Lab 05: SIEM/SOAR/SOC

Network Infrastructure Security (NIS)  
 2024-2025

© Chris Roets

© Henk Brouckxon

© Nico Declerck

## Introduction

# Lab concept

**Security information and event management** (**SIEM**) is a field within the field of computer security, where software products and services combine security information management (SIM) and security event management (SEM). They provide real-time analysis of security alerts generated by applications and network hardware. Vendors sell SIEM as software, as appliances, or as managed services; these products are also used to log security data and generate reports for compliance purposes.

SIEM functions:

* Collect log data from across the organization; leverage data to identify, categorize and analyze incidents and events.
* Provide visibility into malicious activity by pulling data from every corner of an environment, including all network applications and hardware.
* Aggregate all data into a single centralized platform.
* Leverage data to produce alerts, create reports and support incident response.
* SIEM allows organizations to analyze data from all network applications and hardware at any time. This can help organizations recognize potential security threats before they have a chance to disrupt business operations.

[**Security orchestration, automation and response (SOAR)**](https://www.crowdstrike.com/cybersecurity-101/security-orchestration-automation-and-response-soar/)is a collection of software programs developed to bolster an organization’s cybersecurity posture. A SOAR platform enables a security analyst team to monitor security data from a variety of sources, including security information and management systems and threat intelligence platforms.

SOAR functionalities:

* Collect threat information, automate routine responses and triage more complex threats, minimizing the need for human intervention.
* Unite three software solutions — threat and vulnerability management, security incident response and security operations automation — to strengthen and streamline the security posture.
* Leverage both manual and human intervention as well as machine learning (ML) technology to analyze incoming security data and prioritize incident response actions.

The overall goal of a SOAR platform is to collect threat-related data and automate threat responses. Your security team can increase efficiency and response time by using a SOAR platform.

[**Security operations center (SOC)**](https://www.crowdstrike.com/cybersecurity-101/security-orchestration-automation-and-response-soar/) **is** a central control center designed to protect organizations’ IT infrastructure. The SOC is responsible for monitoring security-relevant systems. It also analyses and qualifies threats, as well as initializing and supporting incident response actions. SOC analysts typically work together with cyber-security experts from other areas in coordinated processes using specialized tools.

A typical SOC includes:

* A central control center
* IT security analysts
* Processes
* IT security tools
* Various IT security services

In this Lab you will be part of the SOC central control center and configure a Wazuh manager to implement basic SIEM and SOAR solutions. For both platforms, we’ll start from a pre-installed virtual server. Because Alpine Linux does not trivially support logging to ELK, we’ll first focus on Wazuh.

For more information about the various concepts with a SOC you can go to

[A diagram of a company

Description automatically generated](https://www.protect.airbus.com/blog/cybersecurity-jargon-busting-mdr-soc-edr-xdr-soar-and-siem/) [MDR, SOC, EDR, XDR, SOAR and SIEM - what does it mean? (airbus.com)](https://www.protect.airbus.com/blog/cybersecurity-jargon-busting-mdr-soc-edr-xdr-soar-and-siem/)

## Wazuh architecture

### Elastic Stack

Wazuh is a Soar solution based upon the Elk stack architecture. This architecture (stack) consist of 3 parts :

* Data ingestion (syslog/agents/…): collects information from logfile and command execution
* Data storage (Elastic search) : stores the collected data into a database in a standard format
* Data visualization (Kibana) : web based user interface to view the data

A diagram of a company

Description automatically generated

* **There are different way to ingest data**
  + Integration: Integration typically involves using pre-built connectors or plugins provided by Elastic (the company behind the ELK Stack) or third-party developers. These integrations allow you to gather log data from various sources, systems, or applications and send them directly to Elasticsearch for indexing and analysis. Integrations often provide streamlined configurations and predefined mappings for specific log formats, making it easier to set up data ingestion. Examples of integrations include Beats, Logstash modules, and various Elasticsearch plugins.
  + Beats: Beats are lightweight data shippers developed by Elastic. They can be installed on servers, containers, or edge devices to collect different types of data, including logs, metrics, and network packets. Beats offer specialized modules for specific use cases, such as Filebeat for log files, Metric beat for system metrics, and Packet beat for network data.
  + Logstash modules: Logstash is a powerful data processing pipeline. Logstash modules are pre-packaged configurations designed to collect, parse, and transform logs from specific sources or applications. Modules provide a simplified setup experience and are available for many popular log sources, such as Apache, NGINX, and MySQL.
  + Agents: An agent is a software component installed directly on the source system or application that generates the logs. The agent's purpose is to collect, format, and send log data to Elasticsearch or another centralized log repository. Agents typically offer advanced features and flexibility, allowing for customized log collection and processing. They often provide additional functionalities like log rotation, buffering, compression, and secure transport.
* **Elasticsearch** is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with a HTTP web interface and schema-free JSON documents. Elasticsearch is developed in Java and is dual-licensed under the source-available Server Side Public License and the Elastic license, while other parts[]](https://en.wikipedia.org/wiki/Elasticsearch#cite_note-3) fall under the proprietary (*source-available*) *Elastic License*. According to the DB-Engines ranking, Elasticsearch is the most popular enterprise search engine.
* **Kibana** provides visualization capabilities on top of the content indexed on an Elasticsearch cluster. Users can create bar, line and scatter plots, or pie charts and maps on top of large volumes of data. Kibana also provides a presentation tool, referred to as Canvas, that allows users to create slide decks that pull live data directly from Elasticsearch.

## Wazuh

Wazuh is a customized Elk stack. This customization consists of :

* Wazuh agent as data injection engine. As each OS have different way of logging (syslog/journalctl/windows invent manager/…), each Wazuh-agent install on an endpoint will handle this complexity to ship standard log entries to the elastic search database engine. This customization can be found on /var/ossec/etc/ossec.conf. example entry are :

<localfile>

<log\_format>command</log\_format>

<command>journalctl -u ssh -f >> /var/log/test.log</command>

<frequency>60</frequency>

</localfile>

<localfile>

<log\_format>syslog</log\_format>

<location>/var/log/test.log</location>

</localfile>

Besides this, Wazuh-agent also has additional checks :

* + Rootcheck : check on file integrity. If the checksum of monitored file is changing a log is generated
  + Syscollector : generates an inventory of the system

These are also defined in /var/ossec/etc/ossec.conf

* Kibana has predefined graphs based upon the ruleset defined on the Wazuh-server in the directory /var/ossec/ruleset/rules. Custom rules can be added in /var/ossec/etc/rules. An example of a ruleset is :

<rule id="5760" level="5">

<if\_sid>5700,5716</if\_sid>

<match>Failed password|Failed keyboard|authentication error</match>

<description>sshd: authentication failed.</description>

<mitre>

<id>T1110.001</id>

<id>T1021.004</id>

</mitre>

<group>authentication\_failed,gdpr\_IV\_35.7.d,gdpr\_IV\_32.2,gpg13\_7.1,hipaa\_164.312.b,nist\_800\_53\_AU.14,nist\_800\_53\_AC.7,pci\_dss\_10.2.4,pci\_dss\_10.2.5,tsc\_CC6.1,tsc\_CC6.8,tsc\_CC7.2,tsc\_CC7.3,</group>

</rule>

Please note the existence of <match> with will trigger the rule. the rule id, will later be used in the third customization of Wazuh-server : incident response

* Wazuh-server also has the capability to respond to a ruleid that is triggered. This is also define in the configuration file :

<rule id="5763" level="10" frequency="8" timeframe="120" ignore="60">

<if\_matched\_sid>**5760<**/if\_matched\_sid>

<same\_source\_ip/>

<description>sshd: brute force trying to get access to the system. Authentication failed.</description>

<mitre>

<id>T1110</id>

</mitre> <group>authentication\_failures,gdpr\_IV\_35.7.d,gdpr\_IV\_32.2,hipaa\_164.312.b,nist\_800\_53\_SI.4,nist\_800\_53\_AU.14,nist\_800\_53\_AC.7,pci\_dss\_11.4,pci\_dss\_10.2.4,pci\_dss\_10.2.5,tsc\_CC6.1,tsc\_CC6.8,tsc\_CC7.2,tsc\_CC7.3,</group>

</rule>

<active-response>

<command>firewall-drop</command>

<location>local</location>

<rules\_id>5763</rules\_id>

<timeout>180</timeout>

</active-response>

Important to understand here is the <location> tag. This can be :

* + Local : execute on the node that triggered the event
  + Server : execute on the Wazuh-server itself
  + All : execute on all agents
  + Defined-agent : execute on agent\_id

The command is located in the directory /var/ossec/active-response/bin/ and is again specific to the agent.

# Learning goals

* Start a Wazuh server and navigate through the dashboard
* Add Wazuh agents on the firewall, the webserver and a workstation
* Explore the pre-defined security rules and even
* Create a custom security rule and event
* Implement an automated response for an event
* Implement and alert for an event.

# Practicalities and prerequisites

1. Wazuh is already installed but not enabled on the adminserver. To enable and start the different Wazuh-services, run the following commands:

systemctl enable wazuh-manager

systemctl enable wazuh-dashboard

systemctl enable wazuh-indexer

A screenshot of a computer program

Description automatically generated

systemctl start wazuh-manager

systemctl start wazuh-dashboard

systemctl start wazuh-indexer

A screenshot of a computer program

Description automatically generated

if you need to change the admin password :

/usr/share/wazuh-indexer/plugins/opensearch-security/tools/wazuh-passwords-tool.sh -u admin -p PWD@nis2024?

You can use the GUI (preferred) or access the VM via SSH.

user: wazuh-user

password: wazuh

SSH root user login has been deactivated; nevertheless, the wazuh-user retains sudo privileges. root privilege escalation can be achieved by executing the following command:

sudo -i

**I managed to enter the dashboard (from my adminstation)**

A screenshot of a computer

Description automatically generated

**However the way to access it via SSH does not seem to work but I will stick to what I have right now.**

A screenshot of a computer screen

Description automatically generated

1. Access the Wazuh dashboard:
   1. You need to be able to access the
   2. After starting the VM, the Wazuh dashboard can be accessed from the web interface by using the following credentials:

URL: https://<wazuh\_server\_ip>

user: admin

password: admin

**I updated the password since I could not login with the default credentials**

A screenshot of a computer screen

Description automatically generated

There are different way of using the browser, which one do you think of, the easiest is to use the browser on adminstation. What other ways are possible to access Wazuh from your own desktop ?

**By using the VPN connection we made in the previous labs (and not to forget to enable IP forwarding on Kali)**

A screenshot of a computer

Description automatically generated

**ACCESS SUCCESSFUL after a reboot and change of password**

**TIME TO UPDATE THE PACKAGES, BECAUSE I CAN NOT PROCEED WITH THE LAB DUE TO NEWER PACKAGE ON OPNSENSE**

**This is the current versions of wazuh-indexer/manager/dashboard and filebeat**

**A screenshot of a computer

Description automatically generated**

**Indexer updated**

**A black screen with colorful text

Description automatically generated**

**Manager updated**

**A black screen with white text

Description automatically generated**

**Dashboard updated**

**A screenshot of a computer screen

Description automatically generated**

`sudo systemctl daemon-reload`

**I enabled and started them the same way as was specified in the document, however I kept getting an error saying that “Wazuh dashboard server is not ready yet”**

**To solve it, I ran this command**

`curl -k -X DELETE -u admin:<password> [https://127.0.0.1:9200/.kibana\_2`](https://127.0.0.1:9200/.kibana_2%60)

**https://www.reddit.com/r/Wazuh/comments/17nlhed/wazuh\_dashboard\_server\_is\_not\_ready\_yet\_resolved/**

**A screenshot of a computer

Description automatically generated**

**Now all 3 agents are connected successfully.**

**And the packages on my adminserver are of version 4.9,2**

1. Configuration
   1. All components included in this virtual image are configured to work out-of-the-box, without the need to modify any settings. However, all components can be fully customized. These are the configuration files witch will be discussed later in the LAB.

## Adding a Wazuh agent in Alpine Linux

In order for Wazuh to be able to collect information from a networked device, the device should contain a Wazuh agent that forwards the required logs to the Wazuh server. We’ll first install this agent in Server01.   
**remark**: the latest version of the wazuh agent is not (yet) supported on Alpine Linux. Therefore, we will use the previous version.

1. Add the Wazuh repository in alpine:

wget -O /etc/apk/keys/alpine-devel@wazuh.com-633d7457.rsa.pub https://packages.wazuh.com/key/alpine-devel%40wazuh.com-633d7457.rsa.pub

A screenshot of a computer screen

Description automatically generated

echo "https://packages.wazuh.com/4.x/alpine/v3.12/main" >> /etc/apk/repositories

apk update

**The command given did not work (for some reason, so I did it like this with tee**

`echo "https://packages.wazuh.com/4.x/alpine/v3.12/main" | doas tee -a /etc/apk/repositories`

A screen shot of a computer

Description automatically generated

1. Install the Wazuh agent

apk add wazuh-agent

A screen shot of a computer

Description automatically generated

export WAZUH\_MANAGER="<Wazuh server IP>" && sed -i "s|MANAGER\_IP|$WAZUH\_MANAGER|g" /var/ossec/etc/ossec.conf

A screen shot of a computer

Description automatically generated

**I SET THE WAZUH MANAGER TO** `10.0.0.67`

1. start the agent manually (may be required after each restart of the VM)

/var/ossec/bin/wazuh-control start

A screen shot of a computer

Description automatically generated

1. Disable updates of the Wazuh-agent. This is needed to prevent incompatibilities. The wazuh-manager will work with agents that are at the same software version or previous. An agents with a newer version may not work with an older version of the Wazuh-manager

sed -i "s|^https://packages.wazuh.com|#https://packages.wazuh.com|g" /etc/apk/repositories

A screen shot of a computer

Description automatically generated

**UPDATES DISABLED**

**NOTE FOR SELF – THIS COMMAND JUST COMMENTED OUT THE LINE WITH WAZUH PACKAGES.**

1. Now check if the Server01 VM was added in the web interface of Wazuh, and if logs from this VM are coming in (e.g try to log in to your Server01 VM with the wrong username or password). Test what other messages are sent to the Wazuh server by Server01.

**The agent has been successfully added to the adminserver dashboard.**

A screenshot of a computer

Description automatically generated

**I can not really see a lot of logs coming in. I tried to google how to watch them, and tried to go to agents and see alerts, and other things (in the GUI of course), however it does not seem to work for now, and I do not know why exactly. But I will leave it as is for now.**

**Well actually I can see “Alerts” for wazuh or something like that. Is this the logs though? Do not think so.**

A screenshot of a computer

Description automatically generated

**I managed to open the same tab, but just open the bigger view, and here I can see a logon fail! Then it certainly works**

A screenshot of a computer code

Description automatically generated

1. Make sure that the Wazuh agents is started when the server01 is booting

create following script in /etc/init.d/wazuh-agent :

#!/sbin/openrc-run

start() {

ebegin "Starting ${SVCNAME}"

/var/ossec/bin/wazuh-control start

eend $?

}

stop() {

ebegin "Stopping ${SVCNAME}"

/var/ossec/bin/wazuh-control stop

eend $?

}

A screenshot of a computer program

Description automatically generated

And update the system manager:

rc-update add wazuh-agent default

**It says not executable, hence need to add the right to execute.**

`doas chmod +x /etc/init.d/wazuh-agent`

A screen shot of a computer

Description automatically generated

**And now it can be added**

A screen shot of a computer

Description automatically generated

1. Optionally, add Wazuh agents to some of the other Alpine VMs to monitor them

## Adding a Wazuh agent to OPNSense

A screenshot of a computer

Description automatically generated

$ pkg search wazuh-agent

$ pkg install wazuh-agent

**VERSION INSTALLED – 4.9.2**

$ cp /etc/localtime /var/ossec/etc

change in /var/ossec/etc/ossec.conf

<ossec\_config>

<client>

<server>

<address><Wazuh server IP> </address>

</server>

<config-profile>debian, debian8</config-profile>

<crypto\_method>aes</crypto\_method>

</client>

A computer screen with white text

Description automatically generated

$ service wazuh-agent enable

$ service wazuh-agent start

A screen shot of a computer

Description automatically generated

Add following to /var/ossec/etc/ossec.conf :

<localfile>

<log\_format>syslog</log\_format>

<location>/var/log/audit/latest.log</location>

</localfile>

A screen shot of a computer program

Description automatically generated

$ service wazuh-agent restart

A screenshot of a computer

Description automatically generated

You may get the agent registration complete but the exchange information not working.

Check the ossec.log for the error message.

This may be because the OPNSense distro has per default udp as transport protocol, while the Wazuh agent has tcp as transport protocol. How would you change this ?

<server>

<address>10.0.0.195</address>

<port>1514</port>

<protocol>**tcp**</protocol>

</server>

**The whole story here is that the package manager for FreeBSD updated its wazuh-agent package to 4.9.2 instead of 4.7.5 which was previous and installed by most student (yes I asked a lot of people).**

**Hence, I was trying for a lot of hours how to solve this issue.**

Steps I tried to fix the issues – download wazuh 4.7.5 from Github and selected to install it as type agent and nothing more.

Failed, because I had to install gmake (pkg install gmake)

Failled, because I need cmake (<https://cmake.org/download/>, tar -xvzf, ~~./configure~~, ~~gmake,~~  make install)

Then after cmake is compiled with C, its time to make the rest, and back to `./install.sh`

Got another issue

A computer screen shot of white text

Description automatically generated

`pkg search + pkg install openssl`

I Could not solve the issue. The damn issue that the version installed in adminserver is 4.8.0 and freebsd installs version 4.9.2 is killing me for more than 5 hours already…

The only idea I get right now is to manually update wazuh components for adminserver and that should solve half of my issues.

AFTER INSTALLING LATEST VERSION OF WAZUH COMPONENTS ON ADMINSERVER

A screen shot of a computer code

Description automatically generated

I managed to connect all of them.

A computer screen shot of white text

Description automatically generated

A black background with white text

Description automatically generated

## Some commands to remember

List the agents from the command line :

$ /var/ossec/bin/manage\_agents -l

Remove an agents from the command line :

$ /var/ossec/bin/manage\_agents -r …

And remove client keys out of /var/ossec/etc/client.keys

To reset db to factory defaults :

$ /var/ossec/bin/rbac\_control factory-reset

## Adding a Wazuh agent to Kali-Linux

# curl -o wazuh-agent-4.7.3-1.x86\_64.rpm https://packages.wazuh.com/4.x/yum/wazuh-agent-4.7.3-1.x86\_64.rpm

# sudo WAZUH\_MANAGER==<Wazuh server IP> WAZUH\_AGENT\_NAME='kali' rpm -ihv wazuh-agent-4.7.3-1.x86\_64.rpm

**This Lab looks very poorly prepared unfortunately, however I think that there is an error of using .rpm packages on a Debian based system.**

**`**wget <https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.3-1_amd64.deb>`

**`**sudo dpkg -i wazuh-agent\_4.7.3-1\_amd64.deb`

A screen shot of a computer

Description automatically generated

# systemctl daemon-reload

# systemctl enable wazuh-agent

# systemctl start wazuh-agent

A screenshot of a computer

Description automatically generated

**Finally something worked. However at this moment I will go back to OPNsense and install an older version of wazuh-agent because the current does not work.**

## Preparing Kali Linux to attack with Hydra hacking tool

In the later exercise, we will use hydra to attack our corporate network, both from the inside and the outside. For the sake of the exercise we configure two network interfaces on the Kali Linux. This is of course not done on a corporate network. Check [hydra | Kali Linux Tools](https://www.kali.org/tools/hydra/) for the usage of this tool

1. Install hydra on Adminstation and test it using localhost as host and mickey as username. Add a correct password to the password list and see what happens

A white text with black text

Description automatically generated

Why user mickey? Our users are different in all cases, so I think that was to see the various behaviour of the tool. In this case, it failed for mickey user with correct password, and succeeded for my user with correct password.

## Lab assignments

### Upload the startup page of wazuh

1. Upload a screenshot of the wazuh-agents that are registered in your server: Wazuh-> agents

A screenshot of a computer

Description automatically generated

Here is the dashboard showing my various agents. In this document I also explained all the issues that I had during this lab. hence the versions might vary a bit.

### Invalid ssh login

1. On the Wazuh server, try to login as a non-existing account to localhost and take a screenshot of the events generated in the Wazuh server. Upload this screenshot in Leho

A screenshot of a computer

Description automatically generated

**The instructions are poor, and hence I can not understand what exactly is needed from me, but what I tried – connect to SSH with a non-existent user from my adminstation for my adminserver, and you can clearly see the 2 logs that state authentication failed, and non-existent user.**

### ssh brute force attack

[Blocking SSH brute-force attack with active response - Use cases (wazuh.com)](https://documentation.wazuh.com/current/user-manual/capabilities/active-response/ar-use-cases/blocking-ssh-brute-force.html)

In this exercise you will be part of the red team. You were able to break into the internal network. You did this by adding your Kali to the LAN segment of the management LAN. You will attach the Wazuh-manager but you will see that out of the box, this is prevented by Wazuh. How would you do this.

Create a passwd\_list.txt file with at least 10 lines.

Ping the Wazuh server, it should work

Now run : hydra -t 4 -l “exiting account” -P passwd\_list.txt <ip-addr-off-wazuh-manager> ssh

Make sure the “exiting account” is an account that exists on the wazuh-manager (adminserver)

Ping the Wazuh server again, It will fail

What happened ?

A screenshot of a computer

Description automatically generated

**Passwords created**

A screen shot of a computer code

Description automatically generated

Ping also works great.

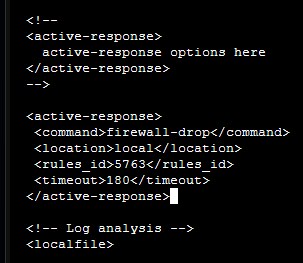
A close up of text

Description automatically generated

I tried running the command, however the wazuh server can be still pinged. The user surely exists, but the right password is not present there.

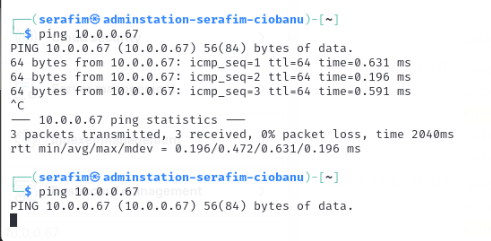
No matter how hard I try, the server is still accessible. I do not understand the possible issue here, however I think it was supposed to block the incoming requests after a certain amount of tries.

I went to read the documentation, and decided to try and add this rule here.



`sudo systemctl restart wazuh-manager`

And here the ping finally stops



So I successfully implemented a stop for SSH attacks, with a timeout of 3 minutes.

Even the page for Wazuh would not load for my Kali machine.

### Invalid http access

For apache this is configured out of the box, for lighttp you need to monitor the logfile :

<localfile>

<log\_format>syslog</log\_format>

<location>/var/log/lighttpd/access.log</location>

</localfile>

/var/ossec/logs # grep access \*

ossec.log:2024/07/26 07:11:19 wazuh-logcollector: INFO: (1950): Analyzing file: '/var/log/lighttpd/access.log'.

/var/ossec/logs #

On your host open a web browser and go to http://<public-ip-of opnsense>/test, check the error and post the event generated in the Wazuh event gui.

**I can not understand this question? What does lighttp has to do with OPNsense and what exactly am I supposed to do? Which server do I have to configure? Am I supposed to post a screenshot of getting an alert of 404 not found?**

A screenshot of a computer

Description automatically generated

I tried to make the lighttpd service to log the access logs, or error logs, however I do not understand the configuration, however I understand that in the end I am supposed to get the 404 not found alert onto my wazuh-manager, but that will take even more of my time that I spent on this lab.

### Block massive web server crawling

You are now part of the blue team and need to defend your web server. A group of cyberterrorists have prepared an attach. For this, they installed a trojan horse on 10.000 enduser pc’s. You are informed that if you don’t pay them a beer, they will release all of those pc’s to attack your web-server.

The DPO is thinking they are bluffing, while the CIO is afraid that if the web-server goes down, the company will stall its operations. The CISO claims that once the attack begins, he needs 5 minutes to counter the victim through the firewall.

Blocking one ip address will not work here, so if the attack happens, you need to bring down the webserver for 5 minutes to allow proper actions to be taken.

For one of the major outages in the Belgium news it was explained, there is no data leak but the web server has been taken offline. Let’s now explore how we can automate this.

Different things need to be done :

* + On the Wazuh manager, create an additional rule that is triggered when ruleid 31101 is triggered more than 5 times
  + On the Wazuh agent, create a script that brings down the web-server on server01 for 5 minutes when the above rule is triggered
  + You need to define the command suspend\_apache.sh so that Wazuh manager understand your previous script
  + You need to define this command as the response to the rules you create in the first bullet.

Now refresh the web page in the browser more than 5 times. After 5 times, you will not get the “not found” error anymore, but a timeout.

Isnt it lighttpd that we use on server01?

A black background with white text and symbols

Description automatically generated

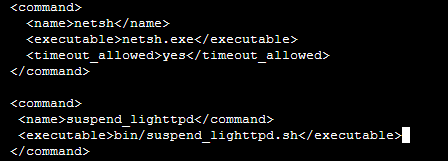
A screenshot of a browser

Description automatically generated

Just to make sure

A computer screen shot of white text

Description automatically generated



A computer screen shot of a black screen

Description automatically generated

AAAAAND none of it worked. I will not proceed with this.

### Optional : monitor an external windows system

For this, you need to configure OPNSense to forward port 1514 1515 and 55000 to the Wazuh server and use the public ip address of OPNSense in the Wazuh agent installation.

You will now see a lot of events coming in. Feel free to explore those further and secure this system towards best practices.

Question : what would take more time, securing your windows or installing Linux on it ?