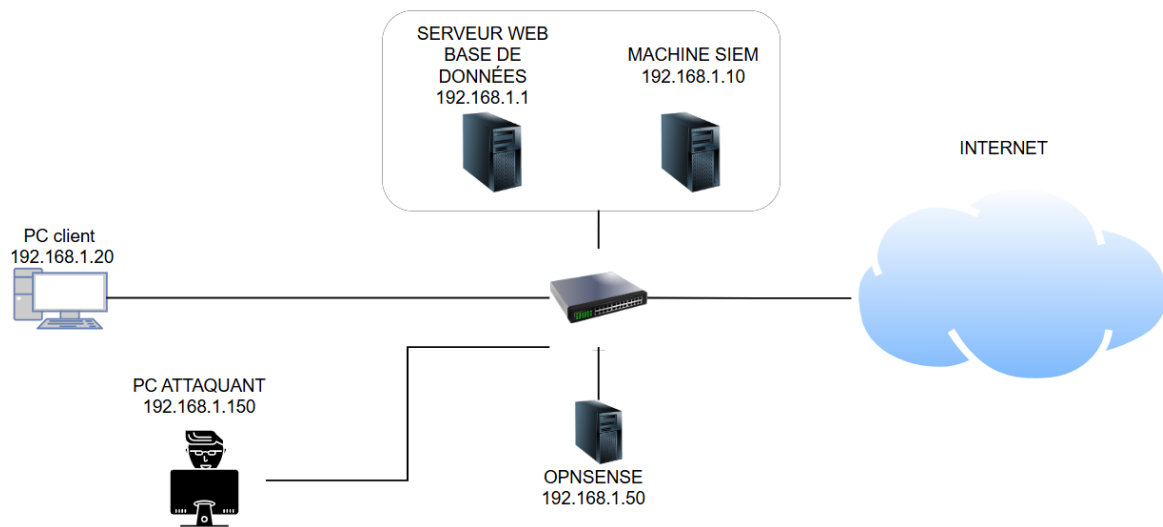


Elmedin SULTANOVIC
Gabin BECU

Dashboard Kibana

Dans le cadre de ce projet, nous avons mis en place une infrastructure de supervision de la sécurité complète s'appuyant sur un SIEM (ELK Stack) et un IDS/IPS (Suricata). Cette phase finale vise à valider la capacité du système à détecter et corrélérer différents types d'activités malveillantes simulées à partir d'une machine attaquante. Les tests réalisés couvrent des scénarios d'intrusion classiques : scans réseau, tentatives de brute-force...

Petite mise en contexte rapide de notre réseaux grâce au schéma :



Dans ce livrable nous allons vous présenter, les résultats des tests d'attaques menées

Pour détecter ces attaques on s'est reposé sur les alertes OPNsense nous avons d'abord téléchargé les règles de bases open source

Description	Last updated	Enabled	Edit
<input type="checkbox"/> abuse.ch/Feodo Tracker	2025/12/04 10:21	✓	Edit
<input type="checkbox"/> abuse.ch/SSL Fingerprint Blacklist	not installed	✗	Edit
<input type="checkbox"/> abuse.ch/SSL IP Blacklist	not installed	✗	Edit
<input type="checkbox"/> abuse.ch/ThreatFox	2025/12/04 10:21	✓	Edit
<input type="checkbox"/> abuse.ch/URLhaus	not installed	✗	Edit
<input type="checkbox"/> ET open/3coresec	not installed	✗	Edit
<input type="checkbox"/> ET open/botcc	not installed	✗	Edit
<input type="checkbox"/> ET open/botcc.portgrouped	not installed	✗	Edit
<input type="checkbox"/> ET open/clarmy	not installed	✗	Edit
<input type="checkbox"/> ET open/compromised	not installed	✗	Edit
<input type="checkbox"/> ET open/drop	not installed	✗	Edit
<input type="checkbox"/> ET open/dshield	2025/12/04 9:10	✓	Edit
<input type="checkbox"/> ET open/emerging-activex	2025/12/04 9:10	✓	Edit
<input type="checkbox"/> ET open/emerging-adware_pup	not installed	✗	Edit
<input type="checkbox"/> ET open/emerging-attack_response	2025/12/04 9:10	✓	Edit
<input type="checkbox"/> ET open/emerging-chat	not installed	✗	Edit

Puis par la suite nous avons créé nos propres règles afin de pouvoir maîtriser parfaitement la sévérité des règles et faciliter nos tests.

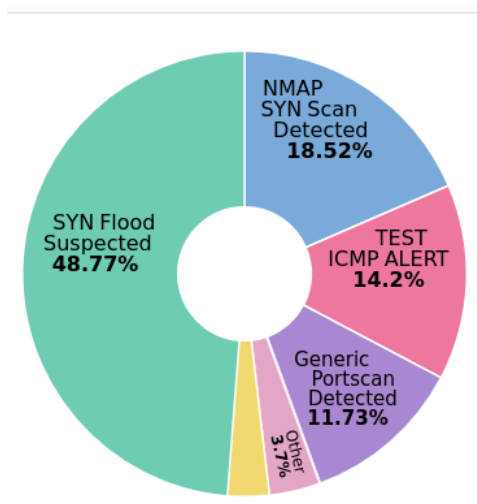
```
GNU nano 8.7 /usr/local/etc/suricata/rules/custom.rules
#détection ICMP/Ping
alert icmp any any -> any any (msg:"icmp alert"; sid:10000001; rev:2;)
alert icmp any any -> any any (type:8; icmp_seq:>10; msg:"ICMP Flood Suspected";sid:10000002; rev:1;)
```

Pour commencer nous avons utilisé une alerte la plus simple possible qui détectait les ping et qui envoyait une alerte. Cette alerte on pouvait la visualiser de deux manières la première directement dans l'interface d'OPNsense dans l'onglet « Alerts »

Settings Download Rules User defined Alerts Schedule									
Search [] [] 2025/12/04 15:12 All [] []									
Timestamp	SID	Action	Interface	Source	Port	Destination	Port	Alert	Info
2025-12-04T16:13:11.789695+0100	10000001	allowed	LAN	fe80:0000:0000:0000:...		ff02:0000:0000:0000:...		TEST ICMP ALERT	[]
2025-12-04T16:11:08.319609+0100	10000001	allowed	WAN	fe80:0000:0000:0000:...		ff02:0000:0000:0000:...		TEST ICMP ALERT	[]
2025-12-04T16:08:17.053243+0100	10000001	allowed	LAN	fe80:0000:0000:0000:...		ff02:0000:0000:0000:...		icmp alert	[]
2025-12-04T16:08:17.053243+0100	10000001	allowed	LAN	fe80:0000:0000:0000:...		ff02:0000:0000:0000:...		TEST ICMP ALERT	[]
2025-12-04T16:07:10.677227+0100	20000016	allowed	LAN	194.57.86.193	53	192.168.1.1	40944	UDP Port Scan Detected	[]
2025-12-04T16:05:09.248430+0100	20000004	allowed	LAN	192.168.1.180	55527	192.168.1.50	80	SYN Flood Suspected	[]
2025-12-04T16:05:08.835932+0100	20000001	allowed	LAN	192.168.1.180	50538	192.168.1.50	80	NMAP SYN Scan Detected	[]
2025-12-04T16:05:07.949106+0100	20000004	allowed	LAN	192.168.1.180	56128	192.168.1.50	80	SYN Flood Suspected	[]
2025-12-04T16:05:06.950471+0100	20000004	allowed	LAN	192.168.1.180	60382	192.168.1.50	80	SYN Flood Suspected	[]
2025-12-04T16:05:05.946460+0100	20000004	allowed	LAN	192.168.1.180	64495	192.168.1.50	80	SYN Flood Suspected	[]
2025-12-04T16:05:05.836144+0100	20000001	allowed	LAN	192.168.1.180	63141	192.168.1.50	80	NMAP SYN Scan Detected	[]
2025-12-04T16:05:04.945129+0100	20000004	allowed	LAN	192.168.1.180	52337	192.168.1.50	80	SYN Flood Suspected	[]
2025-12-04T16:05:03.739254+0100	20000004	allowed	LAN	192.168.1.180	53996	192.168.1.50	80	SYN Flood Suspected	[]
2025-12-04T16:05:03.724789+0100	20000004	allowed	LAN	192.168.1.180	53796	192.168.1.50	80	SYN Flood Suspected	[]
2025-12-04T16:05:02.724211+0100	20000004	allowed	LAN	192.168.1.180	58058	192.168.1.50	80	SYN Flood Suspected	[]
2025-12-04T16:05:02.724211+0100	20000004	allowed	LAN	192.168.1.180	58058	192.168.1.50	80	SYN Flood Suspected	[]


Et la deuxième dans Elastic qui facilite sous forme d'une log qu'on pouvait ensuite traiter avec un Dashboard par exemple

@timestamp [] [] event.category [] [] network.protocol [] [] network.transport [] [] source.ip [] [] rule.name [] [] agent.hostname [] [] service.type [] []							
Dec 4, 2025 @ 16:17:56.974	[network, intrusion_detection]	-	ipv6-icmp	fe80::2	TEST ICMP ALERT	OPNsense.internal	suricata
Dec 4, 2025 @ 16:13:58.291	[network, intrusion_detection]	-	tcp	192.168.1.8	SSH Connection Attempt Detected - ANY SOURCE	OPNsense.internal	suricata



Après un résultat concluant on a utilisé les mêmes protocoles de test pour plusieurs attaques :

Le SSH brutforce on a donc crée un script qui envoyait de nombreuses connexions.
Et on recevait bien une alerte :

2025-12-04T14:35:24.794069+0100	20000005	allowed	LAN	192.168.1.8	53800	192.168.1.50	22	SSH Brute Force Attempt	
---------------------------------	----------	---------	-----	-------------	-------	--------------	----	-------------------------	---

On a ensuite voulu détecter les scans NMAP.
on a donc créé une règle

```
alert tcp any any -> $HOME_NET any (flags:S; threshold:type both, track by_src, count 12, seconds 3; msg:"NMAP SYN Scan Detected"; sid:20000001; rev:1;)
```

et fait un scan nmap qui envoyait bien une alerte

2025-12-04T16:05:05.836144+0100	20000001	allowed	LAN	192.168.1.180	63141	192.168.1.50	80	NMAP SYN Scan Detected	
---------------------------------	----------	---------	-----	---------------	-------	--------------	----	------------------------	--


On avait aussi un cas de scan de port UDP qu'on a su détecter

```
alert udp any any -> $HOME_NET any (threshold:type both, track by_src, count 25, seconds 5; msg:"UDP Port Scan Detected"; sid:20000016; rev:1;)
```

 Dec 4, 2025 @ 16:07:10.677	[network, intrusion_detection]	dns	udp	194.57.86.193	UDP Port Scan Detected	OPNsense.internal	suricata
--	--------------------------------	-----	-----	---------------	------------------------	-------------------	----------

Pour finir on a voulu détecter les attaques flood on a commencé par détecter les attaques SYN flood qui est l'attaque flood de base on a donc une fois de plus créé une règle qui le détecte pour les attaques flood SYN. Puis on a aussi fait les attaques flood UDP avec *hping3 --flood --udp -p 53 192.168.1.50*

```
#SYN Flood
alert tcp any any -> $HOME_NET any (flags:S; threshold:type both, track by_src, count 200, seconds 1; msg:"SYN Flood Suspected"; sid:20000004; rev:1;)
```

 Dec 4, 2025 @ 16:05:09.248	[network, intrusion_detection]	-	tcp	192.168.1.180	SYN Flood Suspected	OPNsense.internal	suricata
--	--------------------------------	---	-----	---------------	---------------------	-------------------	----------

Les multiples tests d'attaque réalisés ont permis de valider le bon fonctionnement de notre système de supervision puisque les scans, les tentatives de brute-force et les attaques par flood ont tous été correctement détectés et remontés par Suricata puis visualisés dans ELK. Les règles, qu'elles soient open-source ou personnalisées, ont permis d'assurer une détection fiable et en adéquation avec les scénarios testés.