DBFST: Detecting Distributed Brute Force Attack on a Single Target

AL-Zwuiany Muhanad M. K., Prof. Huang Dongjun

Abstract— While reading this paper, a lot of servers may be coming under attack by hackers at different sites of the Internet. The actions of these hackers have different motivations and purposes and brute force attack is one of common ways of attack deployed by hackers. Secure Shell (SSH) is one of the widespread attacked servers.

SSH tunnels are commonly used to provide two types of privacy protection. First one is protecting the privacy of the data being exchanged between two peers, such as passwords and so on. The second one is protect the privacy of the behavior of end-users, by preventing an unauthorized observer from detecting which application protocol is being transported by the SSH tunnel.

Wherefore it has become the default remote access method for those using UNIX systems. It is very common for public Internet servers to experience attacks that attempt use brute force to obtain usernames and passwords combinations via SSH. In this paper we shall examine these attacks depending on SSH log file to find unsuccessful logins and then establish if these unsuccessful authorized IP's belong to attackers or to trusted users. Then develop a strategy to determine who should transact with the IP addresses based on the IP kind, and prevent the attackers IP's from attempting to login to the system.

.

Index Terms— Brute force, SSH, Iptables, Fail2ban, Denyhosts and white list.

1 INTRODUCTION

IT is often difficult to prevent brute force attacks because brute force attack tries a very large number of keys, on base try and error method [1].

Unlike the dictionary attack, where there is chosen a target, this kind of attack is a systematically targets the entire search space. Passwords may be easily broken, if the key volume is small. Brute-force attack consumes significant amount of time when the main dimension is big and password is strong. A computer program or ready-made programs are commonly used to perform a brute-force attack. The computer must be configured very well to do the brute force attack faster and more efficiently. The attempt should start with one digit password number and cover all possibilities in a worst case [2].

There are many servers attacked with brute force such as SSH, FTP, SMTP, and more, in this research will address SSH brute force attack.

The SSH protocol is one of the most widely used mechanisms designed to secure many of the activities that are carried out on the Internet[3]. It is used to secure remote shell operations by means of cryptographic authentication mechanisms. Furthermore, it can protect the privacy (through strong cryptography) not only of remote shells and file copying, but also of generic application-layer flows through what is known as "port forwarding"[4].

In general, the privacy of SSH users can be seen as protected on two fronts: the contents of actual data that flows through an SSH session is supposed to be impenetrable by a third-party observer. At the same time, the type of activity that the user is carrying out through the SSH connection, be it web browsing, secure copy, remote shell execution, etc., is supposed to remain private[5]. SSH is most commonly used by administrators to manage servers remotely, and although there are many algorithms to encrypt the passwords such as private key and public keys [6], many people still use weak passwords [7] , and also the attackers still use the traditional brute force attack and dictionary attack[8] to guess the user names and passwords in order to login to a servers. Nowadays there are many kinds of software used to protect servers from this kind of attack, like Denyhosts[9], fail2ban[10] and sshgards[11] . But still weakness in these software cause many problems [12], some of this software can block the users who forgot their passwords if a user failed to login for several time, and these software can block some trusted IP's if the attacker uses this IP's as a user name and failed to login.

Nowadays the attackers use deferent type of attacks to avoid stirring protection programs used to detect brute force attack, one attacker use a range of IP's to execute the attacking with deferent IP in each attempt to login[13].

In general all the above software's depend on one principle that is blocking the IP for the user who failed to login several times [14]. The previous work has ignored individual attacks that do not raise suspicion of different users, as one or two failed attempts from the same host sporadically. However this study seeks to address this kind of attack. To that effect, we develop adaptive machine that can solve the problem in the above software's, and the proposed software (DBFST) has the ability to distinguish between the attacker IP and other trusted IP, and checking the existence of the individual distributed attacks after which the IPtables are automatically updated to block each kind of attack. This paper is organized as follows: Section 1 is the introduction, section 2 describes the related works, Section 3 and 4 provides a detail description of the tools Fail2ban and Denyhosts, while section 5 describes the Methodology of th work, section 6 somrais the DBFST imple-

- *AL-Zwuiany Muhanad is currently pursuing masters degree program in Computre engineering in School of Information Science and Engineering Central South University, Lectuerer in Thi Qar University, IRAQ, PH-+86 15116370922. E-mail: mmkz1981@yahoo.com*
- *Prof. Huang Dongjun School of Information Science and Engineering Central South University, China, E-mail: djhuang@csu.edu.cn*

International Journal of Scientific & Engineering Research Volume 6, Issue 3, March-2015
ISSN 2229-5518

mentation, section 7 contains the evaluation of experimental results and finally section 8 concludes the paper with some final remarks.

2 RELATED WORKS

There have been numerous studies and application that deal with network monitoring, security, and SSH brute force attacks and in this section we shall be highlighting some of these works related to this paper:

A general approach by Mobin Javed and Vern Paxson [12],was proposed for detecting attack in individually sly activities, which operates in unsuspected manner in a SSH Brute-Force attack. This study depends on two elements, site aggregates Analyzer and Attack Participants Classifier. The first element is to observe the activities and attacks which occur in the sites and detect it. The second is to analyze and classify the attack's participant. Fig. (1) illustrate these attributes.

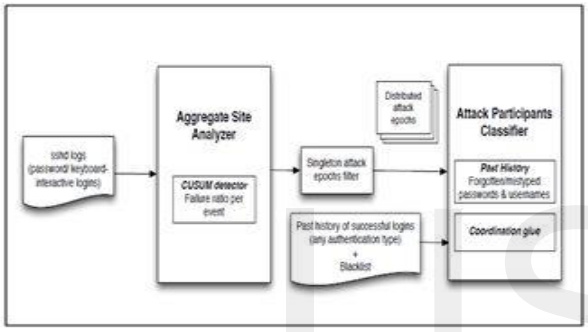

Fig.1 System diagram to detect distributed SSH Brute forcing [12]

Yuvaraj et al.[18] proposed a new Passwords Guessing Resistant Protocol (PGRP) which reconsiders of derivatives previous overtures to tie up this kind of attacks. This system has been divided into three parts namely: a- user & password authentication, b- IP Authentication c- Cookie Authentication. Figures (2) and (3) represent the structure diagram and the block diagram receptively of the PGRP system.

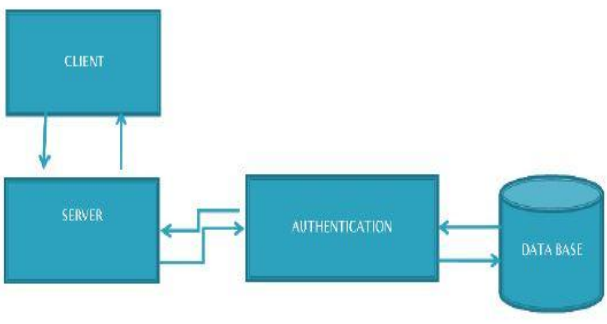

Fig. 2 System structure diagrams [18]

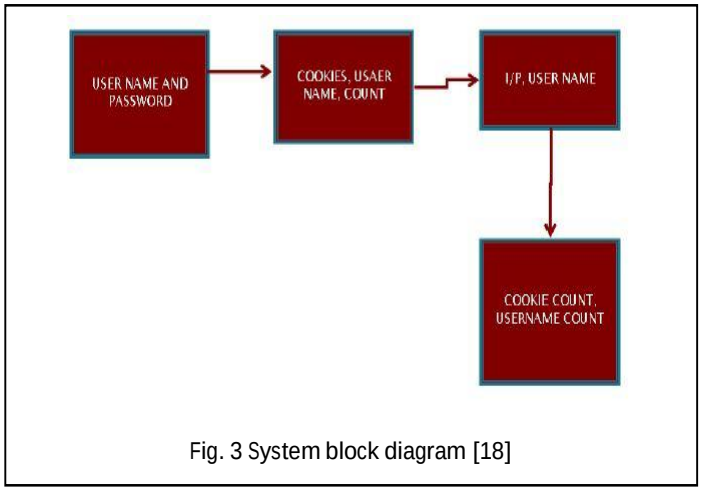

Fig. 3 System block diagram [18]

3 FAIL2BAN

. Fail2ban[4] is one of mostly used software by server administrators to protect the servers from SSH brute force attacks[4][15]. This software focus on monitoring the information in selected log file, then the blocking the users who have failed to authenticate correctly after a set number of times [16], due to adding rule within IPtables.

When fail2ban face lines in a log file that is point out to the attack, and detecting attacks and activating the IPtables blocking mechanism, Fail2ban can send emails to the system administrator with all the actions that are being taken, and inform the administrator about the attack. Fai2ban implementation needs particular preparations such as configuring according the system to protect, and selecting which log file the information will be entered. Therefore, it is prerequisite to analyze the log file to find the line to indicate to the attack. It is not easy to select these lines because there are different types of attack. For that the administrator should be highly skillful to identify the lines indicate to the attack according to his previous experience with deferent kind of attempt to compromise the system. Then fail2ban had to be configured to count the number of failed attempts for each IP during a specific period of time, and then defined the blocking rule for the IP's against attacks. Fail2ban is usually configure to block the IP address after several failure attempts to login[14].

Fail2ban should be used with some precaution, tighten in mind the possibility of the so-called injection attacks, in which the attacker would create a false IP packet and misrepresent themselves as having the IP address of the victim. By generating a massive quantity of such packets, the attacker can make Fail2ban configure the blocking of the victim's IP. For instance, the victim can be the system administrator or another regular user of the server. For that the administrators are recommended to first try and prevent, where possible, the arrival of false IP packets in the local network (i.e. to prevent spoofing attacks). Then, administrators should identify all IP addresses that are important for he work of the service being protected by the Fail2ban tool.

International Journal of Scientific & Engineering Research Volume 6, Issue 3, March-2015
ISSN 2229-5518

4 DENYHOSTS

Denyhosts[9] is an open-source Python script, that maintains a simulated blacklist based on past failed login attempts. It monitoring the ssh log file locate in /var/log/auth.log, and then capture the lines mentioned to the failed attempt login to the system with a specific IP address. Denyhosts has been programmed to block the IP address after passing a threshold of attempts number. The blocking IP address will be prevents from making any further attempts by adding it to the file /etc/hosts.deny on the server. For that when the admin need to authorize a blocked IP he should edit the file /etc/hosts.deny_ssh and manually delete the IP and the comment related to that IP. Fig. (4) shows the /etc/hosts.deny_ssh file[17]

```
# DenyHosts: Thu Feb 14 19:03:30 2013 | 192.168.1.56
192.168.1.56
# DenyHosts: Thu Feb 14 22:36:00 2013 | 192.168.10.21
192.168.10.21
# DenyHosts: Fri Feb 15 08:44:09 2013 | 192.168.16.32
192.168.16.32
# DenyHosts: Fri Feb 15 10:44:39 2013 | 192.168.16.100
192.168.16.100
```

Fig 4 /etc/hosts.deny_ssh file

5 METHODOLOGY

The functionality of framework of this research in general divided into three sections. The first section includes finding the proper technique and method that can be used to detect the brute force attack problem. By analyzing the ssh log file in diverse sites in addition from the related works and the literature review. This study suggests the DBFST as a suitable technique, where the DBFST combine between the characteristics of (Fail2ban and Deny Host), where it's also focused on monitoring the log file of ssh server that's located in /var/log/auth.log, and capture the lines where the failed attempt located, then count the attempts number for specific IP failed to login. Moreover, DBFST has the ability to detect the individual attacks that do not raise a suspicion by different users who use a few shy attempts, from same host and in discontinuous times. The second section concerned about implement and use the new technique based on the characteristics of (Fail2ban & Deny Host).

The third section includes results and evaluation, the performance of DBFST will be evaluated with a multiple attacks whether from group of attackers within the same network or a single attacker use a range of IP addresses or botnet to establish the attacking. In order to describe the DBFST fig. (5) will represent its steps.

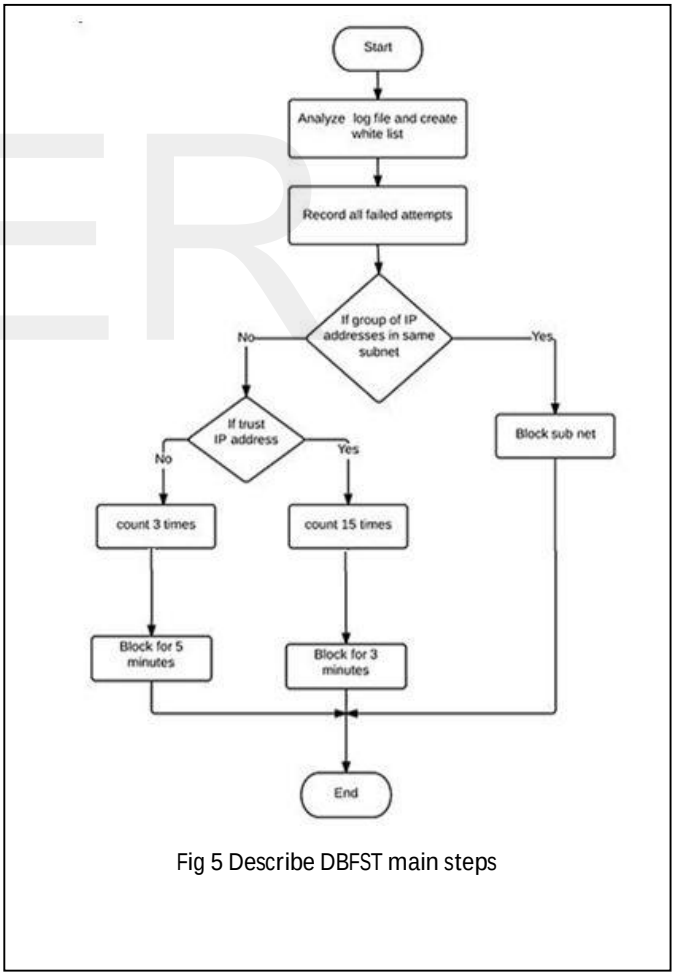

Fig 5 Describe DBFST main steps

6 IMPLEMENTATION

This study proposes the DBFST as a suitable technique, where DBFST combines the strengths

International Journal of Scientific & Engineering Research Volume 6, Issue 3, March-2015
ISSN 2229-5518

and certain characteristically features of both Fail2ban and Denyhosts to improve the performance. Moreover, DBFST has the ability to detect the individual attacks that do not raise a suspicion by different users who use a few shy attempts, from same host and in a discontinuous manner.

In this kind of attack, a group of attackers use a botnet to start an attempt on the target, in this way the first attacker start attempting to login to the server using a range of IP's, then the second one start his attack where the first stop and so on, by so doing, the attacker avoid duplicating the IP and a list of passwords with common usernames are used to execute the attack.

And because the attacker uses a different IP address in each attempt, it eludes attention of the security mechanism of the security software's used to protect against brute force attack even if all attempts failed to login as shown in fi. (6).

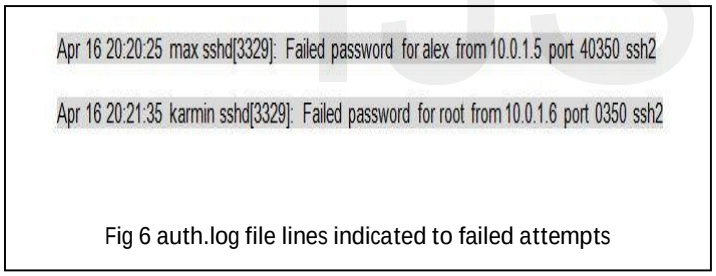

Fig 6 auth.log file lines indicated to failed attempts

In the case where botnet is used by the attacker several hundreds of thousands of attempts will be available to the attackers with wide range of IP's addresses.

In order to have a valid reliability test for this study, there are four main steps to be followed in processing various types of data obtained from the log file.

Furthermore, to improve the security of all hosts in the server and minimize the attacking risk, this study suggests the following steps:

a- Create a white list contains all the trust IP address: In this stage the ssh log file /var/log/auth.log has been analyzed to find every single success login to the system, and record the IP's addresses of each successful attempt to create the white list. In the same time the server's IP will be added as well as the DNS and the getaway of the server. This list can be updated manually by editing from the system administrator to add some trusted IP or remove untrusted ones. Furthermore, the same process can be done automatically when a new user login successfully for the first time.

b- Check the log file for every single failed attempt: In this step of analyzing log file every single failed login will be recorded. Moreover, there are two cases for any failed attempt:

1- Failed attempt of a trusted IP that already listed in the white list. The IP will be given a chance to log in about 15 attempts within 10 minutes. However, when the IP cross the attempts threshold, it will be blocked for 3 minutes.

2- Other IP's will be considered as untrusted IP's. DBFST will block this IP address for 5 minutes, if it tries to login more than 3 attempts within 5 minute. Table (1) illustrate the blocking time of the IP addresses according to white list

TABLE 1
Blocked time for the failed attempts

Failed attempts	Duration Time	IP's in the White list blocked time	Others IP's blocked time
15 times	10 min	3 min	----------
3 times	5 min	----------	5 min

International Journal of Scientific & Engineering Research Volume 6, Issue 3, March-2015
ISSN 2229-5518

c- Creating a failed attempt list: Firstly to setting up this list, every unsuccessful IP address attempt and the attempted time will be recorded. Secondly will establish an archive for all IP addresses that has failed to login.

d- Checking a failed attempt list to find if a group of IP addresses in the same subnet had failed to login in duration time between 3 -10 minutes. DBFST will block all the subnet of these IP addresses. The IP's or subnet blocking will be done through updating the IPtables[19] adding rules. Where the IPtables is section of Netfilter setting in the LINUX kernel, therefore it control the packets filter. The IPtables content has three main chain shooing in fig. (7).

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Fig 7 Iptables chains

While the syntax of blocking source is shooing in figure (8)

```
iptables -A INPUT -j DROP
```

Fig 8 Iptables blocking source

Over all, the mathematical formula (1) describe principles and steps that are been followed in DBFST.

$$g(t) \in White\ list =$$

$$\begin{cases} T \ if\ k \geq 15 \wedge t \leq 600\ then\ t_0 = 180 \\ F \ if (t \leq 300 \wedge k \geq 3) \vee (180 \leq t \leq 600 \wedge f \geq 10)\ then\ t_0 = 300 \end{cases} \quad (1)$$

From the formula (1) where g(t) is mention to the IP address, that had detected failed login, this IP will examine if its belong to the white list that has been created by the DBFST, if this IP is in this list, in this case it's true (T),then the DBFST will count the number of failed attempts of this IP (k) if it is greater than or equal to 15 failed attempts in 10 minute duration of time (t) then the IP should be blocked for 3 minute where (t0) is the blocking time in second.

In the case of the g(t) does not belong to the white list (F), DBFST will count the number of failed attempts of this IP if its greater than or equal to 3 times in a duration of time 10 minutes, then this IP will be blocked for 5 minute. Otherwise the DBFST will check the other attempts from the other IP, and find if 10 of this IP's belong to the same subnet, where (f) mention to these 10 IP's within the same subnet, and if these IP's attempts were within duration 10 minutes, then this subnet will be blocked for 5 minutes.

7 EVALUATION

This study presents DBFST to protect a single target machine from SSH distributed brute force attack, as such the study addresses situation where one or group of attackers use botnet to attack a single target, in which case there are huge numbers of

International Journal of Scientific & Engineering Research Volume 6, Issue 3, March-2015
ISSN 2229-5518

attempts opportunities available to the attacker increasing the possibility of successful maneuvering with IP addresses to penetrate the target.

By using fail2ban or Denyhosts, assumed to block IP after 10 failures for every consecutive 10 minutes so the effective trial rate is 10 attempts every 610 seconds, improving on the unprotected hosts by a factor of 61[13].

For that if the attacker try to login with one deferent IP every second, that means the single IP from each subnet will not repeated for several hours or few days. In this case fail2ban and Denyhosts will not detect this kind of attack. Furthermore DBFST can detect this type of attack and distinguish between trusted and untrusted IP's, then block the failed IP's for disparate time according to trust and untrusted cases, furthermore it will check all the failed attempted IP's to detect the groups of attackers within a specific and convergent time, and then block the subnet of those attackers.

The results are impressive: DBFST examine a few dozen password attempts per minute. But even under heavy attacks, still offers service to most legitimate users. Table (2) illustrates the results obtained from the use of DBFST and fail2ban, and likewise shows the difference between the number of failed attempts before and after deploying the DBFST. In case of the server shut down for any reason, the DBFST can be added to the startup services, to insure the availability of the software. The crone demo was adopted for this job, to force the system to start up the DBFST with every network starting. The commend bellow can be used to configure crone demo to this work.

crontab –e then add @reboot <the path of DBFST >.

TABLE 2
results obtained from the use of DBFST

and fail2ban in one week

Result/week	Using DBFST	Using Fail2ban
Trust IP blocked	21	86
Total IP blocked	547	110
Subnet blocked	59	———

From table (2) which show the result of using the DBFST and Fail2ban in one week duration of time, where Fail2ban is the tool mostly use by the administrators. In this durationof time there are 21 trust IP blocked with DBFST in compare with 68 trust IP blocked with Fail2ban, while the total number of IP's blocked in DBFST was 547 IP's, and 110 IP's had been blocked with Fail2ban, in the same time there were 59 subnets had been blocked with DBFST, while this option is not available in Fail2ban, the total number of failed attempts was 1437 during the using of DBFST, and 6749 failed attempts with Fail2ban using. It's seemed clear that using of DBFST reduced the total numbers of failed attempts, as the subnet blocking mechanism has contributed these attempts numbers. In this table we gave the results for one week, while the DBFST had been examined for two months, and it was correctly worked.

International Journal of Scientific & Engineering Research Volume 6, Issue 3, March-2015
ISSN 2229-5518

8 CONCLUSIONS

This paper present the distribution SSH brute force attack on a single target and the methods used to detect this kind of attack, furthermore the software DBFST has been implemented to improve the SSH server security against the distributed and Individual brute force attacks. The design of this software used the combination of techniques from fail2ban and Denyhosts applications to achieve better performance, in the sense that it DBFST has the ability to detect the Individual and distributed attacks. Unlike fail2ban and Denyhosts, DBFST detects the attacks from the same subnet or network, and block the attackers' networks.

REFERENCE

[1] C. Adams, G.-V. Jourdan, J.-P. Levac, and F. Prevost, "Lightweight protection against brute force login attacks on Web applications," in PST, 2010, pp. 181-188.

[2] A. Jesudoss and N. Subramaniam, "A SURVEY ON AUTHENTICATION ATTACKS AND COUNTERMEASURES IN A DISTRIBUTED ENVIRONMENT," IJCSE, vol. Vol. 5 No.2, 2014.

[3] T. Ylonen and C. Lonvick, "The secure shell (SSH) protocol architecture," 2006.

[4] U. c. knome". (2013, 23 Dec.). SSH / OpenSSH / PortForwarding. Available: https: / / help.ubuntu.com / community / SSH / OpenSSH / PortForwarding

[5] M. Dusi, F. Gringoli, and L. Salgarelli, "A preliminary look at the privacy of ssh tunnels," in Computer Communications and Networks, 2008. ICCCN'08. Proceedings of 17th International Conference on, 2008, pp. 1-7.

[6] S. C. Williams, "Analysis of the SSH key exchange protocol," in Cryptography and Coding, ed: Springer, 2011, pp. 356-374.

[7] 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.

[8] M. Vizváry and J. Vykopal, "Flow-based detection of RDP brute-force attacks," in Proceedings of 7th International Conference on Security and Protection of Information (SPI 2013), 2013.

[9] SourceForge. (2014, Dec. 20). Denyhosts. Available: http: / / denyhosts.sourceforge.net /

[10] Doxygen. (2014, Dec 15). file2ban. Available: http: / / www.fail2ban.org / wiki / index.php / Main_Page

[11] SSHguard. (2014, 12, nov.). sshguard. Available: http: / / www.sshguard.net /

[12] B. Q. M. AL-Musawi, "PREVENTING BRUTE FORCE ATTACK THROUGH THE ANALYZING LOG," vol. 53, No.3, 2012,.

[13] M. Waldvogel and J. Kollek, SIEGE: Service-Independent Enterprise-GradE protection against password scans: Bibliothek der Universität Konstanz, 2014.

[14] M. Javed and V. Paxson, "Detecting stealthy, distributed SSH brute-forcing," in Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013, pp. 85-96.

[15] J. Richter, "DISTRIBUTED PROTECTION AGAINST DISTRIBUTED BRUTE FORCE ATTACKS."

[16] J. Ellingwood. (2013, 7 Dec.). How To Protect SSH with fail2ban on Debian 7. Available: https: / / www.digitalocean.com / community / tutorials / how-to-protect-ssh-with-fail2ban-on-debian-7

[17] F. B. Manolache, Q. Hou, and O. Rusu, "Analysis and prevention of network password guessing attacks in an enterprise environment," in RoEduNet Conference 13th Edition: Networking in Education and Research Joint Event RENAM 8th Conference, 2014, 2014, pp. 1-7.

[18] M. YUVARAJ, A. BHARATHIDASAN, and N. KUMAR, "Implementation of Password Guessing Resistant Protocol (PGRP) to Prevent Online Attacks," 2014.

[19] M. Rash, Linux Firewalls: Attack Detection and Response with iptables, psad, and fwsnort: No Starch Press, 2007.