Measuring the Prevalence of the Password Authentication Vulnerability in SSH

Ron Andrews
EECS Department, ITTC
University of Kansas
Lawrence, KS
terrapin@ku.edu

Dalton A. Hahn
EECS Department, ITTC
University of Kansas
Lawrence, KS
daltonhahn@ku.edu

Alexandru G. Bardas
EECS Department, ITTC
University of Kansas
Lawrence, KS
alexbards@ku.edu

***Abstract*—Securing and hardening network protocols and services is a resource-consuming and continuous effort. Thus, it is important to question how prolific known, mitigable features of those protocols are. The Secure Shell (SSH) protocol is a good example due to its known vulnerability in using password based authentication. We take a closer look at these configurations to identify how prevalent the use of password authentication is at an internet scale. We show that current scanning tools and services provide a starting point in evaluating prevalence, but need to be validated for specific implementations. We also demonstrate that it is possible to augment some of these tools and services to determine the prevalence of password authentication in SSH specifically. As part of our evaluation, we propose a novel method for probing an SSH service to establish if password authentication is allowed, without being intrusive or causing harm to the host. Finally, we show that our analysis has resulted in determining that more than 65% of the over 20 million SSH servers on the public internet allow password authentication.**

***Index Terms*—SSH, Secure Shell, Password Authentication, Authentication, Man-in-the-Middle, Assessment, Analysis, Prevalence, Measurement, Statistically Relevant**

I. Introduction

Vulnerabilities in networks, applications, and services are being identified and exploited everyday. It has become commonplace to hear of large-scale exploits of known and new vulnerabilities in systems and the efforts being put forth to mitigate the loss and impacts. Some systems, such as Continuous Auditing of SSH Servers to Mitigate Brute-Force Attacks (CAUDIT) [1], put forth large-scale and complex efforts to mitigate challenges with known services, acknowledging issues with the base service as well as evolving challenges, such as those described in [2, 3].

A service like SSH is a well-established and key component of many systems such as those described in [1, 4] - for administrators, users, and automated systems alike. A quick look at results from censys.io and shodan.io informs us that there are over 17 million SSH services responding on the public facing internet. This service is interesting as the RFC [5] for the protocol has carried an explicitly defined exploit for decades in that the username and password are transmitted in plain text during the authentication step - leaving the secure service vulnerable to a Man in the Middle attack (MitM). There are alternative ways of implementing SSH as a service to avoid some of these attacks. Even with the explicit vulnerability of password authentication in SSH [5], the practical application of this service continues to allow for its use.

Thus, determining the prevalence of the password authentication vulnerability on an internet-level scale can reveal important insights. Measuring the prevalence of this configuration provides insight not only into how widespread this vulnerability is, but also into the likely use of the classic credentials model in lieu of more secure techniques. First, we explored available resources online and previous literature, to find little in identifying this specific vulnerability at an internet scale. Resources such as Shodan [6] do not provide sufficient information to glean whether or not password authentication is available. Therefore, we looked to developing a method for establishing the prevalence of password authentication in SSH servers and a means for performing an internet scale assessment, while adhering to the tenants for ensuring "Good Internet Citizenship" [7].

The challenge in this was to develop a non-intrusive means to perform the assessment as well as to show whether or not these results were statistically relevant. Showing that the measurement is statistically relevant provides a clear indication that a known, mitigable vulnerability inherent in SSH is widespread on the public-facing internet. The contributions of our work include the following:

- Analysis on the prevalence of a known vulnerable authentication method for SSH, specifically that of password authentication, on an internet scale
- Review of existing tools and services to perform large scale assessments and development of an augmentation for those tools to specifically address the discovery and accounting of the prevalence of SSH services configured to allow password authentication in a non-intrusive and responsible way
- Statistical analysis of our results to ensure that our findings are statistically significant

II. Background and Related Work

There are tools and websites available for performing internet level scans of the internet for performing research on the prevalence of ports and protocols. Websites, such as Shodan and Censys [6, 8] provide interfaces for searching their results from continuously scanning the internet. Additionally, there are software tools available for performing internet-wide scans

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

for ports and protocols such as: ZMAP [9], and NMAP [10]. For our specific needs, we reviewed these tools to identify the most appropriate to meet our needs in performing our evaluation. In order to adequately assess them, we needed to understand what they would provide and what we needed based on how SSH password authentication works.

SSH: Developed and introduced by Tatu Ylonen in 1995 as a replacement for insecure platforms such as telnet [11, 12] and formalized in 2006 via RFCs 4250 through 4254 and RFC 4256 [5, 13]–[17]. This protocol has been widely used and researched over the years, including the identification of vulnerabilities, most with mitigations such as [18]. Unfortunately, there still persists a basic vulnerability - the transmission of user credentials in plain text [19].

An SSH connection, starts with an initial handshake. During this handshake, the server first attempts to establish the key exchange (kex) algorithms to determine the encryption to be used for the connection. Once the algorithm is agreed upon, the host key and cipher algorithms are agreed to followed by the exchange of the host keys. If a password parameter has not been passed in the connection, the server will attempt public-key authentication, by default. If no key is supplied (in addition to no password parameter), the connection is severed.

Shodan and Censys: *Shodan* [6] is a search engine specifically for inter-connected devices. This service performs a continual scan of the public facing internet, recording in their database the results, and then making this data available. Their search capability allows a user to enter in various information to query for, such as protocol, vendor, or service. As an example, entering in SSH in the search criteria returns a page informing us that there were 19,037,202 hosts that return SSH. These results cover banners, a variety of ports (both standard and non-standard) and a list of the IP addresses it found. Each IP address provides a link to a page that gives additional discovery data as well.

Censys [8] provides a similar capability as Shodan, though geared towards research and developed by the ZMAP scanner team. Performing the same search as used above gives 16,977,113 hosts that return SSH. Similar to Shodan, they also provide a link to a page for each IP address found where a user can look at the discovered data. One aspect of the subsequent page provided by Censys is that it does provide the banner data grabbed as well as other discovered metadata.

Scanners: *NMAP* [10] is a scanning tool designed for deep scans on a target machine or subnet of machines. This tool performs a deep scan for all 65,536 TCP ports and attempts to use the packet information to discover services, operating systems, filtering, and other characteristic data of the host being scanned. NMAP also includes a suite of support tools as well, such as nping and ndiff. Discovery can take up to 3 sec per port attempted as the scanner gives time for the host to respond following the SYN packet.

AMAP [20] is a scanning tool that follows the capabilities of NMAP,but goes a step further by adding in functionality to identify applications running on non-standard ports based on their trigger/response database.

ZMAP [9] is another scanning tool, built specifically for performing shallow scans, a single port, at internet level scales. Using a rate of 1.4 million packets per second, ZMAP is able to scan the entire internet in under 45 minutes [7].

MASSSCAN [21] is a shallow scanning tool, also built for quickly scanning the internet. Their method utilizes 10 million packets per second to achieve a full scan in under 6 minutes (done so using a custom TCP/IP stack and configuration).

SCANRAND [22] is a stateless TCP scanning tool which uses two processes to quickly scan the internet. One process sends SYN packets and records the addresses, while the other process leverages libpcap to review and label the responses.

UNICORNSCAN [23] is an asynchronous stateless port scanner that implements its own TCP/IP to quickly scan hosts and then utilize a tool like NMAP to analyze the ports found. This is done speed up the process by not cutting out the wait periods for SYN packet response timeouts.

SSH Implementation: Paramiko [24] is a Python implementation of the SSHv2 protocol as defined in [5, 14]–[16]. This implementation enables us to work with the connection between an SSH server and client so that we could assess communication and dialogue between them.

Previously published works looking at vulnerabilities in SSH focus on preventing Man-in-the-Middle Attacks and SSH brute force attacks as documented in [18, 19, 25]. Other works focus on detection such as [26, 27]. However, there appears to lack of survey or assessment papers to identify the prevalence of the well documented vulnerability of password authentication SSH servers.

III. System Model

In order to determine whether or not an SSH service allows for password based authentication, we needed to construct an environment that would enable us to test and validate responses to queries and requests. To accomplish this, we developed the following tests:

- Socket connection to default SSH server
- SSH probe using Paramiko with no arguments to connect to a default SSH server
- SSH probe with authentication parameters using Paramiko to connect to a default SSH server
- SSH probe with authentication parameters using Paramiko to connect to an SSH server with password auth. disabled[1]
- Assess public facing internet for SSH servers with password authentication enabled

These tests were performed in a controlled environment using virtual machines to create each scenario. We utilized Python to perform the connections. For the socket connection, we utilized a basic socket connection using Python. For the Paramiko probes, we used two configurations:

[1]This was done by setting *PasswordAuthentication* to *no* in the *sshd_config* file for the server

```
client.connect(
    addr,
    port = 22,
    timeout = 1
)
```

```
client.connect(
    addr,
    port = 22,
    password = '',
    timeout = 1,
)
```

The reason for this is due to the fact that without the 'password' argument, the default behavior of an SSH server attempts to process a key for authentication. Sending the empty password argument signals the SSH server to attempt to authenticate via password before falling back to keys.

Our initial testing provided results as expected, based on the SSH RFCs. Specifically, the socket connection and basic SSH proves did not provide any details regarding the specific authentication scheme(s) allowed by the SSH server. In fact, the socket connection did not provide any useful metadata as the host key exchange was not even initiated as access to the raw socket was denied.

The probe using Paramiko with no parameters returned '*Connection Error: No Authentication Methods Available*' with the connection data exchanged between the client and the server yielding no information on what authentication methods are supported. Our second test with Paramiko, included the empty parameter 'password', we are informed by the default SSH server that password authentication is enabled, '*Authentication (password) failed.*'. Executing the same test with password authentication disabled on the server, we clearly see password authentication is not permitted:

```
Authentication type (password) not permitted.
Allowed methods: [publickey]
```

This provided us with sufficient details on what to expect from the banners in the response from SSH servers. With this information, we leveraged both shodan.io and censys.io to search for SSH servers, looking through the metadata to for the identifiers we discovered in our local test bed. At a minimum, these services provide insight into the prevalence of SSH as a service on the public internet, giving us a benchmark to compare our results to.

IV. Problem Formulation

Results from both Shodan and Censys demonstrate that there are millions of SSH servers available on the public facing internet. Since both of these services provide a rolling window of results (each query being a snapshot in time of their database), we visited Shodan aperiodically over a 3 month period looking specifically at their *Total SSH*[2] search totals and specifically those given for *SSH Service*. Based on our findings, the results on Shodan[3] provide a fairly consistent population at a percent deviation from the mean of 0.80% for SSH and 0.88% for SSH services, as shown in Table I and plotted in Figure 1.

[2]Top Services on Shodan give SSH, 2222, 666, 2382, etc.; we elected to constrain our focus to SSH services as we were unable to get consistent results. For example a quick search for SSH may give a total of over 19 million results with over 700 thousand designated as 2222 - though a search for 2222 gives a total of over 53 thousand results.

[3]Note: current shodan.io estimates showed 19,184,084 SSH Services discovered as of 7 Oct 2019 - further demonstrating the consistency in prevalence.

TABLE I
SHODAN.IO CONSISTENCY

Date	SSH Totals	SSH Service	Percent
20 Oct 2018	21,403,815	19,828,963	92.64
27 Oct 2018	21,519,540	19,948,789	92.70
16 Nov 2018	21,438,121	19,871,084	92.69
09 Dec 2018	21,199,695	19,605,874	92.48
10 Dec 2018	21,199,695	19,605,874	92.48
11 Dec 2018	21,170,765	19,576,215	92.47
14 Dec 2018	21,118,762	19,520,895	92.43
19 Dec 2018	21,559,652	19,935,141	92.46
Mean	21,326,256	19,736,604	
Max	21,559,652	19,948,789	
Min	21,118,762	19,520,895	
StdDev	173,051	176,266	
% StdDev to Mean	0.80%	0.88%	
Measurements performed in Oct. - Dec. 2018 at the time of the data gathering phase are consistent with Shodan results from Oct. 2019			

In review of the details of a discrete record from Shodan, there are no metadata parameters or values provided which indicate (explicitly) whether or not password authentication is allowed. There are instances where we might infer that password authentication is allowed based on the algorithms supported in the negotiation of the connection. Based on our test bed results, we did not find this to be the case as the server may still support those algorithms and not allow authentication by password. Thus, the data we were able to glean from Shodan provided a benchmark on the discovered SSH servers on the public internet.

We next turned to Censys, seeking for a complimentary benchmark with explicit identification of whether or not password authentication is allowed by the responding servers. A search for SSH on Censys resulted in a total of 16,990,224 results, refining the search criteria to just those tagged by Censys as SSH servers[4] resulted in 16,298,773 results. We suspect the difference in collected data between the results has to do with the geographical coverage differences between the two services, as shown in Figures 3 and 4. However, we present both in order to provide a comprehensive analysis.

In review of the metadata recovered by the Censys search engine, we see their results do not include the password authentication information that we are seeking. In looking at their raw data results from the queries, it would appear that their probes are implemented similar to that of Shodan and our internal test without the authentication parameter set for password authentication.

With these results, it became necessary to develop a method by which we could perform scans similar to those of Shodan and Censys with the added parameter of *password* in the probe to gain the explicit results we required.

To develop our method we looked to our list of scanners and set forth to identify a scanner that would respect the tenets laid out by the ZMAP team in their internet level scanning paper [7]. In addition to these tenets, we wanted to ensure that we were not flooding our local network or any other network with the traffic. Based on the capabilities of each scanner we deduced the following:

[4](ssh) AND tags.raw: "ssh"

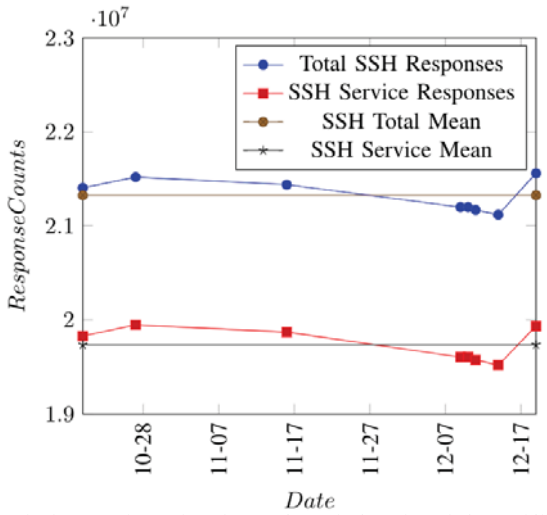

Fig. 1. Shodan Results - Plotted over time, depicts the relative stability in the prevalence of SSH responses on the public internet as discovered by the Shodan search engine.

- NMAP: overkill for scanning for a single port, would need to create a randomization script to select IPs (to ensure we were not generating continuous traffic to a single subnet at a time - spread the scanning), requires second probe of each IP to elicit required response
- AMAP: similar issues found as those with NMAP
- ZMAP: quick single port scanning with random selection within IP space, requires second probe of each IP to elicit required response; provides ability to throttle the scanning in order to not overwhelm the network
- MASSSCAN: similar to ZMAP with higher throughput
- SCANRAND: similar to ZMAP and MASSCAN, requires second probe to elicit required response
- UNICORNSCAN: similar to ZMAP and MASSCAN in probing - leverages NMAP to interpret response

We elected to use ZMAP due to its functionality necessary for our experimentation and the ability to easily throttle the bandwidth used. The ZMap Project [28] also includes a tool to read banner data based on the results of ZMAP, called ZGRAB2 [29]. We reviewed the capabilities by implementing this feature in our test bed and found that it had similar results to the other tools in that the connection request does not appear to include the *password* parameter, therefore not eliciting the required response from the SSH server.

Our solution was to develop a secondary probe following the our scan for possible hosts, using ZMAP, to retrieve the banners from the host. To do this we leveraged our testbed Paramiko script and modified it so that the username it provided was *researchTest* in order to make it clear that the intent of the connection request was not malicious but for research purposes. We stitched this together with our ZMAP results so that after the our initial scan was complete, we then executed the secondary probing. This was done to spread out the connections initiated by our project and to reduce the

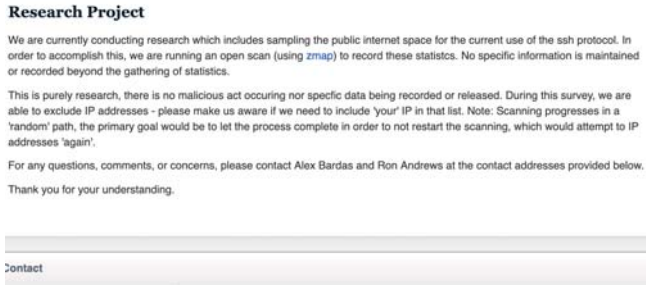

Fig. 2. Research web page - Provided to concisely explain the intent of the research project as well as provide users contact information.

overall impact of the investigation[5].

A. Ethical Considerations

We worked closely with our local network administrators and security operations center to ensure that our intent, goals, and methods were understood and agreed to. As mentioned in the previous section, we payed careful attention to how the tools we elected to employ would impact the network, both ours and external entities. Our implementation used ZMAP for the scanning with the packet rate dialed in to a low rate and then utilized the scan output to drive the order of our secondary probe. Additionally, we chose to implement our secondary probe as a non-threaded application, working each address sequentially, with only one attempt per host.

Our secondary probe, using Paramiko, was implemented with a username of *researchTest* in an effort to make it clear to admins of our intent. We also set up a web page, shown in Figure 2, clearly stating our purpose as well as whom to contact with any questions or requests. The hostname of our server was *researchproject* to further inform any hosts affected by our project.

In addition, which align with the tenets of "Good Internet Citizenship" [7], we also consulted [30, 31] as well as the definitions of sensitive personal data [32], personally identifiable information (PII) [33] in order to ensure that we maintain respect for individuals privacy and their resources. The data collected contains information from the SSH header and the standard handshake data exchange (e.g., kex, HostKey, Cipher, and MAC algorithms), as also seen on both the Shodan and Censys search engines. The only information unique to a specific host is the IP address and any custom banner created by the administrator of the SSH server.

Based on these observations, considerations, and our implementation, we assert that we are observing the privacy, ensured that no harm was incurred to a host during our research, and that we adhered to the tenets of "Good Internet Citizenship".

V. Problem Analysis

Based on the problem identified in the previous section with available tools for determining the prevalence of SSH servers

[5]We recognize that this led to some dynamic hosts not being available, we determined that this strengthens our results on prevalence as the likelihood that we are capturing a stable count of SSH servers is more likely.

Fig. 3. Censys - SSH Data Geographical Coverage [8]

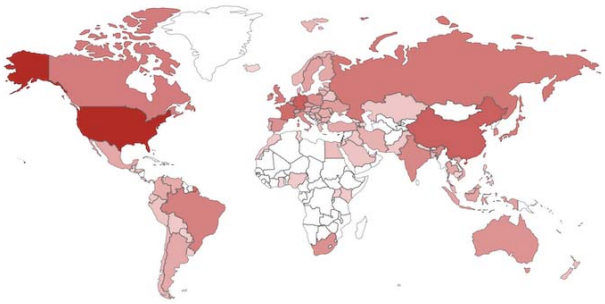

Fig. 4. Shodan - SSH Data Geographical Coverage [6]

with password authentication on the public internet, we implemented our approach, as described in the previous section. Our implementation started by performing a comprehensive scan of the public-facing IP address space using ZMAP via the command:

```
zmap -p 22 -T 1 -B 10M  output-fields=* -i enp0s25
```

The scan was started on 15 Nov 2018 at 11:42:44 and completed on 18 Nov 2018 at 08:49:35 CST with 24,516,371 responses, specifically for port 22, logged. A typical log entry, formatted as comma separated values (CSV), provide the total responding with an SSH occurring in their banner and those which respond as an SSH service.

To ensure that our findings were specific to SSH servers responding as with password authentication being enabled was statistically significant, we looked to identify a representative sampling using Cochrans [34, 35] and Slovins [36] (with Slovins formula being a simplified version of Cochrans) formulae for determining sufficiently large sample sizes based on a given population.

According to Cochran, for populations that are large, the following equation can be used to yield a representative sample size with a 98% confidence level = 2.05, p = estimated proportion that is present in the population for our initial hypothesis = 50%, and q = 1-p = 0.50.

$$n = \frac{Z^2 pq}{e^2} \tag{1}$$

Where:

- n = sample size
- Z = Z-score

This results in a sample size of 2637. Yamane [36] provides a simplified version of Cochrans formula to calculate sample sizes (a.k.a., Slovins formula), which explicitly includes the population in the calculation:

$$n = \frac{N}{1 + Ne^2} \tag{2}$$

Where:

- n = sample size
- N = population = 24,516,371
- e = margin of error = 0.02 (98% confidence)

Resulting in a needed sample size of 2500 entries. With this, we concluded that a sample of size 3000 or greater would be both necessary and sufficient.

We then executed our Python script, using Paramikos SSH client capability, to attempt to probe SSH hosts collected from our zmap experiments. This script executed for two days and logged probes of 43,945 hosts. Of those hosts that responded to our secondary probe, 70% (29,970) were still active on port 22 and responded to our request. We then searched through the logs from our attempts for instances where *userauth is OK* was a response from the SSH server and then excluded those entries which did not allow password based authentication (*password not permitted*). Splitting this data into 8 groups, allowed for 8 sample sets to compare for statistical significance. Figure 5 shows the norm, mean, and standard deviation of the complete dataset results, fit on a normal probability distribution function (PDF).

Our initial hypothesis for which we are trying to establish statistical significance is that more than 50% of SSH services offered on the public internet (answering to port 22) allow for password authentication. For our results, we desire a confidence level of 98% (α = 0.02) to show statistical significance. Therefore we set:

$$Hypothesis : H_a = p > 50\% \tag{3}$$

$$NullHypothesis : H_0 = p \leq 50\% \tag{4}$$

We implemented the Z-Test as defined by:

$$z = \frac{p - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \tag{5}$$

Where:

- z = Test statistics
- n = Sample size
- $p0$ = Null hypothesized value (values $\leq$ 50%)
- p = Observed proportion

Thus, our decision rule for this two-tailed test is: If the result of the z-test, z, is less than or greater than our z-score, Z, then we reject the null hypothesis. For all 8 sets, the null hypothesis, H_0, return false - therefore, resulting in our hypothesis being true. In working with these results, we found that, based on our data, more than 65% of all publicly facing SSH servers allow for password based authentication with a 98% confidence interval. It is important to reiterate here that the density of the sample varies over time, as can be seen in the data from Shodan and our own findings.

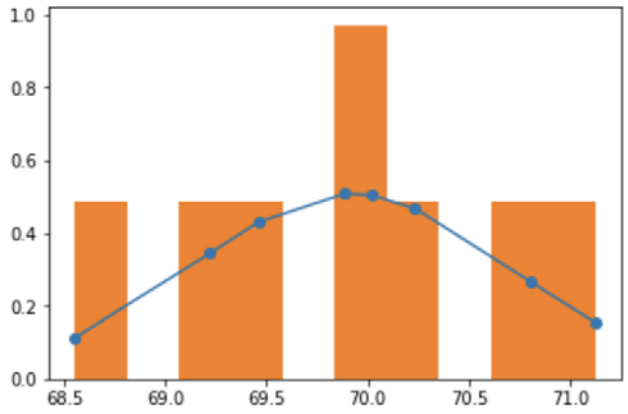

Fig. 5. Normal probability distribution function (PDF) histogram - Constructed from the sampled data with the probability graphed against the percentage initially hypothesized, showing that the resulting percentage (or prevalence) being much higher than the initial estimate of 50%, in fact the results show the prevalence being greater than 65%

Based on our analysis of the data, it is clear that there is statistically significant (more than 65%) number of SSH services offering password based authentication available through the public facing internet.

VI. Discussion and Limitations

In retrospect of our approach to performing the prevalence assessment we make note of the following observations. Due to performing the scan and probe separately, coupled with the probe being performed linearly, our counts of the prevalence are likely lower than the true numbers as there were many servers that responded to the scan but were unavailable during the probing period. Though based on our sample space, using Cochran's formula, this deviation should be mitigated in our significance calculations. Additionally, this also informs us that our finding of 65% is the lower bar of the prevalence - indicating that an instantaneous snapshot would likely yield a much higher penetration.

The results of our analysis begs the question of why is this the case, why are there so many instances of this service configured to allow for the most vulnerable scenario the protocol offers? Is it due to reasons such as 'not knowing any better', 'user preference' to 'just easier to administrate'? This would be an interesting exploration in performing a survey to inquire user and administrator preference and rationale. Results of the survey could be used to guide standard revisions, default configurations, for SSH as well as other services offering classic credentials (username and password) authentication.

VII. Conclusion

Our work demonstrates that the prevalence of a service such as SSH can be established using a non-intrusive approach and that existing frameworks/tools can be augmented for this purpose. We have shown that our analysis has resulted in finding that there is a statistically significant number of the more than 20 million SSH servers on the public internet, over 65% are configured to allow password based authentication.

References

[1] P. M. Cao *et al.*, "Caudit: Continuous auditing of ssh servers to mitigate brute-force attacks," in *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation NSDI 19*, Boston, MA, USA, Feb. 2019, pp. 667–682. [Online]. Available: https://www.usenix.org/system/files/nsdi19-cao.pdf

[2] C. Kolias *et al.*, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7971869

[3] S. Schechter *et al.*, "Inoculating ssh against address harvesting," in *NDSS Symposium 2006*, Feb. 2006. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2017/09/Inoculating-SSH-Against-Address-Harvesting-Stuart-E.-Schechter.pdf

[4] N. DeMarinis *et al.*, "Scanning the internet for ros: A view of security in robotics research," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 8514–8521. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8794451

[5] T. Ylönen, "The secure shell (ssh) protocol architecture," Internet Requests for Comments, RFC Editor, RFC 4251, Jan. 2006. [Online]. Available: https://tools.ietf.org/html/rfc4251

[6] Shodan. (2019, Sep.). [Online]. Available: https://www.shodan.io

[7] Z. Durumeric *et al.*, "Zmap: Fast internet-wide scanning and its security applications," in *Proceedings of the 22nd USENIX Security Symposium*, Washington, D.C., USA, Aug. 2013, pp. 605–619. [Online]. Available: https://www.usenix.org/system/files/conference/usenixsecurity13/sec13-paper_durumeric.pdf

[8] Censys. (2019, Sep.). [Online]. Available: https://censys.io

[9] Zmap. [Online]. Available: https://github.com/zmap/zmap

[10] Nmap. [Online]. Available: https://nmap.org/

[11] D. Barrett *et al.*, *SSH, The Secure Shell: The Definitive Guide, 2nd Edition*. 981 Chestnut Street, Newton, MA 02164, USA: OReilly & Associates, Inc., 2009.

[12] T. Ylonen. (2019, Apr.) The new skeleton key: changing the locks in your network environment. [Online]. Available: https://www.scmagazineuk.com/article/1481613

[13] S. Lehtinen, "The secure shell (ssh) protocol assigned numbers," Internet Requests for Comments, RFC Editor, RFC 4250, Jan. 2006. [Online]. Available: https://tools.ietf.org/html/rfc4250

[14] T. Ylönen, "The secure shell (ssh) authentication protocol," Internet Requests for Comments, RFC Editor, RFC 4252, Jan. 2006. [Online]. Available: https://tools.ietf.org/html/rfc4252

[15] ——, "The secure shell (ssh) transport layer protocol," Internet Requests for Comments, RFC Editor, RFC 4253, Jan. 2006. [Online]. Available: https://tools.ietf.org/html/rfc4253

[16] ——, "The secure shell (ssh) connection protocol," Internet Requests for Comments, RFC Editor, RFC 4254, Jan. 2006. [Online]. Available: https://tools.ietf.org/html/rfc4254

[17] ——, "Generic message exchange authentication for the secure shell protocol (ssh)," Internet Requests for Comments, RFC Editor, RFC 4256, Jan. 2006. [Online]. Available: https://tools.ietf.org/html/rfc4256

[18] D. Wendlandt *et al.*, "Perspectives: Improving ssh-style host authentication with multi-path probing," in *USENIX 2008 Annual Technical Conference (ATC)*, Berkeley, CA, USA, 2008, pp. 321–334. [Online]. Available: http://dl.acm.org/citation.cfm?id=1404014.1404041

[19] S. C. Security. (2019, Apr.) Man-in-the-middle attack. [Online]. Available: https://www.ssh.com/attack/man-in-the-middle

[20] Amap. [Online]. Available: https://github.com/vanhauser-thc/THC-Archive/tree/master/Tools

[21] Masscan. [Online]. Available: https://github.com/robertdavidgraham/masscan

[22] Scanrand. [Online]. Available: https://manned.org/scanrand/b9a07a7a

[23] Unicornscan. [Online]. Available: https://github.com/dneufeld/unicornscan

[24] Paramiko. [Online]. Available: http://www.paramiko.org/

[25] J. Beling, "Conducting ssh man in the middle attacks with sshmitm," Global Information Assurance Certification Paper, SANS Institute, Tech. Rep., 2002. [Online]. Available: https://www.giac.org/paper/gsec/2034/conducting-ssh-man-middle-attacks-sshmitm/103515

[26] R. Hofstede *et al.*, "Ssh compromise detection using netflow/ipfix," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 20–26, Oct. 2014. [Online]. Available: http://doi.acm.org/10.1145/2677046.2677050

[27] R. J. McCaughey, "Deception using an ssh honeypot," Master's Thesis, NAVAL POSTGRADUATE SCHOOL, Monterey, CA, Sep. 2017.

[28] T. Z. Team. (2019, Sep.) The zmap project. [Online]. Available: https://zmap.io
[29] Zgrab2. [Online]. Available: https://github.com/zmap/zgrab2
[30] M. Allman and V. Paxson, "Issues and etiquette concerning use of shared measurement data," in *ACM SIGCOMM/USENIX Internet Measurement Conference*, Oct. 2007. [Online]. Available: http://www.icir.org/mallman/pubs/AP07/AP07.pdf
[31] Z. Durumeric, E. Wustrow, and J. A. Halderman, "Zmap: Fast internet-wide scanning and its security applications," in *Proceedings of the 22Nd USENIX Conference on Security*, ser. SEC'13. Berkeley, CA, USA: USENIX Association, 2013, pp. 605–620. [Online]. Available: http://dl.acm.org/citation.cfm?id=2534766.2534818
[32] E. COMMISSION. (2019, Apr.) The general data protection regulation (gdpr) regulation (eu) 2016/679: European commission submission on us department of commerce?s proposed approach to consumer privacy. [Online]. Available: https://ec.europa.eu/info/law/law-topic/data-protection/
[33] U. S. U. C. of Federal Regulations (CFR). (2019, Apr.) §200.79 spersonally identifiable information (pii). [Online]. Available: https://www.govinfo.gov/content/pkg/CFR-2014-title2-vol1/xml/CFR-2014-title2-vol1-sec200-79.xml
[34] W. G. Cochran, *Sampling Techniques, 2nd Edition*. New York: John Wiley and Sons, Inc., 1963.
[35] G. D. Israel. (2003, Jun.) Determining sample size. [Online]. Available: https://www.tarleton.edu/academicassessment/documents/Samplesize.pdf
[36] Yamane and Taro, *Statistics, An Introductory Analysis*. New York: New York: Harper and Row, 1967.