A Signal Theory Model for Security Monitoring using CheckMK

Iliuță-Alexandru IONEL
Faculty of Electronics, Telecommunications, and Information Technology,
University POLITEHNICA of Bucharest, Romania
iliuta.ionel@stud.etti.upb.ro

Abstract

Continuous monitoring of intelligent systems is used to analyze data and text from various sources. They usually monitor things such as risk, controls, opportunities, competition, and other concerns. While there exists literature that provides information on the capabilities of this kind of system, there has been a limited theoretical development in this field. The information sources monitored by these systems provide signals related to events, activities, or issues. However, selecting the appropriate information sources is not a simple task, because it is influenced by factors such as time, cost, redundancy, reliability, or weak signals. Furthermore, for the monitored signals, it is recommended to generate some analytics to study the flow and have a traceability of the issue we are dealing with. In this paper, a signal theory model is introduced and applied to address some of these issues regarding the SSH brute-force attacks. I will use a tool called CheckMK and its capabilities to implement a signal theory model used for monitoring security of a system [1].

Index terms: Brute force, Monitoring, Security, Signal, SSH

1. Introduction

Performing security audits on large-scale networks is a complex task, as the configurations of networked devices are constantly changing in the ever-evolving environment of today's technology. Many critical devices use the Secure Socket Shell (SSH) protocol to expose their remote access interfaces to the internet, and default usernames and passwords are often used. The issue has been exacerbated by the widespread availability of stolen credentials, which enables attackers to pose as authorized users and gain unauthorized access to internal networks, compromising sensitive data and misusing computational resources to gain some form of leverage from the legitimate users or administrators. Although the success rate of such attempts is low, they have resulted in significant consequences for 51% of the 1,800 organizations surveyed, with a financial impact of up to $500,000 per organization [2]. Brute force attacks are among the most common types of attacks on machines connected to the internet. Brute force attacks are essentially a trial-and-error method of guessing login credentials by attempting numerous combinations of usernames and passwords until the correct combination is discovered.

To protect against these attack vectors, several defense techniques have been developed. One of the most common techniques is to implement a password policy that mandates the use of complex passwords or passphrases, which are more difficult to guess. This can involve requiring a certain length or complexity, and even enforcing regular password changes. Another technique is to implement account lockout policies that temporarily lock an account after a certain number of failed login attempts, which can prevent brute force attacks from being successful [3].

2. Related Research

2.1. Signal Theory

Professor Emeritus Michael Spence from Harvard and Stanford Universities observed, "If the incentives for veracity in reporting anything by means of a conventional signaling code are weak, then one must look for other means by which information transfers take place." This implies that signals provide information about individuals, events, activities, and more, when conventional signaling codes fail to provide accurate information. For example, a college degree can serve as a signal of an individual's capabilities. Professor Brian Connelly et al. provided a definition of signaling theory, stating, "Signaling theory is useful for describing behavior when two parties (individuals or organizations) have access to different information." Information asymmetries can arise due to differences in experiences, opinions, observations, and even incentives, resulting in different information being available from different sources [1][4][5].

Signal theory has been applied in various fields of business research, including accounting, economics, finance, and management. For example, education serves as a signal for potential employees to showcase their market value in human resources. Similarly, faculty members may use information about their research publications and teaching experiences to signal their value to other universities.

In the context of continuous monitoring systems (CMS) for security purposes, signal theory is especially relevant due to the multiple information sources used and the need to choose appropriate analytics. These sources can vary significantly in terms of availability, reliability, informativeness, and veracity, and often contain unstructured data that requires structuring for analysis. Signal theory is useful for aligning analytics with the relevant signals, such as counting the number of relevant topics identified [1].

2.2. SSH (Secure Shell)

Secure Shell (SSH) is a critical remote access protocol that allows users to log into or execute commands on a remote computer, copy files, and perform other functions. Compared to plain-text communication protocols like Telnet, which lack encryption, SSH provides secure management and confidentiality of communications by supporting remote system communications through strong authentication and encryption.

While direct console connections are recommended for initial network system setup, SSH is commonly used to enable remote access. However, allowing SSH service from all origins may expose a system to threats, as not all remote access requests come from managers. Therefore, it is important to establish policies that only permit remote access to SSH from manager IPs and block the others IPs, and to define appropriate firewall start and end points [6].

2.3. Brute-force attack

Brute-force attacks involve submitting all possible account inputs to access the system's account information. These attacks use dictionary or random sequence methods, with the former trying all strings in a pre-arranged listing and the latter trying all possible string combinations in a sequence. To counter brute-force attacks, access can be controlled when an incorrect password is entered beyond a set number of times. For servers, the account lock policy restricts access once the login failure threshold is exceeded. Password policies also help to protect the system and its accounts by setting the password complexity, usage duration, and minimum length. Network systems prevent unauthorized access through the ACL(Access Control Lists) and track unauthorized attempts for reference in the access control policy. While network systems do not block access following login failures exceeding the threshold, logs of many attempts of unauthorized external access on Internet routers can be a valuable reference [6].

2.4. Server Virtualization

A virtual machine (VM) acts as an intermediary layer between the hardware components and the user, enabling the execution of an operating system within a virtual environment. Virtual machines are sometimes referred to as virtual servers and can be hosted on a physical server that shares its hardware resources such as CPUs, memory, and I/O (input/output ports) among the virtual machines. In contrast, a "real machine" or a bare metal system refers to the host operating system and hardware components such as the memory, CPU, motherboard, and network interface. Virtual machines are built on top of the core components of a real machine, and the communication between the virtual and real machines is facilitated by hypervisors or virtual machine monitors (VMMs). Hypervisors provide a layer of abstraction that allows different virtual machine operating systems and configurations to run on the same real machine hardware components. While hypervisors and emulators are similar, hypervisors are more efficient as they have specialized management functions that enable multiple virtual machines to co-exist peacefully and share real machine resources. Therefore, hypervisors and emulators are different primarily in terms of their semantic meanings rather than their functionalities [7].

2.5. Network Security Monitoring

A network monitoring system is utilized to monitor the internal network for identifying slow or failing system components, and reporting and resolving problems. Continuous monitoring of the network helps maintain high-performance networks with little downtime, regardless of whether it is a small business or a large enterprise. Monitoring reports cater to different levels of audiences, including the network and systems administrators and management. Therefore, a monitoring system should have basic reporting and drill-down functionalities, but it should also be easy to understand and use, and not too complex so basic users can understand and properly use the functionalities of the monitoring system. Network monitoring covers every aspect of a networked system, including response time, availability, uptime, and security, making it a difficult and demanding task. To maintain smooth operation, network administrators are constantly striving to optimize data flow and access in a complex and changing environment. The aim of network monitoring for security management is to protect sensitive information on devices connected to a data network by controlling access points to that information, preventing, or denying different forms of malicious attacks. There are various network monitoring approaches to ensure network security. By identifying specific activities and performance metrics, these systems can generate results that help address a range of needs, including compliance requirements, internal security threats, and operational visibility [8].

3. Implementation

This paper proposes to implement a Signal Theory Model for Security Monitoring by provisioning locally two Ubuntu 20.04 Virtual Machines that can communicate with each other and are monitored, mainly for security reasons using the tool called CheckMK.

CheckMK is a comprehensive monitoring solution that provides a unified dashboard to monitor the performance and health of various systems and applications on the network. It allows the administrator to monitor everything from servers and network devices to cloud services and containers. It provides various alerting options, including email, SMS, and mobile push notifications, so the administrator can be notified immediately if any issues arise. Additionally, it offers support for multiple notification channels to ensure that alerts reach the right people at the right time as in the continuous monitoring world any second can make the difference between a fiasco and a good functionality of the system.

CheckMK is a powerful and reliable monitoring solution that is well-suited for businesses and organizations of all sizes. Its comprehensive monitoring capabilities, automatic discovery and

configuration, and flexible alerting options make it an ideal choice for monitoring any network and keeping any systems running smoothly [9].

Ubuntu Server 20.04 (Focal Fossa) is a server-focused operating system that is designed to provide a stable and secure platform for running a wide range of server applications. It is based on the Linux kernel version 5.4 and offers long-term support (LTS) for five years, making it a reliable choice for businesses and organizations. Also, it is very easy to jump from a LTS version to another and all the applications installed on the Ubuntu Server have dependencies that are compatible with the new LTS releases. It is a powerful and versatile server operating system that is well-suited for a wide range of server deployments, from small businesses to large-scale cloud environments. Its extensive software library, comprehensive security features, and long-term support make it a popular choice for businesses and organizations of all sizes [10].

3.1. System Architecture

To simulate this Signal Theory Model, I will use 2 Virtual Machines with Ubuntu 20.04 as Operating System. Both are using static IPs from my internal network so they can communicate with each other. One machine is named "Attacker" with the job of making brute force attacks using SSH towards the other virtual machine. The other one is named "Monitoring Site". On this virtual machine I have installed CheckMK to mainly monitor the security of the server, as shown in Figure 1.

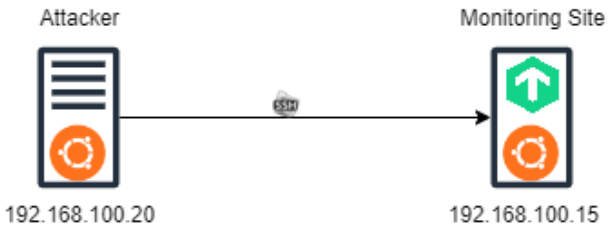

Fig. 1. System Architecture

The CheckMK instance on the "Monitoring Site" is used to count the number of SSH connection attempts that are made and to act depending on the that number. I will explain more when I define the flowchart of the system. The main role of the "Attacker" server is to test the security of the other server by making SSH brute force attacks, while the other server tries to deny the brute force attacks, notify the administrator about the intentions of that server and, if no action is taken by the administrator, to blacklist the IP of the attacker.

3.2. Flowchart

The Flowchart of the system is quite simple. The "Attacker" server will start to make SSH requests towards the "Monitoring Site" Server with the scope of brute force its way into the server. But the server is prepared to combat this kind of action by using the security utilities of CheckMK. So, if the attacker tries to connect via SSH as the root user 3 times in a minute, the server will notify the admin with a warning notification. If the attacker tries to connect via SSH 5 times in a minute, the server will automatically blacklist the IP of the attacker to protect itself from a brute force attack. This is a preventive method and it's mainly used for two reasons. The first reason is not to overwhelm the server with the SSH requests and cause a DDoS (Distributed Denial of Service) attack. The second reason implies human error: maybe the administrator was caught up in something else, did not see the notification involving the possible SSH brute force attack and as a result the server is down as it was overwhelmed by the SSH requests or, worse, the SSH brute force attack was a success.

As shown in Figure 2, the server tries to defend against brute-force attacks if the number of SSH requests it's too big and there is no administrator to take security measures. This practice is a

common one, as daily, if a server is opened to the internet, it will be bombed with SSH requests for a malicious connection on the server and if the resources of the server aren't good enough, the server might get down for some time. So, to save time and money, it is better to invest in a protection system that tries to prevent instead of treating the problem.

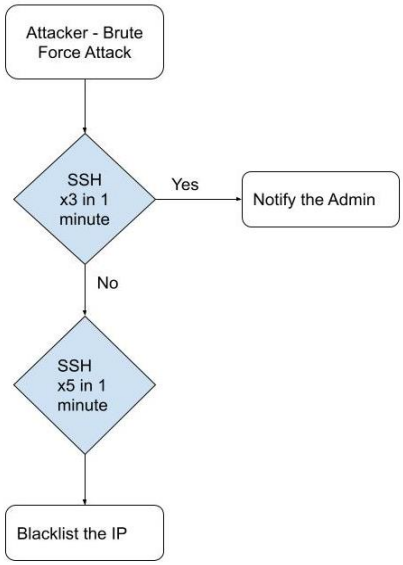

Fig. 2. Flowchart of the System

3.3. Deployment

The deployment of the Virtual Machines was made with the help of VirtualBox. VirtualBox is a powerful and feature-rich virtualization software that allows you to run multiple operating systems on a single computer. It is free and open source. It allows you to create virtual machines (VMs) that run these operating systems, each with its own set of virtual hardware components, such as virtual CPUs, memory, and disk space [11].

I used an Ubuntu 20.04.6 live server image to deploy both the machines. I chose this version because it's the Long-Term Support one, it's well-integrated with CheckMK and it is mature (first released in 2020) so it has a wide and detailed documentation that I can use in my research.

On the virtual machine named "Monitoring Site" I have installed CheckMK 2.1.0p25. This virtual machine has 2 GB of RAM, 2 CPUs and 20 GB of disk memory, and the machine running the "Attacker" server environment has 1 GB of RAM, 1 CPU and 20 GB of disk memory. The first one needs more resources because it runs the CheckMK tool and monitors both machines, just to be sure that the deployment was done correctly, and everything works properly.

3.4. Configuration

On both machines I had to configure the static IPs for the to communicate more easily between them. As an example of an IP configuration, let's look over the next figure, which shows the configuration made for the Monitoring Site.

As shown in Figure 3, I assigned the 192.168.100.15/24 IP to the "Monitoring Site" server with the gateway of 192.168.100.1, which is the default gateway of my internal network, and as a DNS server I've also used the default gateway. Now the machine is open to communicate with any other machine that has allocated an IP from my internal network. The same was done for the "Attacker" server, but with a different IP address (192.168.100.20/24).

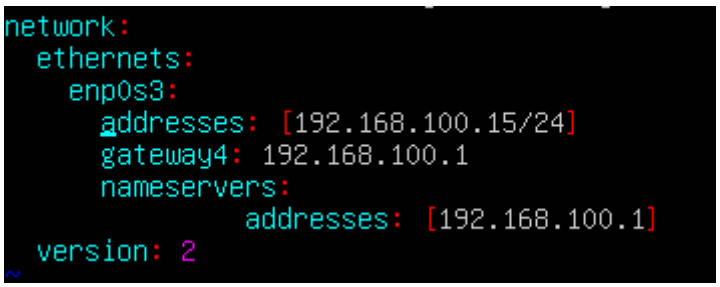

Fig. 3. IP Configuration for "Monitoring Site" Server

3.5. Defining the Rules

CheckMK is a rule orientated tool. To achieve the scope of this paper, I used a feature named Event Console. Event Console in CheckMK is a feature that allows users to view a chronological record of events that have occurred on a monitored system. These events can include system events, application events, and user-defined events. It provides a valuable tool for system administrators to monitor the health and status of their systems, troubleshoot issues, and take appropriate actions to prevent downtime and ensure system availability.

Using the Event Console, I have created two rules that will apply when the server notices that someone tries to SSH brute-force. For example, let's analyze the rule created for blocking the IP when it tries to SSH 5 times as the root user in a minute without having the password.

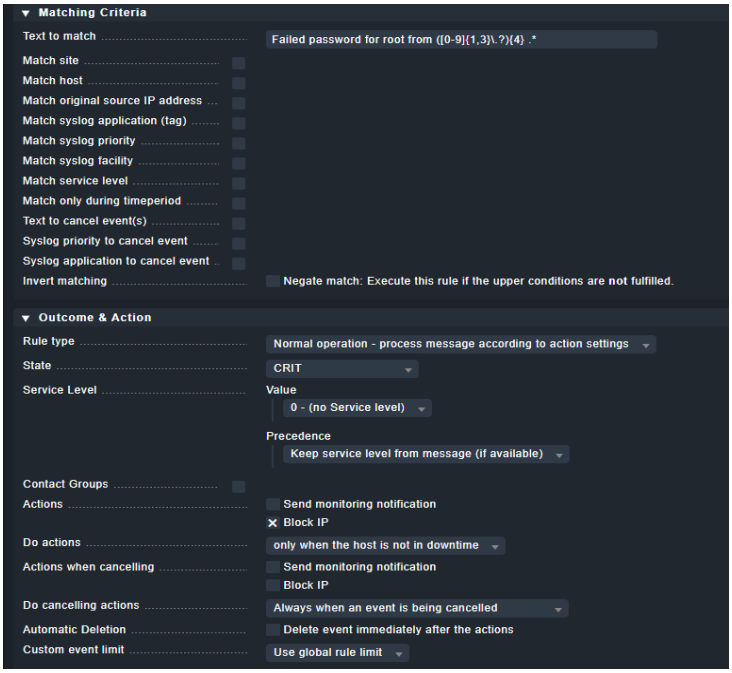

Fig. 4. Rule for Blocking the IP

I have defined a text to match which was taken from the journalctl logs of the "Monitoring Site" Server, I have added those logs for monitoring in CheckMK and I have used a Regex in order to define any possible IP that might try to SSH it's way into the server. I chose a State of CRIT, so the server administrator will know that an IP address was blocked when he reviews the Event dashboard of the server. As Actions I have created one that blocks the IP address using the Linux *iptables* command. The dashboard of the site and the action "Blocked IP" are not the scope of this paper, so I won't delve into explaining them more. I chose to use a Regex to define all the possible IP addresses, instead of specifying the IP address of the attacker (I know it in this case, because I have assigned it to the other server), but in a normal environment the administrator won't know the IP address of an attacker, so using a Regex that understands all the possible IP addresses is better and more proactive.

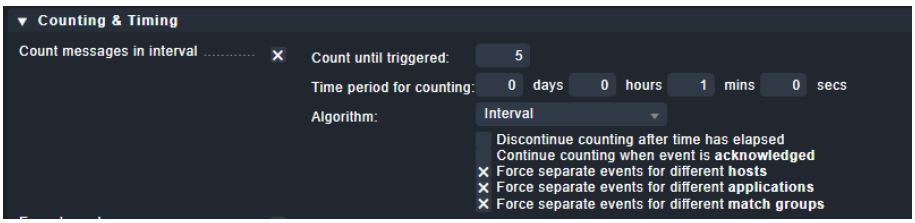

Fig. 5. Counting before blocking the IP

As I have specified earlier, CheckMK will count the SSH log in tries and if in a minute, there will be 3 failed log in tries, the server administrator will be notify. If the count reaches 5(as shown in figure 5) or more in minute, the server will try to protect itself alone by blocking the IP address from whom the SSH requestes came from. This is a proactive tactic that wants to treat the cause, not the effect of an event.

4. Tests and Results

To test the system, I am going to log in as root on the "Attacker" server and start to try to brute-force by SSH my way into the "Monitoring Site" Server and see what happens.

I am going to try to log as root 3 times from the "Attacker" Server to the "Monitoring Site" Server to see if I am going to be notified.

Fig. 6. SSH 3 times as root with the wrong password

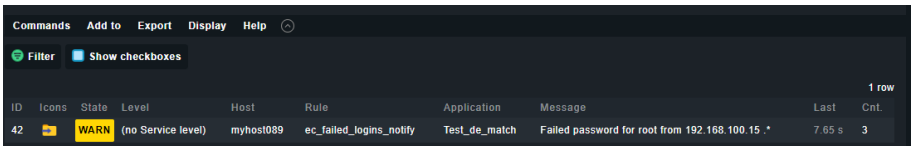

Fig. 7. The Admin was notified

Now I am going to SSH 5 times with the wrong password to see if the IP is going to be blocked:

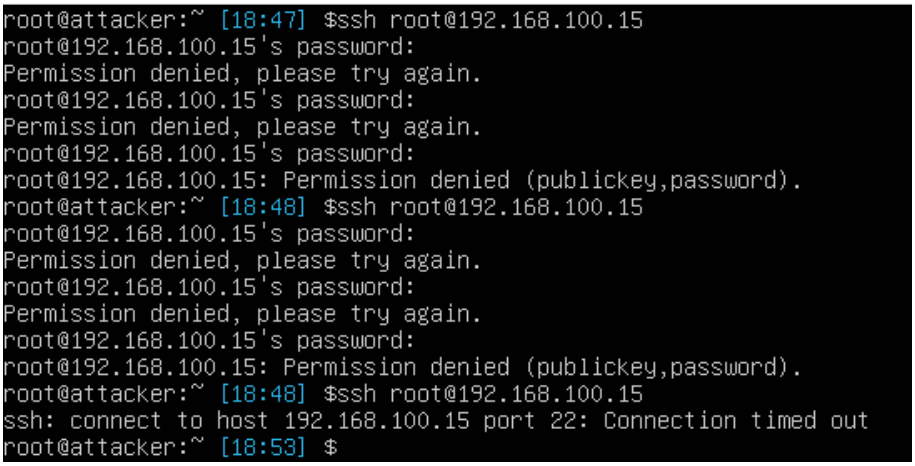

Fig. 8. "Monitoring Site" Server blocks the IP of the "Attacker" Server

As seen in Figure 8, the connection timed out, because the IP was blocked by the other server. The server is now protected against the brute force attacks coming from the "Attacker" server and the server's administrator was notified about the SSH attempts and the blocking of the IP.

5. Conclusion

In conclusion, the use of a Signal Theory model in security monitoring, combined with tools like CheckMK, can provide an effective means of detecting and responding to potential security threats. In this study, I have intended to demonstrate the practical application of this approach by setting up two servers: one with the role of the attacker, and the other one with the role of the defender. On the defender server, I have installed CheckMK to detect and respond promptly to SSH brute force attacks coming from the attacker server.

By applying signal theory concepts to network security monitoring, I was able to identify abnormal patterns of activity and trigger alerts based on the rules I have set. Specifically, I set up rules in CheckMK to detect repeated failed login attempts by an attacker and notify the administrator accordingly. If the attacker persisted in their attempts, the attacked server could automatically block their IP address to prevent further brute force SSH attempts and mitigate unauthorized access.

Overall, my experiment demonstrated the effectiveness of using a Signal Theory model with CheckMK for security monitoring. By leveraging the power of data analysis and pattern recognition, we were able to quickly identify potential threats and take proactive steps to mitigate them. As the threat landscape continues to evolve, this approach will become increasingly important for organizations looking to protect their valuable data and infrastructure.

References

[1]. O'Leary, Daniel E., A Signal Theory Model for Continuous Monitoring and Intelligence Systems (September 9, 2020). Available at SSRN: https://ssrn.com/abstract=3746001 or http://dx.doi.org/10.2139/ssrn.3746001J.

[2]. Phoung M. Cao et al, CAUDIT: Continuous Auditing of SSH Servers To Mitigate Brute-Force Attacks. Available at https://www.usenix.org/conference/nsdi19/presentation/cao.

[3]. Faust, Joshua, "Distributed Analysis of SSH Brute Force and Dictionary Based Attacks" (2018). Culminating Projects in Information Assurance. 56.

[4]. Spence, M., Job Market Signaling, The Quarterly Journal of Economics, Vol. 87, No. 3. (Aug., 1973), pp. 355-374.

[5]. Connelly, B., Certo, S., Ireland, R., Reutzel, C., (2011) "Signaling Theory: A Review and Assessment," Journal of Management, (37.2), January 2011, pp. 39-67.

[6]. Jeonghoon Park, "Network Log-Based SSH Brute-Force Attack Detection Model", Tech Science Press, 2021.

[7]. Daniels, Jeff, "Server virtualization architecture and implementation" XRDS: Crossroads, The ACM Magazine for Students Volume 16Issue 1September 2009 pp 8–12 https://doi.org/10.1145/1618588.1618592.

[8]. Ghafir, I., Prenosil, V., Svoboda, J., & Hammoudeh, M. (2016). A Survey on Network Security Monitoring Systems. 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW). doi:10.1109/w-ficloud.2016.30.

[9]. Official Documentation of CheckMK: https://docs.checkmk.com/latest/en/.

[10]. Official Documentation of Ubuntu: https://releases.ubuntu.com/.

[11]. Official Documentation of VirtualBox: https://www.virtualbox.org/wiki/Documentation.