Lab 4: IDS/IPS Systems

Network Infrastructure Security (CSP)  
 2023-2024

© Henk Brouckon , Chris Roets, Nico Declerck

An Intrusion Detection System (IDS) is like a security guard for your computer network. It keeps an eye on all the data moving in and out of the network to spot any suspicious activity that might indicate a cyber attack.

**How Does an IDS Work?**

1. **Monitoring Traffic**: Think of the IDS as a surveillance camera. It constantly watches the network traffic, looking for anything unusual or potentially harmful.
2. **Detecting Threats**: The IDS uses two main methods to detect threats:
   * **Signature-Based Detection**: This method works like an antivirus program. It has a database of known attack patterns (signatures) and compares incoming data against these patterns. If it finds a match, it raises an alarm.
   * **Anomaly-Based Detection**: This method is more like a detective. It knows what normal network behavior looks like and flags anything that deviates from this norm. For example, if there's a sudden spike in data traffic at 3 AM, the IDS might suspect something fishy.
3. **Alerting Security Teams**: When the IDS detects something suspicious, it sends an alert to the security team. This is like the security guard calling the police when they see something wrong.
4. **Analyzing Activity**: The IDS can also analyze the data it collects to understand the nature of the threat. This helps in identifying whether it's a false alarm or a real attack.

**Types of IDS**

* **Network-Based IDS (NIDS)**: Monitors the entire network. It's like having a security guard watching the whole building.
* **Host-Based IDS (HIDS)**: Monitors individual devices. It's like having a security guard for each room in the building.

**Why is IDS Important?**

An IDS is crucial because it helps detect and respond to potential security breaches before they can cause significant damage. It's a proactive measure to keep the network safe from hackers and other cyber threats.

## Enabling the Intrusion Detection System

In this lab we will use the Intrusion Detection (and Intrusion Prevention) System Suricata, which by default comes pre-installed with OPNsense. To actually use Suricata however, we will first need to activate it in the OPNsense web interface.

1. Open the configuration website of your OPNsense firewall, and go to the Services 🡺 Intrusion Detection 🡺 Administration page. Turn on the following checkboxes:
   * + Enabled
     + Promiscuous mode
     + Enable syslog alerts
     + Enable eve syslog output

A screenshot of a computer

Description automatically generated

1. On the same page, under “interfaces”, select both the LAN and WAN interfaces as interfaces that will be monitored by the IDS system. In general, OPNsense and Suricata recommend applying the IDS on physical interfaces like LAN, and not on virtual (VLAN) interfaces to improve performance and compatibility, especially when the IPS mode is enabled (which we will do later).

A screenshot of a computer

Description automatically generated

## A first test - Detecting facebook App usage

Even though we’ve now enabled Suricata, it will not do anything yet because we have not yet defined any active rules (or signatures) for threat detection. By default, OPNsense comes with a number of rulesets, that vary from the detection of traffic aimed at known C2 servers, to the detection of malicious and not-so-malicious apps based on e.g. IP addresses and known SSL signatures [1]. One of these not-so-malicious apps is for example facebook. We’ll now activate a number of rules that will perform this detection.

1. The IDS “download” page contains the available rulesets for detection. On this page, enable and download the ruleset “OPNsense-App-Detect/social-networking”. This should add a number of rules in the IDS “rules” page. Search and enable all the rules that are related to Facebook. What is the default action that is taken for these rules?

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

As far as I can see, the default action is “alert”

I forgot to enable them

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Now test the IDS system by going to facebook.com in the browser on your Adminstation VM. Can you see this in the logs of the IDS system? What rule numbers (sid) are triggered, and what part of the communication did they trigger on?

**A white background with many small squares

Description automatically generated with medium confidence**

**This is only one request towards the website.**

**Sid’s triggered: 51000009, 51000011**

**Part of communication (I do not really understand what is meant) but it looks like it refers to DNS requests and handshakes, or overall certificate gathering, like for HTTPS, because I see TLS related logs. It is important that the triggers are only seen in IDS “Alerts” page and not Intrusion Detection > Log File**

## Adding a (simple) custom Suricata rule - ftp connections to test.rebex.net

1. Now go to the IDS “User Defined” page, and add a new custom rule that detects traffic with test.rebex.net as destination. For now keep “Alert” as action, and add a suitable description for the rule. Add your first name at the start of the description field.  
   Be careful how this destination needs to be defined in the rule!

First thing I think is I need the IP address of the website, because there is no hostname field  
A screenshot of a computer screen

Description automatically generated

**Dig** and **nslookup** tell me **194.108.117.16**

A screenshot of a computer

Description automatically generated

1. Make an ftp connection from one of your VMs to test.rebex.net and check if the rule triggered an alert.

A screenshot of a computer

Description automatically generated

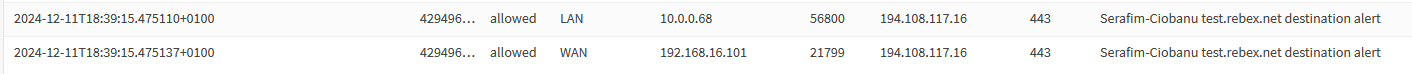
I did it from my **adminstation**, and it worked.

1. Now change the rule’s action to “Drop” and test the ftp connection to test.rebex.net again. Did you notice any difference? Why (not)? Also check the action that was performed in the alerts log.

A screenshot of a computer

Description automatically generated

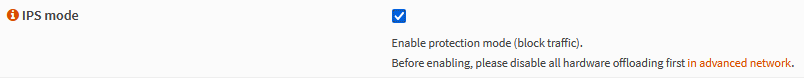
It does not look like anything changed, because I can still access the website (it loads), and that is probably because there is not enabled a module or something that would have rights to drop this kind of packets? The actions in “Alerts” say “allowed”



I even waited a bit for my firewall to apply every single setting but it still seems to work.

## Enabling the Intrusion Prevention System

1. In the IDS administration page, enable the “IPS mode” checkbox. Take care to read the accompanying “help” information, and make sure that you comply with the prerequisites for IPS mode!



A screenshot of a computer

Description automatically generated

As far as I understand, these things should be “ticked” which means that I disabled offloading. This was default when I checked it.

1. Once more, test the ftp connection to test.rebex.net . What happens now? Can you verify this in the alerts log? Insert here a screenshot of the alerts log that shows what action was taken by Suricata, and also showing your name in the description field of the triggered alert.

A screenshot of a computer

Description automatically generated

Now the packets are being blocked, and the whole connection hangs

1. In the IDS administration page, disable the “IPS mode” checkbox, to ensure that the IPS will not interfere with any of the following classes.

A screenshot of a computer

Description automatically generated

Disabled now

## Optional – A more complicated example of custom Suricata rules

Up to now, the detection rules we encountered were relatively simple, e.g. based on the destination IP address or the signatures of the application. More complex rules that combine parameters from different OSI layers and even time-based parameters (e.g. number of sent packets in a given time) are however also possible. Here for example, we’ll look at a set of custom rules that were created to detect potential NMAP scans in the network.

1. Add the custom rules that are described at [2] into the /usr/local/etc/suricata/rules/local.rules files on the OPNsense VM, and reload the Suricata rulesets in the OPNsense web GUI. Enable the new custom rules.

I found the rules

Opened them in raw format

Installed vim and wget onto OPNsense, and then did a **wget** for the raw file into the directory

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

I clicked “Apply” to import the rules, and then they appeared, all enabled by default A screenshot of a computer

Description automatically generated

1. Now perform an NMAP scan from workstation01 to server01. Can the IDS detect this?

A computer screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

Yes, it can detect it.

1. Finally perform an NMAP scan from workstation01 to workstation02. Can the IDS detect this? Why (not)?

A screen shot of a computer

Description automatically generated

It does not seem to detect it, because I do not see alerts about it.

Why? Probably because there is no traffic going from lets say one VLAN to another, because we said to detect traffic on LAN and WAN. But in this case its like packets do not interfere with those interfaces, hence we do not get the trigger.

## References

|  |  |
| --- | --- |
| [1] | "Suricata Rulesets in OPNsense," [Online]. Available: https://docs.opnsense.org/manual/ips.html#available-rulesets. |
| [2] | "Suricata rules for detecting nmap scans," [Online]. Available: https://github.com/aleksibovellan/opnsense-suricata-nmaps. |