|  |
| --- |
| Graphical user interface, text, application  Description automatically generated |
| **Lab Report** |
| COURSE: Cyber 262  Lab : WAZUH HIDS  Submitted BY: Jack Morgan  Date: 02/21/2022  Instructor: Hozza |
| *By submitting this report I certify that this is my work and was not copied from other sources. Any use of public materials is cited.* |

Table of Contents

[INTRODUCTION 3](#_Toc50470226)

[SCREEN CAPTURES 4](#_Toc50470227)

[REFLECTION 9](#_Toc50470228)

# INTRODUCTION

According to the lab directions, we will be utilziing the Security Onion, a free open soruce linux distribution used for intrustion detection, security monitoring and log management. Wazuh, the program we are using, is a free open source host-based intrusion detection (HIDS) system that can be utilized by Windows, Linux and Mac OS X users and is included in the Security Onion. Wazuh gives us the capabilities to block an attack, stop a malicious process or seclude a malware infected file to seperately deal with the issue. Hopefully by performing this lab, we will have a better understanding of intrusion detection systems and how to properly detect and stop a cyber attack on a device.

# SCREEN CAPTURES

***Figure (1.1):*** *Take a screen shot showing at least one of your ssh attempts being logged as above*

Graphical user interface, text, application, email

Description automatically generated

Figure : Screenshot displays attempts to ssh using my username, jvm6656

***Question (1.2):*** *Do some research on Wazuh agents, use the materials referenced in the lab intro or online. In 2-3 sentences, explain what tests the Agents performs to find malicious activity.*

Wazuh agents utilize a variety of checks in order to test whether or not specific activity is considered malicous or not. Some of these checks include; running process checks, checking hidden ports, checking ‘unusual’ files and permissions as well as, checking hidden files using system calls. A multitude of these methods are actually performed through the rootcheck command that we used during this lab as rootcheck has the capability to detect hidden ports!

<https://documentation.wazuh.com/current/user-manual/capabilities/anomalies-detection/how-it-works.html>

***Figure (1.3):*** *Your screenshot should provide detailed level evidecne of the attack “description” with “full\_log” showing the user attempting the logins, source address, port and protocol.*

Graphical user interface, application

Description automatically generated

Figure : Screenshot from the Kibana dashboard displays multiple attempts to ssh under jvm6656.

***Figure (2.1):*** *Show proof from your “tailf” command of at least 3 iterations of rootchecks.*

Graphical user interface, text, application

Description automatically generated

Figure : tailf command displays multiple iterations of rootchecks which can later be seen in the Kibana dashboard

***Figure (2.2):*** *Show proof of rootcheck activity detected in the Kibana Discover screen and the desription field.*

Graphical user interface, text, application, email, website

Description automatically generated

Figure : Screeshot shows live proof from the Kibana dashboard that tailf command was used.

***Question (2.3):*** *What does rootkit do? Explain what type of malicious behavior this module is detecting?*

The entire purpose of rootkit is to protect malware, kind of like an invisibility cloak to hide the code or malicious activity being performed. In this specific module, it is attempt to avoid being detected by the kibana rootchecks and system checks. Typically upon arrival, malware will self install rootkit in order to better protect itself from future detection or future vulnerability.

***Figure (3.1):*** *Prove that your malicious activity was detected in the raw log files per the previous steps*

Graphical user interface, text, application

Description automatically generated

Figure : Screenshot displays the sudo and touch commands used to create the bogus file containing potentially malicious code in the /etc location.

***Figure (3.2):*** *Provide evidence in a screenshot of your dashboard of the malicious activity in /etc. Be sure to scroll down in the dashboard to show commands that were run and who ran them. Use as many screenshots as needed\**

Graphical user interface

Description automatically generated

Figure : Displays the total number of alerts detected from the Kibana dashboard

Graphical user interface, text

Description automatically generated

Figure : Displays all commands utilized from the Kibana dashboard, tried to get the touch command to show, but could not get it after 3-4 attempts.

***Question (3.3):*** *Users who are escelation priviledges can be dangerous. In our lab we have escalated privile to root/super user for severl tasks. What command should we look for in the dashboard to monitor this potentially malicious activity? Try it and explain your results.*

Knowing that the sudo command overrides user priviledges and allows the user to run commands as the root, meaning that almost anything can be achieved as long as the user has acccess to the Sudo command. We should be looking for this command in the Kibana dashboard as well as commands such as touch and downloading esque commands where the root user attempts to download potentially malicious code/files.

# REFLECTION

After completing this lab, its much easier for me to understand how its so difficult yet so easy to detect malware. Using Kibana and similar systems, we can check for malicious actiivty through alerts and the use of user commands, however rootkit and other functions prove that it is much more difficult to detect malware than we think. It is important for cyber and more specifically, SOC analysts, to understand how to detect malware using Kibana and other systems in order to eliminate malware from infected systems. We also utilzied many important commands that can be considered “dangerous” such as grep and more importantly, the sudo command which gives access to root commands that normal users wouldn’t have access to.