

Mining Threat Intelligence from Billion-scale SSH Brute-Force Attacks

Yuming Wu^{*§}, Phuong M. Cao^{*†§}, Alexander Withers[†], Zbigniew T. Kalbarczyk^{*} and Ravishankar K. Iyer^{*}

^{*} University of Illinois at Urbana-Champaign,

[†] National Center for Supercomputing Applications

Abstract—This paper presents a longitudinal study of 11 Billion SSH brute-force attacks targeting an operational system at the National Center for Supercomputing Applications. We report the nature of these distributed attacks in terms of i) persistence (i.e., consecutively attacking over an entire year), ii) targeted strategies (i.e., using stolen SSH keys), iii) large-scale evasion techniques (i.e., using randomized SSH client versions) to bypass signature detectors, and iv) behaviors of human-supervised botnet.

The significance of our analyses for security operators include i) discerning cross-country attacks versus persistent attacks, ii) notifying cloud providers and IoT vendors regarding stolen SSH keys for them to verify the effectiveness of software patches, iii) deterring the above evasion techniques by using anomaly detectors/rate limiters, and iv) differentiating between fully automated attacks versus more sophisticated attacks driven by human.

I. INTRODUCTION

The Secure Shell (SSH) is the universal authentication protocol for managing remote servers. Attacks targeting exposed SSH servers see an exponential growth recently due to the availability of leaked passwords and stolen keys [25], [30], [33], [38], [40], [41]. A successful SSH login typically grants the super-user (root) permission, thus enables persistent access for compromising internal network, exfiltrating sensitive data [26], [32], [37] and causing monetary losses. For example, when being offered 50 Bitcoins by a hacker, a former server administrator at ShapeShift [23], a cryptocurrency company, gave away an SSH private key to the company’s Bitcoin core server for accessing internal Bitcoin’s wallets. This incident eventually led to \$230,000 losses [7], [20].

This paper presents a longitudinal study of 11 Billion SSH brute-force attacks targeting an operational system [22], [31] at the National Center for Supercomputing Applications¹ (NCSA). We report the nature of these attacks in terms of i) persistence (i.e., consecutively attacking over an entire year), ii) targeted strategies (i.e., using stolen SSH keys), iii) large-scale evasion techniques (i.e., using randomized SSH client versions) to bypass signature detectors, and iv) behaviors of human-supervised botnet.

A. Data Overview

Our dataset contains 11 billion attack attempts, including 3.4 billion connections and 7.9 billion SSH password- and key-based brute-force attack records. Each is an attempt to compromise the SSH server and thereby access the internal network and steal sensitive data. The data is collected in an operational honeypot in 1,000 days starting in February

2017, deployed on a /16 IP address space simulating ~65K machines [31]. In total, the honeypot recorded 4.5 million unique, globally distributed, IP addresses of attackers.

B. Analysis Workflow

The main steps in our analyses are to: i) discern the nature of attacks in terms of persistence (Section II), ii) identify coordination and evasion techniques (Section III), and iii) distinguish human-supervised and fully automated botnet attacks (Section IV). Fig. 1 illustrates the logical flow of our approach.

We first load our large dataset (3TB) into Clickhouse, a columnar database [15], for fast and distributed SQL queries on the dataset. Then, we show summary statistics and month-by-month trends of attack techniques, including SSH keys, passwords, usernames, SSH client versions, and IP sources. We focused on trend anomalies (Fig. 2(b)) and persistent footprints over the entire honeypot operation (Fig. 3 and 4). Second, we closely inspected attacker tactics (i.e., through SSH key and client version forensics) in terms of exploitation, coordination, and evasion. Finally, we quantify the degree of automation in attack strategies by comparing attack activities during the weekend and weekday. This analysis allows us to discern and distinguish human-supervised botnets with fully automated ones (Fig. 8, 6, and 7).

C. Findings and Implications

We provide key findings and implications as follows.

• Persistent attacks versus cross-country attacks.

Persistent attackers constituted over 70% of total attack attempts; some of them have been brute-forcing consecutively for over an entire year. While the total number of unique attack sources (4M IPs) does not increase significantly, we saw a 20× increase in attack attempts since the public disclosure of the honeypot [31]. On the other hand, attackers from 20 countries across four continents rapidly exploited an SSH key over four days – 50× faster than a single-country botnet even with more IP sources, implying a global coordination effort. The finding indicates that most attackers (distributed across countries) were rapidly shifting targets after fruitless exploitation within a short period, while a few persistent attackers were relentlessly coming back to the honeypot and posing aggressive threats to production systems.

• Leaked SSH keys exploitation.

Attackers from the major cloud and ISPs (e.g., Google or Charter Communications) used leaked SSH keys of IoT appliances (e.g., Enterprise VA load balancer [3] or VMware data protection appliances [12]). In particular, attackers originated from Google-registered IPs rapidly exploited all seven leaked keys that we have identified (Table II) in one day. This finding enables the opportunity to alert the owner of leaked keys and IoT appliance operators to i) investigate whether their software patches are sufficient, and ii) determine whether

[§]Joint first authors.

¹NCSA hosts the Blue Waters, a sustained petascale supercomputer of 22,640 cores.

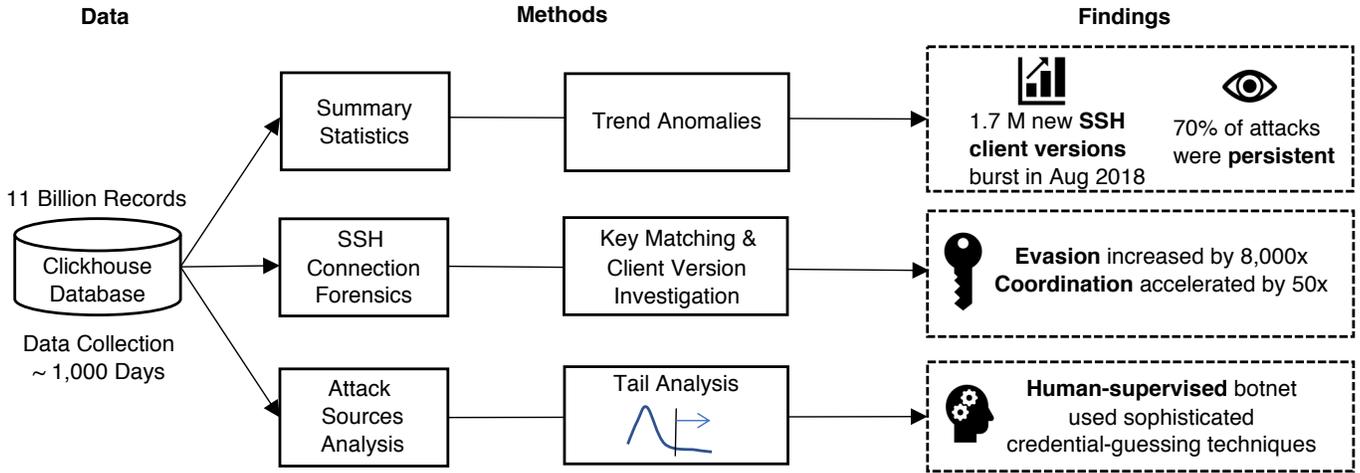


Figure 1: Analysis workflow to mine threat intelligence.

affected devices with outdated vulnerabilities still exist in the wild.

- **Large-scale evasion techniques.** A globally coordinated botnet (from 30 countries across six continents) spoofed millions of unique client versions for three months, which was over 8,000 \times the counterpart in the previous 18 months. This finding implies that new bot blocking techniques such as rate limiting or anomaly-based blocking must be deployed instead of signature-based blocking based on known client versions.
- **Human-supervised botnets.** We found a group of human-supervised bots, operating from the same /8 subnet, that only attacked during human working days. Compared to the fully automated type of botnet that attacks constantly, the human-supervised botnet employed more diverse devices and planned more strategically in credential brute-forcing. Security operators can integrate our approach into defense strategies to build bot-specific defense models.

II. LONGITUDINAL PERSPECTIVE OF ATTACK BEHAVIORS

This section presents key findings from the longitudinal analysis of SSH attack trends.

A. Trend Anomalies

The anomalies in trend, including surgent client versions, dissipating active IPs versus rising active attack attempts, guided us to discover particular attacker tactics behind the scene.

Abrupt upsurge of unseen client versions from new attackers. The honeypot witnessed a dramatic increase of $\sim 50,000\times$ unique new-attempted client versions from August to October 2018. Eventually, as implied in Fig. 2(b), aligned with the total 4 million unique IP sources, the final accumulative number of unique client versions also reached a scale of 2 million in the end. This number is abnormally high because the previous studies both reported just more than 100 version strings [31], [33]. Further inspection reveals attackers from a group of new-emerging, globally distributed IP sources were massively spoofing client versions in coordination. Contrary to the version trend, within the same month in August, increments of new IPs slowed down (Fig. 2(b) and 2(c)). In Section III-C, we present the detailed attack tactics underlying the million-scale SSH version string manipulation.

Increasing scale of attack attempts from fewer attackers. Fig. 2(c) illustrates the monthly progression of active IPs and attack attempts in reverse trends. Due to changes in network policy to no longer block honeypot traffic, monthly attack attempts increased by 600 times in April 2018. On the other hand, while the honeypot did not filter certain IPs,

the overall IPs were still decreasing. This implies that, after fruitless exploitation over a short period, most attackers in the wild were shifting targets rapidly. Combining the two reverse trends, we could conclude that number of attacks per IP were dramatically increasing over time. We speculate that certain attackers remained persistent in exploiting our honeypot by repeatedly coming back or launching a large number of exploits at one time. We will characterize and present the activities and patterns of those persistent attackers below.

B. Persistent Attack Traces

We had observed four million unique IP addresses during the initial 463 days of honeypot deployment [31]. However, this number increased by only 0.5 million (12.5%) over the next 537 days. In contrast, total attack attempts increased by almost 20 times (from 405 million to 7.9 billion). We suspect certain persistent attackers were repeatedly coming back to attack the honeypot. Therefore, in this section, we present the statistics and patterns of attackers in terms of persistence.

Persistent attackers constituted over 70% of all attacks. We categorize two types of IPs in Fig. 4: recurring IPs and monthly new IPs. In contrast with monthly new IPs (with the first occurrence at the current month), recurring IPs repeatedly appear to attack the honeypot across different months. Hence their occurrences in the current month are recurrences. For recurring IPs, the maximum value of monthly average attempts per IP is 0.2 million, which is 5 \times that of new IPs. As Fig. 4 implies, Starting from May 2018, the trend of actively recurring IPs started to catch up with monthly new IPs. In general, total attacks from the recurring IPs are 2.5 \times the overall contribution from monthly new IPs, and those recurring IPs constituted more than 70% of all the attack attempts in the dataset.

Persistent attackers continuously attacked for over one year. To quantify recurring IP's degree of persistence, we investigated three metrics: attack time span, effective attack days, and the longest span of consecutive attack days. For each IP, We define "attack time span" as the time between the dates of that IP's first and last attack, "effective attack days" as the number of days with at least one attack attempt by that IP, and "longest consecutive days" as the highest number of consecutive effective attack days for that IP.

Even within a long time span, attackers may not actively attack every day. Our results show that four IPs have the longest attack time span of 1,000 days, which is as long as the time span of the honeypot's operation. However, their effective attack days range from 62 to 221 days, which means these four IPs were only effectively attacking at most 22% of their time spans.

We then switched focus to attackers with a large number

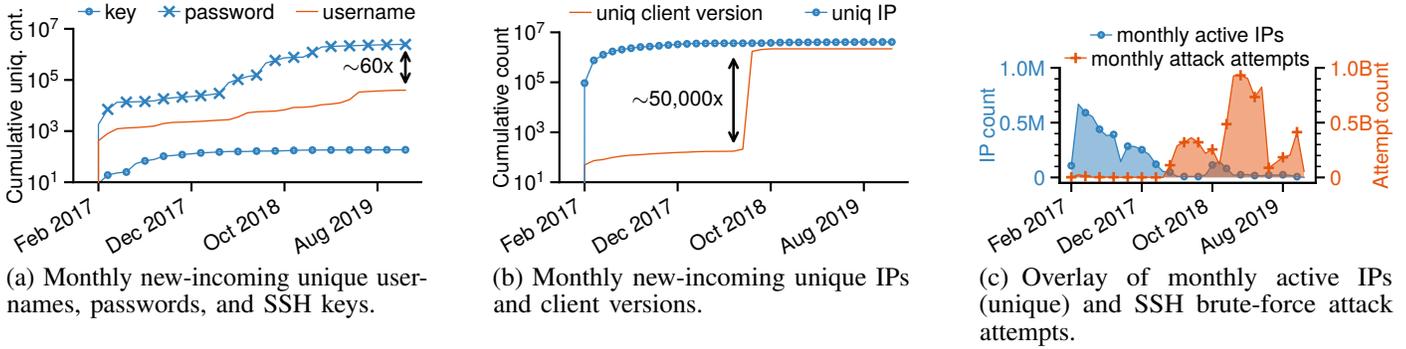


Figure 2: Cumulative trend plots depicting key features in more than 8 billion attack attempts spanning 1,000 days.

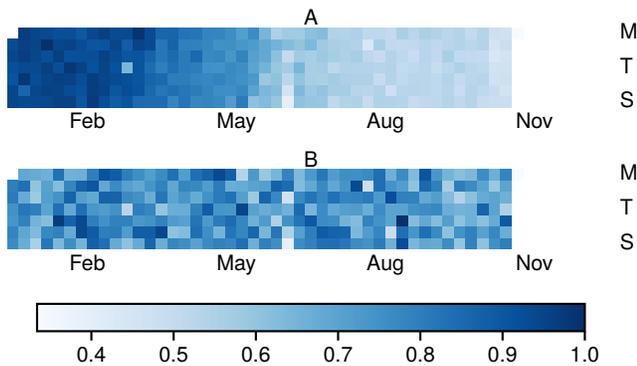


Figure 3: Normalized attack activities between two persistent IPs (A&B) in 2019 [M: Monday, T: Thursday, S: Sunday]. Both IPs adopted naive password strategies: A rotated over 42 unique passwords almost every day; while B alternated almost equally between only two passwords every single day.

of effective attack days. The most significant value is 555 days, which was achieved by two IPs. Moreover, both IPs were actively attacking for 384 consecutive days – over an entire year. The top five IPs with the most effective attack days were all attacking consecutively over an entire year. In particular, these five IPs came from four different ISPs. In Fig. 3, we select two persistent IPs from the top five to illustrate the different patterns in their normalized attack activities.

Implications. Persistent IPs are more prone to access a range of internet-connected sites and devices [36]. Therefore, we advocate sharing the long-term threat intelligence with peer sites, so that corresponding network operators can preempt the aggressive threats by monitoring or flagging these persistent attackers in advance, saving both defense resources and time.

III. EXPLOITATION, COORDINATION, AND EVASION

This section presents the exploitation, collaborating, and evasion strategies of advanced adversaries.

Table I: Details of the seven leaked SSH keys (sorted by public disclosure year).

SSH Key (SHA256)	Key Owner	Appliance Type	Public Disclosure Year	1st Attack Attempt Year	Username
1M4Rz...qu0ZA	Vagrant [1]	Base box for development environments	2010	2018	root
9prMb...Ghro4	F5 [2]	BigIP appliances	2012		
MEc4H...UfTww	Loadbalancer [3]	Virtual load balancer	2014		
VtjqZ...PiQPc	Quantum [5]	Virtual deduplication backup appliance			
/JLp6...P0Cc0	Array Networks [4]	Virtual application delivery controllers Secure access gateways			
Z+q4X...8kIxM	Ceragon [6]	IP traffic router	2015		mateidu
f+1oG...zEDhc	VMware [12]	Data Protection appliances	2016		admin

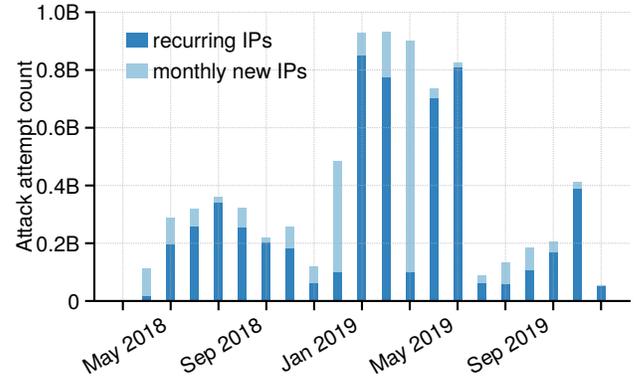


Figure 4: Monthly attack attempts comprising of monthly new-incoming IPs and cross-month recurring IPs. Total attempts from recurring IPs accounted for more than 70% total attack attempts.

A. Exploitation of Leaked SSH Keys

In total, 185 unique SSH public key fingerprints (in the SHA-256 hash) found their way into our honeypot. By matching each of the keys with a public database and online files of bad keys [11], [21], we discovered and recovered the identities of seven keys that were publicly leaked due to vulnerabilities. Further investigations implied that cybercriminals were trying to gain root permission to vulnerable production appliances and devices in the wild using these leaked keys, even years after the key-pertinent vulnerability disclosure. Their exploitation strategies are revealed in more detail below.

Attackers were targeting production devices using leaked keys. The seven leaked keys belonged to seven different enterprises (see Table I). All these keys granted attackers with root permission in the targeted systems eventually. Among those keys in Table I, the top four have direct root privileges. With the bottom three keys, attackers gain access as non-root users initially. However, either by exploiting local vulnerabilities (e.g., Array Networks [4] and Ceragon [6]) or becoming a sudoer without a password (e.g., VMware [10]), attackers eventually

Table II: A summary of the ASes that exploited the seven leaked SSH keys.

Autonomous System	Client Version [SSH-2.0-]	SSH Key (SHA256) & Key Owner						
		1M4Rz...	9prMb...	Mec4H...	VtjqZ...	/JLp6...	Z+q4X...	f+1oG...
		Vagrant	F5	Loadbalancer	Quantum	Array Networks	Ceragon	VMware
Google LLC	libssh_0.7.0	✓	✓	✓	✓	✓	✓	✓
Charter Communications	Ruby/Net::SSH...		✓	✓	✓	✓	✓	✓
Portlane	libssh-0.6.1			✓	✓			

Ruby/Net::SSH... refers to Ruby/Net::SSH_5.0.2 x86_64-linux-gnu.

escalate privilege to root.

The attackers used the privilege level related to each corresponding leaked key when targeting our honeypot. In Table I, the top four keys were all attempted with root permission, while the bottom three were attempted using the exact user names issued from key owners and related to disclosed vulnerabilities. For example, username `admin` from VMware can escalate to root without a password [8]. This observation indicates, instead of randomly using leaked keys to brute-force, the attackers have adequate details about pertinent vulnerabilities when plotting the targeted attacks.

Attackers were rapidly exploiting the leaked keys. Attacks that originated from Autonomous System (AS) such as Google LLC (Google) [18], Charter Communications [14], and Portlane [17] participated in exploiting the seven leaked keys. Table II presents an overview. As can be inferred from Table II, two of the seven keys were used by all three ASs; four of the seven keys were used by both Google- and Charter Communications-registered IPs. In particular, attackers from Google tried all seven identified, leaked keys. In addition to the seven leaked keys, Google-registered IPs also exploited four other keys with unknown identities. We suspect that those four keys, though not identified yet, also belong to production devices with disclosed vulnerabilities. Coincidentally, all seven keys were attempted on the same day (Dec 14, 2018) by attackers from Google-registered IPs. In addition, IPs registered by Charter Communications attempted to exploit the five known leaked keys, together with three other unidentified keys, over two days (July 28-29, 2018). Portlane-registered IPs used two and only two known keys on another day (Mar 24, 2018). Therefore, we speculate that these attackers were rapidly switching targets for massive exploits of (outdated) vulnerabilities in order to access compromised devices in the wild by collecting and reusing old, sensitive SSH keys.

Challenges and limitations. Our previous work on hundred million scale attempts [31] claimed no evidence of leaked keys. We suspect that was either due to a mismatched comparison between different hashing algorithms in SSH protocol (e.g., SHA256, DSS, RSA) or insufficient keys for analysis. In our work, we did the exact SHA256 conversion (`ssh-keygen -lf id_rsa.pub`) of the publicly available keys in online database and files. As a result, we have successfully pinpointed matches

Table III: Top ten SSH key fingerprints from most diverse IP sources (reversely sorted by number of originating countries).

SSH Key (SHA256)	# Countr(y/ies)	# AS(es)	# IPs	Client Version [SSH-2.0-]
qLIN/...	20	38	64	Go
B6kr4...	}	}	25	}
mumiE...			49	
jSCqa...			42	
V600C...			28	
zPA6Y...			23	
NH5Y7...			19	
OyHmn...			17	
8bLLD...			16	
+UJNI...			71	
				kthrssh_x00

between public datasets and recorded attempts at our honeypot. Therefore, these matches further allowed us to trace attack origins, behaviors, and strategies.

In addition to a proper conversion method, the challenges in SSH key matching also lie in the limited availability of a public, formatted database of leaked SSH keys. We matched our keys with one public bad SSH key database [11]. To expand the search scope, we also manually looked for leaked keys listed in the files at Packet Storm [21]. The entire search, however, is far from automated and complete, and thus leaves the remaining 178 keys unidentified.

B. Key-based Collaboration

We inspected the diversity of attack sources using SSH keys in general, from which we uncovered global coordination.

An SSH key was exploited by 20 countries. We sorted key-based attempts characterized by the number of the originating IP address, presenting the top ten in Table III. Each of the ten keys originated from more than 15 distinct IP addresses, with the highest number being 71. However, most attackers originated from just a single country or AS. The only exception is the first key in Table III, which was used by 64 IPs scattered over 20 countries from four continents (including Asia, Europe, North America, and Oceania).

A persistent, single-country botnet versus a rapidly, globally colluding botnet. Further inspection revealed that this globally coordinated botnet exploited a single SSH key 90 times within only four days (Dec. 11 to Dec. 14, 2017). On the other hand, the last key in Table III originated from 71 IPs, yet all from a single country and AS. In contrast with the global botnet, this botnet persistently used one key for 2,700 times spanning five months (Feb. 2017 to July 2017). Compared with this single-origin bot, we can conclude that the globally coordinated bot wrapped up its fruitless attacks and shifted targets 50× faster.

C. Client Version-based Coordination and Evasion

Starting from August to October in 2018, the honeypot witnessed an unprecedented emergence of unseen client versions. More than 1.7 million new client versions sprang up in August alone, which was over 8,000× more than the total number of unique client versions in the previous 18 months. Further, inspection revealed that only several hundred IPs spoofed client versions by randomizing over one million OpenSSH version banners. This is unusual because, among all attackers, about 90% of IPs advertised only one client version. We speculate these randomizations were the attackers' mimic technique responding to our honeypot's deceitful defense mechanism.

Attackers randomized SSH client version banners at high frequencies. We sorted those unique client versions by the number of occurrences in August. The distribution of occurrences turned out to have an extremely long tail: 88 out of 1.7 million client versions occurred over 2,000 times each; ten out of 1.7 million occurred over ten times, while the rest 1.7 million distinct client versions just occurred at most 9 times each. Moreover, the top-spoofing IP advertised 400,000 unique client versions during its 200-hour attack campaign,

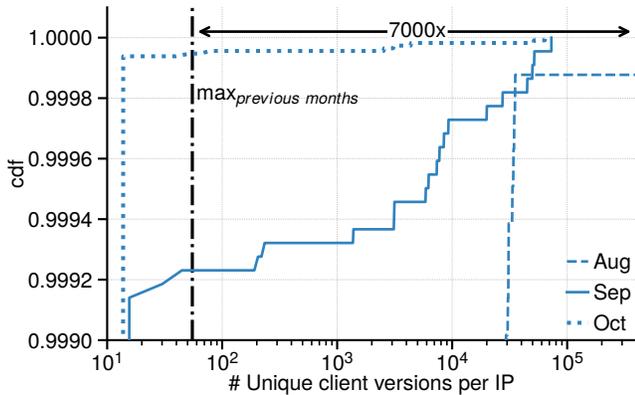


Figure 5: CDFs indicate over million unique client versions in total during August - October 2018 (contrast with previous 18 months: $\max_{\text{previous months}}(\# \text{ unique client versions/IP}) = 55$).

implying varying an average of 2,000 client versions per hour. This attacker advertised SSH-2.0-OpenSSH_7 within first several days of attack, then switched to massive spoofing by appending SSH-2.0-OpenSSH_ with 5-character random strings (e.g., +qlfH). Plus, contributions from other attackers adopting the same technique spoofed random strings had more than a million permutations in total.

A globally-coordinated botnets were involved in forging a million permutations of client versions. As illustrated in Fig. 2(b), the pace of new-incoming IPs even slowed down in August 2018. We initially suspected a large number of recurring botnets were scheming the large-scale randomization. However, only several hundred IPs were involved, and over 85% of them were new-incoming IPs in August. On the other hand, the earliest IPs involved in spoofing launched its first attack in February 2017 – as early as the honeypot’s initial operation, yet these IPs did not start spoofing client versions until August.

Around 90% IPs advertised only one client version. In the long tail of the unique number of client version per IP distribution, further investigation showed that less than 300 IPs, yet globally coordinated from over 30 countries across six continents (all excluding Antarctica), actually accounted for the million-scale random permutations of client versions to masquerade their attack traces. Fig. 5 presents the CDFs of client versions per IP tail distribution during Aug – Oct 2018, with a maximum number of 400,000 in August, which was over $7,000\times$ its counterpart in all previous 18 months.

Defense-targeting evasion. The honeypot deceives attackers by randomizing key fingerprints for each of the 65,536 servers on the entire /16 IP address space [31]. We, therefore, suspect that the attackers were mimicking our honeypot’s defense mechanism responsively. Besides, attackers were massively hiding essential attack signatures, which would generally invalidate signature-based detection. As a result, it calls for needs to deploy new defense strategies such as rate-limiting or anomaly-based detection.

IV. HUMAN-SUPERVISED ATTACK TECHNIQUES

This section analyzes routine human activities embedded in a large scale of time-series attack data (Section IV-A). After discovering routine patterns of human attackers on a weekly basis, we further provide case studies to compare and contrast the distinctive behavior patterns and strategies between fully automated botnets and human-supervised botnets (Section IV-B).

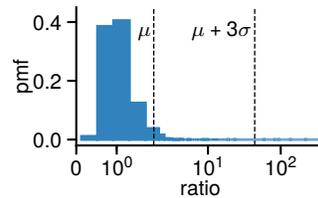


Figure 8: Distribution for ratio of weekday to weekend average attempts in June 2019.

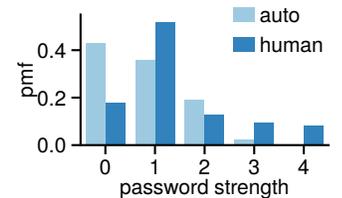


Figure 9: Password score distribution from two types of botnets.

A. Why understanding human-supervised attacks is important and our data-driven methodology

Human interaction played an essential role in cybersecurity attacks, e.g., ransomware propagation, advanced tool development at targeted victims [13], [39]. Revealing human attacker evidence aid in the detection of sophisticated underlying strategies that automated bots alone cannot accomplish. However, it’s non-trivial to distinguish both techniques in large-scale security data.

Current work [32], [34] implemented additional features to capture human-generated activities, e.g., keyboard/mouse typing/clicking, window resizing. However, these methods introduced overhead to networking system design. Instead of modifying or adding features to the current design, The billion-scale attack attempts motivate us to come up with a data-driven methodology for mining human activity patterns.

Tail analysis of attack distributions. We take regional time differences into account when investigating evidence of routine human activities. In practice, we pinpointed one specific time zone and focused only on IP addresses originating from this time zone. Then we chose a month with the most attack attempts, to increase the chance of finding evidence of human activities among source IPs.

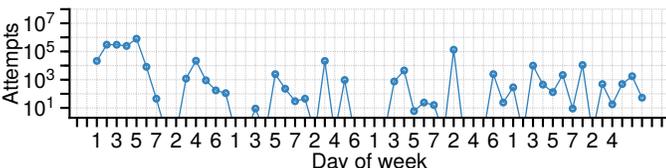
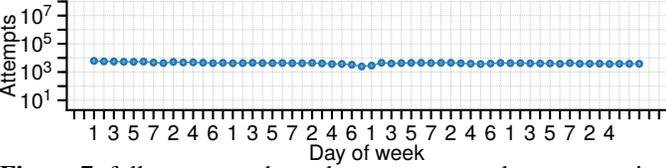
After grouping by IP, we computed average weekday and weekend attempts for each IP during the selected month. To quantitatively capture routine human evidence, we calculated a ratio of weekday to weekend average attempts for each IP. Since we aimed to find relatively long-term (4–6 weeks) evidence, we filtered out IPs with the number of active weekdays lower than 15. Fig. 8 presents the ratio distribution after filtering, which drew our attention to the tail. Specifically, we then focused on IPs with the ratio Z-score [24] greater than three standard deviations (3σ) from the mean (μ), as shown by the tail on the rightmost part of the distribution.

Activity patterns of the human-supervised botnet. It turned out that all IPs in the tail, with similar activity patterns, originated from the same /8 subnet, indicating organized routine management of botnet by the human attacker(s). These IPs also used the same client versions, passwords, and usernames. The daily intensity of these bots indeed aligned with human social routine on a weekly basis: periodic variations with decreasing activities on weekends (especially Sundays). As an example, we illustrate daily activities of one such IP over eight weeks in Table IV (Fig. 6), a representative of a human-supervised bot. Further inspection revealed that its average attack attempts during working hours (i.e., 9 am–5 pm) are over $2\times$ the counterpart outside working hours.

B. Case studies of two botnet types: human-supervised and fully automated

To fairly compare between two distinct botnet behaviors, over the same duration, we selected another IP with a weekday to weekend average attempt ratio equaling to one. Its daily activities are illustrated in Table IV (Fig. 7), a representative of a fully automated bot. In general, Table IV provides an overview of feature-wise comparisons in terms of main authentication

Table IV: Human-supervised versus fully automated botnet.

Type	Illustration of Daily Attempts [May 27 – July 21, 2019 (8 weeks)]	List of Unique Client Versions [SSH-2.0-]	List of Unique Username(s)	# Unique Passwords
Human-supervised	 <p>Figure 6: Human-supervised attack attempts declined on weekends [1: Mon. ~ 7: Sun.].</p>	PuTTY OpenSSH_5.3 OpenSSH_6.2p2... nssh2_4.0...	root	35,952
Fully automated	 <p>Figure 7: fully automated attack patterns were almost unvarying over different days of the week [1: Mon. ~ 7: Sun.].</p>	sshlib-0.1	root, user, admin, ubnt, usuario, pi, supervisor, support, service, mother	42

OpenSSH_6.2p2... refers to OpenSSH_6.2p2 Ubuntu-6;
 nssh2_4.0... refers to nssh2_4.0 NetSarang Computer, Inc.

credentials. The duration we have selected is representative of the entire attack campaign for both IPs. Furthermore, this section offers detailed case studies to distinguish the attack strategies adopted by both botnet types.

Human-supervised botnet is more resourceful in terms of attack devices. The entire human-supervised botnet shared the four client versions listed in Table IV. All bots iterated over these four client versions with equal distribution for each. There were cases when these four client versions were used at the same time by one bot. On the other hand, the fully automated bot advertised one and only one commonly-used client version. Therefore, human-supervised botnet employed a more diverse handful of devices to launch attacks.

Human-supervised botnet is more ambitious and strategic in terms of credential brute-forcing. We used Dropbox zxcvbn [42] to measure password strength. Fig. 9 summarizes the score distribution for both botnets. For a fully automated bot, only one password (7ujMko0admin) scores 3, which is the highest among all 42 unique passwords it attempted, with the majority scoring 0. On the other hand, around 3,000 passwords used by the human-supervised botnet score 4.

Notably, one password used by human-supervised botnet begins with Branch:masterFindfileCopypath, and ended with a path in a Github repository [9]. This Git repo contains a wide range of passwords collected from backdoors, web shells, mail servers, WebLogic, Linux OS, dictionaries, etc. In addition to passwords, we also found collections of database and backdoor file paths, plus a script for brute-forcing enterprise mail servers, including Exchange [16], etc.

On the other hand, fully automated bot rotated all 42 passwords every day over the entire attack campaign. Most passwords are commonly-used default passwords in Linux OS, IoT devices, routers, and firewalls.

V. FUTURE WORK

We hereafter plan out the future work as follow.

SSH keys. The identified SSH keys (especially disclosed in recent years) alert key owners and IoT appliance operators to i) investigate the coverage of patches for outdated vulnerabilities and ii) examine whether affected devices and users still exist in the wild. On the other hand, for unknown keys, we can speculate their identities or targeted devices based on associated

usernames (e.g., raspberry) and client versions, so that we can accordingly broadcast unknown keys to in advance as precautions for zero-day exploits towards targeted devices.

Client versions. Attackers’ massive evasion techniques to bypass signature detection motivate rate-limiting or anomaly-based detection. Moreover, we plan to detect fake and spoofed SSH version banners with [33] and [19]. Both methods require additional SSH information (e.g., key exchange algorithms, encryption methods), which is not available at present. On the other hand, apart from fake banners, attackers also advertised over 50 seemingly legitimate client versions (the vertical line in Fig. 5). A further inspection into the landscape and dynamics of such resourceful attackers will benefit the design of effective defense strategies and mitigation operations.

Privacy-preserving insight sharing. In terms of threat intelligence sharing across different sites, we suggest employing Private Intersection-Sum [35] to 1) preserve the privacy of sensitive information to be shared and 2) establish anonymous hacker identifiers. In the long term, the intelligence gathered from the honeypot can be leveraged in the automatic learning and upgrading of defense systems, thus building up to AI-driven intrusion detection [27]–[29] that can respond automatically to unknown threats based on past evidence.

VI. CONCLUSION

We investigated a broad scope of attack strategies in billion-scale SSH brute-force attacks. We discover the great potential in attackers to launch large-scale, persistent, and evasion attacks that are accompanied by strategic human supervision. Also, we contribute methodology to reveal bot collaboration campaign in the wild, offer a scientific data-driven approach to differentiate between human-supervised versus fully automated botnet, as well as motivate privacy-preserving threat intelligence sharing for AI-driven intrusion detection in the ultimate goal.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No 1535070, 1547249. The opinions, findings, and conclusions stated herein are those of the authors and do not necessarily reflect those of the sponsors. We thank the NCSA security team, the SDAIA project, SUTD for collaborating with the SDAIA project, DEPEND group members, anonymous reviewers, and Prof. Pekka Nikander for providing valuable feedback. We also thank Frances Rigberg Baker for proofreading.

REFERENCES

- [1] "Vagrant is a tool for building and distributing development environments." 2010, <https://github.com/hashicorp/vagrant/tree/master/keys>.
- [2] "Scanning for vulnerable f5 bigips with metasploit," 2012, <https://blog.rapid7.com/2012/06/11/scanning-for-vulnerable-f5-bigips-with-metasploit/>.
- [3] "Loadbalancer.org enterprise va 7.5.2 static ssh key," 2013, <https://packetstormsecurity.com/files/125754/Loadbalancer.org-Enterprise-VA-7.5.2-Static-SSH-Key.html>.
- [4] "Array networks vxag / xapv privilege escalation," 2014, <https://packetstormsecurity.com/files/125761/Array-Networks-vxag-xAPV-Privilege-Escalation.html>.
- [5] "Quantum dxi v1000 2.2.1 - static ssh key," 2014, <https://www.exploit-db.com/exploits/32372>.
- [6] "Ceragon fibear ip-10 ssh private key exposure (cve-2015-0936)," 2015, <https://gist.github.com/todb-r7/5d86ecc8118f9eeccc15>.
- [7] "Shapeshift lost \$230k in string of thefts, report finds - coindesk," 2016, <https://www.coindesk.com/digital-currency-exchange-shapeshift-says-lost-230k-3-separate-hacks>.
- [8] "Vmware vdp known ssh key," 2016, https://www.rapid7.com/db/modules/exploit/linux/ssh/vmware_vdp_known_privkey.
- [9] "Blasting_dictionary," 2017, https://github.com/rootphantomer/Blasting_dictionary.
- [10] "Cve-2016-7456," 2017, <https://packetstormsecurity.com/files/cve/CVE-2016-7456>.
- [11] "Ssh bad keys," 2017, <https://github.com/rapid7/ssh-badkeys>.
- [12] "Vmware vdp known ssh key," 2017, <https://packetstormsecurity.com/files/143883/VMware-VDP-Known-SSH-Key.html>.
- [13] "Cisco 2018 annual cybersecurity report," Tech. Rep., 2018.
- [14] "Charter communications inc," 2019, <https://db-ip.com/as20001>.
- [15] "Clickhouse — open source distributed column-oriented dbms," 2019, <https://clickhouse.yandex/>.
- [16] "Enterprise email service for business - ms exchange email," 2019, <https://products.office.com/en-us/exchange/email>.
- [17] "Glesys ab," 2019, <https://db-ip.com/as42708>.
- [18] "Google llc," 2019, <https://db-ip.com/as15169>.
- [19] "'hassh' - a profiling method for ssh clients and servers." 2019, <https://github.com/salesforce/hassh#hassh---a-profiling-method-for-ssh-clients-and-servers>.
- [20] "Looting of the fox: The story of sabotage at shapeshift," 2019, <https://news.bitcoin.com/looting-fox-sabotage-shapeshift/>.
- [21] "Packet storm," 2019, <https://packetstormsecurity.com/>.
- [22] "Sdaia," 2019, <https://wiki.ncsa.illinois.edu/display/cybersec/SDAIA>. [Online]. Available: <https://wiki.ncsa.illinois.edu/display/cybersec/SDAIA>
- [23] "Shapeshift," <https://shapeshift.io/#/coins>, 2019.
- [24] "Z-score definition," 2019, <https://www.investopedia.com/terms/z/zscore.asp>.
- [25] "Expert found a hardcoded ssh key in fortinet siem appliances," 2020, <https://securityaffairs.co/wordpress/96649/security/hardcoded-ssh-key-fortinet.html>.
- [26] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis *et al.*, "Understanding the mirai botnet," in *26th {USENIX} Security Symposium ({USENIX} Security 17)*, 2017, pp. 1093–1110.
- [27] P. Cao, "On preempting advanced persistent threats using probabilistic graphical models," *arXiv preprint arXiv:1903.08826*, 2019.
- [28] P. Cao, E. Badger, Z. Kalbarczyk, R. Iyer, and A. Slagell, "Preemptive intrusion detection: Theoretical framework and real-world measurements," in *Proc 2015 Symposium and Bootcamp on the Science of Security*. ACM, 2015, p. 5.
- [29] P. Cao, E. C. Badger, Z. T. Kalbarczyk, and R. K. Iyer, "A framework for generation, replay, and analysis of real-world attack variants," in *Proc Symposium and Bootcamp on the Science of Security*. ACM, 2016, pp. 28–37.
- [30] P. Cao, H. Li, K. Nahrstedt, Z. Kalbarczyk, R. Iyer, and A. J. Slagell, "Personalized password guessing: a new security threat," in *Proceedings of the 2014 Symposium and Bootcamp on the Science of Security*, 2014, pp. 1–2.
- [31] P. M. Cao, Y. Wu, S. S. Banerjee, J. Azoff, A. Withers, Z. T. Kalbarczyk, and R. K. Iyer, "CAUDIT: Continuous auditing of SSH servers to mitigate brute-force attacks," in *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. Boston, MA: USENIX Association, Feb. 2019, pp. 667–682. [Online]. Available: <https://www.usenix.org/conference/nsdi19/presentation/cao>
- [32] F. Dang, Z. Li, Y. Liu, E. Zhai, Q. A. Chen, T. Xu, Y. Chen, and J. Yang, "Understanding fileless attacks on linux-based iot devices with honeycloud," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2019, pp. 482–493.
- [33] V. Ghi ette, H. Griffioen, and C. Doerr, "Fingerprinting tooling used for {SSH} compromise attempts," in *22nd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2019)*, 2019, pp. 61–71.
- [34] R. Gummedi, H. Balakrishnan, P. Maniatis, and S. Ratnasamy, "Not-a-bot: Improving service availability in the face of botnet attacks," in *NSDI*, vol. 9, 2009, pp. 307–320.
- [35] M. Ion, B. Kreuter, E. Nergiz, S. Patel, S. Saxena, K. Seth, D. Shanahan, and M. Yung, "Private intersection-sum protocol with applications to attributing aggregate ad conversions." *IACR Cryptology ePrint Archive*, vol. 2017, p. 738, 2017.
- [36] D. Mashima, Y. Li, and B. Chen, "Who's scanning our smart grid? empirical study on honeypot data."
- [37] P. Muncaster, "Poorly secured ssh keys exposing firms to breaches - infosecurity magazine," 2017.
- [38] J. Owens and J. Matthews, "A study of passwords and methods used in brute-force ssh attacks," in *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2008.
- [39] H. N. Security, "More than 99% of cyberattacks rely on human interaction," [Online]. <https://www.helpnetsecurity.com/2019/09/10/cyberattacks-human-interaction/>, 2019.
- [40] D. X. Song, D. A. Wagner, and X. Tian, "Timing analysis of keystrokes and timing attacks on ssh," in *USENIX Security Symposium*, vol. 2001, 2001.
- [41] D. Wendlandt, D. G. Andersen, and A. Perrig, "Perspectives: Improving ssh-style host authentication with multi-path probing," in *USENIX Annual Technical Conference*, vol. 8, 2008, pp. 321–334.
- [42] D. Wheeler, "zxcvbn: Realistic password strength estimation," *Dropbox Tech Blog*, Apr. 2012.