THREATX

From ThreatX Labs:

Trends in Credential Stuffing and How to Identify It

Alex Gatz, Senior Security Researcher

Contents

Credential Stuffing Defined

There are various, distinct forms of brute force-based attacks. In this report, we focus on a variant called distributed botnet-based credential stuffing.

The MITRE Corporation provides an excellent definition for credential stuffing from a single IP:

> *Adversaries may use credentials obtained from breach dumps of unrelated accounts to gain access to target accounts through credential overlap. Occasionally, large numbers of username and password pairs are dumped online when a website or service is compromised and the user account credentials accessed. The information may be useful to an adversary attempting to compromise accounts by taking advantage of the tendency for users to reuse the same passwords across personal and business accounts.*

The full definition, including additional details, can be found here. OWASP also has a similar definition here.

Credentials are often bought, sold, stolen, and leaked. Recent examples of leaks are the Uber breach announced on Sept 16, 2022 (the official announcement can be found here), and the American Airlines breach that was reported the day before on the 15th (more information can be found here). Each data breach allowed attackers to aggregate massive lists of emails, usernames, and passwords, which are then supplied to distributed networks of compromised devices in combination with carefully written software. This software, called command and control software, takes the previously breached information to conduct targeted and distributed credential stuffing campaigns by coordinating the attack across all the compromised devices at once.

When stolen information is leaked or becomes publicly available, there are sites and services dedicated to helping consumers know they have been part of a breach. One such service is have i been pwned?. Users can search using their personal information. The results from the search provide a detailed list of anywhere the personal information has been associated with breached data. Security professionals can also obtain access to leaked information. They use it to assess the security posture of their own organization. A popular example would be SecLists, which is used during ethical hacking or penetration testing.

Distributed Botnet-Based Credential Stuffing in the Wild

The unfortunate truth is that widely distributed botnet-based credential stuffing attacks occur quite often. Based on ThreatX's internal data collected over the past six months across a sample size of 13 credit unions, banks are attacked on average at least once per week. The success of these attacks varies, and any success means that someone could lose access to their account, lose funds, or, worst of all, lose their identity.

Rent-a-botnet

If an attacker doesn't have a collection of compromised devices to carry out attacks, they can rent the compromised devices instead. It has been possible to rent a variety of botnets on the dark web for many years now, and Mirai-based botnets remain some of the most widespread. Other notable types of botnets include botnets using mobile devices as well as IoT devices. The examples provided below are specifically related to mobile-based botnets, or botnets emulating mobile devices.

How is it possible for rent-a-botnets to exist? First, cryptocurrency makes it easy to pay for these services anonymously. Second, it's now incredibly easy to set up a botnet attack. It's as simple as menu selections, plugging in some values (hostname and endpoints to target), uploading a file, and scheduling a time for the attack to be initiated. Further discussion around the topic of how these botnets are rented and/or configured and paid for is beyond the scope of this paper. If there is interest in further reading, here is an article on the topic.

The pattern of a distributed botnet-based credential stuffing attack

Every case of distributed credential stuffing shares the same distinct pattern. The pattern is similar to other distributed attacks (DDoS for example):

1. There is a large spike in RPS (requests per second), far above any previous baseline of traffic. The attacks vary in whether they have an initial spike that is much larger than the plateau of traffic that follows.
2. A plateau of mostly consistent RPS follows. Some have more variability than others.
3. A sharp decrease in RPS occurs.

Note that there are several situations that will feature this pattern, and it should not be used solely as a method of attack identification. For example, a marketing campaign could result in a similar initial spike in traffic; therefore, alerting people to an initial spike in traffic may produce many false positives. Another example would be deploying new sites or integrating sites to a new platform. Each of these cases can produce a pattern similar to at least item 1 above.

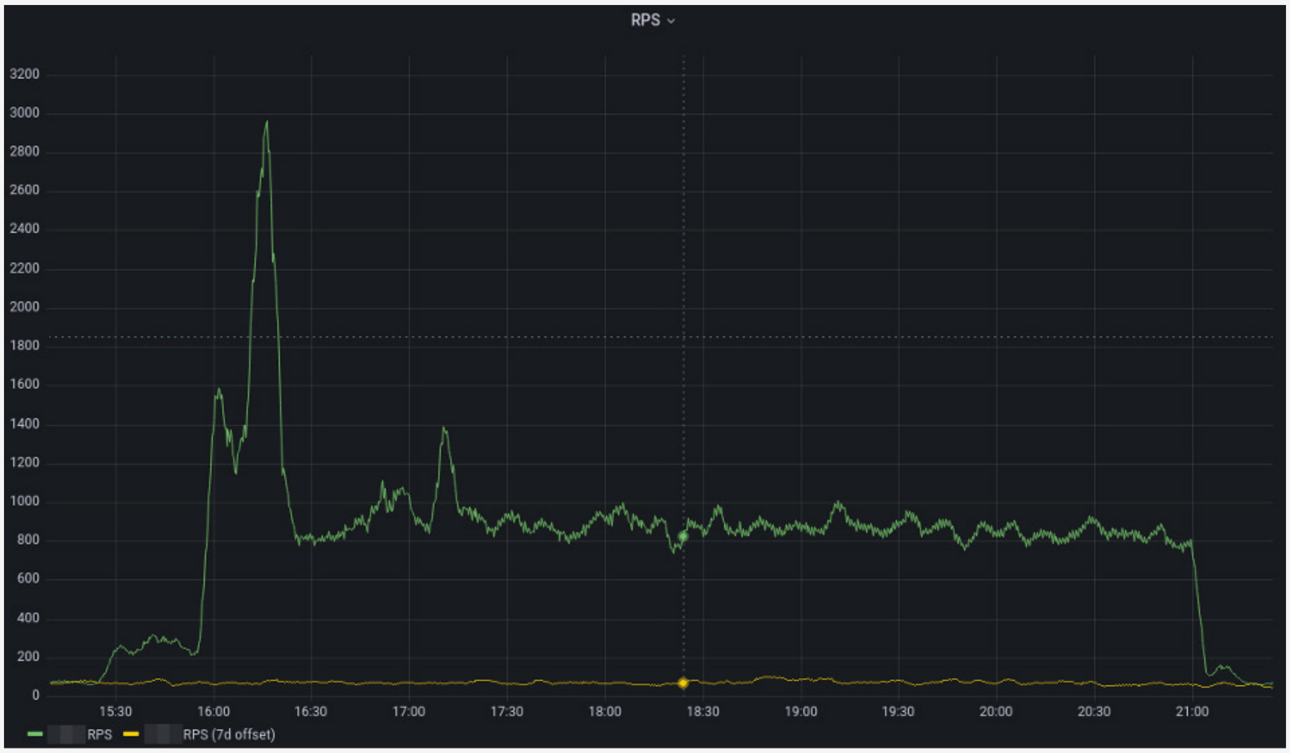
Attack examples from the wild

Both of the following examples highlight attacks from mobile-based botnets. It is difficult to identify whether the attacks originated from mobile emulators vs. compromised mobile devices; however, it was clear that the attacks originated from the same mobile application. Each attack aggressively targeted mobile authentication endpoints. We introduce the method used to come to these conclusions in the "Analysis" section.

Q4 2022 - Example of an attack from 1,062 unique IPs — a single hostname and multiple URIs were involved:

In Figure 1, there is an average baseline of ~80 RPS (requests per second) with an initial spike up to ~3,000 RPS and sustained average of ~1,000 RPS. The traffic eventually sharply returns to normal. The attack was sustained for 6 hours. The rate of change of the largest overall difference in RPS was ~4 RPS/S; in 6 minutes, the total RPS increased from 200 up to 1,600. Stated another way, the total RPS increased by 8 times its original value in 6 minutes. Overall, the attack was sustained for ~6 hours.

Figure 1: Example attack from 1,062 IPs

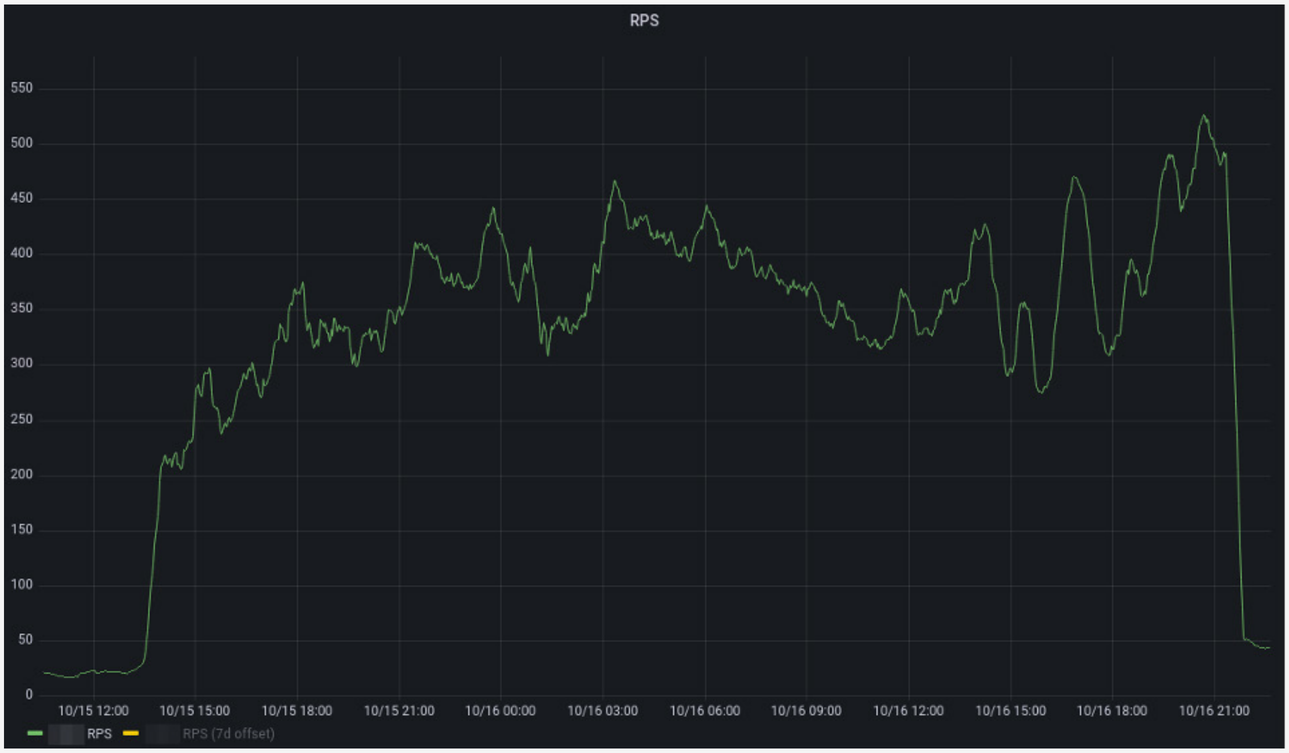


Stated another way, the total RPS increased by 8 times its original value in 6 minutes.

Q4 2022 - Example of an attack from 16,157 unique IPs — a single site and multiple endpoints were involved:

In the example attack highlighted in Figure 2, there is an average baseline of ~20 RPS with an initial spike of only 200 RPS from baseline. This attack was unique in that it had a significantly higher variability than previous examples this year. While unconfirmed currently, it is possible the increased variability in traffic rates is a new tactic. Despite the variability, the average sustained RPS was ~380, and again we see a sharp decrease in RPS once the attack concludes. This attack was sustained for exactly 33 hours, to the minute.

Figure 2: Example attack from 16,157 IPs



While unconfirmed currently, it is possible the increased variability in traffic rates is a new tactic.

How to Identify Botnet-Based Distributed Credential Stuffing Attacks

This section breaks down components involved with answering the most important question: What is the best way to identify key pieces of information that may be used to subvert these types of attacks?

Due to the significant difference in volumes of overall requests, looking at the frequency of occurrence of data elements provides clear insight into what the attackers are doing. This is because of the vast difference in the total volume of normal traffic when compared to the total volume of attack traffic; instead of having to look for small anomalies, we can just look for the largest occurring pieces of information (the attack traffic). *For some initial information on frequency analysis in general, Wikipedia provides clear and concise details here.*

You can apply a similar method of analysis to any attack where there is a significant difference in baseline RPS (or total volume) and RPS from an attack. All attacks that were evaluated using this method had at least a difference in overall RPS of 10x (as highlighted in the attack examples).

Frequency distribution

Since the exact values are often important, we will stick with viewing the data in the form of frequency distribution tables. These tables are quite simple and consist of a column of unique values with a corresponding column of total counts of those unique values.

Starting with unique counts of individual data elements harvested from requests, a sense for things like UA (user agent) rotation can become clear, and we can identify a disproportionate IP count when compared to overall requests that were made by comparing unique request IDs to the total IP count (Figure 3).

By evaluating the number of malicious requests made per IP, you can identify a request frequency that is a strong indicator of malicious activity. You can then use this rate in a counting based rule that can allow for blocking of a particular source if they exceed this static threshold. Just looking at IPs is not always helpful for several reasons, and often the more distributed the attack is, the less significant a standalone IP becomes. With the following example, we can clearly see an even distribution of attempts made across thousands of IPs (result truncated). Keeping in mind that these should be individual login attempts, anything over a few is significant (Figure 4).

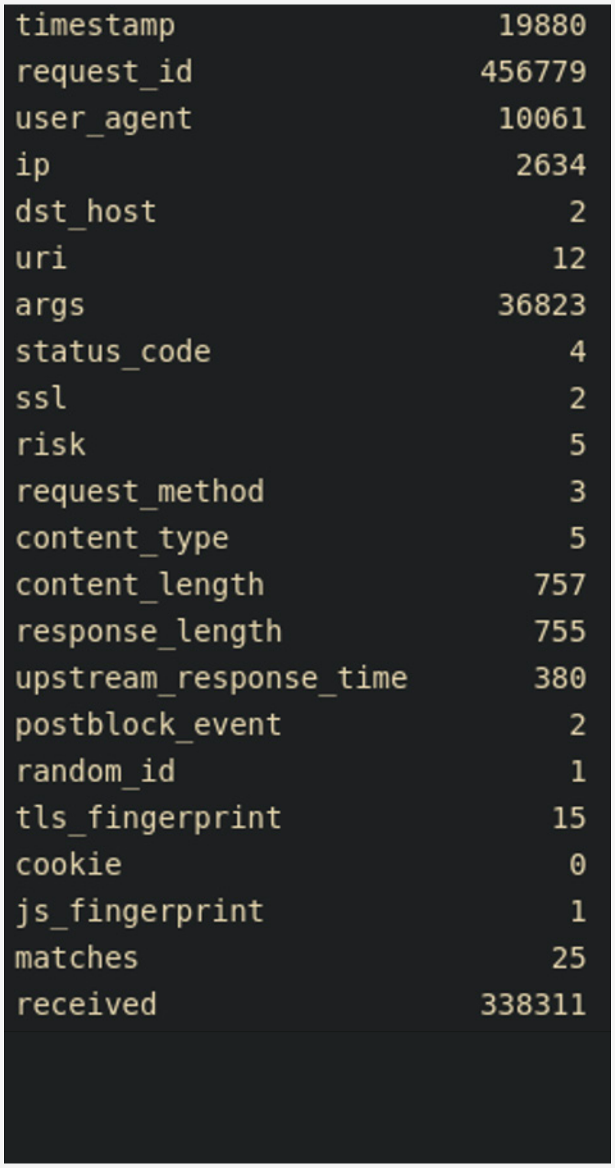


Figure 3: Frequency distribution table to identify malicious activity

```
timestamp                       19880
request_id                     456779
user_agent                      10061
ip                               2634
dst_host                            2
uri                                12
args                            36823
status_code                         4
ssl                                 2
risk                                5
request_method                      3
content_type                        5
content_length                    757
response_length                   755
upstream_response_time            380
postblock_event                     2
random_id                           1
tls_fingerprint                    15
cookie                              0
js_fingerprint                      1
matches                            25
received                       338311
```

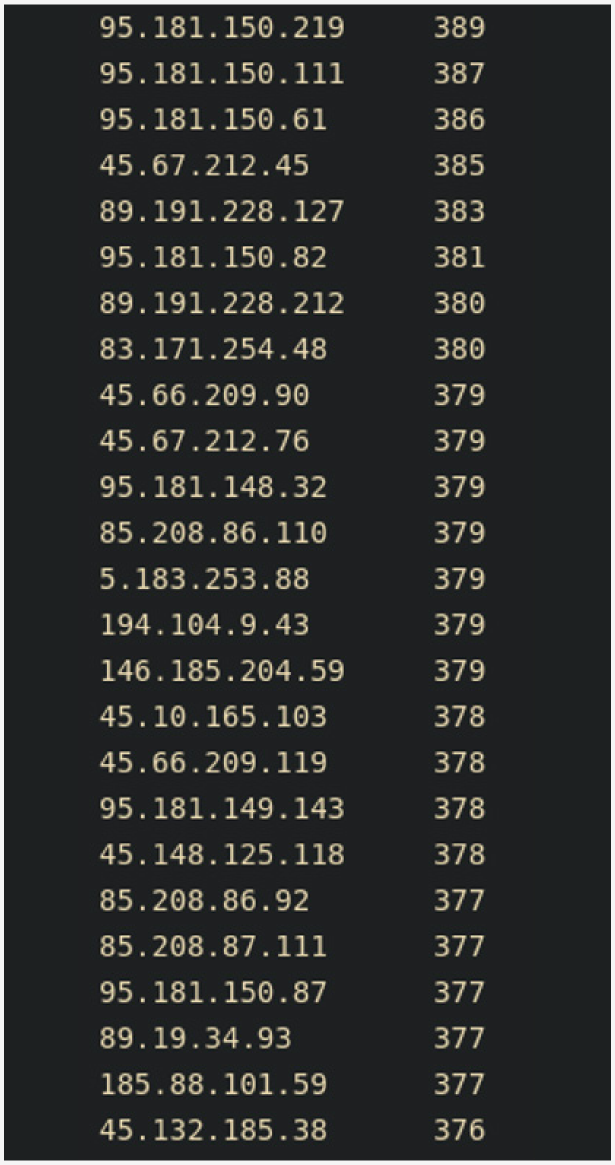
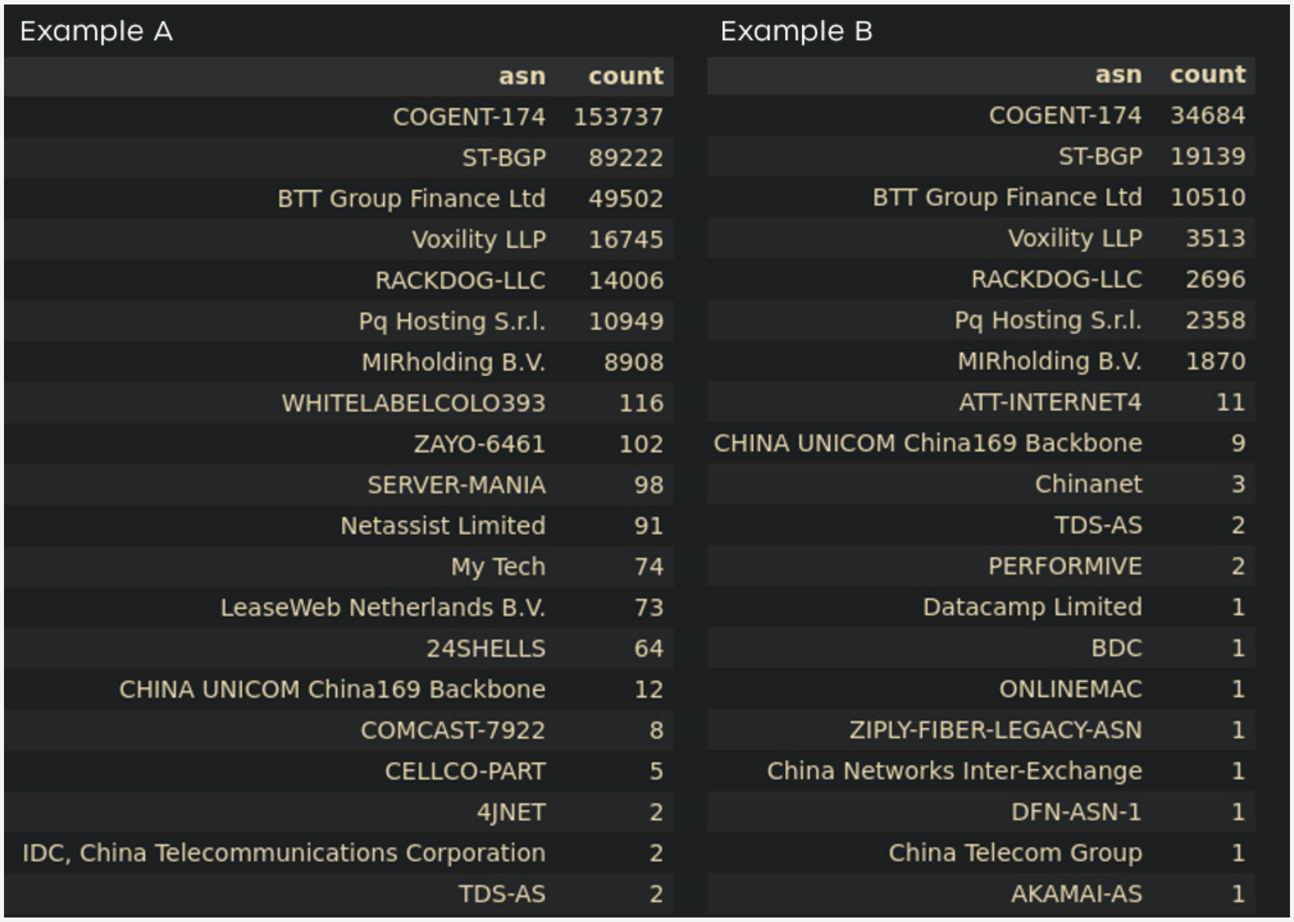


Figure 4: Frequency distribution table highlighting a distributed attack

```
95.181.150.219     389
95.181.150.111     387
95.181.150.61      386
45.67.212.45       385
89.191.228.127     383
95.181.150.82      381
89.191.228.212     380
83.171.254.48      380
45.66.209.90       379
45.67.212.76       379
95.181.148.32      379
85.208.86.110      379
5.183.253.88       379
194.104.9.43       379
146.185.204.59     379
45.10.165.103      378
45.66.209.119      378
95.181.149.143     378
45.148.125.118     378
85.208.86.92       377
85.208.87.111      377
95.181.150.87      377
89.19.34.93        377
185.88.101.59      377
45.132.185.38      376
```

It is helpful to group requests and IPs into their respective ASNs (Autonomous System Numbers). New information often presents itself when rolling up information in this way. Interestingly, these two attacks occurred within 24 hours of each other and on two separate, mostly unrelated, credit unions. The overlap between the two is clear, and the line denoting which ASNs were most used is also clear.

Figure 5: Two separate, unrelated attacks on two separate unrelated organizations



Example A

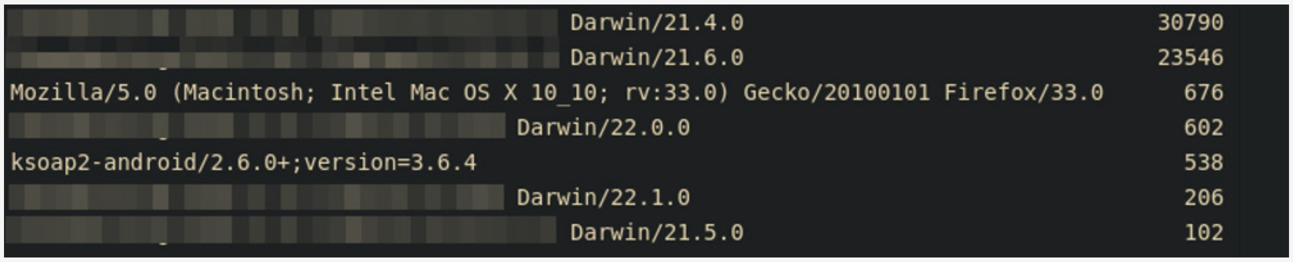
asn	count
COGENT-174	153737
ST-BGP	89222
BTT Group Finance Ltd	49502
Voxility LLP	16745
RACKDOG-LLC	14006
Pq Hosting S.r.l.	10949
MIRholding B.V.	8908
WHITELABELCOLO393	116
ZAYO-6461	102
SERVER-MANIA	98
Netassist Limited	91
My Tech	74
LeaseWeb Netherlands B.V.	73
24SHELLS	64
CHINA UNICOM China169 Backbone	12
COMCAST-7922	8
CELLCO-PART	5
4JNET	2
IDC, China Telecommunications Corporation	2
TDS-AS	2

Example B

asn	count
COGENT-174	34684
ST-BGP	19139
BTT Group Finance Ltd	10510
Voxility LLP	3513
RACKDOG-LLC	2696
Pq Hosting S.r.l.	2358
MIRholding B.V.	1870
ATT-INTERNET4	11
CHINA UNICOM China169 Backbone	9
Chinanet	3
TDS-AS	2
PERFORMIVE	2
Datacamp Limited	1
BDC	1
ONLINEMAC	1
ZIPLY-FIBER-LEGACY-ASN	1
China Networks Inter-Exchange	1
DFN-ASN-1	1
China Telecom Group	1
AKAMAI-AS	1

UAs (user agents) can be misleading due to the ease of spoofing a UA during an attack, as well as the ease of rotating through a static list of UAs as an evasion technique. In this case, the primarily used UAs lead to further validation that this attack is in fact originating from a mobile application.

Figure 6: UA analysis reveals attack origin



```
[redacted] Darwin/21.4.0                                          30790
[redacted] Darwin/21.6.0                                          23546
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10; rv:33.0) Gecko/20100101 Firefox/33.0   676
[redacted] Darwin/22.0.0                                            602
ksoap2-android/2.6.0+;version=3.6.4                                 538
[redacted] Darwin/22.1.0                                            206
[redacted] Darwin/21.5.0                                            102
```

TLS Fingerprinting

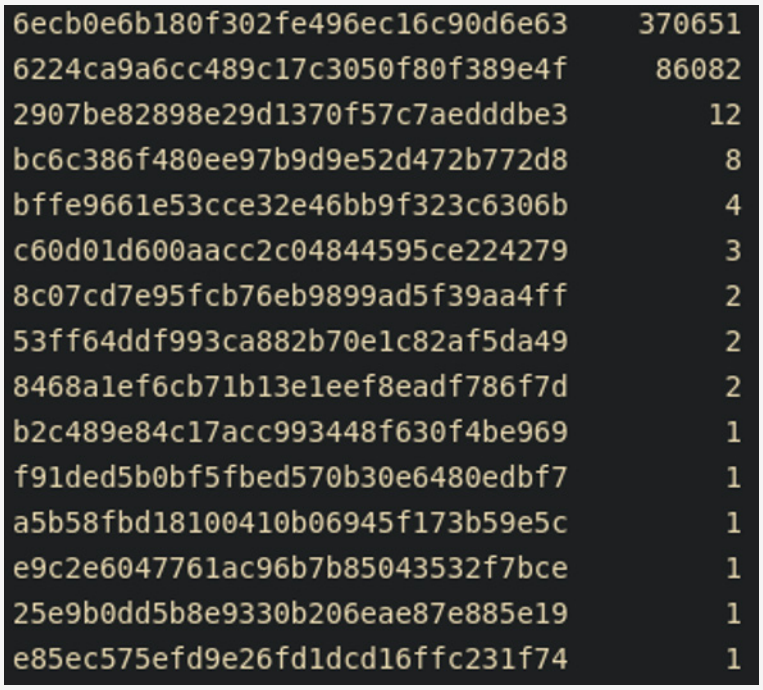
Lee Brotherston developed a technique of fingerprinting a TLS handshake in 2015. His original talk from DerbyCon can be viewed here on YouTube. The biggest problem with this technique is typically either 1, 2, or all of the following:

1. The device collecting the information is not at the edge; therefore, the device is unable to collect information directly from the client. This will be the case when a CDN (Content Delivery Network) or other IP consolidation is involved.
2. The device is at the edge; however, a mobile application is involved. Therefore, the information is mostly fixed to versions of that mobile application and provides misleading information.
3. The fingerprint is associated with a browser that is very commonly used; therefore, using the frequency of occurrence provides misleading information.

An advanced evasion technique for TLS fingerprinting is TLS fingerprint rotation. The evasion technique requires the ability to modify some components involved in a TLS handshake. The attacker may rotate through valid options of the following fields: TLS version, ciphers (cipher suites), extensions, elliptic curves, or elliptic curve point formats.

In this case, we see option 1 from the above list of problems (as also noted with the ASN example)..

Figure 7: Mobile apps evading TLS fingerprinting

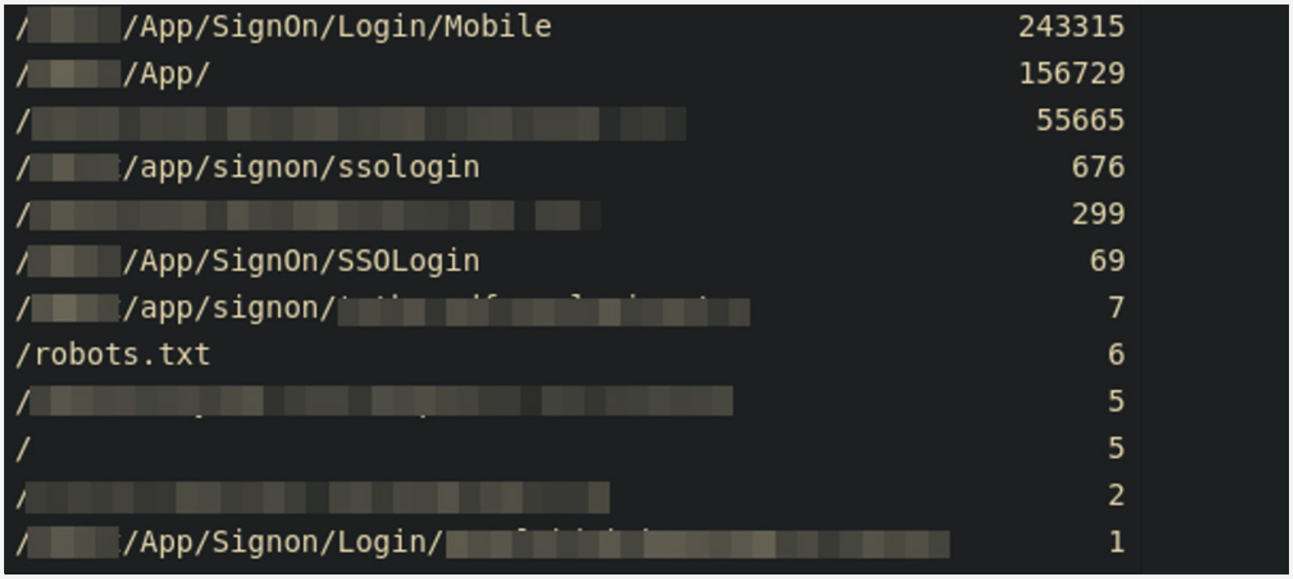


```
6ecb0e6b180f302fe496ec16c90d6e63    370651
6224ca9a6cc489c17c3050f80f389e4f     86082
2907be82898e29d1370f57c7aedddbe3        12
bc6c386f480ee97b9d9e52d472b772d8         8
bffe9661e53cce32e46bb9f323c6306b         4
c60d01d600aacc2c04844595ce224279         3
8c07cd7e95fcb76eb9899ad5f39aa4ff         2
53ff64ddf993ca882b70e1c82af5da49         2
8468a1ef6cb71b13e1eef8eadf786f7d         2
b2c489e84c17acc993448f630f4be969         1
f91ded5b0bf5fbed570b30e6480edbf7         1
a5b58fbd18100410b06945f173b59e5c         1
e9c2e6047761ac96b7b85043532f7bce         1
25e9b0dd5b8e9330b206eae87e885e19         1
e85ec575efd9e26fd1dcd16ffc231f74         1
```

Analyzing User Flow

URIs (Uniform Resource Identifiers) can be used to determine the path, or flow, the user is taking through an application. By identifying the flow of a normal user, it can become apparent when an attack is accessing an endpoint that is not typically accessed directly. As with many sites, this mobile application requires the user to load an initial page before receiving the fields to enter a username and password.

Figure 8: Unusual user flow points to an attack

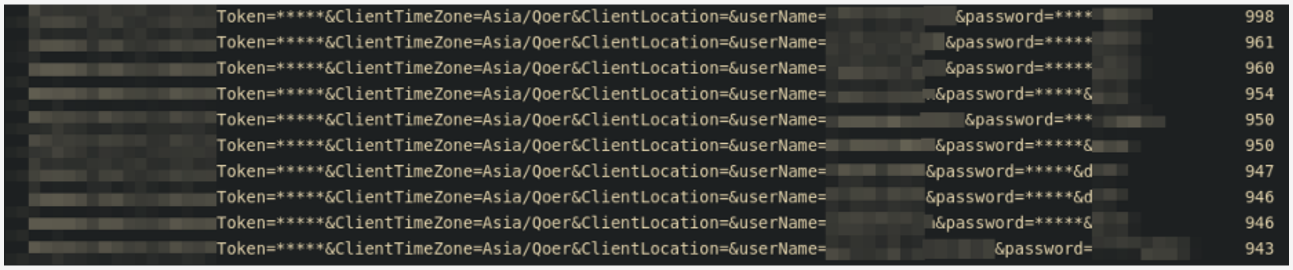


```
/[redacted]/App/SignOn/Login/Mobile             243315
/[redacted]/App/                                156729
/[redacted]                                      55665
/[redacted]/app/signon/ssologin                    676
/[redacted]                                        299
/[redacted]/App/SignOn/SSOLogin                     69
/[redacted]/app/signon/[redacted]                    7
/robots.txt                                          6
/[redacted]                                          5
/                                                    5
[redacted]                                           2
/[redacted]/App/Signon/Login/[redacted]              1
```

Arguments

Finally, we get to ARGs (arguments) and can clearly see examples of credential stuffing attempts being made with the same ClientTimeZone. In this case, when looking at the unique counts, the expectation for credential stuffing is to see an even distribution of attempts being made, assuming you pre-obfuscate unique tokens and passwords.

Figure 9: Credential stuffing attempt evident from ClientTimeZone



```
[redacted]Token=*****&ClientTimeZone=Asia/Qoer&ClientLocation=&userName=[redacted]&password=****[redacted]      998
[redacted]Token=*****&ClientTimeZone=Asia/Qoer&ClientLocation=&userName=[redacted]&password=*****[redacted]     961
[redacted]Token=*****&ClientTimeZone=Asia/Qoer&ClientLocation=&userName=[redacted]&password=*****[redacted]     960
[redacted]Token=*****&ClientTimeZone=Asia/Qoer&ClientLocation=&userName=[redacted]&password=*****&[redacted]    954
[redacted]Token=*****&ClientTimeZone=Asia/Qoer&ClientLocation=&userName=[redacted]&password=***[redacted]       950
[redacted]Token=*****&ClientTimeZone=Asia/Qoer&ClientLocation=&userName=[redacted]&password=*****&[redacted]    950
[redacted]Token=*****&ClientTimeZone=Asia/Qoer&ClientLocation=&userName=[redacted]&password=*****&d[redacted]   947
[redacted]Token=*****&ClientTimeZone=Asia/Qoer&ClientLocation=&userName=[redacted]&password=*****&d[redacted]   946
[redacted]Token=*****&ClientTimeZone=Asia/Qoer&ClientLocation=&userName=[redacted]&password=*****&[redacted]    946
[redacted]Token=*****&ClientTimeZone=Asia/Qoer&ClientLocation=&userName=[redacted]&password=[redacted]          943
```

Analysis summary

To conclude and summarize the information identified using this method:

1. The primary attack being reviewed had a total of 456,779 total malicious requests that were blocked, with a total of 2,624 unique IPs being used.
2. We see a mostly even distribution of malicious requests being rotated through ~1000 IPs.
3. When rolling up the IPs into their respective ASNs, we see that ~75% of the attack is coming from the top 7 ASNs. This highlights that aggregating data in a more macroscopic way can provide new insights and shows an overlap between two attacks within 24 hours of each other.
4. Two UAs were identified as being used primarily. Further research identified that these are in fact due to the use of a mobile application presenting these UAs. *It was not shown in the example, but there turned out to be an aligning number of UAs that were being rotated emulating a Huawei tablet.* Though it was not tested directly, it is very likely that this same method could be applied to any volumetric attack such as layer 7 DDOS or otherwise.
5. Two TLS fingerprints, the fingerprints of the mobile application, were used during the attack. This further strengthened our initial suspicion that a mobile application was used directly in some way.
6. The unique counts of each URI highlight that not only is this attack targeting mobile endpoints, it is specifically targeting the mobile *login* endpoints.
7. Reviewing the ARGs in this case provided confirmation that this was in fact credential stuffing.

Looking at the big picture, we can conclude that not only was this credential stuffing, but this was also an attack using either mobile emulators, compromised mobile devices, or mobile device farms. We can also conclude that the attack was distributed across 2,624 IPs. Therefore, we have created a repeatable process that is a quick and straightforward way to identify key pieces of information from distributed botnet-based attacks that can be used to subvert these attacks. Though it was not tested directly, it is very likely that this same method could be applied to any volumetric attack such as layer 7 DDOS or otherwise.

Final Thoughts

Large and well distributed botnet-based credential stuffing attacks occur every day. They are typically unseen by the general public, and they are abusing the reuse of usernames, emails, and passwords. Attackers do not need to harvest their own devices anymore; they are sharing already compromised devices for a price via rent-a-botnet services. Use a password manager, enable two-factor authentication, and follow the strategies from above when going to investigate these attacks.