range of $0h - 24h$.

We did not come up with a conclusive analysis of the numbers in Figure 5. The samples in the $0h - 2h$ production range seem to be somewhat plausible and can be explained by an automated tool chain that first compiles the payload, let's say on one machine, and then passes on the payload to a packer machine. Yet we would expect this production times to be somewhat constant and we have no good explanation why the production process of some samples apparently takes many hours. Maybe a deeper analysis of the samples and the packers used in Trickbot production could shed some light on this issue.

Last but not least we would like to point out that it is uncertain whether the numbers in Figure 5 indeed reflect the production times: (i) Unlike for normal compilers, we do not

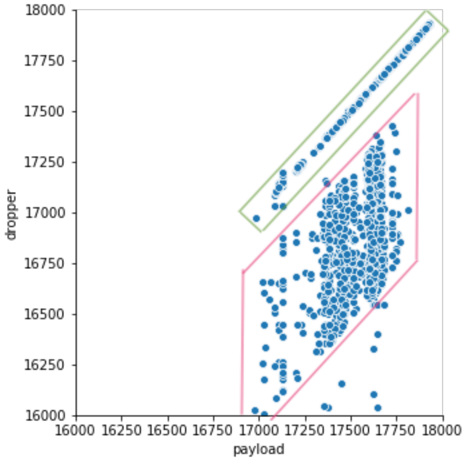


Figure 4: PE timestamps for payloads on x-axis, for droppers on y-axis (measured in days since 1970).

know how packers set the PE timestamps of the dropper files they produce. (ii) We have found that for 8% of the samples the dropper PE timestamp is $0h$ to $24h$ *older* than the payload timestamp. The existence of such samples can be hypothetically explained by clock synchronization issues between multiple machines or services used for compilation and subsequent sample packing. This however implies that we cannot necessarily trust PE timestamps, even for the samples whose timestamp is not intentionally forged (iii) As we said earlier, PE timestamps can be forged at will.

Production time	Number of samples
0h-1h	262
1h-2h	160
2h-3h	97
3h-4h	81
4h-5h	54
5h-6h	43
6h-7h	37
7h-8h	42
8h-9h	22
9h-10h	5
10h-11h	3
11h-12h	1
12h-13h	1
13h-14h	3
14h-15h	8
15h-16h	3
16h-17h	0
17h-18h	2
18h-19h	1
19h-20h	4
20h-21h	2
21h-22h	3
22h-23h	3
23h-24h	1

Figure 5: Number of Trickbot samples with production times of 0 – 24 hours in 1 hour intervals.

3 Infrastructure Analysis

In this section we are going to have a deeper look at the networking infrastructure of Trickbot based on the information we collected during approximately 5 months. We do not go into details about Trickbot networking protocol as the focus lies on the temporal analysis. However a brief introduction of the way Trickbot communicates might be helpful for the further understanding, Figure 6 shows a high-level schema of how Trickbot communicates.

The most common infection vector are weaponized Office documents that trigger the download of the Trickbot binary or a dropping of Trickbot after an Emotet infection has happened. The first method is using Powershell code that is embedded in the Office document. The Powershell scripts download the binary directly from a webserver and executes it. The second is commonly seen during targeted ransomware attacks such as reported by Trend Micro [10] and us [11].

After the successful infection, Trickbot begins to communicate with the first stage C2 servers that are in the configuration delivered within the binary. These first stage C2 servers are mostly compromised systems. The communication is encrypted and uses either TCP port 443 or (often) TCP port 447 or 449. Interestingly, the certificates used for these communications are self-signed and use the default parameters of OpenSSL (”organizationName=Internet Widgits Pty Ltd”). The malware then downloads the next actual configuration file (we name it main.cfg) with a list of C2 servers to connect to. Communication however remains identical using SSL with the aforementioned ports. Depending on the module, additional C2 servers come into play that are contained in additional configuration files. In the following we focus on the configuration file of the injectDll module (or more precisely injectDll32 or injectDll64 depending on the platform), which is used for credential theft and injects within the browser.

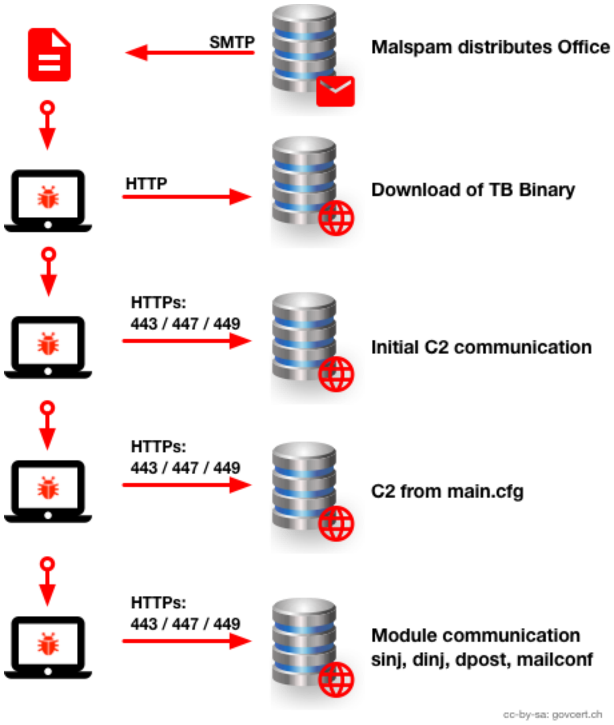


Figure 6: TrickbotNetwork

There are 3 types of configuration files shown in Table 1 that are going to be discussed later in this document.

File	Description	No of files
sinj	Static injects, contains targets, C2 servers, used by injectdll	1566
dinj	Dynamic injects, contains targets, C2 servers, used by injectdll	1559
dpost	Password Grabber, contains exfiltration IP addresses, used by injectdll	1697
mailconf	Email stealer, contains exfiltration IP addresses, used by mailsearcher	1648
main	Main configuration of Trickbot	7156

Table 1: Overview of collected configuration files

We have analyzed the configuration files and extracted IP addresses, domain names and targets for their temporal and spatial traits and are going to present them in the following sections.

3.1 Analysis of C2 Servers

Information about C2 servers is stored in the configuration files mentioned above. We have extracted the IP addresses, Autonomous System (AS), geolocation and their temporal behavior. The term temporal behaviour explains how the infrastructure elements are changing over time. In the following chapters we are going to analyze the C2 servers for basic configuration, static injects, dynamic injects, mail exfiltration and credential theft.

3.1.1 Analysis of Main Configuration

We collected a total of 316 IP addresses in the main configuration files. These show interesting patterns as there are some hosting providers that are often used. In Listing 7 an excerpt of a typical main configuration file of Trickbot is shown. In the context of the network analysis, the `<srv>` tags are important, as they consist of the IP address and the port number. We extracted and analyzed the IP addresses and are introducing the results in the subsequent sections. The `<gtag>` displays the campaign ID. After the `<servs>` section the module configuration follows. In our example, the System Reconnaissance and the Browser Inject modules are configured.

```
<mcconf>
<ver>1000292</ver>
<gtag>tt0002</gtag>
<servs>
<srv>51.68.170[.]58:443</srv>
<srv>68.3.14[.]71:443</srv>
<srv>174.105.235[.]178:449</srv>
<srv>195.54.162[.]247:443</srv>
<srv>181.113.17[.]230:449</srv>
 ...
</servs>
<autorun>
<module name=”systeminfo” ctl=”GetSystemInfo”/>
<module name=”injectDll”/>
</autorun>
</mcconf>
```

Figure 7: Excerpt from main configuration file

This example shows C2 servers hosted on TCP port 443 and TCP port 449, but no usage of TCP port 447 which is also known to be used by Trickbot. We extracted and analyzed the IP addresses for their AS as shown in Figure 8.

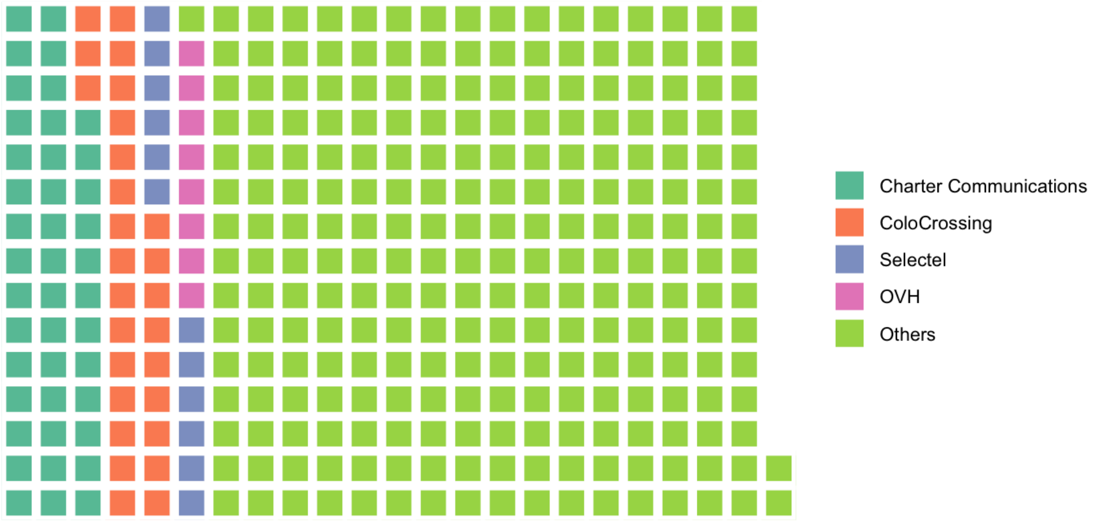


Figure 8: C2 servers in Main Configuration

To get the temporal context, we took the timestamp of the first and last appearance of a given IP address within a configuration file. Figure 9 shows that the lifetimes vary. The majority of the IPs are very short lived, while others have a lifetime of several days, or even weeks. We do not know the exact reason for this pattern, but we assume that most IP addresses are only short-lived because they are blacklisted or used for detection of an infection after a very short time thus forcing the attackers to change them quickly. Why other IPs have a longer lifetime cannot be answered, perhaps these are just testing systems that only appeared in pre-production configuration files. It would also be interesting to correlate the disappearance of IP addresses with their appearance in blacklists, but this was out of the scope of this article.

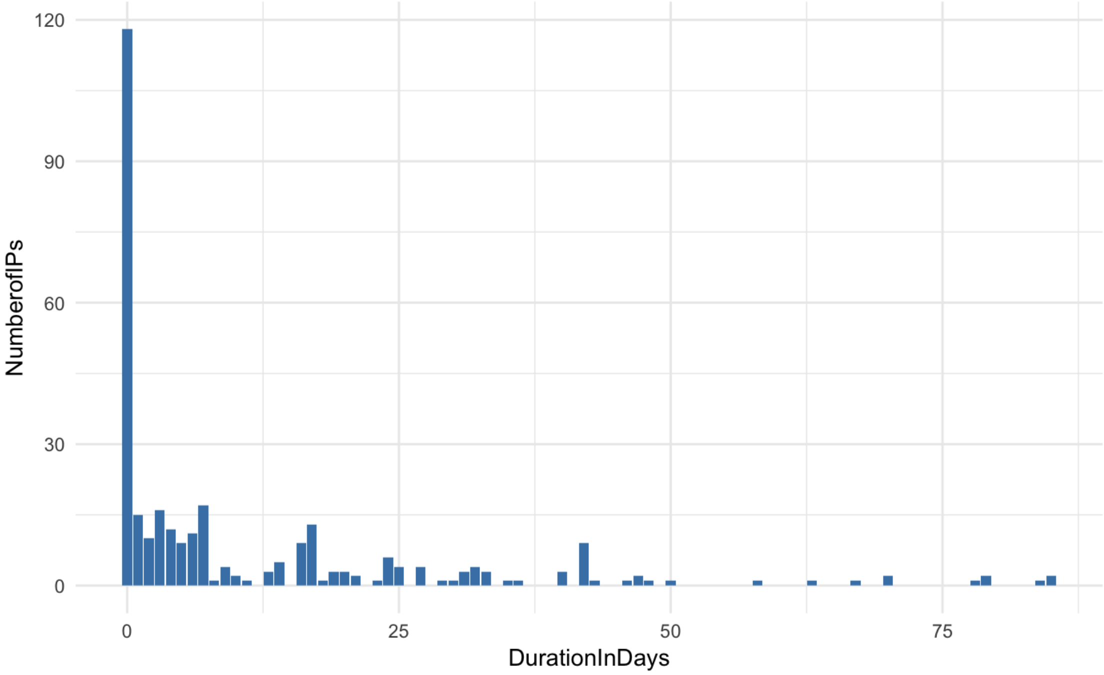


Figure 9: C2 servers in Main Configuration

A small extract from October and November shows this pattern in more detail in Figure 10.

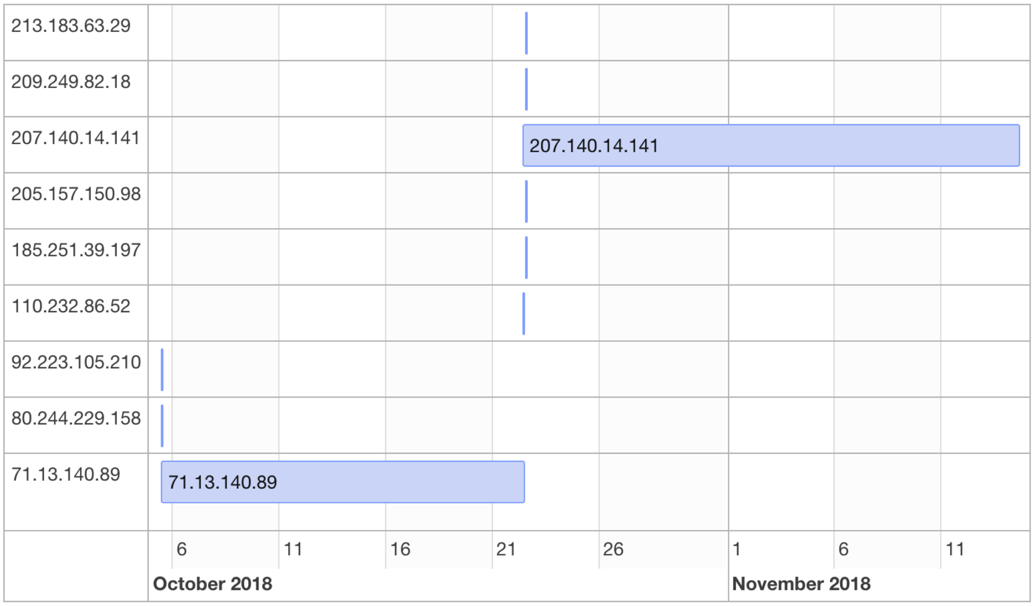


Figure 10: C2 servers in Main Configuration

3.1.2 Analysis of Static Configuration

Sinj files describe the Static Configuration of Trickbot with an example shown in Figure 11

```
<slist>
<sinj>
<mm>hXXps://www.rbsidigital[.]com*</mm>
<sm>hXXps://www.rbsidigital[.]com/default.aspx*</sm>
<nh>krsajxnbficgmrhtwsoezpklqvyd[.]net</nh>
<url404></url404>
<srv>162.248.225[.]103:443</srv>
</sinj>
```

Figure 11: Sinj Configuration Example

The parameter `<mm>` describes the target host, the `<sm>` the target URL and the `<srv>` the IP address of the server that is contacted for the injects.

We have analyzed the destination IPs of the sinj configuration files. We do not know for sure whether these are hacked systems or owned by the attackers. However, there are a few traces that may indicate the latter. If these were hacked systems, one would expect a more random distribution of registrar information which is clearly not the case as can be seen in Figure 13. Many of these IP addresses seem to have been running Nginx and are showing its default webpage. However we do not have enough evidence to either verify or falsify this.

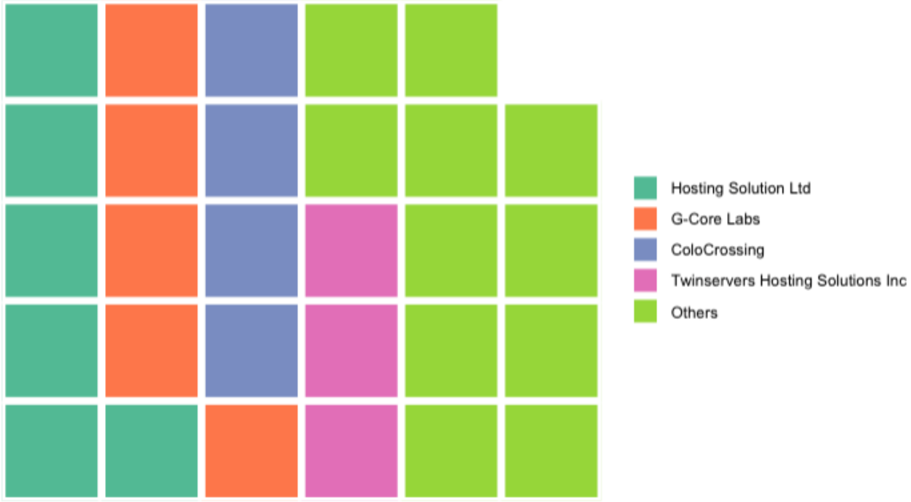


Figure 12: C2 servers in Static Configuration

It is interesting that many of the servers in our dataset are located either at Hosting Solution or G-Core Labs. Table 2 shows the IPs and their respective ASNs.

IP	Country	AS	AS Description
104.149.50[.]68	US	40676	Psychz Networks
107.174.15[.]76	US	36352	ColoCrossing
108.174.60[.]156	US	36352	ColoCrossing
131.153.19[.]122	NL	60558	Phoenix Nap, LLC.
131.153.19[.]58	NL	60558	Phoenix Nap, LLC.
154.16.195[.]34	NL	49981	WorldStream B.V.
162.247.155[.]116	US	30235	Twinservers Hosting Solutions Inc.
162.247.155[.]128	US	30235	Twinservers Hosting Solutions Inc.
162.247.155[.]155	US	30235	Twinservers Hosting Solutions Inc.
162.248.225[.]103	US	14576	Hosting Solution Ltd.
162.248.4[.]55	US	62838	Reprise Hosting
165.231.102[.]50	NL	41564	Packet Exchange Limited
185.180.197[.]117	US	14576	Hosting Solution Ltd.
185.180.197[.]35	US	14576	Hosting Solution Ltd.
185.180.197[.]36	US	14576	Hosting Solution Ltd.
185.180.198[.]147	US	14576	Hosting Solution Ltd.
185.20.184[.]74	NL	50673	Serverius Holding B.V.
192.252.210[.]19	US	46562	Total Server Solutions L.L.C.
192.99.178[.]144	CA	16276	OVH SAS
198.46.160[.]190	US	36352	ColoCrossing
198.8.91[.]37	US	46562	Total Server Solutions L.L.C.
204.155.31[.]137	US	14576	Hosting Solution Ltd.
23.94.160[.]49	US	36352	ColoCrossing
31.131.27[.]213	US	56851	PE Skurykhin Mukola Volodumurovuch
92.38.149[.]175	US	202422	G-Core Labs S.A.
92.38.149[.]45	US	202422	G-Core Labs S.A.
92.38.149[.]50	US	202422	G-Core Labs S.A.
92.38.149[.]52	US	202422	G-Core Labs S.A.
92.38.149[.]53	US	202422	G-Core Labs S.A.

Table 2: Static Configuration Country and ASN Distribution

Most of the IPs are short lived and can only be observed during one day as can be seen in Figure 13. However there are a few that last longer, but not more than 6 days which is in sharp contrast to the IPs in the main config which have some very long-living elements. Whether the 6days are merely coincidental or if these are really longer lived elements is difficult to tell. Nevertheless, the difference to the temporal pattern of the main config is noteworthy even if we cannot provide a good explanation.

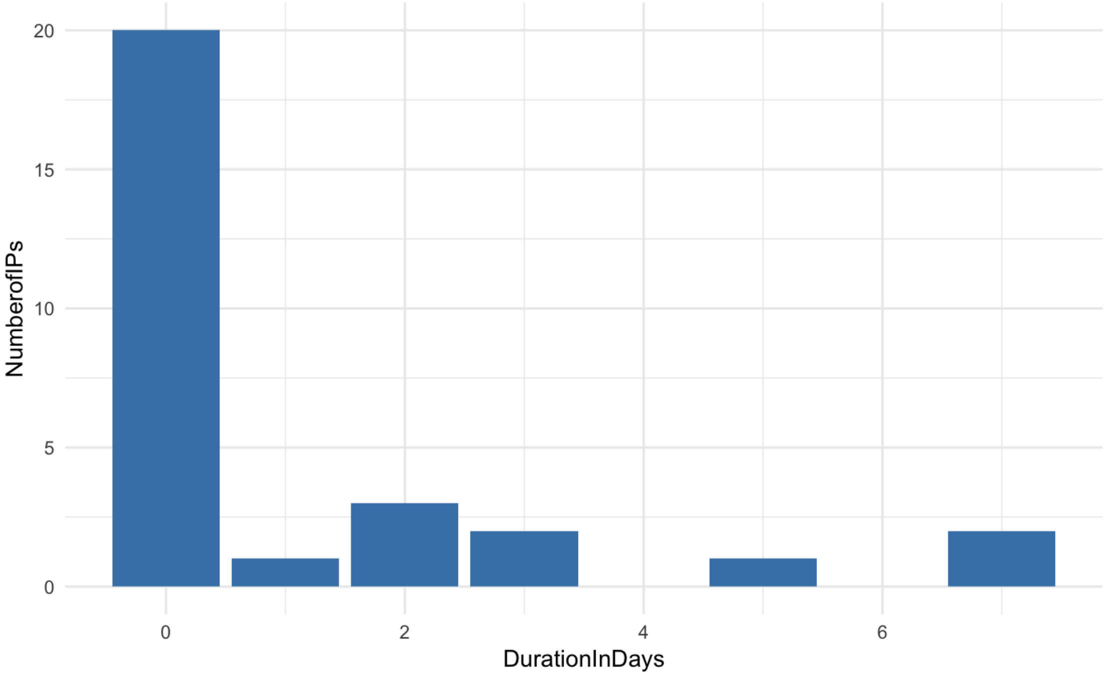


Figure 13: C2 servers in Static Configuration Lifetime

Plotted on a timeline (see Figure 14) there are three remarkable elements:

- The lifetime of IP addresses essentially does not overlap. IP addresses are rather used sequentially.
- The diagram shows a sequence of IPs used for static injects.
- Some of them were seen for several days, others were just used in one occasion, the longest period was 7 days.

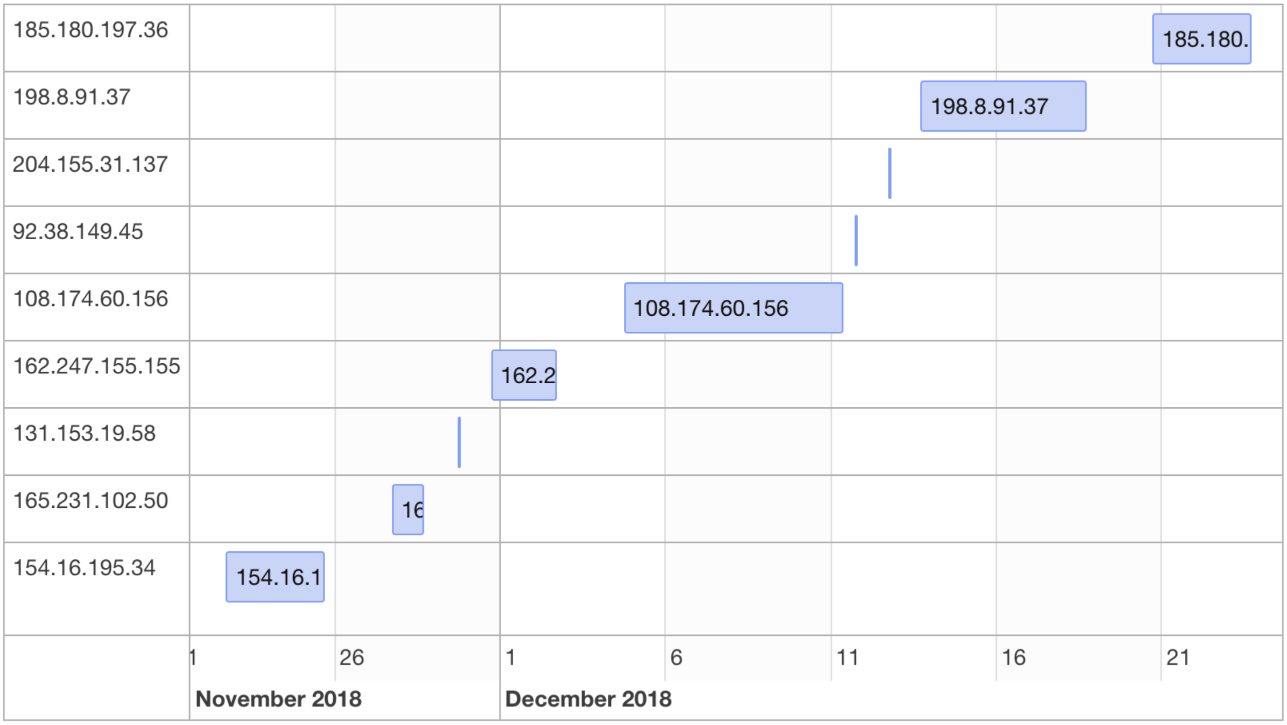


Figure 14: C2 servers in Static Configuration over Time (extract)

3.1.3 Analysis of dpost

As already mentioned, dpost configuration files contain exfiltration points for stolen credentials. The configuration files have the following format as shown in Listing 18. The format is pretty self-explanatory as it just has the handlers (C2 servers) where the stolen credentials are sent to. Interestingly this is done using plain http with the stolen data sent out in cleartext. See also the blog post by Fortinet about the pwgrab module [12].

```
<dpost>
<handler>hXXp://24.247.181[.]125:8082</handler>
<handler>hXXp://96.36.253[.]146:8082</handler>
<handler>hXXp://46.146.252[.]178:8082</handler>
...
</post>
```

Figure 15: Listing of dpost configuration file (extract)

If we plot the IP addresses over time as in shown in Figure 16, the pattern is different from the other configuration files:

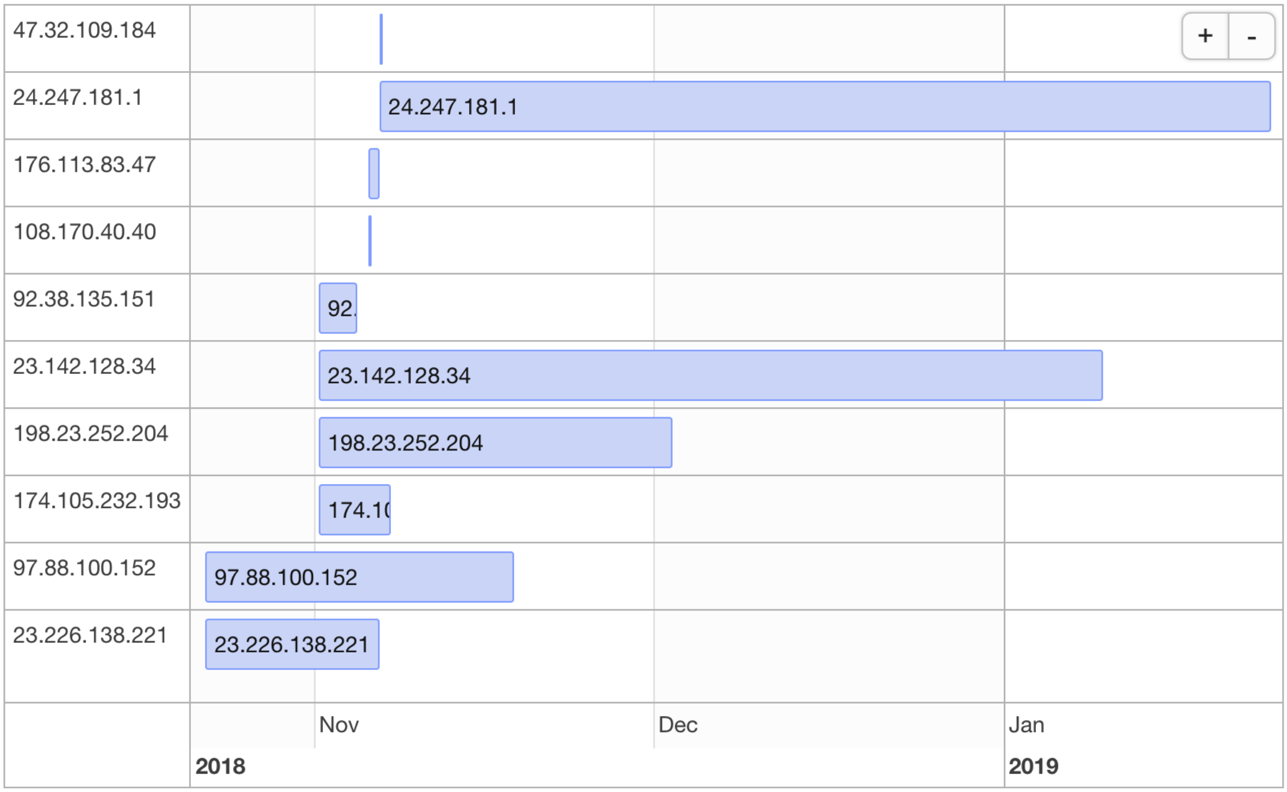


Figure 16: C2 servers in dpost config over time

We see that some IPs have slong lifetimes whereas others are only very short lived. This gets much more evident when we plot it in a histogram (see Figure 17) showing the count of IPs with a certain lifetime. Most IP addresses are short lived, meaning one day or less while some are active for a longer time, the longest being 127 days (46.146.252.178/ASN12768/ER-TELECOM-AS, RU).

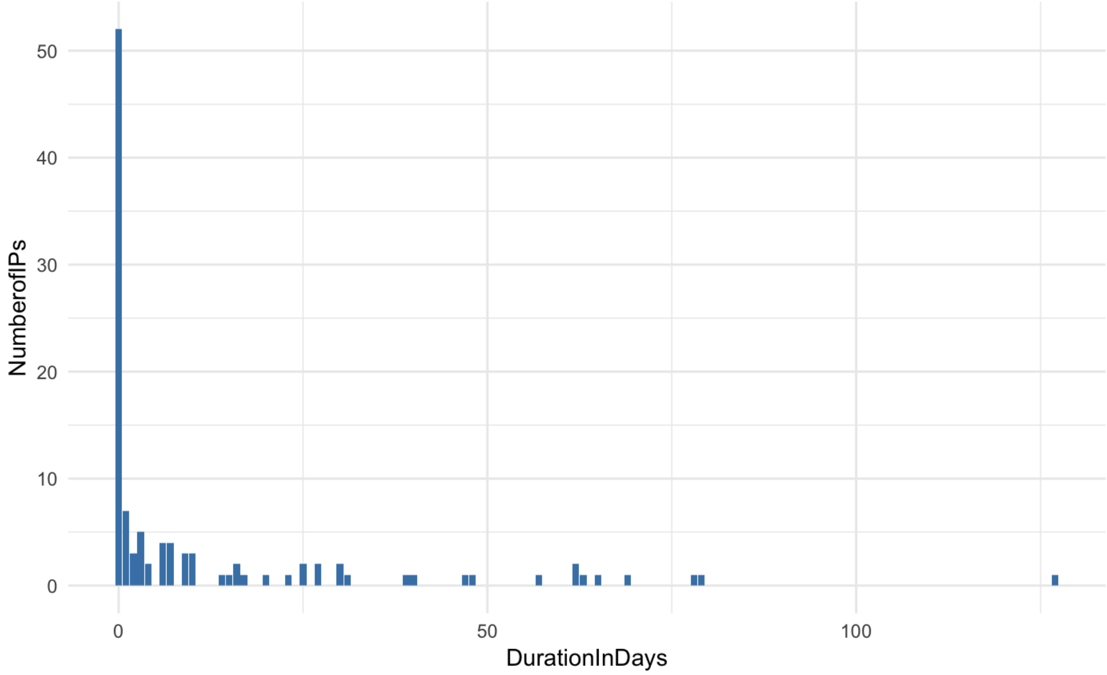


Figure 17: C2 servers in dpost config lifetime

3.1.4 Analysis of Mailconf

We harvested various mailconf files of Trickbot which configure one of the possible data exfiltration points. These are simple configuration files similar to the dpost configs. The `<handler>` denotes the C2 server where the harvested email is sent to. An example is shown in Listing 18.

```
<mail>
<handler >195.123.245[.]131:443</handler>
</mail>
```

Figure 18: Listing of dpost configuration file (extract)

Figure 19 shows the IP addresses plotted onto a timeline. One can clearly see that the IPs are seldom used at the same time but are replacing one another after lifetime of a few days to a few weeks. It seems that they have only one IP address active at a given time. As we have only monitored the actors over 4 months we do not have enough data to make a histogram showing the lifetime distribution.

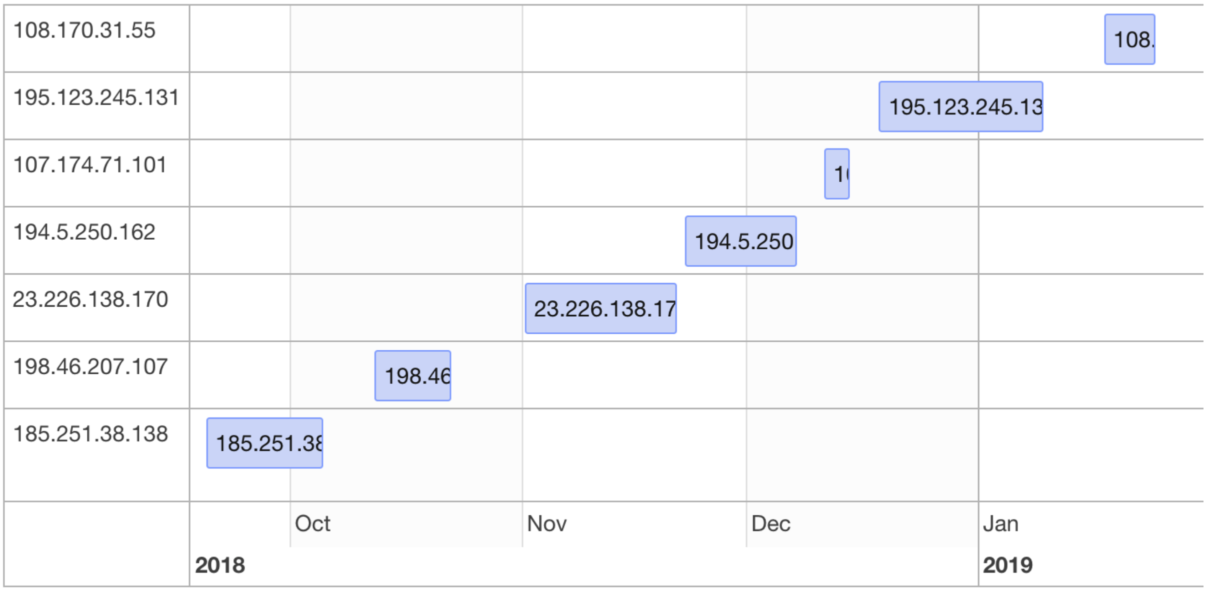


Figure 19: C2 servers in Mailconf

In Table 3 the networks and countries of these servers are listed. We can see that there is some tendency to use hosters in the US and in Eastern Europe but apart from that we have not enough data to draw any conclusions.

IP	Country	AS	AS Description	Country of AS
107.174.71[.]101	US	36352	AS-COLOCROSSING - ColoCrossing	US
108.170.31[.]55	US	20454	SSASN2 - SECURED SERVERS LLC	US
185.251.38[.]138	NL	48282	MCHOST-AS	RU
195.123.245[.]131	CZ	204957	LAYER6	UA
198.46.207[.]107	US	36352	AS-COLOCROSSING - ColoCrossing	US
23.226.138[.]170	US	8100	Quadranet	US

Table 3: Mailconf Country and ASN Distribution

3.2 Targets

For the determination of targets, we focused on the Static Configuration. In order to determine the country of the target, we looked where the web server was located assuming that most banks position their ebanking servers in the country of their most relevant customer base. We checked the result manually and made adjustments where necessary. We took the top 5 values as there is a gap between the 5th and 6th country. We have observed the following illustrated in Figure 20:

- Trickbot has a lot of targets in the US region.
- Switzerland is currently not a target (apart from big international financial institutes).
- The campaigns are spread widely and are neither targeted to a region nor done in a way that tries to adapt to the victims.

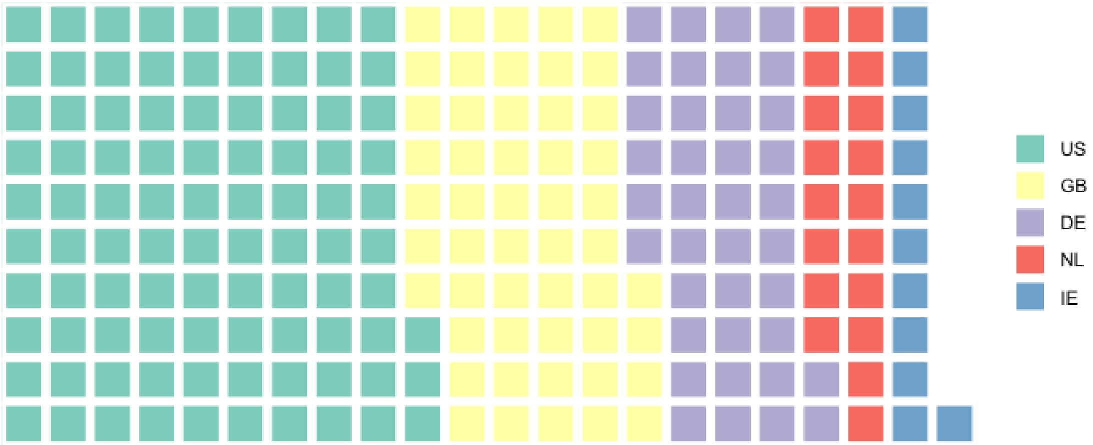


Figure 20: Targets per Country

Our results are matching with the results published by Fortinet at Botconf 2018 [8] even though there might be some minor discrepancies, probably based on differences in the method of determining the target's country.

Having a look at the temporal distribution of the target countries to time in Figure 21, we can observe a few noteworthy points:

- The number of targets remains stable over an extended time period.
- In November, we observe a steep rise in the number of targets.
- The scattered points at the beginning and the end is probably due to lack of data from our side and has no special meaning.
- Switzerland is currently not a target (apart from big international financial institutes).

By looking deeper into the data, we observed that Germany became a target on November 7th 2018: when we started the tracker in autumn 2018, we had a stable rate of 250 - 260 targets in the list. This went on until November the 7th when we noticed a large increase to 318 targets. When we compare the list of the attacked organizations we can see that nearly all of them are located in Germany. We believe that the attacker began targeting German financial institutes at this point. After that, the target list remained stable again. The decrease at the end of the measurement is most probably due to our reduced visibility because the criminals made significant changes in Trickbot. In contrast to other malware families such as Dridex, Gozi or Retefe, there seems to exist only one configuration file used for all countries.

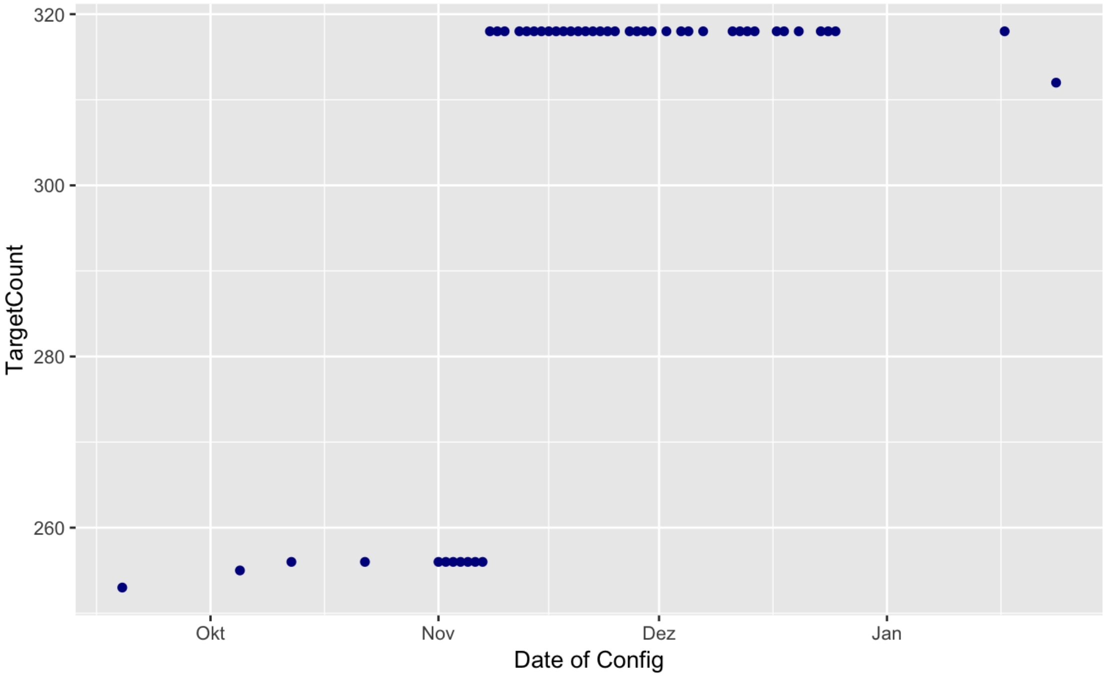


Figure 21: Number of targets over time

3.3 Conclusions Network Analysis

As we have shown, there seems to be some kind of coordination about the networking infrastructure. Even though there is a lot of uncertainty yet, we believe that the analysis proofs that the actors are actively managing their infrastructure and exchanging it on a regular base. We observe a clear sequence in the data of the static inject servers as well as in the mailconf servers. For the larger amount of C2 servers used in the main config, the sequence is less clear, there is more overlapping (as one would expect). Although we do not have that much data when it comes to dpost configuration, the pattern seems to be similar to the one seen with C2 servers from the main config. The lifetime of how long a server is being used greatly varies. However, most IP addresses are used for a very short time period but there are several IPs with a much longer malicious lifespan. We can also see that there is a preference for certain AS (Colocrossing, Charter and G-Core Labs), but as these are huge providers, it cannot be told if this is on purpose as the attackers prefer these networks or if it is merely a coincidence. One of the most important part of the work of a CERT is to determine which organisations of its constituency are at risk. This is why we try to extract configuration files that contain the target lists on a regular base. When analyzing the Trickbot target list we can see that the attackers have a strong focus on the US, Great Britain and Ireland, Germany and the Netherlands. In the analyzed configuration files we saw a sharp rise in the number of targets on November 7th when a lot of German targets were added to the target list.

References

[1] https://www.virustotal.com

[2] https://abuse.ch/

[3] Trickbot: We Missed you, Dyre. https://www.fidelissecurity.com/threatgeek/threat-intelligence/Trickbot-we-missed-you-dyre

[4] Introducing Trickbot, Dyreza's successor. https://blog.malwarebytes.com/threat-analysis/2016/10/trick-bot-dyrezas-successor/

[5] A Nasty Trick: From Credential Theft Malware to Business Disruption. https://www.fireeye.com/blog/threat-research/2019/01/a-nasty-trick-from-credential-theft-malware-to-business-disruption.html

[6] The Business of Organized Cybercrime: Rising Intergang Collaboration in 2018. https://securityintelligence.com/the-business-of-organized-cybercrime-rising-intergang-collaboration-in-2018/

[7] Spring Dragon – Updated Activity. https://securelist.com/spring-dragon-updated-activity/79067/

[8] Fortinet - Trickbot: The Trick is on you. https://www.botconf.eu/wp-content/uploads/2018/12/2018-F-Bacurio-Junior-J-Salvio-Trickbot-The-Trick-is-On-You-presented.pdf

[9] Cyberreaon - A one-two punch of Emotet, Trickbot, & Ryuk Stealing & ransoming data. https://www.cybereason.com/blog/one-two-punch-emotet-Trickbot-and-ryuk-steal-then-ransom-data

[10] Trendmicro - Emotet-Distributed Ransomware Loader for Nozelesn Found via Managed Detection and Response. https://blog.trendmicro.com/trendlabs-security-intelligence/emotet-distributed-ransomware-loader-for-nozelesn-found-via-managed-detection-and-response/

[11] GovCERT.ch - Severe Ransomware Attacks Against Swiss SMEs. https://www.govcert.admin.ch/blog/36/severe-ransomware-attacks-against-swiss-smes

[12] Fortinet - Deep Analysis of Trickbot New Module pwgrab. https://www.fortinet.com/blog/threat-research/deep-analysis-of-Trickbot-new-module-pwgrab.html