irpSSHa: Identifying and Reporting Potential SSH Attackers from IP Flow Logs

Jenny Louthan

January 16, 2018

Abstract

ABSTRACT HERE

1 Introduction

The Secure Shell (SSH) protocol is a popular security tool commonly enabled on hosts to supply trusted users with essential capabilities like remote login, file transfer, remote command execution, and other features requiring secure access. SSH helps guarantee security by encrypting traffic to prevent eavesdropping, providing client-side host key validation to protect against host IP spoofing, employing session integrity checks to render connection hijacking ineffective—excepting denial-of-service attacks—and more [1]. However, SSH alone cannot prevent all possible attacks. Since SSH operates over TCP, weaknesses in TCP and IP that allow denial-of-service attacks can be exploited to compromise legitimate SSH connections, for example, with a SYN flood or spoofed TCP reset packet [1]. Another potential vulnerability for many hosts using SSH is the ubiquitous brute force attack. In these attacks—often preceded by unwelcome port scans probing for open SSH servers—an adversary attempts to gain access to a host by automating a process to guess username and password combinations, often using a dictionary attack to try more common combinations first. An attacker commonly makes many login attempts within a relatively short time window, or they may mount a stealthy distributed attack campaign that is more difficult to distinguish from legitimate users rarely failing to authenticate over time [2]. The tool presented in this paper focuses on the former, although potential future work could deploy the proposed reporting framework for use with existing algorithms used to detect the more subtle attacks.

If an IP is reachable over the Internet and the host has enabled SSH, it will assuredly be a target for large numbers of brute force attack attempts. One honeypot server that was set up to monitor brute force SSH attack attempts experienced 6899 login attempts in 22 days, observing a total of 2741 unique usernames and 3649 unique passwords guessed [3]. Another organization's honey pot server logged 15000 brute force attempts in 7 days (in both cases, the most common username tried was "root" by a majority) [4]. In another experiment, the organization configured five IPv4 servers with credentials root/password to monitor how much time would pass before becoming compromised; the first was hacked in 12 minutes [5].

Of course, if the proper security measures are taken, these attacks can be mitigated, but attempts to gain unauthorized access will not cease just because they are unsuccessful. A host can eliminate the possibility that a brute force attack will be successful by disabling username/password authentication and instead requiring public-key authentication for SSH access. To reduce successful port scans, the administrator can use a port other than 22 for SSH traffic. Additionally, the SSH port can be configured to reject all traffic originating from an IP not included in a specified whitelist.

However, even if a host is not vulnerable to SSH brute force attacks, malicious login attempts may still occur, albeit unsuccessfully. Although this may be of no direct consequence to the secured host, the malicious source IPs instigating the attacks have many targets, some of which may remain vulnerable. Secure hosts can go a step beyond protecting themselves and help to identify malicious IPs via their TCP/IP activity and report them to organizations and communities dedicated to maintaining public blacklists of abusive IPs. Thus, there is a need for a tool that can run on an end host, analyze IP logs to identify IPs engaged in SSH brute force attacks, and, when authorized, automatically file reports on these IPs with a reputable public blacklist.

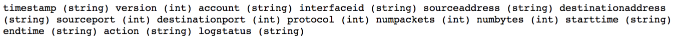
One such blacklist is maintained by the AbuseIPDB project, a public online database dedicated to combatting abusive activity on the internet [6]. AbuseIPDB maintains a blacklist of abusive IPs available for webmasters, sys admins, and anyone interested in IP security. The tool introduced in this paper, irpSSHa, allows users to query large sets of IP traffic flow data for potential SSH attackers and cross references these results with AbuseIPDB to identify potential adversarial IPs. Once identified, the user can use irpSSHa's interactive command prompt to file reports for any and all of the suspicious IPs with AbuseIPDB directly.

2 Design

2.1 Service Integrations

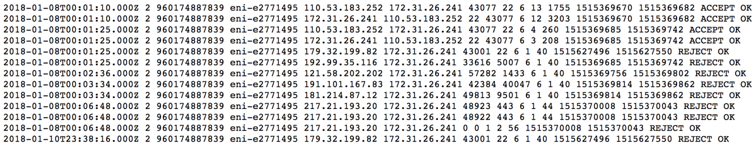
irpSSHa is designed to be as simple as possible, while still providing users the ability to query large datasets, i.e., flow logs containing thousands or millions of records, efficiently, as well as the option of storing the logs and query results securely in the cloud for future reference. This is achieved by integrating with the services Amazon Simple Secure Storage (S3) and Amazon Athena, both products offered by Amazon Web Services (AWS). S3 stores objects consisting of files and metadata in online storage containers called buckets that offer fine-grained access control rules. Athena is a service that allows running standard SQL queries against formatted data stored in S3; Athena is serverless and scales automatically to ensure ad-hoc queries remain performant regardless of dataset size. Integrating with these services means that users of irpSSHa must be able to provide valid AWS credentials for an account with access to both S3 and Athena. This restriction is acknowledged as a tradeoff for the performance, reliability, and ease of querying and storing large amounts of data that the services bring to irpSSHa. Future work could involve exploring ways to evolve irpSSHa to have fewer or no third party dependencies.

As necessitated by the automatic abusive IP reporting functionality in irpSSHa, the tool also integrates with the AbuseIPDB API. In addition to reporting IPs, irpSSHa queries AbuseIPDB for data on file about IPs identified as potential attackers from the input data in order to corroborate suspicions that a given IP is engaging in an SSH brute force attack. Using a free account with AbuseIPDB incurs a rate limit of 1000 requests per day, so future extended work would require an account tier upgrade.



```
timestamp (string) version (int) account (string) interfaceid (string) sourceaddress (string) destinationaddress
(string) sourceport (int) destinationport (int) protocol (int) numpackets (int) numbytes (int) starttime (string)
endtime (string) action (string) logstatus (string)
```

Figure 1: Default input format

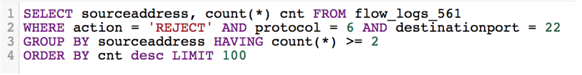


```
2018-01-08T00:01:10.000Z 2 960174887839 eni-e2771495 110.53.183.252 172.31.26.241 43077 22 6 13 1755 1515369670 1515369682 ACCEPT OK
2018-01-08T00:01:10.000Z 2 960174887839 eni-e2771495 172.31.26.241 110.53.183.252 22 43077 6 12 3203 1515369670 1515369682 ACCEPT OK
2018-01-08T00:01:25.000Z 2 960174887839 eni-e2771495 110.53.183.252 172.31.26.241 43077 22 6 4 260 1515369685 1515369742 ACCEPT OK
2018-01-08T00:01:25.000Z 2 960174887839 eni-e2771495 172.31.26.241 110.53.183.252 22 43077 6 3 208 1515369685 1515369742 ACCEPT OK
2018-01-08T00:01:25.000Z 2 960174887839 eni-e2771495 179.32.199.82 172.31.26.241 43001 22 6 1 40 1515627496 1515627550 REJECT OK
2018-01-08T00:01:25.000Z 2 960174887839 eni-e2771495 192.99.35.116 172.31.26.241 33616 5007 6 1 40 1515369685 1515369742 REJECT OK
2018-01-08T00:02:36.000Z 2 960174887839 eni-e2771495 121.58.202.202 172.31.26.241 57282 1433 6 1 40 1515369756 1515369802 REJECT OK
2018-01-08T00:03:34.000Z 2 960174887839 eni-e2771495 191.101.167.83 172.31.26.241 42384 40047 6 1 40 1515369814 1515369862 REJECT OK
2018-01-08T00:03:34.000Z 2 960174887839 eni-e2771495 181.214.87.12 172.31.26.241 49813 9501 6 1 40 1515369814 1515369862 REJECT OK
2018-01-08T00:06:48.000Z 2 960174887839 eni-e2771495 217.21.193.20 172.31.26.241 48923 443 6 1 44 1515370008 1515370043 REJECT OK
2018-01-08T00:06:48.000Z 2 960174887839 eni-e2771495 217.21.193.20 172.31.26.241 48922 443 6 1 44 1515370008 1515370043 REJECT OK
2018-01-08T00:06:48.000Z 2 960174887839 eni-e2771495 217.21.193.20 172.31.26.241 0 0 1 2 56 1515370008 1515370043 REJECT OK
2018-01-10T23:38:16.000Z 2 960174887839 eni-e2771495 179.32.199.82 172.31.26.241 43001 22 6 1 40 1515627496 1515627550 REJECT OK
```

Figure 2: Example of valid input in default format

2.2 Input Requirements

IP flow logs are required as input to irpSSHa in order to report on attackers. Because SSH uses TCP, logs containing only TCP flows would be sufficient. To simplify the tool, rather than reading streams of data at the packet resolution, irpSSHa requires packets to be aggregated into flows prior to running the analysis. Output generated from existing tools like NetFlow [CITE?] and AWS VPC Flow Logs can be used as input, or individual packet traffic could be preprocessed before running irpSSHa. The format of the input must match the expected format of tool in order to run ad-hoc SQL queries as needed. The default format is modeled after the output from VPC Flow Logs and can be seen in Figure 1. Sample input adhering to this format can be seen in Figure 2. The current format may appear a bit verbose; however, changing the format to match flow input data from different sources requires only a trivial change to the `create_table` method in `athena_helper.py`. More sample input files can be found in the GitHub repository.



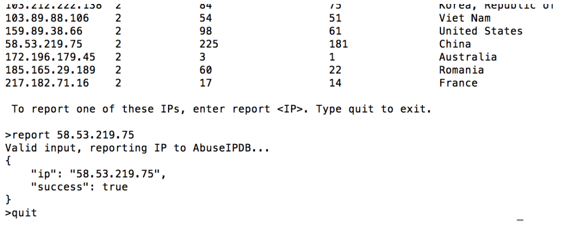
```
SELECT sourceaddress, count(*) cnt FROM flow_logs_561
WHERE action = 'REJECT' AND protocol = 6 AND destinationport = 22
GROUP BY sourceaddress HAVING count(*) >= 2
ORDER BY cnt desc LIMIT 100
```

Figure 3: Default SQL query used to identify potential attackers

2.3 Detailed Implementation

irpSSHa is written in Python and relies on the AWS SDK for Python, boto3, and the requests module for its service integrations.

The tool begins execution by uploading the input file to S3. This involves creating a bucket for the irpSSHa data if it does not exist yet, and uploading the input to a new uniquely named folder. Next, a database is created in Athena if one for irpSSHa does not exist, and then a table is created in the database with the default or supplied field format. This table automatically imports the data from the appropriate bucket and folder in S3. At this point, the flow logs can be queried with standard SQL. irpSSHa executes the query in Figure 3 to obtain a list of potential threats.

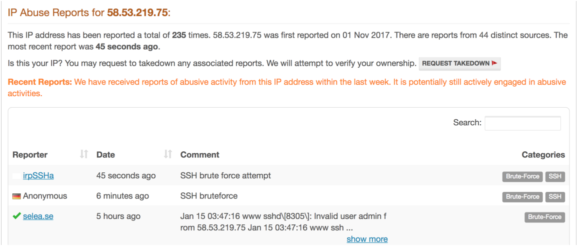


```
103.212.222.138   2          84                 75                Korea, Republic of
103.89.88.106     2          54                 51                Viet Nam
159.89.38.66      2          98                 61                United States
58.53.219.75      2          225                181               China
172.196.179.45    2          3                  1                 Australia
185.165.29.189    2          60                 22                Romania
217.182.71.16     2          17                 14                France

 To report one of these IPs, enter report <IP>. Type quit to exit.

>report 58.53.219.75
Valid input, reporting IP to AbuseIPDB...
{
    "ip": "58.53.219.75",
    "success": true
}
>quit
```

Figure 4: Example command prompt interaction



IP Abuse Reports for **58.53.219.75**:

This IP address has been reported a total of **235** times. 58.53.219.75 was first reported on 01 Nov 2017. There are reports from 44 distinct sources. The most recent report was **45 seconds ago**.

Is this your IP? You may request to takedown any associated reports. We will attempt to verify your ownership. REQUEST TAKEDOWN

Recent Reports: We have received reports of abusive activity from this IP address within the last week. It is potentially still actively engaged in abusive activities.

Search:

Reporter	Date	Comment	Categories
irpSSHa	45 seconds ago	SSH brute force attempt	Brute-Force SSH
Anonymous	6 minutes ago	SSH bruteforce	Brute-Force SSH
✓ selea.se	5 hours ago	Jan 15 03:47:16 www sshd\[8305\]: Invalid user admin from 58.53.219.75 Jan 15 03:47:16 www ssh ... show more	Brute-Force

Figure 5: Auto-generated report from irpSSHa on AbuseIPDB site

Once these IPs are identified by the query, irpSSHa issues a GET request to the AbuseIPDB API for each one and records the number of times it has been reported for abusive activity in the past 60 days, as well as the number of those reports that implicated the IP in attacks with the "SSH" or "Brute Force" label, or both. After this information is obtained from the API for all IPs returned by the query, the findings are presented to the user, along with an interactive prompt on the command line as in Figure 4. The user then has the option to review the results and select which, if any, of the source IPs to report. Entering `report <source IP>` with one of the IPs from the output will submit a report to AbuseIPDB via an HTTP POST. The report is submitted with the default or custom comment and is automatically tagged with categories "SSH" and "Brute Force". The report is then immediately viewable on the AbuseIPDB site, see Figure 5.

2.4 Additional Requirements

It is recommended that the destination hosts for the input flows be secured enough to reject unauthorized SSH traffic. In particular, ssh login with username/password should be disabled and only whitelisted IPs should be allowed, as discussed in section [SECTION 1 REFERENCE].

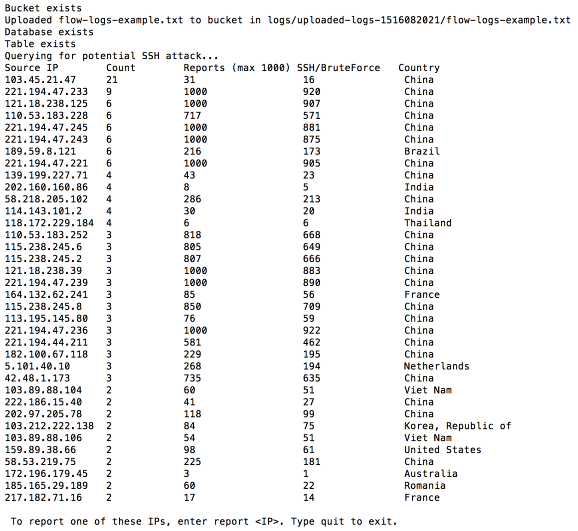
2.5 Source Code

The irpSSHa source code is available on GitHub: https://github.com/jlouthan/irpSSHa.

3 Evaluation

To evaluate irpSSHa, two IP flow log datasets were used as input. The flow logs were collected from VPC Flow Logs monitoring on an AWS EC2 t1.micro instance running Ubuntu 14.04 in the US East (N. Virginia) region. The first input dataset spanned a time period of four days, from midnight GMT on January 8th, 2018, to midnight GMT on January 12th, 2018. The second spanned a time period of four hours, from 1:00am GMT on January 15th, 2018, to 5:00am GMT on January 15th, 2018. The datasets included 13058 and 476 flow records, respectively. The irpSSHa output for each can be seen in Figure 6 and Figure 7

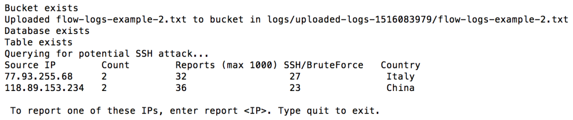
The output is separated into five columns. The Source IP column contains distinct IPs that were identified as potential SSH brute force attackers within the input flows. The Count column contains



```
Bucket exists
Uploaded flow-logs-example.txt to bucket in logs/uploaded-logs-1516082021/flow-logs-example.txt
Database exists
Table exists
Querying for potential SSH attack...
Source IP        Count        Reports (max 1000) SSH/BruteForce   Country
103.45.21.47     21           31                  16                China
221.194.47.233   9            1000                920               China
121.18.238.125   6            1000                907               China
110.53.183.228   6            717                 571               China
221.194.47.245   6            1000                881               China
221.194.47.243   6            1000                875               China
189.59.8.121     6            216                 173               Brazil
221.194.47.221   6            1000                905               China
139.199.227.71   4            43                  23                China
202.160.160.86   4            8                   5                 India
58.218.205.102   4            286                 213               China
114.143.101.2    4            30                  20                India
118.172.229.184  4            6                   6                 Thailand
110.53.183.252   3            818                 668               China
115.238.245.6    3            805                 649               China
115.238.245.2    3            807                 666               China
121.18.238.39    3            1000                883               China
221.194.47.239   3            1000                890               China
164.132.62.241   3            85                  56                France
115.238.245.8    3            850                 709               China
113.195.145.80   3            76                  59                China
221.194.47.236   3            1000                922               China
221.194.44.211   3            581                 462               China
182.100.67.118   3            229                 195               China
5.101.40.10      3            268                 194               Netherlands
42.48.1.173      3            735                 635               China
103.89.88.104    2            60                  51                Viet Nam
222.186.15.40    2            41                  27                China
202.97.205.78    2            118                 99                China
103.212.222.138  2            84                  75                Korea, Republic of
103.89.88.106    2            54                  51                Viet Nam
159.89.38.66     2            98                  61                United States
58.53.219.75     2            225                 181               China
172.196.179.45   2            3                   1                 Australia
185.165.29.189   2            60                  22                Romania
217.182.71.16    2            17                  14                France

 To report one of these IPs, enter report <IP>. Type quit to exit.
```

Figure 6: Output from running irpSSHa with the first example IP flow dataset



```
Bucket exists
Uploaded flow-logs-example-2.txt to bucket in logs/uploaded-logs-1516083979/flow-logs-example-2.txt
Database exists
Table exists
Querying for potential SSH attack...
Source IP        Count        Reports (max 1000) SSH/BruteForce   Country
77.93.255.68     2            32                  27                Italy
118.89.153.234   2            36                  23                China

 To report one of these IPs, enter report <IP>. Type quit to exit.
```

Figure 7: Output from running irpSSHa with the second example IP flow dataset

the number of flows in the input that contained unauthorized SSH access attempts from the IP. The Reports column indicates how many times in the past 60 days the source IP has been reported to AbuseIPDB. A limitation of the AbuseIPDB API is that the maximum value returned for this field is 1000, so source IPs with 1000 reports in the output were very likely reported a significant number of times *more* than 1000. The data in the fourth column represents the number of the reports that indicated the IP was suspected of an SSH attack, brute force attack, or both. The last column contains the name of the country in which the source IP originates.

As seen in the output, the smaller dataset indicated two IPs potentially involved in SSH brute force attack attempts. In both executions of irpSSHa, the minimum threshold for observing unauthorized SSH access attempts was two flows. Each IP in the smaller input's results was reported to AbuseIPDB at least 30 times within the last 60 days, with the majority of each implicating the IP in SSH and/or brute force attacks. The output for the larger dataset identifies 36 different source IPs as potential attackers, with 8 of them having been reported at least 1000 times within the last 60 days to AbuseIPDB. The potential attackers were responsible for a total of 145 unauthorized SSH access attempts (the sum of the Count column). If we decrease the minimum threshold from two to one, AbuseIPDB rate limits are quickly exceeded, but 101 more IPs are identified from the larger dataset, bringing the total up to 246 potential brute force attack attempts in four days. While not quite as large as some reports using dedicated honeypots, these numbers are consistent with what would be expected. In addition, we observe that, of the 36 source IPs in the larger input's table, 23 or about 64% of them originate in China.

4 Related Work

Full featured intrusion detection systems (IDSs) such as Snort [7] and Bro [8] can run real time packet analysis on an end host with filters to implicate traffic in potential SSH brute force attacks. Adding the ability to run queries on traffic in real time is included in the scope of future work for irpSSHa. However, for use cases where historical traffic data from any host needs to be analyzed for attacks and reported upon, irpSSHa would be ideal over complex IDS systems running on destination hosts.

The network telemetry system Sonata [12] offers a query interface to allow administrators performant access to traffic analytics that could feasibly be used to identify potential SSH attackers in a similar manner as irpSSHa's Athena queries. However, Sonata does not run on the end host, and a reporting mechanism would need to be added on top of Sonata to add abusive IPs to a blacklist.

Several smaller services providing intrusion prevention are perhaps the most comparable to irpSSHa in goals and functionality. These services, including tools like DenyHosts [9], sshguard [10], and the popular Fail2Ban [11], scan log files on the end host for signs of malicious activity and block IPs of repeat offenders. Some have configuration options for reporting attacker IPs to a system administrator. The main goals of these tools is to secure the host on which they are running; this is in contrast to irpSSHa, which aims primarily to identify abusive IPs for the purpose of sharing with reputable public blacklists.

5 Conclusions

References

[1] Daniel J. Barrett and Richard E. Silverman. *The Secure Shell: The Definitive Guide*. O'Reilly Associates, Inc, Sebastopol, CA, 2011.
`https://docstore.mik.ua/orelly/networking_2ndEd/ssh/copyrght.htm`

[2] Mobin Javed and Vern Paxson. Detecting Stealthy, Distributed SSH Brute-Forcing. in *ACM CCS*. 2013.

[3] Christian Seifert. Analyzing Malicious SSH Login Attempts.
`https://www.symantec.com/connect/articles/analyzing-malicious-ssh-login-attempts`

[4] Daniel Cid. SSH Brute Force – The 10 Year Old Attack That Still Persists.
`https://blog.sucuri.net/2013/07/ssh-brute-force-the-10-year-old-attack-that-still-persis`

[5] Daniel Cid. SSH Brute Force Compromises Leading to DDoS.
`https://blog.sucuri.net/2016/09/ssh-brute-force-compromises-leading-to-ddos.html`

[6] AbuseIPDB.
`https://www.abuseipdb.com/`

[7] Snort.
`https://www.snort.org/`

[8] Bro.
`https://www.bro.org/`

[9] Deny Hosts.
`http://denyhosts.sourceforge.net/`

[10] sshguard.
`https://www.sshguard.net/`

[11] Fail2Ban.
`https://www.fail2ban.org`

[12] Arpit Gupta, Rob Harrison, Ankita Pawar, Marco Canini, Nick Feamster, Jennifer Rexford, Walter Willinger. Sonata: Query-Driven Streaming Network Telemetry. 2017.