Summary

In the paper *SSHkex: Leveraging virtual machine introspection for extracting SSH keys and decrypting SSH network traffic*, the authors proposed a new way to decrypt SSH traffic, by extracting the encryption keys from the SSH server using VMI (Virtual Machine Introspection).

Abstract

SSH is a well-known application layer protocol for remote access. In digital forensics, forensic personnel need to investigate SSH traffic, especially when an attacker connects to a honeypot server. Since the traffic between SSH server and client is encrypted, forensic personnel needs to extract the traffic in plaintext. Although the forensic personnel has full control over the server VM, current implementation of traffic extraction either modifies the server system, or has large overhead by pausing the server frequently. The authors implement a new workflow called SSHkex. Using VMI (Virtual Machine Introspection), SSHkex does not modify the system and has small overhead by only pausing the server twice.

Introduction

Securing a remote communication was and still a big concern of network protocols developers. In the early days, many exploits and vulnerabilities were used easily to perform harmful and malicious actions against a victim. Some of the methods used by attackers such as Man-In-The-Middle, Bruteforcing or binary exploitation, are actively used by threat actors. SSH falls in the middle of this problem to "solve" these issues, by using certain defense mechanisms and secure configurations. It also serves as an encryption solution for the remote connections, since it follows a symmetric, asymmetric cryptography scheme and some extra hashing operations. When a client decides to initiate a remote connection to an SSH server, the whole process starts with a key exchange (KEX), and those keys will be used for encrypting and decrypting the packets during communication. The researchers of this paper came up with a way to exploit this process, using VMI, by setting up breakpoints in-memory processes and extracting authentication keys. This will help uncovering some structs that contain the keys.

Main Contribution

Design

The authors have designed the following workflow, that extracts the decryption keys for SSH traffic, while satisfying:

1. Stealthiness: the investigation setup cannot be easily detected by the attacker.
2. Preservation: the server system is not modified.
3. Evidence Integrity: the investigation outcome is correct and accurate.

Note that, other implementation that has a lot of overhead will violate stealthiness, because the attacker can detect the abnormal pauses or delays. Similarly, other implmentation that modifies the server system or `sshd` binary will violate preservation.

Key Ideas

The design based on the assumption that traffic between server and attacker is passively captured thoroughout the whole process.

1. Using VMI, insert breakpoints before execution of `kex_derive_keys` and `do_authentication2` functions of `sshd`.
2. When the first breakpoint is hit, use VMI to read the first argument passed into the `kex_derive_keys` function. The argument is the memory address of the `ssh` struct.
3. When the second breakpoint is hit, the keys are already stored in the `ssh` struct. Dump the `ssh` struct at the previously obtained address.
4. Extract keys and IVs from the struct and decrypt captured traffic with Wireshark.

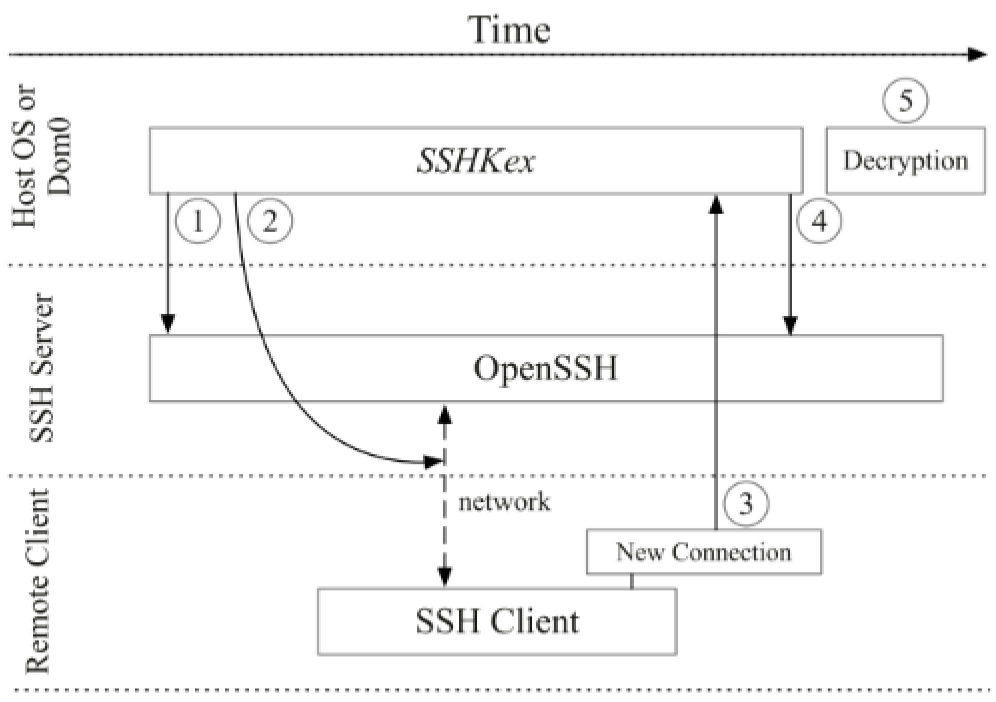

Fig. 4. *SSHkex*'s high-level design.

Implmentation

The authors' implementation of SSHkex is available at https://github.com/smartvmi/SSHKex.

Experiment using this implementation shows that SSHkex can extract decryption keys accurately, with a small overhead of less than 100 millisecond.

Backgrounds

While SSH has been extensively studied, limited attention has been given to the extraction of SSH keys and the decryption of SSH network traffic using virtual machine introspection techniques. Traditional approaches rely on monitoring network packets or exploiting vulnerabilities in the SSH implementation, but these methods often have limitations in terms of effectiveness or practicality. By exploring the potential of virtual machine introspection, which allows for deep introspection into the runtime behavior of virtual machines, this research addresses an important knowledge gap and offers a novel perspective on SSH security.

Few studies have examined the possibilities of extracting SSH keys and decrypting SSH network traffic through virtual machine introspection. This research paper builds upon the foundations laid by previous studies in SSH security while exploring a unique approach to extract keys and decrypt traffic using virtual machine introspection.

The primary objective of this research paper is to investigate the feasibility of using virtual machine introspection to extract SSH keys and decrypt SSH network traffic. The research aims to address the following key research questions: (1) Can virtual machine introspection techniques be utilized to extract SSH keys from a running virtual machine? (2) Is it possible to leverage virtual machine introspection to decrypt encrypted SSH network traffic? By answering these research questions, this study intends to provide valuable insights into the security of SSH and contribute to the ongoing efforts to enhance SSHkex defenses.

Conclusion

In summary, this research paper investigated the use of virtual machine introspection to extract SSH keys and decrypt SSH network traffic. The study identified a gap in existing knowledge regarding these techniques. The findings demonstrated that virtual machine introspection can successfully extract SSH keys and decrypt encrypted SSH network traffic. This research contributes to enhancing SSH security and provides valuable insights for system administrators and researchers. Further research is needed to develop more advanced intrusion detection systems and tools tailored for SSH security analysis. Overall, this study highlights the importance of continuous efforts to improve SSH security and protect sensitive information in modern computing environments.