Automation Attacks at Scale - Credential Exploitation

Will Glazier
Stealth Security, Inc.

Mayank Dhiman
Stealth Security, Inc.

Abstract

Automation attacks are currently plaguing organizations in industries ranging from fina ncial and retail, to gaming and entertainment. These attacks exploit stolen credential leaks, use black market & custom attack toolkits, and massively scalable infrastructure to launch widely distributed attacks that are extremely difficult to detect, let alone attribute. In this paper we will inform the audience of the scale of this problem, discuss a detection methodology to counter these attacks, and walk through real-world examples of how attackers created and monetized the distributed infrastructure they require to launch these attacks.

1 What are Automation Attacks at Scale?

Credential exploitation attacks are a new class of ATO (account-take-over) attacks, where the goal is to test known leaked credentials, *at scale*, against different targets (usually, in parallel). Since people reuse passwords; if enough leaked credentials are tested on enough websites, many of them will lead to successful account compromises.

The problem of **credential reuse** is by nature, a by-product of expecting users to create and remember strong passwords. Additionally, users are also advised against reusing existing passwords. The rationale is that leaked credentials from one website can lead to compromise of accounts on other websites where the same credentials are reused.

In practice, however, each user has too many accounts. This makes it hard for users to construct and remember strong passwords for each account. Proposed solutions like password managers (even though they sound promising), are not widely adopted and are themselves riddled with security issues. Hence, credential reuse is an ever-present problem and, users are reusing the same credentials across multiple websites. According to a study from Microsoft Research, a typical user has ˜6.5 different passwords for ˜25 accounts [12]. This study is based on data from 2006, and the current extent of password reuse is unknown.

The rise of publicly accessible **credential dumps** has aggravated the credential reuse problem. Databases containing user credentials, PII (personally identifiable information) and other private information are publicly dumped by attackers upon successful compromise of a website or an organization. These credential dumps are extremely handy for attackers intending to launch credential exploitation attacks as these are known "working credentials" of users identified in the breach.

This is worrying both for users and *all* organizations trying to protect user accounts as a user's account security is directly dependent on the security of each organization where the user reuses the same set of credentials. Hence, data that is compromised in what appears to be a low-impact breach like a video gaming account or online forum, can have vast ripple effects on industries with sensitive data like banking, healthcare and retail. Additionally, with each new compromise, attackers will likely test the newly leaked credentials against accounts on high profile websites.

The credential exploitation attack methodology relies heavily on the ***law of large numbers***. Specifically, if millions of these credentials are tried out on a large number of websites, inevitably many of these attempts will end up being successful logins. If an attacker can try 1,000,000 credentials, and successfully login at a paltry 0.01% success rate, they will have compromised 100 accounts in one attack - yielding them potentially substantial profits depending on the type of account compromised.

The attacks themselves are also easily parallelizable; as different sets of credentials can be tested against the same target or across different targets in parallel. Therefore, these attacks tend to be launched on a large scale.

In this paper, we will explore this problem in more depth. Section 2 introduces the relevant background work. In Section 3, we explore the attacker's perspective and delve deeper into how they are acquiring the tools as well as infrastructure to launch such attacks in the wild. In Section 4, we explore the defensive side. Motivated by the attacker behavior observed in the wild, we propose a novel detection methodology. We then tie these two together in Section 5.

2 Background

Credential verification attacks are an evolution of *credential brute-force attacks* where the goal is to break into an account by testing different set of credentials. One popular variant is a dictionary attack where all dictionary words (and their variants) are tried as passwords. Rainbow tables often attempt to decipher the plain text password from a hashed password, as password files are often (but not always) stored in some form of hashed way. Such attacks are still extremely popular against network services where many vulnerable services are known to run with known or easily guessable usernames and weak passwords.

An extensive amount of research has been conducted into the password reuse problem and the habits of typical users online. Florencio et al. monitored password habits of half a million users over a three-month period. Their study revealed that the average user has 6.5 passwords, each of which is shared across 3.9 different sites. Furthermore, Bonneau et al. estimate that 43%-51% of users reuse the same password across multiple sites. They also identified a few simple tricks users often employ to transform a basic password between sites which can be used by an attacker to make password guessing vastly easier [9]. They developed the first cross-site password-guessing algorithm, which is able to guess 30% of transformed passwords within 100 attempts compared to just 14% for a standard password-guessing algorithm without cross-site password knowledge. Jakobsson et al. developed a password strength meter based on how users create passwords [13].

The password reuse problem renders even the most stringent password requirements – which are often burdensome – useless, as the effects from one data breach will radiate throughout the internet. Bauer et al. showed that an increase in complex password requirements like special characters, typically caused users to create passwords that fell into easily guessable patterns [18]. An example of this would be users including a single digit at the end of a password or stringing multiple complete words together. These behaviors drastically increase the chances for attackers to guess passwords, as each character in a string is not independent of the others, changing the probabilistic calculations.

Additionally, we looked into prior research on how attackers scale their attack infrastructure. Feamster and Ramachandran focused on infrastructure to send spam [15]. They found that IP ranges sending spam tend to be persistent and that spam is not evenly distributed throughout different infrastructure. This is consistent with our findings that infrastructure is often used in groups or network ranges, leaving volumes from individual IPs lower allowing them to fly under the radar. Also, Feamster and Ramachandran find that detection methods looking at aggregate behaviors will go much further than looking at individual IPs, as network level characteristics are less malleable than content level characteristics.

We also looked into documented cases of BGP hijacking of IP ranges. This technique allows attackers to commandeer the route to infrastructure that has higher reputations, thereby providing spammers (or other malicious actors) more cover. In this case, it happened to a Swiss government IP range. The practice is not limited to simple spammers or financially motivated actors, as the Turkish government would use BGP to hijack popular DNS servers to block traffic they found politically unsavory [19].

Other observations of attacker infrastructure include the myriad of studies on the Mirai botnet, and other large IoT botnets which were behind the DYN DDoS attack in late 2016 [16, 14]. As Schneier discusses, the motivation for the use of these botnets can vary widely, and portions of them can be rented out to different actors aiming to accomplish different goals [16]. These can include massive network level DDoS designed for extortion, protest or an act of cyberwar. Also, they may include application level DDoS designed to limit availability of a service for monetary purposes.

The problem of credential exploitation/verification attack and the associated tools has been described earlier [4, 8]. Troy Hunt mentions how attackers are using tools like Account Hitman along with the leaked credential dumps to launch such attacks [8]. However, there isn't a lot of previous work on this specific problem.

3 Attacker's Perspective

We first outline the key requirements to launch a successful credential exploitation attack:

1. A set of stolen credentials
2. Attack tools configured for a particular target
3. Compute power
4. Ability to rotate over multiple IP addresses

5. Ability to bypass any deployed defenses

3.1 Stolen Credentials

Stolen credentials aren't very hard to obtain given the plethora of publicly accessible credential dumps [8].

We analyzed large samples of traffic to find patterns in stolen credential use. We found that for credentials tried from a particular tool, 32% appeared in Myspace, 32% in RiverCityMedia, 25% in LinkedIn and 22% in Adobe breach. Each username tried appeared in an average of 3.4 credential dumps. For SentryMBA, over 42% of credentials appeared in RiverCityMedia, 23% in Myspace, 19% in Adobe and 17% in LinkedIn.

For comparison, we analyzed traffic that this retailer had called legitimate. For these requests 42% of usernames tried did not appear in any credential dumps. 27% appeared in RiverCityMedia and 15% appeared in LinkedIn. Furthermore, each username tried appeared in an average of only 2.6 credential dumps which is substantially lower than the known attack tool traffic.

3.2 Tools

Brutus, released in early 2000, was one of the first credential brute force tools which could target HTTP based authentication flows [17]. Subsequently, many advanced tools like nCrack, Hydra, and Medusa were written in early-mid 2000s [6, 1]. Most of these were command line tools written to target specific protocols and simple web authentication flows. Hence, most of these tools lack the ability to deal with modern authentication flows. For example, they are unable to parse login forms to extract CSRF tokens (cross site request forgery) before making form requests.

This gap was filled by cracking tools which originated from the cracking underground like Sentry MBA, Account Hitman, Vertex, and others [4, 7]. Out of these Sentry MBA seems to be the most popular attack tool as it is quite sophisticated and is highly configurable. The origins of Sentry MBA are unclear. However, it started circulating in the underground cracking forums in 2012. The last known version of Sentry MBA (version 1.4.1) was likely written in late 2011.

Attackers have also developed an advanced repertoire of attack tools that can be easily re-configured to attack any given target. These tools and their configurations can be obtained freely or purchased for paltry sums of money - an average of $2.73 - on underground forums [11].

3.3 Compute Power

3.3.1 Cloud hosting providers as proxy farms

Open proxy feeds have been around for a long time. For attacker, one of the dangers of proxies listed in those feeds is that the reputation of those IPs, and the ISPs and Organizations that host them, quickly deteriorate and appear on blacklists fed into many security solutions. Often, those providers are located in offshore locations, have a low volume of legitimate traffic, or are otherwise easy for defenders to flag as suspicious.

Attackers have adapted to leverage the existing good reputations of cloud providers like AWS, Azure, and other well-known organizations. These cloud services easily provide the ability to spin up multiple virtual instances with different IP addresses and launch widely distributed attacks in a short period of time. In our dataset, which we collected over the course of 3 months at a large United States retailer, we observed approximately 11.34% of all traffic comes from cloud providers. The top providers by volume include – QuadraNet (3.5%), Choopa LLC (2.9%), OVH (2.8%), Linode (1.4%) and Amazon (.25%).

We also analyzed this same dataset in smaller batches to understand common traffic patterns on any given day. During one such representative day in September at a United States retailer, the traffic was spread across hundreds of IP addresses - with varying volumes by provider. For instance, attacks from Linode were spread across 1003 IPs while attacks from OVH only came from 95 IPs. Attackers from OVH sent twice as much volume of traffic than Linode despite having fewer IP addresses.

Conservatively, less than 2.5% of all traffic from these cloud providers was legitimate during this day-long period. More popular cloud providers like Amazon, Microsoft and Google fared slightly better, with 15% of their traffic being legitimate. Fig. 1 represents a snapshot of traffic from Amazon during this day, all of which was marked bad.

3.3.2 IoT Botnets and Open Routers

A recent trend for attackers has been to compromise IoT devices and use them as a jump board to launch other malicious activities. Here we present data from an attack we observed during December 2016 through January 2017.

In this attack campaign, we witnessed a credential exploitation attack using known attack tools, routed through IP addresses that belonged to compromised devices. There were over 578 IP addresses that came from 119 different ISPs/Organizations and 39 different countries. We attempted to reach these IP addresses via the public internet on common ports like 80, 8080 and 443. All of these devices were reachable on at least one of

Figure 1: Details of an attack cluster using AWS as a proxy to launch the attack. Attackers will try to blend in with legitimate traffic from high reputation cloud providers, but signatures of attack tools and the presence of stolen credentials are suspicious. This attacker rotated through over 364 User Agent strings from 493 IP addresses.

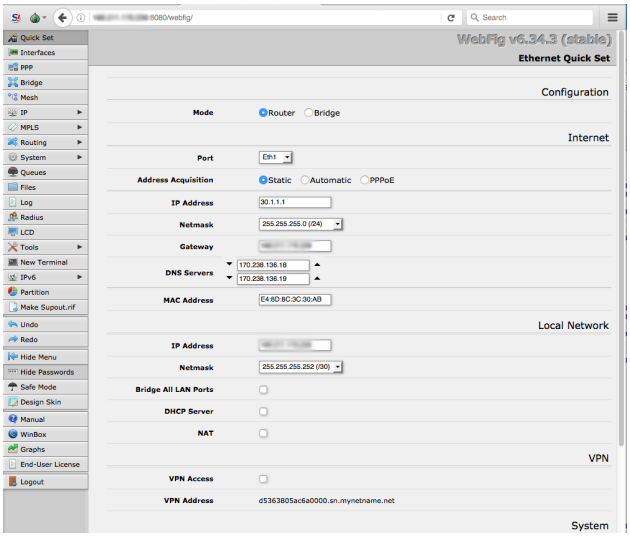

Figure 2: Publicly accessible router with no authentication. This router was being used to carry out credential exploitation attacks.

those ports, and we examined the responses to determine the type of device.

Of the device types, there were at least 175 IPs open home routers, 10 DVR or camera systems, 10 open web servers (including Apache Tomcat), 4 webcams, and one SCADA system. Of all these devices, approximately 25% belonged to the Mexican residential ISP Telmex. Other common ISPs included VDC (Vietnam), Claro Domincan Republic, Link Egypt, Telefonica del Peru, TE Data (Egypt), and Qubee (Pakistan). These ISPs owned anywhere from 1%-3% of IPs belonging to these devices. The most popular types of routers included Mikrotic (v6.36.4 and v6.34.3), Huawei HG532 and HG8245H, GPON Home Gateway and D-Link routers. Other interesting devices included an Intelbras camera system and an Advantech WebAccess browser-based HMI/SCADA software system.

Many of these devices were trivial to exploit for attackers. We observed many devices with web admin pages sitting open on HTTP ports with no controls for brute force attacks. Default credentials for these devices are easily accessible online with only a Google search. Additionally, we found long existing exploits for the D-Link, Mikrotic and Huawei routers used in these attacks [3, 5]. These included a directory traversal attack against Huawei discovered in late 2015, and an unauthenticated remote command execution vulnerability against D-Link routers [2].

For a few of these devices who had open admin portals, we were able to observe evidence of a third party attempting an SSH brute force login attack onto that device. This displays the tug-of-war game between attackers where they will attempt to corral compromised infrastructure, in order to corner the market for resale of "proxy" IP addresses.

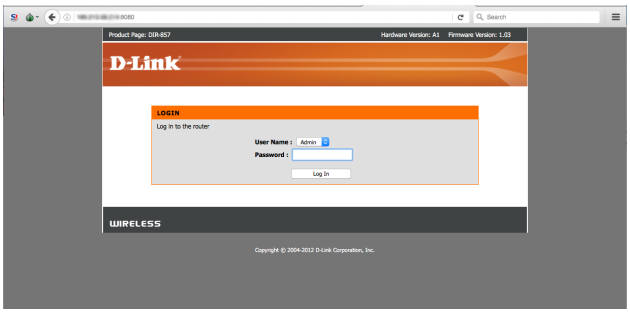

Figure 3: Publicly accessible D-Link router which was responsible for credential exploitation attacks.

3.4 Artificially Geo-Distributed Proxy Farm

We also discovered an ***artificially geo-distributed proxy farm***, which was run by one actor who goes by the name "Ilya Trusov Igorevych". This actor was able to trick many third party geolocation providers like Maxmind such that the geolocation services presumed that many of the IP networks appear to come from countries where they didn't belong. The attacker would fool geolocation services like Maxmind by spoofing the ISP and Org locations in DNS.

More specifically, the actor owned more than 40,000 IP addresses, mostly located in Kaluga (Russia) and the US. However, according to geolocation services like Maxmind, these IP addresses were located in 61 different countries. This allows attackers to blend in amongst the traffic from a target's most popular country for legitimate users. The actor can "lease" IPs in different IP spaces which would be geo-located to different countries, hence, behaving as a geo-distributed proxy network.

Maxmind According to Maxmind, IP addresses from more than 150 /24 CIDR ranges were coming from 61 different countries and 84 different cities. Over 35% were registered to Russia, 16.3% registered to the United States, 11.5% to Ukraine, 4.5% to the Netherlands, 2.75% to both Germany and the UK, and 1.4% each to Latvia, Czech Republic, Kazakhstan, Poland, Canada, Turkey, Sweden, and France. Some of the frequently appearing ISPs and Organizations include:

1. Petersburg Internet Network ltd. - 38.7% (ISP)
2. Transit Telecom LLC - 15.6% (ISP)
3. Atomohost - 15% (ISP)
4. Link Telecom LLC - 7.5% (ISP)
5. PP Trusov Ilya Igorevych - 4.8% (ISP)

```
organisation:     ORG-TII6-RIPE
org-name:         Trusov Ilya Igorevych
org-type:         LIR
address:          Moscow Street 258, office 16
address:          248021
address:          Kaluga
address:          RUSSIAN FEDERATION
phone:            +79533100064
mnt-ref:          RIPE-NCC-HM-MNT
mnt-ref:          MNT-DEPO40
mnt-by:           RIPE-NCC-HM-MNT
abuse-mailbox:    abusemail@depo40.ru
descr:            Kaluga Data Center Depo
created:          2013-11-08T11,14,03Z
last-modified:    2017-03-29T11,44,15Z
source:           RIPE
e-mail:           iluxa85@inbox.ru
abuse-c:          AC28994-RIPE

person:           Trusov Ilya Igorevych
remarks:          Depo Data Center Kaluga
address:          248021, Russia, Kaluga region, Moscow Street 258, o
phone:            +79533100064
nic-hdl:          TII0-RIPE
e-mail:           noc@depo40.ru
abuse-mailbox:    abusemail@depo40.ru
mnt-by:           MNT-DEPO40
created:          2013-07-19T09,32,30Z
last-modified:    2017-03-26T13,29,22Z
source:           RIPE
```

Figure 4: The Whois registration data of one network under the control of the actor hosting and managing this artificially geo-distributed proxy farm.

6. DepoDataCenter - 25% (Organization)
7. net for depo40.ru - 25% (Organization)
8. Atomohost - 11.5% (Organization)
9. Petersburg - 9.5% (Organization)

Traceroute Experiment In order to validate our hypothesis about the spoofed geolocation data, we conducted a distributed traceroute experiment on a random sample of IPs from each network. This actor took many of their larger /21 or /22 CIDR ranges, and broke them down into subsets of /24 and /25 networks, each purporting to come from different countries. We conducted the traceroute experiment through 6 independent trials. Each trial had random samples of 3 different IPs in each subnet. The traceroute origination locations were: 1) The United States (east coast) 2) Switzerland 3) Russia (Moscow) 4) Japan

The results of our experiment validated our hypothesis, and illuminated which ASNs in particular were responsible for most of the "spoofed" traffic. Of traffic claiming to come from the United States on MMDB, 48% came from Russia, while 47% actually came from the United States, and an additional 5% from the Netherlands. Of all traffic claiming to come from Russia, 74% actually did, while another 18% came from the United States. Traffic claiming to come from Germany was evenly split between actually originating in Russia and the United States, while 75% of traffic claiming to come from UK, Latvia, and Canada actually originated in Russia. All traffic appearing to come from the 53 other countries in these networks actually originated in Russia.

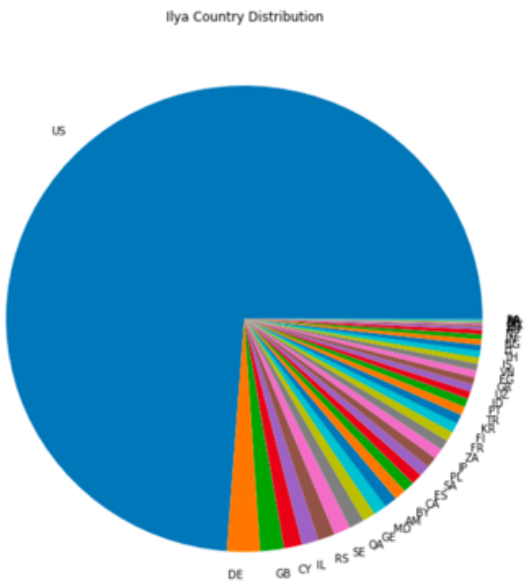

Figure 5: The country distribution of traffic (by volume) from this proxy farm hitting a United States retailer. The attackers try to hide amongst the United States traffic - in reality all this traffic is originating from Russia.

.200.0/21	DEPO-NET	net for depo40.ru	RU
.200.0/25	CapeTownNet	South Africa Network	ZA
.200.128/25	SeoulNetwork	South Korea Network	SK
.201.0/25	CairoNet	Egypt Cairo Network	EG
.201.128/25	Jakarta Indonesia	Jakarta Indonesia Network	ID
.202.0/25	IslamabadNet	Islamabad Network	PK
.202.128/25	Hanoi Network	Hanoi-Vietnam-Network	VN
.203.0/25	SG-Net	Singapore Network	SG
.203.128/25	Las Vegas Network	Nevada Las Vegas Network	US
.204.0/25	TehranNet	Iran Network	IR
.204.128/25	OgdenNet	Ogden Utah Network	US
.205.0/25	VancouverNetwork	Canada Vancouver Network	CA
.205.128/25	ParisNetwork	France Network	FR
.206.0/25	AlaskaNetwork	Network Alaska US	US
.206.128/25	Helsinki Network	Helsinki Finland Network	FI
.207.0/25	Miami Network	Network Infrastructure to Miam	US
.207.128/25	SwedenNetwork	Sweden Network	SE

Figure 6: A snapshot of some IP networks owned by this actor. These networks demonstrate the pattern of larger /21 or /22 networks subdivided into /25 networks that purport to come from different locations around the world.

Other results that solidified our hypothesis of the geolocation spoofing was the identical traceroute round traceroute round trip times (RTT) for IPs registered to 15 different cities across the United States. These cities ranged from Honolulu to Anchorage, Baltimore and Los Angeles. All of the IPs from these cities took an average of 3.6ms to ping from Moscow and 135ms to ping from the east coast of the United States, which is consistent with average ping times between those locations.

We were able to effectively separate out the traffic actually coming from Russia and the United States by the ASNs. 3 ASNs that routed approximately 70% of the IPs we analyzed come from Russia - those are ASN 50896, 200557 and 44050. One ASN that routes approximately 20% of the IPs we analyzed comes from the United States - that ASN is 29802. And finally, there were 6 ASNs routing a combined 10% of IPs that did not spoof their geolocation. Most of these IPs were located in either Sweden, Netherlands or Russia.

While it is no secret that geolocation data is sometimes incorrect, the scale at which these actors were able to manipulate the geolocation databases should give pause to detection and mitigation methods that rely on geolocation by third parties. At a large United States retailer, we observed that more than 2% of all login traffic during a 3-month period originated from these IPs with spoofed geolocation.

In the case of this actor, the email address used to register many of the deceptive /24 and /25 network ranges is the same email used to register billingproxy[.]net, which is supposedly registered by an Ilya Trusov in Kaluga, Russia and hosted by CloudFlare. A user going by the same pseudonym as the email address can be found on searchengines[.]guru, a Russian language forum, marketing proxy services at buy[.]neproxy[.]org, which supposedly rents access to hundreds of thousands of proxies in the US, EU, North & South America. According to this user and the website itself, the service hosts its data centers in Kaluga Russia. We suspect that these data centers are the ”Depo40 Data Centers”, found at depo40[.]ru, further evidenced by the presence of the abuse email contact in many of the IP network registrations being one from depo40[.]ru.

A short Yandex search of the physical address from the vast majority of the whois registrations leads us to the municipal trolley bus management building of Kaluga. Terms alluding to ”Depo” or ”Region 40” are present throughout many of the IP network registrations we have observed launching attacks, and in fact are the main source of IP addresses with spoofed geolocations that truly originate from Russia. The word “depo” is used to refer to autobus depots in many languages, and Region40 of Russia also corresponds to the region of Kaluga as dictated by the first 2 digits on municipal vehicle license plates in Russia. Another common thread throughout the registration data of these offending IP networks is the “mnt-by:” field consistently reading “MNT-DEPO40”.

The sheer volume of attacks from these IP networks, in combination with the variety of attack tools and attacker patterns we have observed from the networks, leads us to believe that this infrastructure must be rented out to multiple attackers. We observed sustained attacks using all of the attack tools described in Section [**?**] -

Ping time from Russia

MM_City	Max_RTT mean	Max_RTT median	IP count
Albuquerque	69.753600	3.9270	15
Anchorage	602.785067	3.9070	15
Baltimore	4.049400	3.8800	15
Cedar Falls	3.695688	3.7005	16
Dallas	3.818667	3.8140	15
Detroit	356.223118	3.7030	17
Honolulu	5.079800	3.8020	15
Las Vegas	318.735211	3.6900	19
Los Angeles	3.841933	3.8420	15
Miami	203.213533	3.6720	15
None	4.649216	3.8670	51
Ogden	3.766667	3.7300	15
Orlando	3.828545	3.8160	22
Portland	3.900625	3.8340	16
Seattle	3.903438	3.8260	16

Ping time from USA

MM_City	Max_RTT mean	Max_RTT median	IP count
Albuquerque	113.620067	133.8720	15
Anchorage	735.037600	137.0600	15
Baltimore	132.246600	131.7100	15
Cedar Falls	135.660375	134.7540	16
Dallas	136.748600	133.9290	15
Detroit	138.642529	134.1860	17
Honolulu	131.118867	130.6670	15
Las Vegas	455.526316	139.1030	19
Los Angeles	134.830067	133.8160	15
Miami	335.337733	133.4570	15
None	136.346333	135.5950	51
Ogden	136.639333	136.7310	15
Orlando	137.369773	134.7120	22
Portland	133.129375	131.1810	16
Seattle	135.956437	134.1655	16

Figure 7: A snapshot of results from a distributed traceroute experiment. All the IPs shown here were supposedly geo-located to different cities across the United States. However, when IPs from all these cities were pinged from Moscow, the median RTT was roughly 3ms while it took nearly 135ms to ping from The United States, regardless of the city (they were located all over the country).

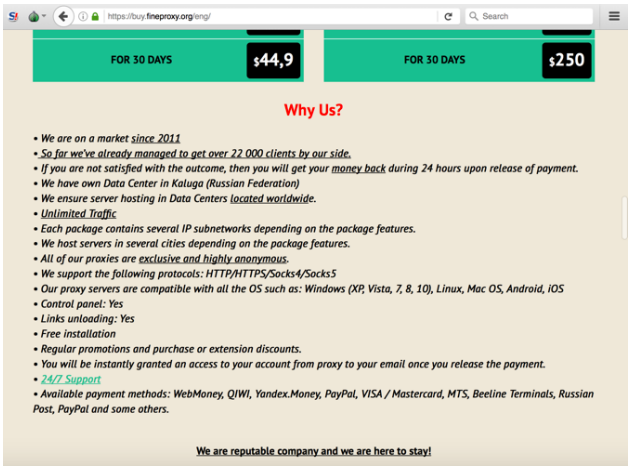

Figure 8: The homepage of buy[.]fineproxy[.]org advertising its' proxy services. They claim to have thousands of proxies from dozens of countries, and disclose that their data centers are in Kaluga, Russia.

SentryMBA, Vlad, Drago, and other known attack tools like Medusa. Substantial percentages of traffic included stolen credentials being reused in these attacks. The most common breaches from which credentials appeared in these attacks were: Myspace, RiverCityMedia, LinkedIn & Adobe.

4 Defender's Perspective - Detection Methodology

So far, we have discussed what an attacker needs to launch a successful credential exploitation attack. As the security community seeks to adapt to these changing attacks, it is important to take stock of existing defensive capabilities and seek to improve on them. Currently, many existing defenses attempt to combat this problem at the level of the individual transaction. This means analyzing client signals such as mouse movement, window size, input speed and other micro-components of an individual web request to try and determine if it is automated or not. While these techniques are effective at fingerprinting users, they face certain limitations that impact us negatively from a defensive perspective.

An ideal defensive solution should not alert attackers to its' presence and should be hard to bypass. Simply blocking attacks isn't enough - attackers can learn from this behavior and reverse engineer security solutions and find ways to bypass them. For example, they can test until they find the rate limit, and continue their attacks "low and slow" under the radar. Or they can simply change their IP address and/or User Agent string and return.

The detection methodology we introduce contains many actionable insights for network admins that can improve detection today, as well as ideas for future development and investment in tools that aid in detection and

mitigation.

Our detection methodology rests on the following pillars:

1. Analysis of HTTP/HTTPs requests and headers to fingerprint attack tools as they come across the wire.
2. Machine learning models to detect forged browser behavior and other suspicious activity.
3. A Threat Intelligence component designed to starve the attackers of the resources (compute power, IP addresses and stolen credentials) they need to execute the attacks.
4. Data analytics beyond the individual transaction level to detect attacker behaviors such as reconnaissance, account verification and "low & slow" on a wider scale.
5. Technology that covers web, mobile and API flows to detect attackers as they move across channels.

We will illustrate our pillars of detection through examples of live attack data, beginning with the popular tool - SentryMBA.

Figure 9: Most Targeted Sites by total number of config downloads.

Attacker's tool of choice: SentryMBA

SentryMBA is one of the most popular attack tools in the underground, and there is a thriving ecosystem selling configuration files for different target sites - more than 1,800 on the most popular underground forum. More than 10% of all websites in the Alexa Top 1000 have configuration files available [11].

We call this tool a "Plug and Play" attack tool, because it has an easy-to-use GUI, enabling novice attackers to launch highly effective campaigns. An attacker will plug in three components to the tool, first the configuration file, then a set of stolen credentials procured from the underground, finally followed by a set of proxy IP addresses to route traffic through. After loading these components, the attacker selects the amount of threads they will use via a slider, allowing them to distribute the attack traffic.

Attackers have used SentryMBA in activity ranging from high powered brute force attacks to low & slow distributed attacks. These campaigns can be conducted over a period of days to verify millions of credentials - we have observed sustained successful login ratios of less than 0.01%, days before an attacker will return with a verified list and login at a rate greater than 95% successful. Fig 10 shows the traffic pattern of SentryMBA against a large United States retailer over the course of an entire day's traffic. There is a mixture of sustained bursts and low & slow attacks. Over the course of this day, the attackers were successful with more than 4% of logins.

Detection

The SentryMBA attack tool can be fingerprinted in a handful of ways:

- First, the tool comes pre-programmed with a set of default UserAgent strings – 5 of which are regularly rotated, and another which is the default test UserAgent string. Observing the "test UA" in the wild is a good indicator of reconnaissance behavior, and observing regular rotation of the 5 default UAs is a good low hanging fruit indicator of SentryMBA. [1]
- While low-hanging fruit indicators are easy to act on, more sophisticated attackers can easily hard-code different User Agent strings into the tool. To adapt along with the attackers, we can use the idea of our first detection pillar and fingerprint the HTTP request headers. We analyzed the HTTP request headers of the default configuration and compared it with the request headers of about 1500 configs in the wild [10]. We found that SentryMBA traffic is often missing certain headers or attributes that are common in legit traffic.

Drago & Vlad - The "Forged Browser Family"

A key pillar of our detection methodology is machine learned models to detect forged browser behavior. A **forged browser** is a request that advertises itself as a certain browser and operating system, while in reality is behaving much like another. This behavior is extremely suspicious because we would rarely, if ever, expect to see it in legitimate human traffic – why would a normal user try to hide their browser family and model?

[1] See Appendix for the full list of SentryMBA default UserAgent strings.

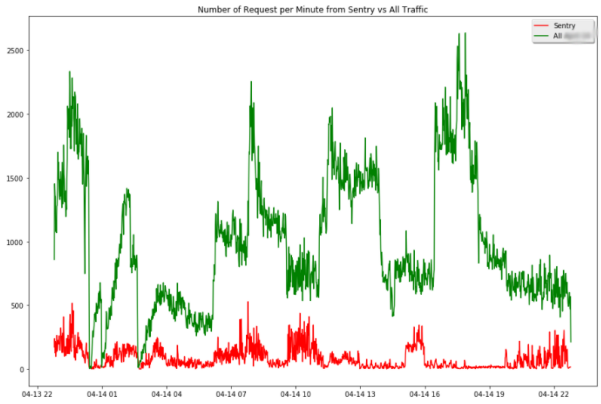

Figure 10: Time-series of SentryMBA attack traffic vs all traffic on a representative day. The tool contains both high velocity and low & slow attack patterns.

Drago Drago is an attack tool impersonating Chrome 56 on Windows 8.1. Drago originated from over 3,769 ISPs, 4,160 Organizations and more than 150 countries, with no single ISP/Organization being responsible for more than 3.5% of the tool's traffic. This is a small example of how widely distributed these attacks are, and how a reactive defensive posture is totally ineffective.

Vlad Vlad is an attack tool impersonating Firefox 40 on Windows 10. Attack tool Vlad appeared to have 100% of its' traffic originating from the US. In reality, the Accept-Language value was always set to "ru-RU".

Both of these tools logged in at a measly 0.4% successful rate, yet these attacks were sustained over a period of months. This resulted in thousands of accounts compromised every week. Fig. 12 (from a representative full day) demonstrates that these attack tools were responsible for a sustained large portion of attack traffic hitting the website throughout the day, including every large spike in traffic. This is an example of how infrastructure is often over-provisioned to deal with these spikes of automated traffic. This also shows how Machine Learning and Anomaly Detection can play an important role to detect badness.

Threat Intelligence Framework

Even though there are multiple issues with threat intelligence feeds like unreliability of data, duplicity, mislabeling etc., we argue that a curated list of feeds is critical to help defend against credential exploitation attacks.

One such complaint about using threat intelligence feeds is due to the dynamic nature of cloud providers, attempting to blacklist or even trigger an alert on known bad IPs from cloud providers is risky. It would cause false positives as soon as the IP address is reallocated and recycled to a legitimate user, and would create a barrage of alerts undermining analyst trust.

From a defender's perspective, one question we must ask is how quickly do these attacking IPs get repurposed and used in legitimate traffic? As cloud providers frequently recycle IPs, one would assume that using threat feeds and blacklists to detect these would be a futile exercise. However, we were surprised to find that these IP addresses often appear in blacklists and threat feeds for months before these attacks. Over 80% of them appeared in at least one threat feed in our database, while more than 40% appeared in attack traffic at other customers we analyzed.

This shows that these IPs can be procured by multiple attackers running many different tools configured to attack different targets. 92% of these IPs showed up at least 1 day before the attack, and 86.5% showed up at least 1 week before the attack in publicly assessable threat feeds. The average period when these IPs first appeared was over 4 months prior to the attack.

"CoolPad" & FireFox 51 – Low Hanging Fruits for Detection across API & Web Channels

In many situations, there is some sort of defense installed to protect websites against credential exploitation attacks. However, other interfaces like the Mobile and API channels go unprotected. Attackers don't discriminate and prefer to go through the path of least resistance. That's why our final pillar of detection - visibility across Web, Mobile & API channels - provides us with insights to protect against attacks across all channels.

Here we describe two attack tools which we uncovered, one of which were targeting channels other than Web. These two tools alone were responsible for 40% of web login traffic and 95% of REST API login traffic sustained over a period of months. This represents a massive amount of over-provisioning of resources to handle this traffic - amplifying the financial losses that result from automation and credential exploitation attacks.

FireFox 51 Tool

The first of these tools advertises itself as FireFox 51, and behaves in a similar manner, with the following user agent string:

```
Mozilla/5.0 (Windows NT 10.0; WOW64; rv:51.0)
Gecko/20100101 Firefox/51.0
```

This tool has been observed coming from more than 210 different countries over a period of months,

Figure 11: Details of a sustained cluster of SentryMBA, of over 150,000 requests coming from 3,385 IP addresses and 1,293 Organizations. The attackers are using the default SentryMBA user-agent strings, and we also observe the presence of credentials being tested that were leaked in MySpace, LinkedIn and Yahoo breaches. Occasional velocity spikes also tip us off to attacker behavior, and some of the IPs have been seen before in other attacks. This behavior is typical of SentryMBA users.

yet its Accept-Language header value is always "en-US,en;q=0.5".

From the FireFox 51 tool, we see each username being tried only once. Legitimate users however often mistype their passwords or trying a few passwords for the same username. For example, legitimate traffic from the Chromium family tends to see a ratio of 1.15-1.3 login requests per unique username. This metric can be used by network admins to alert for potential presence of credential exploitation attacks. The calculations can provide more insight especially when analyzing traffic grouped by ISP/Organization or ASN.

CoolPad

This tool was attacking a REST API of a retailer and was responsible for more than 95% of all traffic to that API. This tool had a UserAgent string consistent with a device that should rarely be seen in that corporate's environment.

5 Discussion

There are many lessons to be learned from these three different examples of attacker's infrastructure. They include the fact that attack infrastructure can be created in many ways, and the techniques will continue to adapt. Attempting to detect attacks from this infrastructure in a reactive manner is becoming impossible due to the scale. Sophisticated malware is not a prerequisite to acquire any of the infrastructure needed to launch automated attacks at scale, and criminals will get creative in order to "hide in plain sight" amongst legitimate traffic.

Another important takeaway is that open sourced threat feeds are quite effective and provide great value for defenders. During one attack campaign at a United States retailer, which used primarily cloud provider infrastructure to attack, over 80% of IP addresses appeared in an open source threat feed. Those IP addresses appeared in an average of 5 different threat feeds, suggesting a powerful network effect. 92% of those IPs showed up in the feeds more than one day before the attack campaign began, and 86.5% showed up more than one week prior to the attack. A system that aggregates and tracks these threat feeds can provide network admins with a surprisingly effective early warning system.

With respect to the attack tools being used, these findings should inform security analysts and network admins that traditional detection methodologies are inadequate for many of these attack tools. Effective detection re-

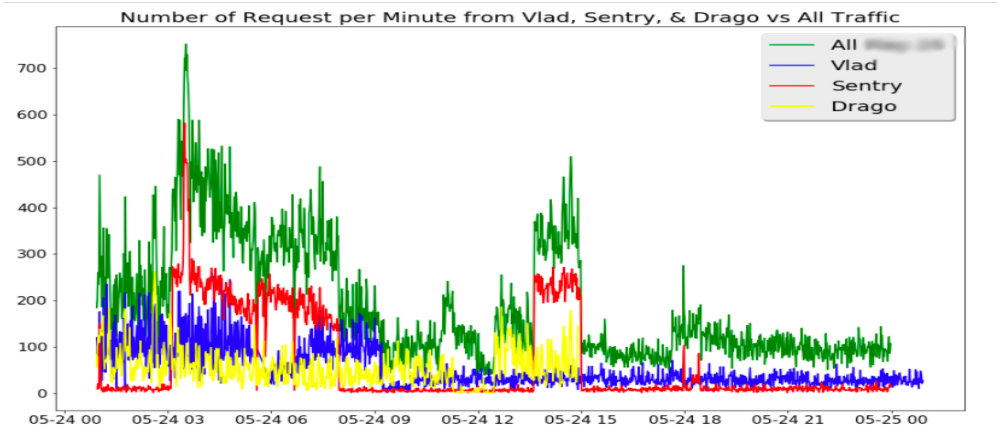

Figure 12: A time-series of traffic coming from 3 different attack tools (SentryMBA, Vlad & Drago) vs all traffic on a representative day. This shows that these attack tools were responsible for all large spikes in traffic, and a large percentage of traffic as a whole.

quires a holistic view of the traffic across all layers of the TCP/IP stack. The five pillars of detection outlined above place an emphasis on network traffic analysis and packet inspection, with the ability to cover all channels and analyze data beyond the individual transaction level.

Conclusion

Automation attacks and abuse from bots across web, mobile and API channels is a problem that continues to grow and cause significant losses due to account takeover, fake account creation, content scraping and other fraud. Attackers need only procure stolen credentials, an attack tool and enough proxy IP addresses to launch their attacks, and current detection methods are struggling to keep up. We have displayed a passive detection methodology relying on analysis of the HTTP request headers that allows fingerprinting of the attack tools. When combined with the other pillars of detection we discussed, this represents an improvement on existing technologies. The tools we have described such as SentryMBA, CoolPad, Drago and others all contain a non-trivial amount of low hanging fruit available for detection purposes.

Using our methodology, network admins and security professionals can implement quickly provided they have the proper tools. We have displayed a multitude of ways attackers procure the infrastructure they need for these attacks, demonstrating both the scale of the problem, and the low barriers to entry for some new attackers. Massive data breaches continue to hit the news seemingly every day, providing further fuel to this fire. This leads us to believe that the problem of automated attacks at scale will continue to grow until defenders can adapts as quickly as the attackers.

References

[1] Brute Forcing Passwords with ncrack, hydra and medusa. https://hackertarget.com/brute-forcing-passwords-with-ncrack-hydra-and-medusa/.

[2] D-Link Devices Unauthenticated Remote Command Execution. https://www.rapid7.com/db/modules/exploit/linux/http/dlink_command_php_exec_noauth.

[3] D-Link DIR-600M Wireless N 150 - Authentication Bypass. https://www.exploit-db.com/exploits/42039/.

[4] How Cybercrooks Put the Beatdown on My Beats. https://krebsonsecurity.com/2017/04/how-cybercrooks-put-the-beatdown-on-my-beats/.

[5] Huawei Routers Vulnerable to Directory Traversal Attacks (CVE-2015-7254). https://www.redpiranha.net/huawei-routers-vulnerable-directory-traversal-attacks-cve-2015-72

[6] Medusa. http://foofus.net/goons/jmk/medusa/medusa.html.

[7] Password reuse, credential stuffing and another billion records in Have I been pwned. https://www.troyhunt.com/password-reuse-credential-stuffing-and-another-1-billion-records-

[8] BURGESS, M. Want to know if you've been hacked? Troy Hunt has all the details. http://www.wired.co.uk/article/troy-hunt-interview-pwned-security, 2017.

[9] DAS, A., BONNEAU, J., CAESAR, M., BORISOV, N., AND WANG, X. The tangled web of password reuse.

[10] DHIMAN, M. Detecting Credential Verification Attacks: An Analysis of SentryMBA Configs. https://www.stealthsec.com/resources/Detecting-SentryMBA-Credential-Verification-Attacks.pdf.

[11] DHIMAN, M., AND WILL, G. SentryMBA: A Peek into the underground economy. https://s3-us-west-2.amazonaws.com/stealthsec-www/resources/SentryMBA-eBook.pdf.

[12] FLORENCIO, D., AND HERLEY, C. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 657–666.

[13] JAKOBSSON, M., AND DHIMAN, M. The benefits of understanding passwords. In *Mobile Authentication*. Springer, 2013, pp. 5–24.

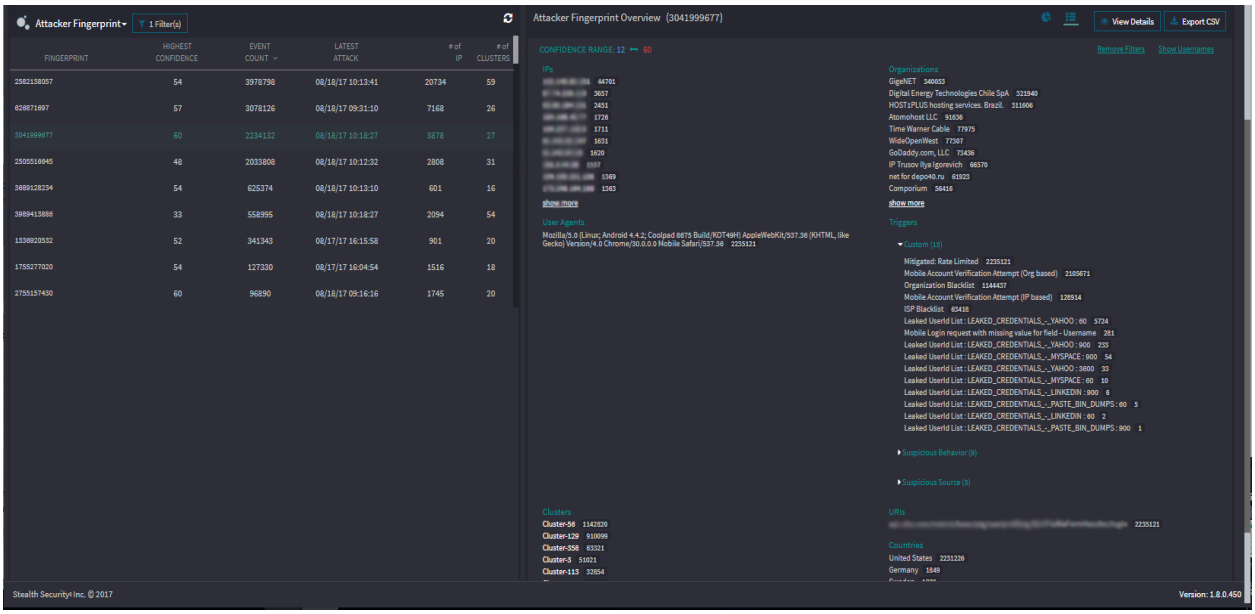

Figure 13: Insight into the characteristics of the Coolpad attack tool, including its' user-agent string and some of the attack infrastructure used to launch the campaign.

[14] KHANDELWAL, S. 100,000 Refrigerators and other home appliances hacked to perform cyber attack. `https://thehackernews.com/2014/01/100000-refrigerators-and-other-home.html`, 2014.

[15] RAMACHANDRAN, A., AND FEAMSTER, N. Understanding the network-level behavior of spammers. In *ACM SIGCOMM Computer Communication Review* (2006), vol. 36, ACM, pp. 291–302.

[16] SCHNEIER, B. Lessons From the Dyn DDoS Attack. `https://www.schneier.com/essays/archives/2016/11/lessons_from_the_dyn.html`, 2016.

[17] SECTOOLS. Brutus. `http://sectools.org/tool/brutus/`.

[18] SHAY, R., BAUER, L., CHRISTIN, N., CRANOR, L. F., FORGET, A., KOMANDURI, S., MAZUREK, M. L., MELICHER, W., SEGRETI, S. M., AND UR, B. A spoonful of sugar?: The impact of guidance and feedback on password-creation behavior. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), ACM, pp. 2903–2912.

[19] TOONK, A. Turkey Hijacking IP addresses for popular Global DNS providers. `https://bgpmon.net/turkey-hijacking-ip-addresses-for-popular-global-dns-providers/`, 2014.

Appendix

SentryMBA Default User Agents

```
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0;
.NET CLR 1.1.4322; .NET CLR 2.0.50727;
.NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)

Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1;
.NET CLR 1.1.4322; .NET CLR 2.0.50727;
.NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)

Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
rv:1.9.0.11) Gecko/2009060215 Firefox/3.0.11

Mozilla/5.0 (Windows; U; Windows NT 5.1; en)
AppleWebKit/522.11.3 (KHTML,, like Gecko)
Version/3.0 Safari/522.11.3

Opera/9.80 (Windows NT 6.0; U; en) Presto/2.2.0
Version/10.00

Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1) **Testing UA**
```