Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/jisa

A flow-based detection method for stealthy dictionary attacks against Secure Shell☆

*Akihiro Satoh**, *Yutaka Nakamura*, *Takeshi Ikenaga*

Kyushu Institute of Technology, 1-1 Sensui-cho, Tobata-ku, Kitakyushu-shi, Fukuoka, Japan

ARTICLE INFO

Article history:
Available online 26 September 2014

Keywords:
ssh dictionary attack
Flow analysis
Network operation
Machine learning algorithm

ABSTRACT

SANS has warned about the new variants of SSH dictionary attacks that are very stealthy in comparison with a simple attack. In this paper, we propose a new method to detect simple and stealthy attacks by combining two key innovations. First, on the basis of our assumptions, we employ two criteria: *"the existence of a connection protocol"* and *"the inter-arrival time of an auth-packet and the next"*. These criteria are not available, though, owing to the confidentiality and flexibility of the SSH protocol. Second, we resolve this problem by identifying *"the transition point of each sub-protocol"* through flow features and machine learning algorithms. We evaluate the effectiveness through experiments on real network traffic at the edges in campus networks. The experimental results show that our method provides high accuracy with acceptable computational complexity.

1. Introduction

Secure Shell (SSH) is run on many hosts with various scopes other than just operation, so a dictionary attack against SSH services is a common security threat. In addition to the attack, the SysAdmin, Audit, Network, Security (SANS) Institute (SANS Internet Storm Center) has warned about new variants, namely slow-motion SSH dictionary attack and distributed SSH dictionary attack. The variants are very stealthy in comparison with a simple one. Since even one success of these attacks causes serious problems, such as leaks of confidential information, transmission of spam email, and deployment of phishing sites, administrators should be prepared to cope with them.

SSH dictionary attacks have been detected in two basic ways that rely on either log files (Thames et al., 2008; Su et al., 2011) or network traffic (Sperotto et al., 2009; Takemori et al., 2009). The first approach parses the log file of all hosts in large networks, and thereby imposes heavy maintenance costs on administrators. The second approach limits the above costs because its requirement is to only capture traffic through a few observation points. However, this approach cannot distinguish between successful and unsuccessful attacks. Of more immediate concern, both approaches are ineffective in the case of stealthy attacks that have little impact on log files and network traffic. An ideal method should be able to detect stealthy attacks and to distinguish between their success and failure, based on network traffic of each connection.

In this paper, we focus on realizing such a method to enable secure networks. We have developed the method by combining two key innovations (Satoh et al., 2012). First, we employ two criteria in our assumptions that are derived by reference to SSH protocol specifications. The specifications show that an SSH handshake consists of three major sub-protocols — i.e., transport layer, user authentication, and connection protocols — and that an auth-packet contains a

☆ This article belongs to the special issue Security, Privacy and Trust in Future Networks and Mobile Computing.

* *Corresponding author.*
E-mail addresses: satoh@isc.kyutech.ac.jp (A. Satoh), yutaka-n@isc.kyutech.ac.jp (Y. Nakamura), ike@ecs.kyutech.ac.jp (T. Ikenaga).

http://dx.doi.org/10.1016/j.jisa.2014.08.003
2214-2126/

username and password pair for authentication. A summary explanation of the two criteria is as follows:

The existence of a connection protocol — a criterion to estimate whether a username and password pair is accepted in authentication, to distinguish between successful and unsuccessful dictionary attacks;
The inter-arrival time of an auth-packet and the next — a criterion to estimate whether a username and password pair is entered by user's keystrokes, to detect dictionary attacks.

These criteria are not available, however, owing to the confidentiality and flexibility of the SSH protocol. Second, we resolve this problem by identifying *the transition point of each sub-protocol* through flow features (Moore et al., 2005) and machine learning algorithms (Jain et al., 1999). A transition point is the point at which an SSH handshake shifts to the next sub-protocol in a flow. A flow is bi-directional packet exchanges between a client and a server with the same source address, source port number, destination address, destination port number, and protocol number, and its features are statistical patterns — in terms of, for example, packet size, packet inter-arrival time, and packet order — in externally observable packets taken from a flow. The reason for using flow features rests on two perspectives: (1) the transition point of each sub-protocol typically has the features distinct from those of non transition points; (2) the features are observable without direct packet inspection.

We evaluate the effectiveness of our method through experiments on real network traffic at the edges in campus networks. The experimental results show that our method provides high accuracy with acceptable computational complexity. The significant contribution is a means to alleviate the threat of simple and stealthy attacks that administrators will face in the future.

This paper is organized as follows. Section 2 summarizes related work and their limitations. Our findings from analysis of SSH dictionary attacks at flow level are given in Section 3. On the basis of these analytical results, our method is proposed in Section 4, and this method is evaluated in Section 5. We conclude and look at future work in Section 6.

2. Background

In this section, we describe the details of SSH dictionary attacks and SSH protocol specifications. We then discuss related work and their limitations regarding SSH dictionary attack detection.

2.1. SSH dictionary attack

An SSH dictionary attack is defined as a login attempt to gain fraudulent access by guessing a username and password pair. The attack relies on the fact that many users tend to choose their password from a small domain. A malicious client tries all possible username and password pairs until the correct one is found. As a result, these attacks contaminate log files and flood network traffic.

New variants have emerged as an invisible security threat: slow-motion SSH dictionary attack and distributed SSH dictionary attack. The former type is made by a malicious client, and its target changes one after the other. Specifically, the destination address varies with the login attempts though the source address is constant. The latter type is made by a large coordinated group of malicious clients, such as botnets. Each of the clients perpetrates login attempts against their target at an interval. To be precise, the source address varies with the login attempts though the destination address is constant. In both cases, malicious login attempts leave little impact on log files and network traffic because their number never exceeds single digits over a long time period. Consequently, these attacks are very stealthy in comparison with a simple one.

2.2. SSH protocol specification

An SSH handshake consists of three major sub-protocols: transport layer, user authentication, and connection protocols (Ylonen and Lonvick, 2006a, 2006b, 2006c).

A transport layer protocol negotiates encryption, integrity, compression, and key exchange algorithms to establish secure connections between a client and a server. For example, the encryption algorithms are AES-CBC and 3DES-CBC; the integrity algorithms are HMAC-MD5 and HMAC-SHA1. Note that encryption, integrity, compression algorithms are immediately applied after finishing this sub-protocol. Then, the SSH handshake shifts to a user authentication protocol. The role of this sub-protocol is to determine whether the server allows the client to establish connections via SSH. The client notifies the server of an authentication method name with its attribute values by sending an auth-packet, and the server returns the result of authentication. For example, the method names are password and public-key; the attribute values are username and password. Finally, a connection protocol provides various functions such as remote login, file transfer, X11 forwarding, TCP/IP forwarding, and so on.

The SSH protocol has two notable properties: confidentiality and flexibility. Confidentiality means encrypting connections, checking integrity, and authenticating each other. Flexibility means choosing suitable algorithms according to circumstances. In the transport layer protocol, for example, encryption, integrity, compression, and key exchange algorithms are negotiated independently for each host, so each host chooses its own algorithms from a set it supports.

2.3. Related work

Numerous studies are related to SSH dictionary attack. Traffic causality graphs (Asai et al., 2011) were proposed for visualizing and analyzing the temporal and spatial causality of flows to profile network applications without direct packet inspection. The results helped administrators identify the root that cause various attacks, including SSH dictionary attacks. Another line of work (Goyal et al., 2006; Alsaleh et al., 2012) was to design a secure protocol for preventing dictionary attacks. For example, Goyal et al. (Goyal et al., 2006) improved an authentication protocol by adding fast one-way hash functions and challenge-response exchanges, and the protocol