AN EFFICIENT COMBINED SECURITY MODEL FOR NETWORK USING HYBRID PORT KNOCKING TECHNIQUE

VENKATESWARLU SUNKARI, ATO DANIEL ABEBE

[1]Assistant Professor, Centre for ITSC, Addis Ababa Institute of Technology, Addis Ababa.
[2]HOD, Centre for ITSC, Addis Ababa Institute of Technology, Addis Ababa.

Abstract- The main objective of this work is to develop and evaluate the performance of a new port knocking(PK) technique, which can avert all types of port attacks and meets all network security requirements. The new technique utilizes three well-known concepts, these are: port-knocking (PK), steganography, and mutual authentication, therefore, it is referred to as the hybrid port-knocking (HPK) technique. It can be used for host authentication to make local services invisible from port scanning, provide an extra layer of security that attackers must penetrate before accessing or breaking anything important, act as a stop-gap security measure for services with known un-patched vulnerabilities, and provide a wrapper for a legacy or proprietary services with insufficient integrated security. The performance of the proposed technique was evaluated by measuring the average authentication time, which also compared with the average authentication time for a number of currently used port authentication techniques.

Keywords- Network security, Firewalls, Authentication, Port-knocking, Steganography, Cryptography.

I. INTRODUCTION

The Internet can be seen as a huge network of different nodes connected together providing different services. The difference in every service provided is that it is given to whom? Some services are for the public whereby others are for some specific users. The problem is how can we control this access? A first solution that might come in mind is using a firewall [1]. Firewalls are a good solution but they can only provide control based on IP addresses and some other characteristics. Unfortunately, firewalls cannot dissipate between users connecting from the same IP, and for sure different IP's. It can only see IP addresses and its characteristics but not a user name and password for example. So we can only consider firewalls as the first level of defence [2]-[3].

Also, there are common attacks against which a firewall cannot protect. For example, firewalls do not protect against attempts to exploit bugs in application-level software. Such vulnerabilities occur because the Internet architecture assumes that services bound to a port should be accessible by any machine using the Internet protocols. Another problem that shall face an online service is the zero days (0-day) exploit attacks [4].
There are many terminologies used in this research, the upcoming section will describe the most related terminologies. The PK concept has been around for a while, and there are many different PK implementations. In computer networking, PK is a method of externally opening ports on a firewall by generating a connection attempt on a set of pre-specified closed ports. Once a correct sequence of connection attempts is received, the firewall rules are dynamically modified to allow the host which sent the connection attempts to connect over specific port(s).

Originally, this was simply conceived as a series of connection attempts to closed ports in a specific order. For example, a client attempt to connect to a certain port knocks on ports 10001, 220202, 4444, or any other ports sequence. The PK server checks the sequence of incoming packets, if they are in the correct order that the client and server have agreed on, the PK server informs the firewall to open port 22 (SSH) to the client requesting the service.

The problem today is the world is full of security threats, it should be assumed that all traffic is monitored by an unknown third party as it travels across a network. Doggedly adhering to this viewpoint provides us with the fact that our knock sequence can be passively observed by an eavesdropping person in the middle of our connection and just replays the knock sequence to get the same response from the server. This problem is called “TCP replay attack”.

So we have to find a solution were the knock sequence is not re-playable. Any host connected to the Internet needs to be secured against unauthorized intrusion and other attacks. Unfortunately, the only secure system is one that is completely inaccessible, but, to be useful, many hosts need to make services accessible to other hosts. While some services need to be accessible to anyone from any location, others should only be accessed by a limited number of people, or from a limited set of locations. The most obvious way to limit access is to require users to authenticate themselves before granting them access.

Problems Associated with PK Techniques

In order to increase network security, it is sometimes desirable to allow access to open ports on a firewall only to authorized external hosts (users) and present closed ports to all others. The most obvious way to limit access to open port is to require users to

authenticate themselves before granting them access. There are a number of techniques that have been developed by many researchers to create port authentication, such as: PK, single packet authentication (SPA), or use a lightweight concealment protocol.

The investigations on the performance of these techniques in avoiding all possible types of port attacks (e.g., 0-day attacks, TCP-replay attacks, dictionary-based attacks, root-kit attacks, and brute-force attacks) have demonstrated that most of these techniques suffer from either one or more of the following problems:

- 0-day attacks.
- The sequence replay attack.
- Minimal data transmission rate.
- Knock sequences and port scans.
- Knock sequence busting with spoofed packets.
- Failure if a client is behind a NATed network.
- Failure if packets are received/delivered in out of order.
- A lack of association between authentication and connections being opened
- Flaws in how cryptography is applied to provide authentication.
- Data extraction from eavesdropped packets.

PK techniques have been studied by many researchers and they developed their models trying to avoid all possible types of port attacks that may threat network security. The next section presents a description of a new PK technique that can be used for efficient, reliable, and cost-effective host authentication, called the hybrid PK (HPK) technique.

The Proposed HPK Technique

The HPK technique consists of seven main steps. In what follows, is the description of the seven steps [5].

Traffic monitoring

In this step, a PK server is installed behind the network firewall, as shown in Fig. 1, monitoring and checking traffic arrived to firewall (gateway).

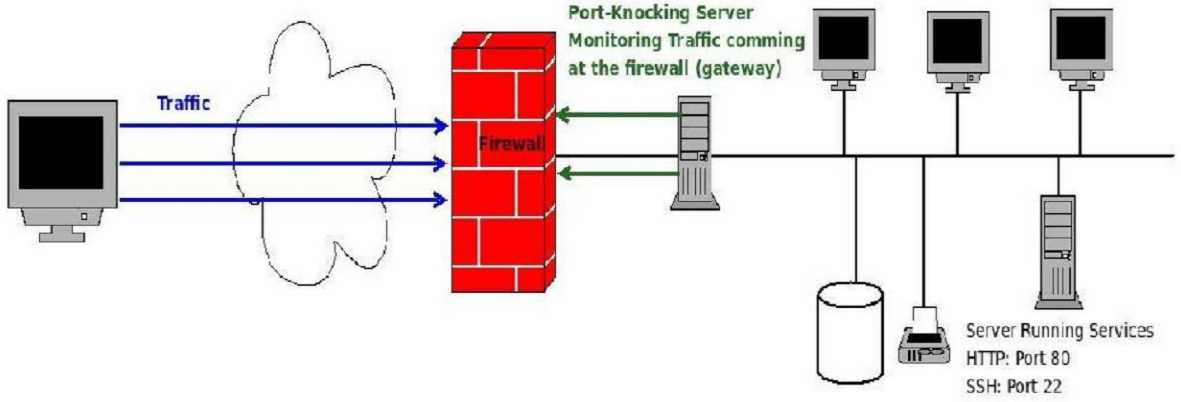


Figure 1: Traffic monitoring

Traffic capturing

In this step, the PK server captures only the traffic holding a payload (image) for further processing, as shown in Fig. 2. In this figure, for example, only Traffic #3 is captured for further processing because it contains an image.

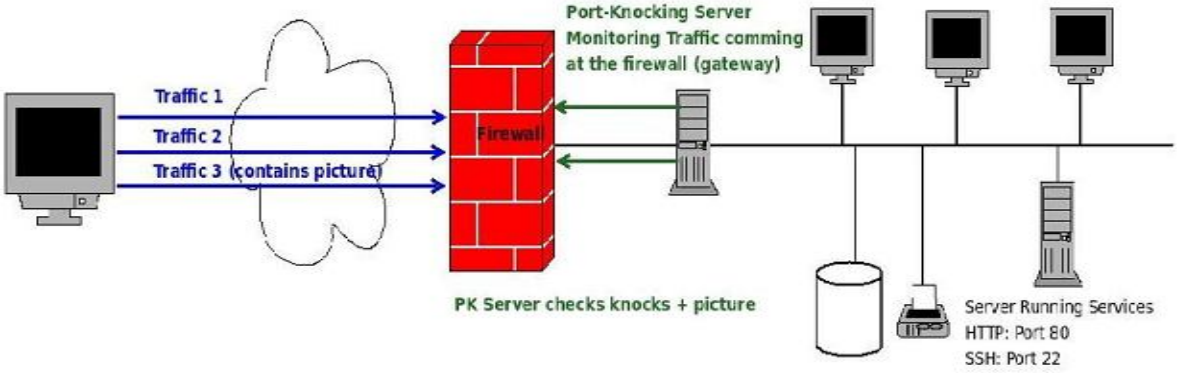


Figure 2: Traffic capturing

Image processing

In this step, the PK server extracts the payload (image) from the received packet. The payload is supposed to hide some information using Steganography that can be used to prove the knockers identity and request.

If the payload, contains intended information, which is either to demand the firewall to open/close a port for the client as shown in Fig. 3, or execute a command remotely on the appropriate server as shown in Fig. 4. Otherwise, if the result of the image processing fails to reveal valid authentication parameters, the PK server blocks the IP address of the source that sent the knocks and the payload (image). No port open/close or remote command execution is done in this step, only ensuring that the received payload holds a request which needs further authentication.

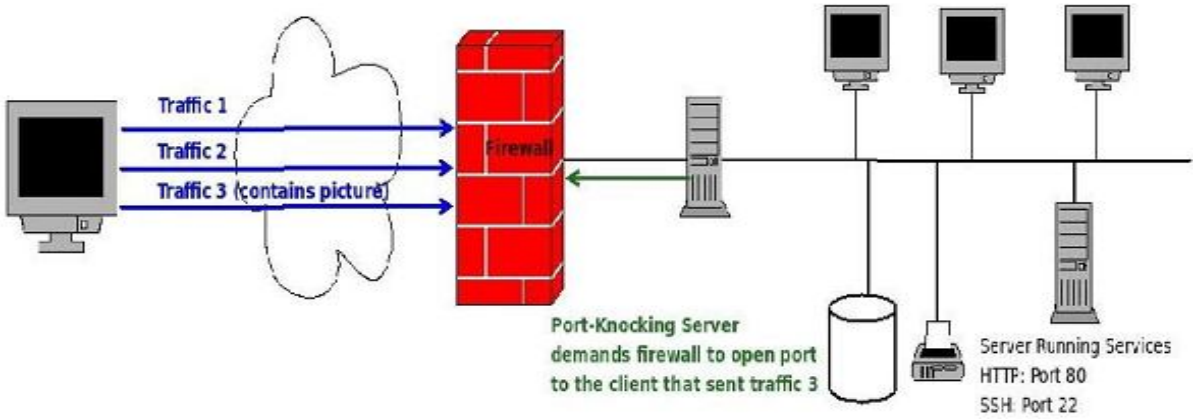


Figure 3: PK server demanding firewall to open port

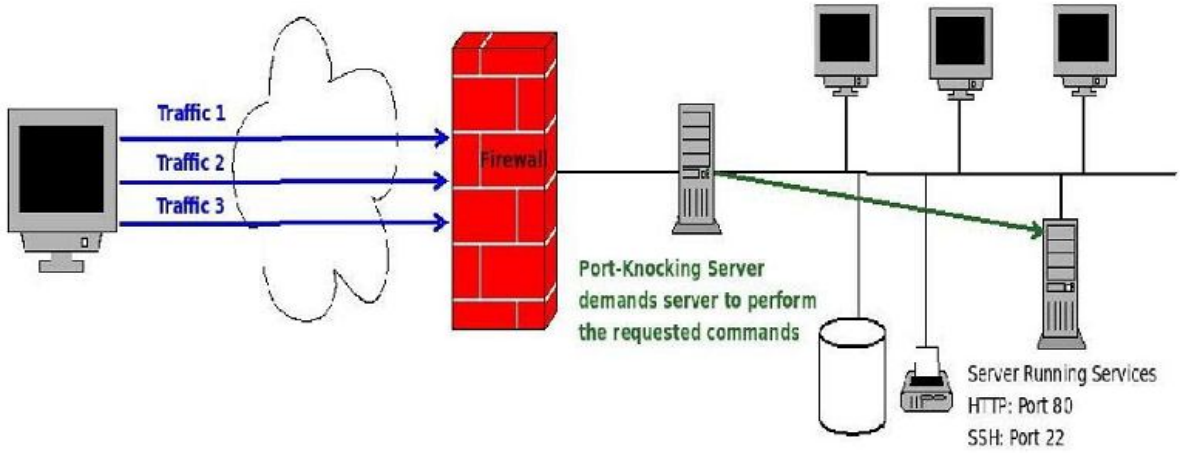


Figure 4: PK server executing clients command request

Client authenticating

After the PK server makes sure that the payload was carrying an intended request, it needs to make sure that it is communicating with the correct client, so it takes a

Random number and encrypts it using the clients GnuPG public key and sends it as a payload to the client.

Server authentication

The client now receives the packet carrying the encrypted payload, extracts it and decrypts it using the servers GnuPG public key. Then the client sends the random number as a payload back to the PK server to ensure its identity.

Proving the identity of the client

The PK server is still in the monitoring/sniffing state and receives the reply from the client to its random number check. The server extracts the payload and checks if the received message holds the same number as the one randomly generated and sent to the client. If the message is identified then the PK server executes the opening/closing of the requested port on the firewall, or executes the remote command based on the client's request.

Port closing

Finally, in this step, after the task is completed, either the client informs the PK server to close the port, or the PK server decides to close the opened port after specified silent period on that open port as shown in Fig. 5.

In any of these two cases, the PK server demands firewall to close the open port. In this case, if the client wants to access the system again, it needs to initiate new access or authentication request, i.e., start from phase #1

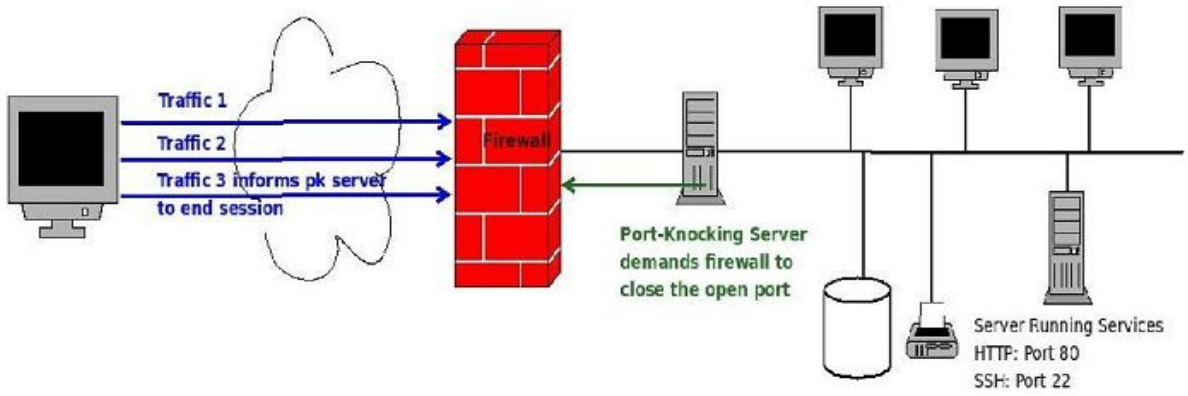


Figure 5: Port closing

Security Measurements and Evaluation

In order to evaluate the security of the HPK technique, a number of hacking scenarios were compared with two other well known port-knocking techniques (Traditional Port-knocking, and Single Packet Authorization). The scenarios performed were the following:

- TCP replay attack
- DoS attack

The technique has been implemented using Python programming language [6] on a Fedora GNU/Linux operating system [7]. Also, a number of standard tools will be used to perform port monitoring, checking, processing, and closing, and also simulating various port attacks, such as:

- Wireshark for network protocol analysis [8].
- Tcpreplay for replaying captured network traffic [9].
- Nmap for network exploration and port scanning [10].

Test Results

When investigating the TPK technique, it found that it is vulnerable to DoS attack, because the technique doesn't have any detection capability and is by default vulnerable to a TCP replay attack. The SPA technique is also found to be vulnerable to DoS attack, because the technique can only detect DoS attack, but cannot countermeasure against the host causing the attack. But, the HPK technique is found to be immune to flooded spoofed knock packets or DoS attack, because the technique has a built-in detection capability that can be adjusted to countermeasure after a specific number of failure attempts.

CONCLUSION

The main innovative idea in the HPK technique is that it is designed to work in two different modes without pre-adjustment or setting, namely the interactive mode and the non-interactive mode. In any of the above modes, the HPK client does not send TCP SYN packets to initialize the service on the HPK server as in TPK techniques; instead it sends TCP packets with sophisticated payloads. The payloads send within the TCP packets represent the content of the service or task that needs to be performed on the accessed network or any of its servers.

The main conclusions of this work can be as follows:

- The HPK technique can be easily implemented on any GNU/Linux firewall box.
- The HPK technique is immune to a TCP replay attack, because it uses cryptography and Steganography within the TCP payload, and mutual authentication to authenticate both parties together.
- The HPK technique is immune to a denial-of-service (DoS) attack, because it has a built in detection system with the ability to counter measure against hosts causing such attacks.
- The HPK technique is much more secure than the traditional PK (TPK) and the single packet authentication (SPA) techniques, because solved problems that others failed in.
- Starting a MySQL service and adding a new user to the system remotely using the HPK technique was performed to prove that HPK can execute and perform remote tasks, not just open/close firewall ports.
- The communication protocol used is a simple secure encryption scheme that uses GnuPG keys with Steganography constructions.
- The HPK technique is implemented using threads technology in case more HPK processes are needed (i.e., more clients requests are received).
- The HPK technique is highly configurable to suite network needs and completely open source, and uses GNU General public license version 3 (GPL3).

REFERENCES

[1] B. Rudis. The Enemy Within: Firewalls and Backdoors. Securityfocus, June 2003. Available at http://www.symantec.com/connect/articles/enemy-within-firewalls-and-backdoors. Access Date 16-07-2010.

[2] W. Sonnenreich, and T. Yates. Building Linux and OpenBSD Firewalls, Wiley, New York, 2000.

[3] A. Tongaonkar, A. Tongaonkar, N. Inamdar, and R. Sekar. Inferring Higher Level Policies from Firewall Rules. Proceedings of 21st Large Installation System Administration Conference (LISA '07), USENIX Association, pp. 17-26, Dallas, USA, November 2007.

[4] J. Song, H. Takakura, and Y. Kwon, A Generalized Feature Extraction Scheme to Detect 0-Day Attacks via IDS Alerts Proceedings of the 2008 International Symposium on Applications and the Internet - Volume 00, IEEE Computer Society Washington, DC, USA, 2008

[5] Ali Hussein, 2010, "A Hybrid Port-Knocking Technique for Host Authentication", Ph.D. Thesis, University of Banking and Financial Sciences.

[6] Python Programming Language http://www.python.org/

[7] Fedora GNU/Linux Operating System https://fedoraproject.org/wiki/Fedora_Project_Wiki

[8] Wireshark, Network Protocol Analyzer http://wireshark.org.

[9] tcpreplay, Replay captured network traffic, http://tcpreplay.synfin.net/trac/

[10] Gordon Lyon (aka Fyodor Vaskovichr), Nmap ("Network Mapper") utility for network exploration or security auditing, Phrack Magazine, Vol. 7, Issue 51, article 11 of 17, September 01, 1997.

★★★