Detection of SSH Host Spoofing in Control Systems Through Network Telemetry Analysis

Stanislav Ponomarev
Louisiana Tech University
Ruston, LA, 71270
spo013@latech.edu

Nathan Wallace
Louisiana Tech University
Ruston, LA, 71270
nsw004@latech.edu

Travis Atkison
Louisiana Tech University
Ruston, LA, 71270
atkison@latech.edu

ABSTRACT

Modern networking architecture is designed with high scalability in mind. Different protocols can be encapsulated to support different systems. Machine identifiers (IP and MAC addresses) in network packets can be modified easily. This modification prevents servers from determining whether the connecting machines are allowed to communicate. Cryptographic functions have been used in protocols such as Secure Shell (SSH) to establish network node authenticity, but they can be circumvented by social engineering and brute force attacks. This research effort created a new classifier that processes network telemetry to determine authenticity of SSH clients in a control systems network. Developed classifier, within the control systems network, was able to differentiate with a 100% accuracy SSH connections from machines that were transmitting identical MAC and IP addresses, and had the same RSA key for authentication.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General —*Security and protection*

General Terms

Algorithms, Design, Security

Keywords

Network telemetry, Secure shell, Network latency

1. INTRODUCTION

Network models currently implemented allow for easy access to various networked systems through a multitude of mediums, devices, and implementations. This approach has allowed computer networks to spread across the globe. Network models create abstraction layers, and keep some information away from the application. This reduces the ability to establish authenticity of the host transmitting the data. [9] Because of this, tools have been developed that allow an attacker to perform complex data manipulations on the network to achieve penetration of target machines.

The SSH protocol was created to allow the use and management of remote machines, without the need for the user to be physically present. By having means to execute commands on a remote machine, an attacker can perform many tasks from planting a virus, to getting users' sensitive data or joining the machine to the botnet. Many approaches exist to secure the data being transmitted by SSH, which include various encryption schemes, and host filters. Christodorescu calls the problem of malicious code as a "game between malicious code writers and researchers working on malicious code detection" [6], but the same principle applies to many aspects of security, including the network security.

A typical usage scenario will include an SSH server, behind a firewall, using a set of private and public keys to limit the access to the server only for a set of specific people on the network. However, as the security research grows, so does the research to circumvent security measures. Firewalls can be penetrated [11], passwords can be decrypted [16], and new techniques are always found to break into a secure system.

This research effort proposes a new scheme for detecting a network intrusion based on the meta-data that is not directly transmitted in the packet, but rather data that is associated with it, for example latencies, throughput, sequence timings, etc. Unlike packet data, which can be easily altered by an attacker, the network's meta-data is difficult to spoof, which makes it a perfect target for a host fingerprinting method.

2. BACKGROUND

Secure shell (SSH) protocol operates on top of the TCP/IP stack and provides a secure connection to a machine over an unsecured network [15]. IP protocol is connection-less network protocol that does not guarantee the data to arrive to the destination, and if it does arrive, it may arrive out of order. IP protocol implements an address field which allows IP packets to be routed to a proper destination, and lets the destination machine know where to send the reply. However, these fields do not guarantee that the machine at a given address is a machine that is supposed to be there. Changing the address of IP packets is called an IP spoofing and allows for many different attacks [10].

IP protocol commonly utilizes Ethernet frames to allow pack-

CISR '14, April 8-10, Oak Ridge, Tennessee, USA Copyright C 2014 ACM
978-1-4503-2812-8... $15.00

ets to be forwarded between multiple nodes of the network until they reach their final destination. To achieve this, Ethernet frame headers contain the source and destination media access control (MAC) addresses. While IP addresses are being assigned to different machine interfaces by network administrators or self-configuration protocols, MAC addresses are unique to each network interface and are assigned during the manufacturing process of the interface's controller circuit [8].

Both, IP and MAC addresses can be spoofed by software, preventing the server from being able to determine whether the packets are arriving from an authorized location. [3, 13] That's why SSH implements encryption based algorithms to prevent illegal access and enforce data integrity. [4] However, no matter how secure a cryptographic function can be, it can be attacked with at least a brute force attack. [2] Encryption, while effective to deter most attackers, can be broken in many different ways. Hashes can be reversed, users can be tricked into giving their passwords to a phishing sites, weak passwords can be guessed. [16] To improve security scheme of SSH protocol, features that the attacker has low control of should be used.

There have been various studies in anomaly detection which use data mining and machine learning facilities to detect anomalies. [5] NetMine is a data mining type application that specializes in understanding traffic data correlations and interactions. [1] While implementing methodologies similar to the ones used in this research effort, both of these studies focus on feature generation to improve traffic quality, and network stability rather then network security.

In their works, Erman et al. [7] were able to successfully cluster similar packet types by analyzing transport layer statistics. By using K-Means and DBSCAN algorithms they were able to successfully identify protocols being used, without extracting the data from packets [7]. Sheng et al. showed that it is possible to detect host spoofing by analyzing statistical fluctuations in received signal strength of the packets transmitter over wireless networks [13].

Wireshark is a software network analyzer, which allows to capture network traffic, and visualize it in real time. It also allows system administrators to save all of the received packets for future analysis, and extract useful information about these packets. Wireshark was used in this research effort to extract packet arrival times into a comma separated values format used to generate graphs and interpret the data. [12]

Though the detection approach presented in this article may not withstand the dynamics of a typical enterprise LAN, the approach will be beneficial in the detection of spoofed hosts in control system LANs. Control system LANs are unique in that hosts generally communicate in set intervals passed on the polling protocol utilized. The developed detection scheme will be able to determine when communication is initiated and maintained outside of these intervals and upon detection will alert on a possible intrusion. Furthermore, with the use of open source tools that automate the attack process against control systems [14], this work proves to be fruitful as it can distinguish between legitimate and spoofed packets.

3. METHODOLOGY

The attack model developed for this research effort was composed of benign and malicious clients trying to communicate with a secure server (Figure 2) similar to methodologies of Sheng at al. [13] . However, instead of using received signal strength of the wireless transceiver, this research effort is aiming to detect malicious hosts, not only on a wireless network, but in wired networks and networks of various infrastructures. To achieve this, inter-packet delays were used.

Unlike the data in an Ethernet packet, the attacker has less control of the timings of the packets being transmitted. All of the server processed packets were captured using wireshark. Those packets were then graphed by time of arrival. Figure 1 shows the generated graph. All of the protocols that wireshark was able to identify are separated into individual categories. After observing SSH handshake pattern, it became evident that a classifier can be made to differentiate between hosts.

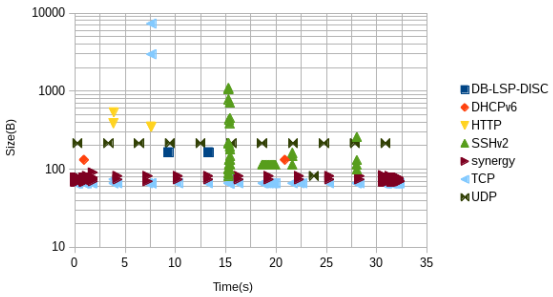

Figure 1: Server's normal operation

By looking at the relationships between packets, Erman et al. had to use various features to be able to cluster similar packets. [7] However, a node fingerprint can be built simply by observing differences in sequential packet arrival times. There are many different parameters that can affect packet arrival times. They range from the CPU load of the node to the load on the network between nodes. Because of this, methodologies presented by this research are better suited to secure connections over networks of constant or near constant load, such as control system LANs.

To account for variance in the network and machine CPU load, standard deviation was calculated over the received samples of benign client (Figure 1). The authenticated window was built by $p \in [\bar{x} - S_N, \bar{x} + S_N]$ where p is the inter-packet arrival time, S_N is standard deviation, and $\bar{x}$ is the average inter-packet arrival time of the benign client.

$$S_N = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \quad (1)$$

4. EXPERIMENT

In the experimental attack scenario, an attacker was able to gain access to a control system's local area network. A portable computer was attached to the system's LAN and

used to execute the attacks. The attacker was able to gain access to the encryption keys, an authorized user had, as well as spoof the workstation's MAC and IP addresses, to prevent the secure server from recognizing or logging any potentially malicious activity. Once the spoofing detection algorithm was established, a second trial from a different machine is executed to test the intrusion detection algorithm.

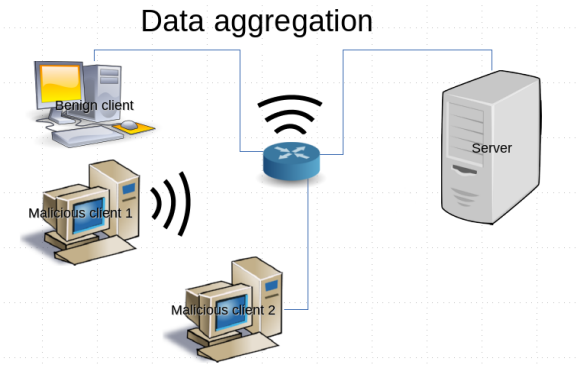

Figure 2: Experimental Setup

4.1 Experimental setup

Secure server was running Gentoo Linux environment on Intel Core 2 Duo E7500 CPU, 2GB RAM, and Intel's 82567LM-3 ethernet controller. Benign client was running on a separate machine of the same configuration, while the malicious client was running Debian Linux, on a BCM2835 SoC controller connected to the local area network using RTL8188CUS 802.11n WLAN Adapter. For the second attack a Windows 7 machine running on AMD Phenom II X2 555 CPU, 8GB of RAM, and a Realteck PCIe GBE family network controller was used. All of the machines were connected with each other though a Netgear WNDR3500 wireless router (Figure 2).

5. RESULTS

Tcpdump utility was used on the secure server to log all of the packets captured by the server's network card during the experiments. First experiment determined a pattern of the proper SSH RSA handshake. The timings of SSH RSA handshake for the benign machine can be found in Figure 3.

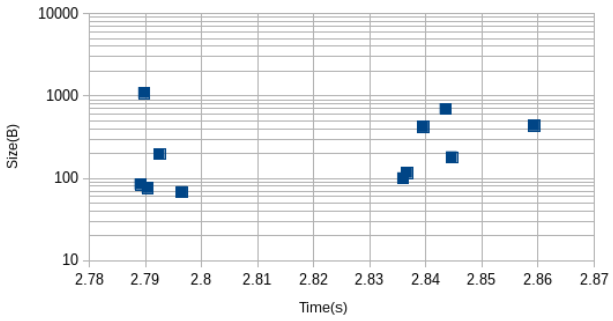

Figure 3: SSH RSA handshake

To remove the human input timings, an *ssh-agent* program was used to cache the RSA decryption key. This allows the RSA public key to be encrypted and does not require a user input to decrypt it while calling the *ssh* command.

After the handshake pattern was established, both benign and malicious clients executed an SSH connection command to the secure server, while capturing all the packets with tcpdump utility. SSH software was instructed to establish a connection using a public/private key pair, execute a *uname -a* command which returns a string of text and exits:

$$\#\ ssh\ \ secure\_server\ \ uname\ -a$$

Once the experiments were finished, the differences between packet arrival times were recorded and graphed (Figure 4). First three benign trials are SSH sessions from the benign machine to the secure server. Login Delta highlights the packets that are used in authentication. Malicious trials 1 through 3 are from the portable computer, while Malicious trial 4 is run on the verification machine. Packets of the benign connection had very small deviation, which improved detection accuracy. A classifier was then designed to measure these packet arrival differences, and decide whether a connected host was authorized or not.

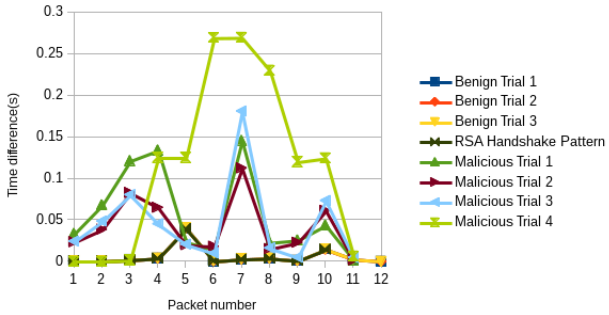

Figure 4: Packet arrival time differences

The variance in packet arrival times from authorized host was measured, and standard deviation was used to determine whether the connection is authorized or not. Figure 5 shows an average arrival time, as well as the standard deviation window. Each column represents a packet in a sequence of SSH handshake authentication. Column hight shows average time between each packet being processed. Window on top of each column represents the window of benign authentication derived by $p \in [\bar{x} - S_N, \bar{x} + S_N]$.

Same process was applied to the data extracted after a malicious host attempted to log into the secure server. Figure 6 shows packet arrival differences for one of the malicious login attempts. All of the times between processed packets were different from the times in benign trial, allowing to classify this connection as malicious.

6. CONCLUSIONS

Packet arrival times can be used as features in network intrusion detection systems from hosts within the local area network of the system being protected. For a wired network with constant average load, the range of values for trusted

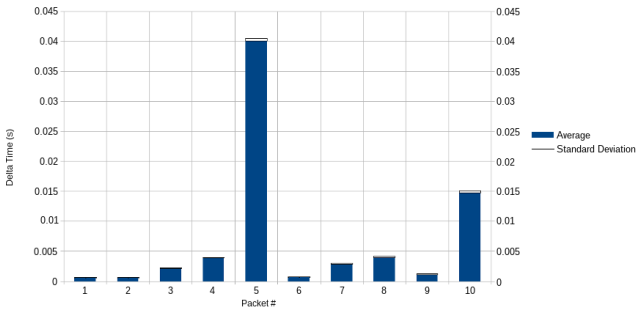

Figure 5: Standard Deviation of packet arrival differences

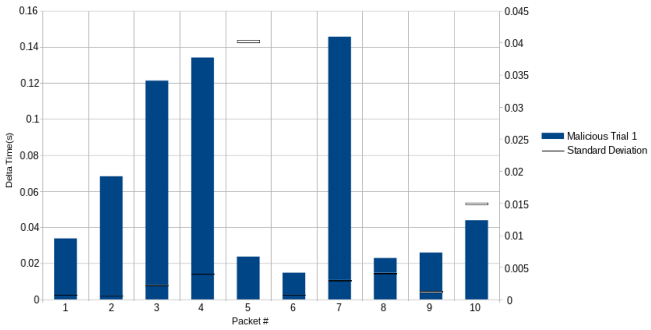

Figure 6: Standard Deviation of packet arrival differences during an attack

host is low, under 0.010 seconds, allowing for a high accuracy detection of malicious SSH authorizations.

Inter-packet arrival times can be varied due to many external factors, including software versions and operating systems being used, CPU load of the host machine, and the load of the network between the host and server. This makes it hard for the attacker to spoof these values, but a high variance of those features may also lead to a large allowed authentication window, which will decrease the accuracy of this method.

7. REFERENCES

[1] D. Apiletti, E. Baralis, T. Cerquitelli, and V. DâĂŹElia. Characterizing network traffic by means of the netmine framework. *Computer Networks*, 53(6):774–789, 2009.

[2] K. Apostol. *Brute-force Attack*. SaluPress, 2012.

[3] A. Belenky and N. Ansari. Ip traceback with deterministic packet marking. *IEEE Communications Letters*, 7(4):162–164, 2003.

[4] M. Bellare, T. Kohno, and C. Namprempre. Authenticated encryption in ssh: provably fixing the ssh binary packet protocol. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 1–11. ACM, 2002.

[5] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[6] M. Christodorescu and S. Jha. Static analysis of executables to detect malicious patterns. Technical report, DTIC Document, 2006.

[7] J. Erman, M. Arlitt, and A. Mahanti. Traffic classification using clustering algorithms. In *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, pages 281–286. ACM, 2006.

[8] F. Guo and T.-c. Chiueh. Sequence number-based mac address spoof detection. In *Recent Advances in Intrusion Detection*, pages 309–329. Springer, 2006.

[9] L. T. Heberlein and M. Bishop. Attack class: Address spoofing. In *Proceedings of the 19th National Information Systems Security Conference*, pages 371–377, 1996.

[10] B. Ho and T. T. Vu. Ip spoofing.

[11] J. LIU and G.-y. QIU. The firewall penetrating techniques based on the inverse connection, http-tunnel and sharing dns. *Journal of Zhengzhou University of Light Industry (Natural Science)*, 5:014, 2007.

[12] A. Orebaugh, G. Ramirez, and J. Beale. *Wireshark & Ethereal network protocol analyzer toolkit*. Syngress, 2006.

[13] Y. Sheng, K. Tan, G. Chen, D. Kotz, and A. Campbell. Detecting 802.11 mac layer spoofing using received signal strength. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1768–1776. IEEE, 2008.

[14] N. Wallace and T. Atkison. Observing industrial control system attacks launched via metasploit framework. In *Proceedings of the 51st ACM Southeast Conference*, ACMSE '13, pages 22:1–22:4, New York, NY, USA, 2013. ACM.

[15] T. Ylonen and C. Lonvick. The secure shell (ssh) protocol architecture. 2006.

[16] Y. Zhang, F. Monrose, and M. K. Reiter. The security of modern password expiration: an algorithmic framework and empirical analysis. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 176–186. ACM, 2010.