

An Enhanced Port Tunneling and Device Tracking Authentication Mechanism

¹* S. Yamini, ²Dr. D. Maheswari

¹ School of Computer Studies - PG, RVS College of Arts & Science, Coimbatore, India

² School of Computer Studies - PG, RVS College of Arts & Science, Coimbatore, India

Email: ¹ yamini@rvsgroup.com, ² maheswari@rvsgroup.com

Abstract. Port knocking is a technique by which only a single packet or special sequence will permit the firewall to open a port on a machine where all ports are closed by default. It is an unresisting authorization technique which offers firewall-level authentication to ensure authorized access to possibly unprotected network services. This method is liable to attacks when attackers detect the network. This paper suggests a new method which is called “Enhanced Port Tunneling & Device Tracking (EPT & DT)” to banish both DOS-Knocking and NAT- Knocking attacks. The source IP address where an annoyed activity had originated is of limited value because it does not specify a physical locality, besides an endpoint in a network for the exclusive conviction of routing. Furthermore, people and their devices move across the network, changing IP address as significance. It is proficient to have some hints about where a device was at the time the offending action was accomplished. Nevertheless, it would be prudent to connect different pieces of evidence to ascertain additional information, such as IP addresses worn by the corresponding device. Devices constantly accessing a private network, at different times, can be outlined by analyzing and associating Network and Port Address Translation (NAPT) logs, in order to acclaim recurring activity patterns. It is feasible to recognize some of the users from their traffic abnormalities without considering the exposed IP addresses. Experiments were conducted on NAPT logs accumulated in a campus network, with DHCP data providing control points for validation. The main purpose of using NAPT logs is for device tracking.

Keywords: Port knocking, Network Address Translation, Tunneling, Port security, DOS knocking attacks, Log analysis, Device tracking, tracing

* Corresponding Author:

S. Yamini

School of Computer Studies - PG, RVS College of Arts & Science, Coimbatore, India.

Email: yamini@rvsgroup.com

1. Introduction

The importance of securing the hostile world of internet has increased now, because there have not been such deadly risks before. Reason of this increased security risk is the introduction of Internet. One way to limit access to selected users is by using an authentication method, but this is not a perfect solution. One usual method of limiting the hosts is to use firewall.

Firewall selectively accepts and rejects network packets by considering their source address and other important characteristics. Some dangerous attackers are capable enough to hide the source of packets sent by them. Users having unpredictable IP can also easily pass through firewall. So firewall is also not a complete solution as well [19].

Port knocking is a kind of security mechanism installed over firewall of secure computer systems. Basically what port knocking does is that it provides with another security layer over the security we already have. As shown in Figure 1, all well-known ports of secure server are closed by Port Knocking(PK) firewall so when an attacker try to directly get connected to port

22, the firewall will simply drop its packet and will not allow attacker to access any secure port directly.

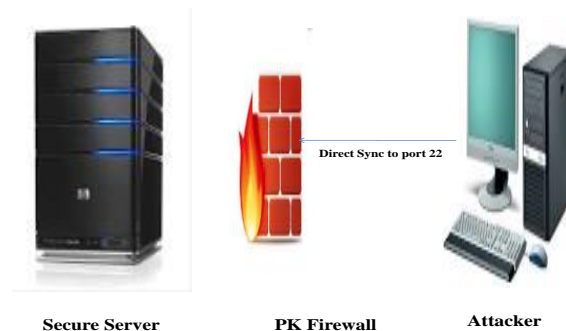


Figure 1. Port Knocking Firewall

Once a correct sequence of connection venture is accepted, the firewall rules are aggressively modified to allow the host which sent the connection attempts to connect over specific ports. Actually, client who wants to use services should start an authentication process with sending non-reply packet to server [1]. Therefore, an attacker who is monitoring the network cannot detect server. There is a monitoring system in the server-side that stores the log of knocking process. When the authentication pattern is completed then server opens a port for the valid user and the trusted connection is established between client and the server. As shown in Figure 2, an authentic client who knows the predefined sequence of knocks which will act as a key will send tcp sync packets to those predefined sequence of ports.

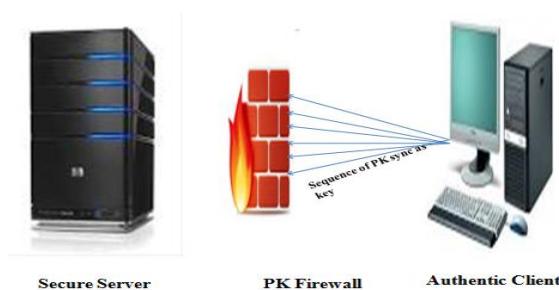


Figure 2. Predefined sequence of Knocks

There are some attacks that can affect PK interpretation which enables a malicious user rebuke the connection. While, PK can build the authentication process safer than before, it encounters some conditions, which make the network unprotected. DOS-Knocking and NAT-Knocking attacks are some of the popular attacks on PK techniques [20]. One of these circumstances occurs when attackers redirect random packets to the server often. Server should allot a buffer for enduring log of each client until PK finish. Consequently, DOS-Knocking steers to occupy the remarkable amount of memory [2]. The other state happens when monitoring system cannot discriminate trusted users from others. This situation emerges when Network Address Translate (NAT) is used in the network. As a culmination, all the users have the same address outside the local network. Consequently, when one user finishes the

PK process and obtains permission for accessing to the server, all the clients which are revealed behind the similar NAT can use the service [3, 4].

There are predominantly three methods to implement NAT, including Static NAT, Dynamic NAT and NAPT (Network Address Port Translation) [22]. Static NAT is to transmute private internal network IP addresses into public IP address, and IP address is a one-to-one, meaning that a private IP address can exclusively be revamped to a fixed public IP address. With the static NAT, external network can acquire to specific internal network equipment (such as servers) [21]. NAPT is to revamp the source port of outgoing packets and port translation, manipulating the port multiplexer mode. All hosts in the internal network can share only one sustainable external IP address to acquire the Internet, and accordingly can save the IP address to maximize. Simultaneously, we can conceal all hosts within the network, constructively intercepting the attack from the Internet. Hence, NAPT is the stupendous application in the network.

This paper ventilates a novel port knocking proposal in which PK authentication process is divided into two phases. First phase eliminates the DOS-Knocking while the second part abolishes the NAT-Knocking problem. This new method is known as EPT & DT: Enhanced Port Tunneling & Device Tracking, which is an enhanced port security authentication mechanism. To the best of the authors' knowledge, there are not enough studies in PK. Therefore, it can be a suitable field for researchers who are working on the network security and want to use a new method for combating attackers or anonymous users.

While it is relatively easy to track down the source IP address where a specific malicious or offending activity had originated, this piece of information is of scarce value in practice. Typically, an Internet Protocol (IP) address is not associated with a person, and is only loosely associated with a device. IP addresses are not meant as a way to specify physical locations; they correspond to an endpoint in a network for the sole purpose of routing [5]. In addition, people and devices move across the network, both physically and virtually. When traveling from one place to another, they dynamically acquire new IP addresses depending on the associated attachment point to the network (e.g., WLAN or 3G cellular coverage area) [23].

On the other hand, they can have different IP addresses by using proxies that masquerade their real address. Thus, the usefulness of an IP address to determine the real identity, or geographical location of an offender could be very limited. Furthermore, such information is anyway restricted to the location the device was at when an action was performed or, even better, when a record was taken. Instead, it would be desirable to learn about the specific device/user behind a particular network activity.

Worse yet, as a result of mobility or malicious address spoofing practices, several outwardly unrelated, IP addresses may be found as the source of each phase of a multistage action which is performed in different steps and (sometimes apparently) at different locations. The critical factor is tracking the user or at least the device rather than the IP address. Given an evidence (such as a packet trace, a Netflow dump or a set of log entries) known to be associated with a device under observation (for example, sending a threatening email message or launching a DoS attack to a third party organization), it would be desirable to determine if other Internet activities were also originated from the same device. By that means, it could be possible to identify the real endpoint whose addressing information might, voluntary or involuntary, have been changed. Such correlation would be also useful even if it were just able to roughly describe the different locations traversed by the device and hence to gain some clues about the movements of the user under attention.

NAPT devices appropriately modify the source or the destination transport addresses of traffic traversing them. They maintain state information about the translation so that the reverse translation can be applied to traffic flowing in the opposite direction and NAPT translations can be logged and in some countries there is a legal obligation to do so. A NAPT log entry contains several fields describing a single translation. In this work, analysis of NAPT logs provides a way to suggest a possible recognition of devices repeatedly accessing a network across multiple visits. The idea is to determine a device profile or "fingerprint" in order to track down the device movements independently of the IP address it has. Matching the internal addresses associated with these common profiles, when combined with other collected logging

information (e.g., DHCP logs), may generate stronger evidence that can be used for a more reliable identification in a network investigation process.

2. Related Work

Hussein Al-Bahadili [6]; develops and evaluate parameters of newly built PK implementation referred with the name of hybrid, which have the capability to defeat previous knocking techniques. This new technique uses concepts of PK, mutual authentication and steganography. Renniede-Graaf, Improved Port Knocking with Strong Authentication[7]; studies existing PK implementation, improves existing PK techniques, builds a new PK technique which is refers to as novel port knocking technique.

Ben Maddock [8]; defines portknocking and its benefits in detail, elaborates features of existing portknocking techniques, future offer exploration and PK conclusion. Sebastien Jeanquier [9] in, "An Analysis of Port Knocking and Single Packet Authorization"; analyzes PK and SPA as network security mechanisms, performs compatibility as firewall authentication schemes and talk about drawbacks and outcomes in current PK implementations, critical evaluation of FWKNOP, outlining its outcomes and suggesting some remedies.

Several research works focus on Payload Attribution, i.e., the identification, given a certain excerpt of a payload, of the sources and destinations of all recorded packets whose payload contains that excerpt. As the size of full packet traces hinders both storage and analysis, recent research on the retention of network data for forensic investigations concentrates on techniques which reduce data footprint while, at the same time, supporting queries on payloads. The storage savings provided by data representation are counterbalanced by a controllable number of false positives in the query results.

Those methods include Bloom filters [10], arithmetic coding for data compression [11], as well as storage of partial flow information [12]. A recent study by D. Worth [1] has combined finger print and port knocking for authentication method. Also a firewall knock operator, which is a tool that can support both shared (plain) and encrypted port sequence, was introduced. In 2005, researchers explored the limitations of PK and highlighted the issues which can put the network in danger [13]. Between 2005 and 2010, most of the papers worked on the encryption method for port sequence [2, 14].

The Silent Knock method is the result of the conducted studies during that time. In this approach, AES block cipher and MD4 hash function are applied to increase the security of proposed PK but the simulation results shown that Silent knock has a reasonable overhead [15].

Compared to the existing algorithms like ALDABA, SIG², FWKNOP, the proposed EPT&DT algorithm is giving better performance in the form of platform, Implementation, Protocols, Out of order delivery, NAT, Encryption etc., Most of the methods that are mentioned above cannot eliminate the two well know attacks: NAT-knocking and DOS-Knocking. The proposed PK mechanism in this paper EPT & DT, can achieve this goal and it is described in the next section.

3. EPT & DT

Enhanced Port Tunneling & Device Tracking (EPT & DT) is the new method which is presented in this paper. It is proposed to counter back NAT-knocking and DOS-Knocking attacks and also it can increase the protection of the authentication process. EPT & DT has three phases for securing the authentication mechanism, which are port knocking, tunneling & Device tracking. First stage can solve the DOS-Knocking attack while the second one removes the NAT-Knocking problem and in third it uses the DHCP log for tracing the device.

Figure 3 illustrates a connection in which client want to establish a connection to SSH server after passing the EPT & DT authentication. Client starts the EPT & DT process as a port knocker via sending a UDP packet to the server.

As an example which is shown in figure 4, the PK sequence is completed after four knocks. In this example, the source node with 123.123.123.123 IP address starts EPT & DT and send UDP packet on port 3456. Server checks the passphrase because the port number is valid. Then if passphrase is similar to sec-pass1, server buffers the information for 10 second in the list that is called temprory1 in the example. Otherwise it means the malicious user sends packet and server does not allocate memory space for it and drops it. Therefore, DOS- Knocking problem does not occur anymore. For next knock besides checking the secret text server must check whether the IP address exists in temporary list or not. This process continues until information of four knocks store in the buffer.

The whole process should take only 40 seconds. If each packet cannot arrive to server side before 10 seconds, buffer will flush automatically and the process should be started once again. But if the PK process was successful then the second phase will start. In the tunneling part authenticated user who passes the PK process, should connect to the SSH server through tunnel. Therefore, client must bypass the VPN authentication.

Each session will be open for 30 minutes then it will be closed automatically. User who wants to use the channel for a long time should send port sequence again before the threshold time expires.

In order to maintain the security of the network environment, to monitor and analyze the status of the various network devices and systems in the network environment is an important method. According to analyze of the log files, we can know the operational status of various network devices [16]. The role of the log is to record the operating system, applications and user behavior [17]. Real-time monitoring and analysis of log records, we can analyze suspicious behavior in the network, and promptly take appropriate measures.

When the system has been compromised, we can analyze the log files to find loopholes in the system, and we can even trace to the network address of the attacker. When the system was crash, we can make the system to be restored according to the records in the logs.

Logs of NAT are used to solve this security issue. NAT logs are system information generated by the NAT devices when conducting address translation. The logs are formulated by timestamp, source IP address, source port, destination IP address, destination port, the IP address after the translation, the port after translation as well as the operations of the users. The NAT logs only used to record the user behaviors of internal network users to access the external network and cannot record the user behaviors of external network users to access internal network servers. When the internal network users access to the external network through the NAT device, many users share only one public IP address, resulting that we cannot locate the users accessing to a specific external network server. With the logs of NAT, we can track the behaviors of internal network users, and enhance the network security.

The choice of NAPT logs, instead of other sources about traffic information was motivated by two reasons. Firstly, NAPT log lines provide a compact representation of traffic flows, much lighter than packet traces. A single translation is registered as two simple text lines, which can be efficiently compressed. Data relative to longer periods can be stored, and this yields a considerable advantage over other traffic recording methods, especially because this results in the ability to compare samples taken weeks, or even months, apart. Secondly, while packet traces and flow-based dumps are usually produced only when the circumstances demand them, logs are generated routinely. This may allow investigators to scan for prior activities. However, it should be remarked that there are two drawbacks of using NAPT log lines instead of using flow- based data or packet traces.

The first is that the technique is only applicable to private networks, whereas the other two are general. The second is that easurements about traffic volume are missing. Note that, as far as the translation duration is concerned, log lines are accurate, in contrast to flow-based statistics, which are sampled. Experiments demonstrate that promising detection accuracy can be achieved even without these measurements. Another piece of information that NAPT logs fall short of (but the same is also true for flow-based data) is the indication of the Fully Qualified Domain Name used. This is important especially for web sites. In fact, some servers support more than one web site by using the Virtual Host directive of most web servers daemons which, in turn, implement the Host: header field defined in the HTTP/1.1 standard [10]. Even if the IP address is the same, the host portion of the URL determines

which web site was intended by the user.

In response to these issues, we use the port range mapping program to optimize the NAT transform mechanism [18], in order to achieve the purpose of reducing the amount of the logs, thus improving the efficiency of the querying of logs.

Pros and Cons of the Proposed System

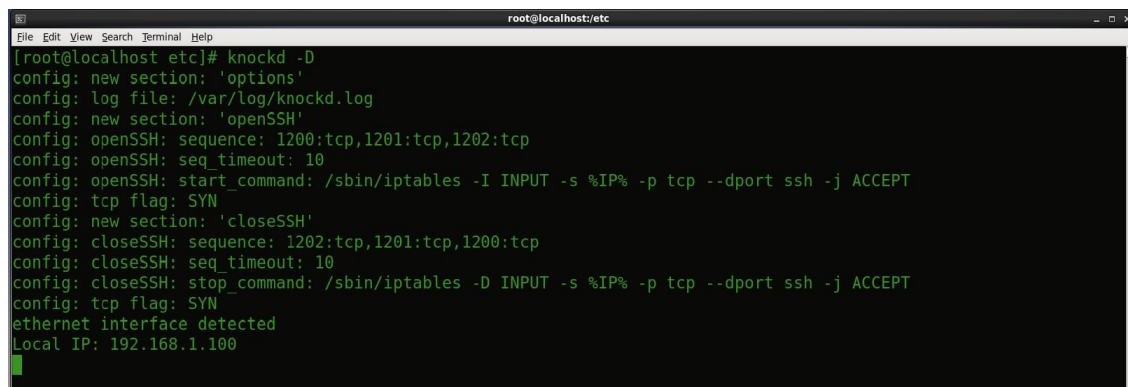
The proposed method EPT & DT is the added layer of Security which reduces the risk of network attacks. It is firewall-based method for user authentication. It initiates connections to hosts with no open ports through disruptive use of closed ports. It seal off network hosts and avert remote profiling. It is unendurable to govern whether port tunneling is implemented and also detection by sniffers both onerous and piercing. It uses encrypted sequences for expanded security. One time encryption pads proffer maximum possible shield. Benefits may be sustained from access control provided by firewall and IDS systems.

Any implementation of this proposed method intended at production systems with a large number of connection attempts needs to inspect misordered/missing ports in a knock sequence caused by network latency, discernment and clarification of simultaneous knock sequences and influence on system and network stuff.

4. Implementation Results

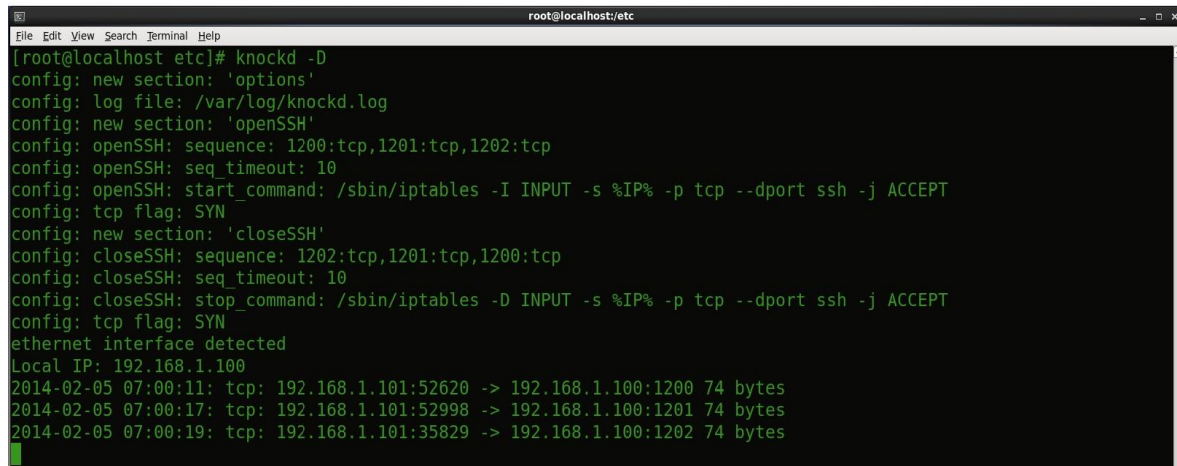
The Proposed method is implemented in Redhat enterprise Linux 6. The port knocking sequence to be used to open the SSH port. Then login to SSH port of REDHAT server and it will be successfully connected. While using the port, the iptables can be seen that a particular ip of the client (UBUNTU) is added to the Accept list. The SSH port to be used and to close the port the close port sequence to be used. In this case it is just the reverse of the open sequence-1200, 1201, 1202 that is 1202, 1201, 1200. After using the close sequence the port will be closed and it won't allow any connection from any client.

This port knocking service can be enabled to any port to make it even more secure, because the combination cannot be guessed easily with 65000 ports approximately. The Sample code performing this technique is shown in the figures from 5 to 8.



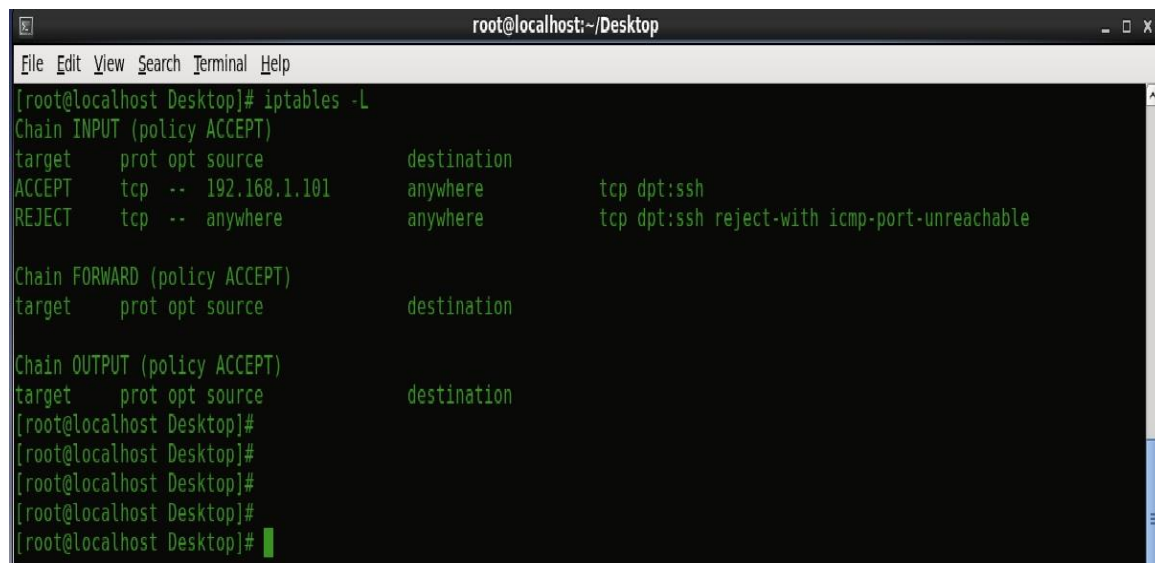
```
root@localhost:etc
File Edit View Search Terminal Help
[root@localhost etc]# knockd -D
config: new section: 'options'
config: log file: /var/log/knockd.log
config: new section: 'openSSH'
config: openSSH: sequence: 1200:tcp,1201:tcp,1202:tcp
config: openSSH: seq_timeout: 10
config: openSSH: start_command: /sbin/iptables -I INPUT -s %IP% -p tcp --dport ssh -j ACCEPT
config: tcp flag: SYN
config: new section: 'closeSSH'
config: closeSSH: sequence: 1202:tcp,1201:tcp,1200:tcp
config: closeSSH: seq_timeout: 10
config: closeSSH: stop_command: /sbin/iptables -D INPUT -s %IP% -p tcp --dport ssh -j ACCEPT
config: tcp flag: SYN
ethernet interface detected
Local IP: 192.168.1.100
```

Figure 5. Port sequence



```
root@localhost:/etc
[root@localhost etc]# knockd -D
config: new section: 'options'
config: log file: /var/log/knockd.log
config: new section: 'openSSH'
config: openSSH: sequence: 1200:tcp,1201:tcp,1202:tcp
config: openSSH: seq_timeout: 10
config: openSSH: start_command: /sbin/iptables -I INPUT -s %IP% -p tcp --dport ssh -j ACCEPT
config: tcp flag: SYN
config: new section: 'closeSSH'
config: closeSSH: sequence: 1202:tcp,1201:tcp,1200:tcp
config: closeSSH: seq_timeout: 10
config: closeSSH: stop_command: /sbin/iptables -D INPUT -s %IP% -p tcp --dport ssh -j ACCEPT
config: tcp flag: SYN
ethernet interface detected
Local IP: 192.168.1.100
2014-02-05 07:00:11: tcp: 192.168.1.101:52620 -> 192.168.1.100:1200 74 bytes
2014-02-05 07:00:17: tcp: 192.168.1.101:52998 -> 192.168.1.100:1201 74 bytes
2014-02-05 07:00:19: tcp: 192.168.1.101:35829 -> 192.168.1.100:1202 74 bytes
```

Figure 6. Log details



```
root@localhost:~/Desktop
[root@localhost Desktop]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT    tcp  --  192.168.1.101          anywhere           tcp dpt:ssh
REJECT     tcp  --  anywhere              anywhere           tcp dpt:ssh reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

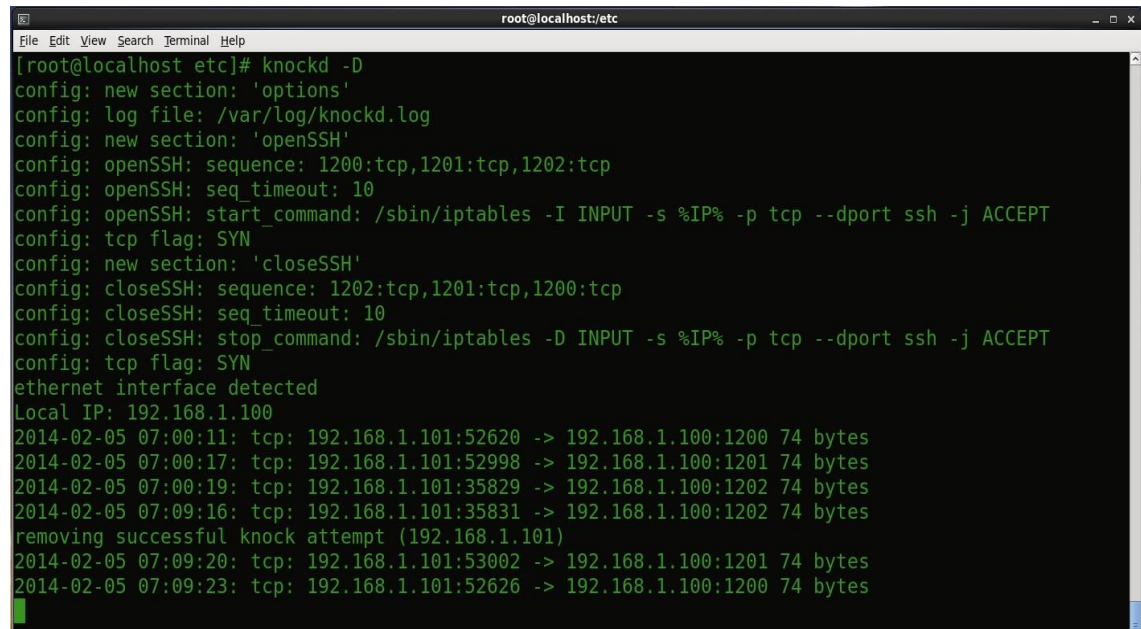
Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
[root@localhost Desktop]#
[root@localhost Desktop]#
[root@localhost Desktop]#
[root@localhost Desktop]#
[root@localhost Desktop]#
```

Figure 7. SSH details

The steps for the above process are as follows.

- installing knock server in REDHAT EL6
- flushing the iptables and adding rules to reject all the connections on the particular port in this case SSH port 22.
- Starting the SSH service
- editing the knockd.conf file in the /etc folder.
- Checking the ip of the REDHAT EL6 server
- In the client machine, in this case UBUNTU, trying to login to REDHAT server using SSH remote access with its ip 192.168.1.100.
- Connection will be refused because of the firewall iptables rule set to reject all connections to the SSH port.
- Then starting port knocking server.
- In the client machine and trying again to connect to the SSH port. Again refused

- Then the port knocking sequence to be used to open the SSH port
- Now login to SSH port of REDHAT server will be successfully connected.
-



```
root@localhost:/etc
File Edit View Search Terminal Help
[root@localhost etc]# knockd -D
config: new section: 'options'
config: log file: /var/log/knockd.log
config: new section: 'openSSH'
config: openSSH: sequence: 1200:tcp,1201:tcp,1202:tcp
config: openSSH: seq_timeout: 10
config: openSSH: start_command: /sbin/iptables -I INPUT -s %IP% -p tcp --dport ssh -j ACCEPT
config: tcp flag: SYN
config: new section: 'closeSSH'
config: closeSSH: sequence: 1202:tcp,1201:tcp,1200:tcp
config: closeSSH: seq_timeout: 10
config: closeSSH: stop_command: /sbin/iptables -D INPUT -s %IP% -p tcp --dport ssh -j ACCEPT
config: tcp flag: SYN
ethernet interface detected
Local IP: 192.168.1.100
2014-02-05 07:00:11: tcp: 192.168.1.101:52620 -> 192.168.1.100:1200 74 bytes
2014-02-05 07:00:17: tcp: 192.168.1.101:52998 -> 192.168.1.100:1201 74 bytes
2014-02-05 07:00:19: tcp: 192.168.1.101:35829 -> 192.168.1.100:1202 74 bytes
2014-02-05 07:09:16: tcp: 192.168.1.101:35831 -> 192.168.1.100:1202 74 bytes
removing successful knock attempt (192.168.1.101)
2014-02-05 07:09:20: tcp: 192.168.1.101:53002 -> 192.168.1.100:1201 74 bytes
2014-02-05 07:09:23: tcp: 192.168.1.101:52626 -> 192.168.1.100:1200 74 bytes
```

Figure 8. Port Knocking Configurations

5. Conclusion and Future Work

The analysis of port knocking authentication methods has revealed both some design flaws and implementation problems that could provide access to unauthorized users. EPT & DT is the novel method presented in this paper that improves port knocking authentication mechanism. It can easily remove the DOS-knocking and NAT-knocking attacks. Therefore, the connection which is established based on the EPT & DT is more reliable than previous methods. This method has a four knock scheme that should be finished in the specific period otherwise the process should start again. Working on the port sequence selection suggested as a future work.

The analysis of NAPT logs provides an interesting capability to identify devices characterized by similar network behaviour. In addition, data mining concepts and tools can be applied to this problem to complete and widen the scope of the presented approach.

References

- [1] D. Worth, COK: Cryptographic one-time knocking, 2004, Talk slides, Black Hat USA, pp. 19-25.
- [2] A. I. Manzanares, J. T. Marquez, J. M. Estevez-Tapiador, J. Cesar Hernández Castro, Attacks on port knocking authentication mechanism, Computational Science and Its Application, ICCSA 2005, pp. 1292-1300.
- [3] T. Popeea, V. Olteanu, L. Gheorghe, R. Rughinis, Extension of a port knocking client-server architecture with NTP synchronization, 10th Roedunet International Conference (RoEduNet), 2011, pp. 1 - 5.
- [4] S. Jeanquier, An Analysis of Port Knocking and Single Packet, MSc Thesis, Information Security Group, Royal Holloway College, University of London, 2006.

- [5] V. P. Kafle and M. Inoue, Locator ID separation for mobility management in the new generation network, *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 1, no. 2/3, pp. 3–15, 10 2010.
- [6] Hussein Al-Bahadili and Ali H. Hadi, Network Security Using Hybrid Port Knocking, *IJCSNS International Journal of Computer Science and Network Security*, VOL.10 No.8, August 2010.
- [7] Rennie deGraaf, John Aycock, and Michael Jacobson, Improved Port Knocking with Strong Authentication, *Proceedings of the IEEE 21st Annual Computer Security Applications Conference* 2005.
- [8] Ben Maddock, Port Knocking: An Overview of Concepts, Issues and Implementations, *SANS GIAC GSEC Practical* 2004.
- [9] Dawn Isabel, Port Knocking: Beyond the Basics, *GIAC Security Essentials Certification (GSEC) Practical Assignment Version 1.4c Option 1 - Research on Topics Information Security* 2005.
- [10] M. Ponc, P. Giura, J. Wein, and H. Brönnimann, New payload attribution methods for network forensic investigations, *ACM Trans. Inf. Syst. Secur.*, vol. 13, pp. 15:1–15:32, March 2010.
- [11] Y. Tang and T. Daniels, A simple framework for distributed forensics, in *25th IEEE International Conference on Distributed Computing Systems Workshops*, 2005. IEEE, 2005, pp. 163–169.
- [12] S. Kornblum, V. Paxson, H. Dreger, A. Feldmann, and R. Sommer, Building a time machine for efficient recording and retrieval of highvolume network traffic, in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. USENIX Association, 2005, pp. 23–23.
- [13] R. DeGraaf, J. Aycock, M.J. Jacobson, Improved Port Knocking with Strong Authentication, *21st Annual Computer Security Applications Conference*, 2005, pp. 451- 462.
- [14] P. Iyappan, K. S. Arvind, N. Geetha, S. Vanitha, Pluggable Encryption Algorithm In Secure Shell (SSH) Protocol, *Second International Conference on Emerging Trends in Engineering Technology*, 2009, pp. 808-813.
- [15] E. Y. Vasserman, N. Hopper, J. Laxson, J. Tyra, SilentKnock: practical, provably undetectable authentication, *International Journal of Information Security*, Vol. 8, No. 1, February 2009, pp. 121-135.
- [16] Sun Pengcheng, Zhou Lihua, Research on Log Management System in Linux, *Electronic Sci.&Tech.*2007(07),72-74.
- [17] S.Sivakumar, Logging of NAT Events, *Internet-Draft*, Expires: April 26, 2012.
- [18] Stuart Cheshire, NAT Port Mapping Protocol (NAT- PMP), *Internet-Draft*, Expires 16th October 2008.
- [19] Z. A. Khan, N. Javaid, M. H. Arshad, A. Bibi, B. Qasim, Performance Evaluation of Widely used Portknocking Algorithms, *IEEE 14th International Conference on High Performance Computing and Communications*, 2012.
- [20] Mehran Pourvabab, Reza Ebrahimi Atani,, Laleh Boroumand, SPKT: Secure Port Knock-Tunneling, an Enhanced Port Security Authentication Mechanism, *IEEE Symposium on Computers & Informatics*, 2012.
- [21] Hong Ma, Yongjuan Wu, Yan Ma, Zhenhua Wang, Optimization Scheme of CGN logs, *Proceedings of IEEE CCIS*2012.
- [22] Zeng Chuanhuang, Hu Haonan, Analysis Of The NAT-PT Gateway, *International Conference on Computer Science and Service System*, 2012.
- [23] Aniello Castiglione, Alfredo De Santis, Ugo Fiore, Francesco Palmieri, Device Tracking in Private Networks via NAPT Log Analysis, *Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2012.