Chapter 23
A Hybrid Port-Knocking Technique for Host Authentication

Ali H. Hadi
Philadelphia University, Jordan

Hussein Al-Bahadili
Petra University, Jordan

ABSTRACT

This chapter presents the detail description of a Port-Knocking (PK) technique, which should avert all types of port attacks and meets all other network security requirements. The new technique utilizes four well-known concepts, these are: PK, cryptography, steganography, and mutual authentication; therefore, it is referred to as the Hybrid Port-Knocking (HPK) technique. It is implemented as two separate modules. One is installed and run on the server computer, either behind the network firewall or on the firewall itself, and the other one is installed and run on the client computer. The first module is referred to as the HPK server, while the second is the HPK client. In terms of data processing, the technique consists of five main processes; these are: request packetization and transmission, traffic monitoring and capturing, mutual authentication, request extraction and execution, and port closing. The HPK technique demonstrates immunity against two vital attacks, namely, the TCP replay and Denial-of-Service (DoS) attacks.

INTRODUCTION

The network security has become a primary concern on the Internet in order to provide protected communication between hosts/nodes in a hostile environment. In order to protect network resources, each service provider pose a number of nontrivial challenges to security design and set its own policies for accessing resources on the network. These challenges make a case for building security solutions that achieve both broad protection and desirable network performance in terms of minimum data overhead and delay. It is so crucial to have computationally cheap and simple defense mechanisms that allow early protection against all types of attacks. In particular, it becomes very

DOI: 10.4018/978-1-4666-2919-6.ch023

common and useful to have multiple progressively stronger layers of security, rather than attempting to have a single perfect security layer.

The Internet can be seen as a huge network of different nodes (computers or any microprocessor-based devices) connected together providing different services for a wide range of users. A node on the Internet can act as a host and/or client. It is classified as a host when it provides services to other nodes on the Internet, while it is classified as a client when it is a user to a service provided by a host on the Internet. The services on the Internet can be classified in different ways depending on the classification criterion. Based on who can access these services, services on the Internet are classified into public and private services. Public services are those services that can be accessed by all users on the Internet, while private services are those who can be accessed by authorized users only.

Any host connected to the Internet to provide private services needs to be secured against unauthorized intrusion and a number of network attacks. Example of these attacks may include: illegally access private resources, flood the network with redundant messages to deny other users from using the available network resources, impersonate some legal users, modify or re-direct exchanged messages, etc.

A first defense solution is implemented on the Internet represented by using the firewall (Rudis 2003). Firewalls are usually running on the network layer, so that it can only see IP addresses and its characteristics but not a user name and password, which can only be seen by the application layer. In addition, 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.

As a result a mechanism is required to open ports on a firewall to authorized users, and blocking all other traffic users (Krzywinski, 2003a). The best way to perform such a mechanism is to run port authentication service on firewalls, which validates the identity of remote users and modifies firewall rules according to per-user access policies. Such a mechanism was used in a number of applications, such as: block SSH guessers (Google Groups, 2006), Symantec security response, fast port scan detection (Jung et al 2004), exploration of modern network threats (Krivis, 2004), remote operating system detection (Fyodor, 1998), etc.

There are a number of techniques that have been developed by many researchers to create port authentication, such as: Port-Knocking (PK) technique, which will be referred to as Traditional PK (TPK) technique (Krzywinski, 2003b); PK with Single Packet Authentication (SPA) (Rash, 2006a; Rash, 2006b); a lightweight concealment protocol (Barham, et al., 2002; Murdoch & Lewis, 2005), etc.

The concept behind the PK-based techniques can be explained as follows: When a client requires accessing a certain service or performs a specific task on a server through the Internet (which we refer to as a request). For example, a network administrator requires to access a certain task on his network remotely or a client wish for accessing a music server, and that particular service or task is hidden behind an unknown or closed port, which is also hidden behind a firewall. At the same time an attacker is presented on the network.

In PK-based techniques, in order to gain access to the network resources and pass the firewall and access a server behind the firewall, the client sends a sequence of TCP SYN packets, which are called "knocks," to closed-ports, each of them has a different destination port number. In other work, the client sends a specified and agreed on ports sequence through a number of A TCP SYN packets to the server to gain access to the network resources. The number of SYN packets sent by the client is equal to the number of ports in the ports sequence. For example, for a port sequence of four ports: 1111, 2222, 3333, 4444, the client