

Consigne

- L'ensemble des réponses doivent être documentées et appuyées de captures d'écran. Même si la réponse est correcte si elle n'est pas documentée votre note sera de **0 point**.
- Vous devez faire preuve d'analyse et de recherche dans les réponses que vous fournissez.
- Veuillez soumettre vos devoirs sous forme de fichiers PDF ou DOC et présenter vos réponses sous les questions ci-dessous en les copiant telles quelles avec leur numéro de question.
- **Attention, utilisez votre propre compte utilisateur afin d'effectuer les opérations dans la plateforme Security Onion. (Voir Annexe A)**
- **La longueur du devoir ne doit pas dépasser 25 pages**

Exercice #1 (10p) – Formation Elastic

Allez sur <https://www.elastic.co/training/kibana-fundamentals> et créez-vous un compte. Suivez la formation sur les aspects fondamentaux de Kibana et joignez une image du certificat de complétion. Attention le certificat doit inclure votre prénom et votre nom.



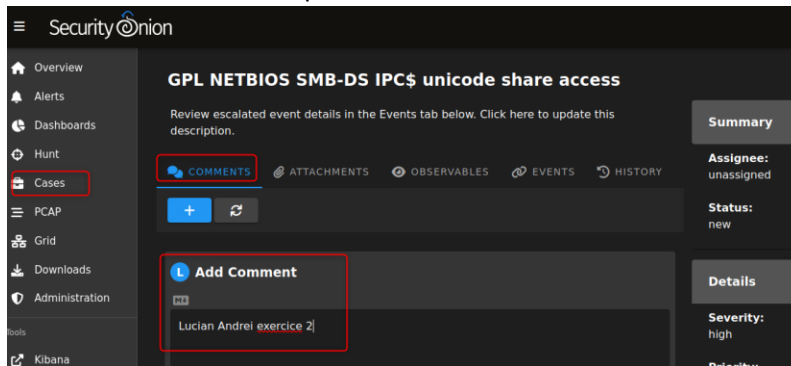
Exercice #2 (10p) – Formation Security Onion (max 5 pages)

Effectuez la formation en ligne disponible via le lien (en commençant par Intro to Analyst Tools) :

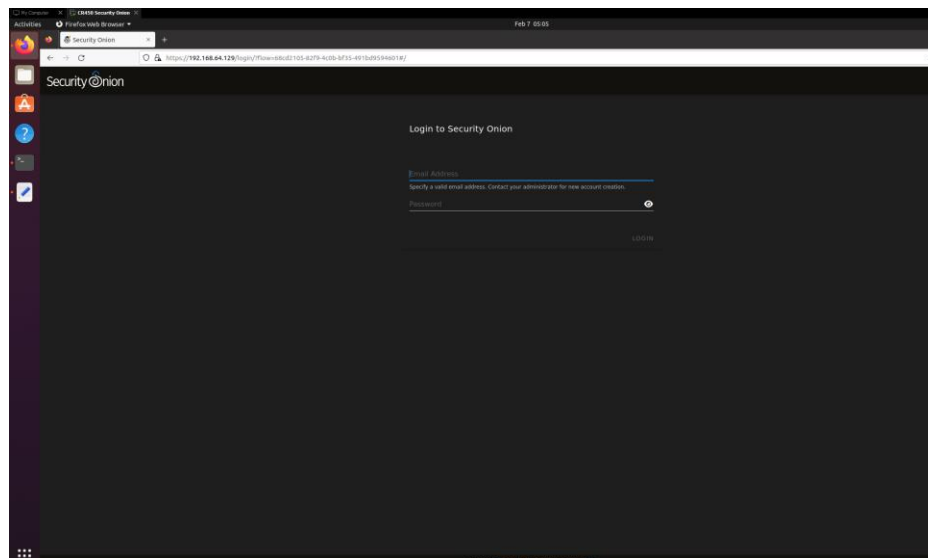
<https://www.youtube.com/watch?v=4PrwSuEEN8&list=PLIjFITO9rB155aYBjHw2InKkSMLuhWpxH&index=4>

Vous devez joindre à votre document de réponses :

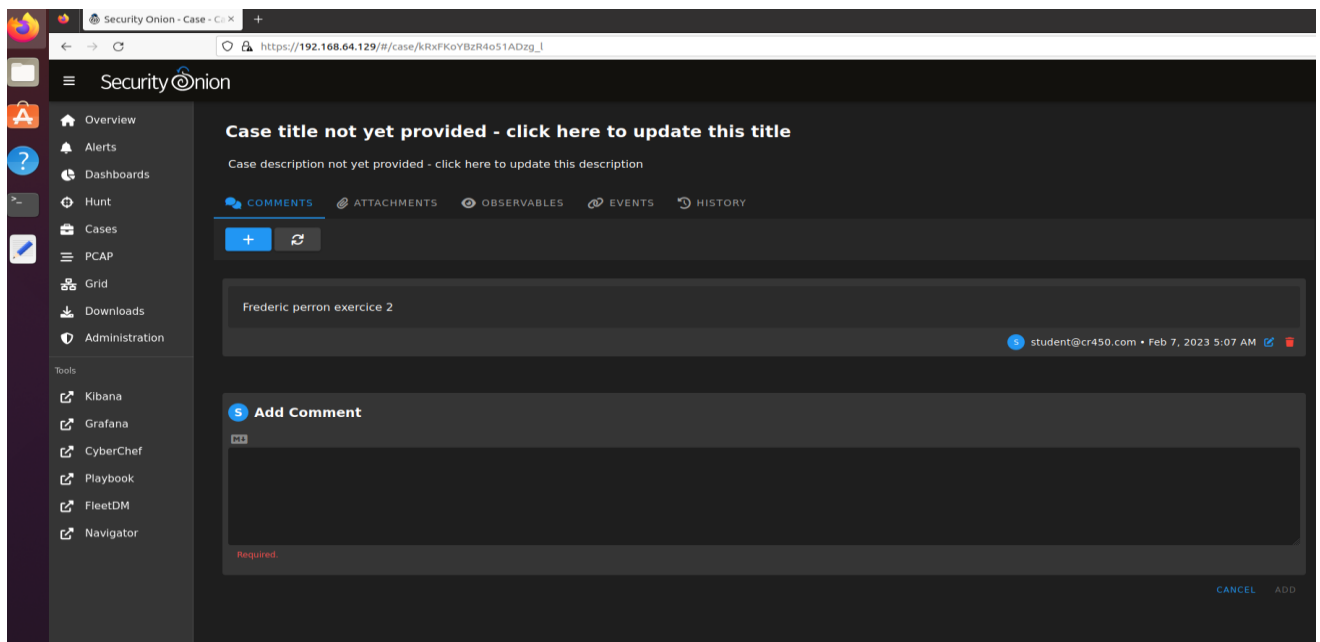
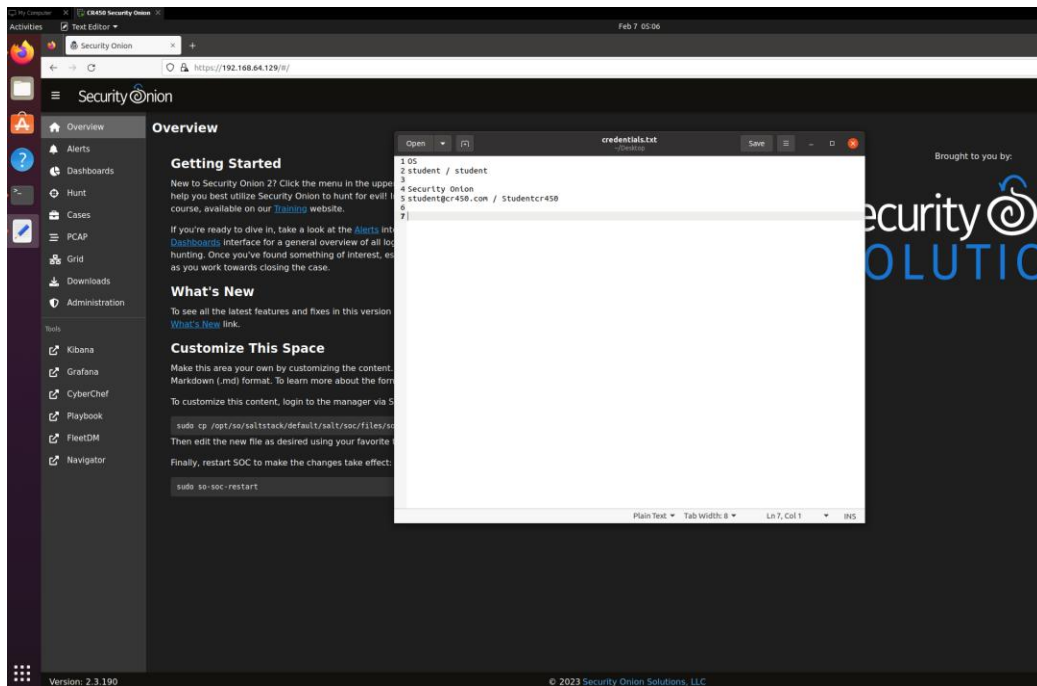
- Des captures d'écran des commandes et résultats obtenus (ex : sudo so-test etc.)
- Créez un cas d'escalade qui contiendra votre nom dans les commentaires



```
student@cr450:~$ sudo so-status
[sudo] password for student:
Checking Docker status
Docker ----- [ OK ]
Checking container statuses
so-aptcacherng ----- [ OK ]
so-curator ----- [ OK ]
so-dockerregistry ----- [ OK ]
so-elastalert ----- [ OK ]
so-elasticsearch ----- [ OK ]
so-filebeat ----- [ OK ]
so-fleet ----- [ OK ]
so-grafana ----- [ OK ]
so-ldstools ----- [ OK ]
so-influxdb ----- [ OK ]
so-kibana ----- [ OK ]
so-kratos ----- [ OK ]
so-mysql ----- [ OK ]
so-nginx ----- [ OK ]
so-playbook ----- [ OK ]
so-redis ----- [ OK ]
so-sensoronl ----- [ OK ]
so-soc ----- [ OK ]
so-soctopus ----- [ OK ]
so-steno ----- [ OK ]
so-strelka-backend ----- [ OK ]
so-strelka-coordinator ----- [ OK ]
so-strelka-filestream ----- [ OK ]
so-strelka-frontend ----- [ OK ]
so-strelka-gatekeeper ----- [ OK ]
so-strelka-manager ----- [ OK ]
so-suricata ----- [ OK ]
so-telegraf ----- [ OK ]
so-wazuh ----- [ OK ]
so-zeek ----- [ OK ]
student@cr450:~$
```



Après avoir entré les infos de credentials.txt nous avons pu rentrer :



Exercice #3 (15p) – tcpdump

À l'aide de votre machine Security Onion, effectuez ceci:

3.1 Démarrez une capture via tcpdump qui répond aux critères suivants :

- Désactive la résolution de nom (-n)
- Active la sortie détaillée (-v) pour afficher les champs d'en-tête IP.
- Capture les paquets ICMP
- Enregistre les paquets capturés dans un fichier appelé exercice3.pcap

```
student@cr450:~$ sudo tcpdump -nvtc 10 -w exercice3.pcap -i ens33 host 192.168.2.10 and host 192.168.64.129
[sudo] password for student:
tcpdump: listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
^C0 packets captured
0 packets received by filter
0 packets dropped by kernel
student@cr450:~$ sudo tcpdump -nvtc 10 -w exercice3.pcap -i ens33 host 192.168.2.10 and host 192.168.64.129
tcpdump: listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
Got 4
10 packets captured
10 packets received by filter
0 packets dropped by kernel
student@cr450:~$
```

```
ping: write error
student@cr450:~$ ping 192.168.2.10
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data.
64 bytes from 192.168.2.10: icmp_seq=1 ttl=128 time=0.504 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=128 time=0.791 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=128 time=0.402 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=128 time=0.441 ms
64 bytes from 192.168.2.10: icmp_seq=5 ttl=128 time=0.605 ms
64 bytes from 192.168.2.10: icmp_seq=6 ttl=128 time=0.387 ms
64 bytes from 192.168.2.10: icmp_seq=7 ttl=128 time=0.399 ms
64 bytes from 192.168.2.10: icmp_seq=8 ttl=128 time=0.537 ms
^C
--- 192.168.2.10 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7161ms
rtt min/avg/max/mdev = 0.387/0.508/0.791/0.128 ms
student@cr450:~$
```

3.2 Générer des paquets (à partir d'un autre terminal) à n'importe quel hôte (sur votre réseau ou Internet) qui entraînera le renvoi d'un message ICMP TTL exceeded. N'oubliez pas d'utiliser tcpdump pour capturer les deux paquets (stimulus et réponse). Vous pouvez utiliser ping avec l'option -t pour modifier le TTL

io.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.64.129	192.168.2.10	ICMP	98	Echo (ping) request id=0x0002, seq=1/256, ttl=64 (reply in 2)
2	0.000467	192.168.2.10	192.168.64.129	ICMP	98	Echo (ping) reply id=0x0002, seq=1/256, ttl=128 (request i...
3	1.015904	192.168.64.129	192.168.2.10	ICMP	98	Echo (ping) request id=0x0002, seq=2/512, ttl=64 (reply in 4)
4	1.016664	192.168.2.10	192.168.64.129	ICMP	98	Echo (ping) reply id=0x0002, seq=2/512, ttl=128 (request i...
5	2.041018	192.168.64.129	192.168.2.10	ICMP	98	Echo (ping) request id=0x0002, seq=3/768, ttl=64 (reply in 6)
6	2.041393	192.168.2.10	192.168.64.129	ICMP	98	Echo (ping) reply id=0x0002, seq=3/768, ttl=128 (request i...
7	3.064016	192.168.64.129	192.168.2.10	ICMP	98	Echo (ping) request id=0x0002, seq=4/1024, ttl=64 (reply in ...
8	3.064415	192.168.2.10	192.168.64.129	ICMP	98	Echo (ping) reply id=0x0002, seq=4/1024, ttl=128 (request ...
9	4.088218	192.168.64.129	192.168.2.10	ICMP	98	Echo (ping) request id=0x0002, seq=5/1280, ttl=64 (reply in ...
10	4.088771	192.168.2.10	192.168.64.129	ICMP	98	Echo (ping) reply id=0x0002, seq=5/1280, ttl=128 (request ...

Une fois que vous aurez capturé les paquets, répondez aux questions suivantes :

- Quel est le type de message ICMP envoyé ? (3p)

Echo Request

- ii. Quel est le format de message ICMP? Quelles sont les valeurs de type et de code ? (3p)

Format 20 bytes (header length), Type 8, code 0

- iii. Qu'avez-vous fait pour obtenir ce message (incluez la commande + capture d'écran que vous avez utilisée pour créer le paquet) ? (3p)

`sudo tcpdump -nv -w exercise3.pcap -i ens33 host 192.168.2.10 and host 192.168.64.129` (while pinging 192.168.64.129 from another terminal)

- iv. Quelle est la commande tcpdump que vous avez utilisée pour capturer les deux paquets ? Incluez le filtre que vous avez utilisé pour isoler les deux paquets (stimulus et réponse) ainsi qu'une capture d'écran.

`Sudo tcpdump -nvtc 10 -w exercise3.pcap -i ens33 host 192.168.2.10 and host 192.168.64.129`

- v. Dans la cellule vide du tableau ci-dessous, collez une capture d'écran de la sortie tcpdump (hexadécimal et ASCII) montrant les deux paquets (stimulus et réponse). (3p)

Après avoir eu un problème avec mes VM et mon router, les adresses IP sont malheureusement différentes dans cette exercice. J'ai du désinstaller et réinstaller Security Onion, mais voici le screenshot que j'ai fais avec les nouvelles adresses IP de mes machines.

```
student@cr450:~$ sudo tcpdump -nvtxc 10 -i ens33 host 192.168.64.131 and host 192.168.2.10
[sudo] password for student:
tcpdump: listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes

IP (tos 0x0, ttl 64, id 29157, offset 0, flags [DF], proto ICMP (1), length 84)
 192.168.64.131 > 192.168.2.10: ICMP echo request, id 1, seq 1, length 64
 0x0000: 4500 0054 71e5 4000 4001 04e6 c0a8 4083
 0x0010: c0a8 020a 0800 82de 0001 0001 e5d7 e263
 0x0020: 0000 0000 e210 0c00 0000 0000 1011 1213
 0x0030: 1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
 0x0040: 2425 2627 2829 2a2b 2c2d 2e2f 3031 3233
 0x0050: 3435 3637
IP (tos 0x0, ttl 128, id 5185, offset 0, flags [none], proto ICMP (1), length 84)
 192.168.2.10 > 192.168.64.131: ICMP echo reply, id 1, seq 1, length 64
 0x0000: 4500 0054 1441 0000 8001 628a c0a8 020a
 0x0010: c0a8 4083 0000 8ade 0001 0001 e5d7 e263
 0x0020: 0000 0000 e210 0c00 0000 0000 1011 1213
 0x0030: 1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
 0x0040: 2425 2627 2829 2a2b 2c2d 2e2f 3031 3233
 0x0050: 3435 3637
IP (tos 0x0, ttl 64, id 29163, offset 0, flags [DF], proto ICMP (1), length 84)
 192.168.64.131 > 192.168.2.10: ICMP echo request, id 1, seq 2, length 64
 0x0000: 4500 0054 71eb 4000 4001 04e0 c0a8 4083
 0x0010: c0a8 020a 0800 eb4c 0001 0002 e6d7 e263
 0x0020: 0000 0000 78a1 0c00 0000 0000 1011 1213
 0x0030: 1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
 0x0040: 2425 2627 2829 2a2b 2c2d 2e2f 3031 3233
 0x0050: 3435 3637
IP (tos 0x0, ttl 128, id 5186, offset 0, flags [none], proto ICMP (1), length 84)
 192.168.2.10 > 192.168.64.131: ICMP echo reply, id 1, seq 2, length 64
 0x0000: 4500 0054 1442 0000 8001 6289 c0a8 020a
 0x0010: c0a8 4083 0000 f34c 0001 0002 e6d7 e263
 0x0020: 0000 0000 78a1 0c00 0000 0000 1011 1213
 0x0030: 1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
 0x0040: 2425 2627 2829 2a2b 2c2d 2e2f 3031 3233
 0x0050: 3435 3637
IP (tos 0x0, ttl 64, id 29236, offset 0, flags [DF], proto ICMP (1), length 84)
 192.168.64.131 > 192.168.2.10: ICMP echo request, id 1, seq 3, length 64
 0x0000: 4500 0054 7234 4000 4001 0497 c0a8 4083
 0x0010: c0a8 020a 0800 92fd 0001 0003 e7d7 e263
 0x0020: 0000 0000 cfe0 0c00 0000 0000 1011 1213
 0x0030: 1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
```

Exercice 4 (15p) – Analyse d'une capture en Wireshark I

Via l'outil Wireshark, ouvrez le fichier "Exercise4.pcap"

- Assurez-vous qu'il n'y a pas de filtre d'affichage appliqué.
- Analysez la capture et référez-vous à la matière du cours afin de déterminer ce qu'y a d'étrange dans les communications capturées.
- Documentez votre hypothèse via la matière du cours.

Avec la matière vue en classe, nous pouvons former une hypothèse sur les paquets. Premièrement, un très grand nombre de communications ICMP dans un court délai de temps devrait être considéré comme suspectueux. Des paquets avec un nombre plus élevé de bytes qu'à l'habitude devrait également être un drapeau rouge. Après avoir analyser les paquets, nous pouvons déduire qu'avec certaines requêtes également, il est possible de flagger ces échanges pour une enquête approfondie puisqu'elle contient des paquets suspectueux. Les 2 analyses numéros 3 et 4 diffèrent puisque dans l'analyse 4, il y a de l'info d'inséré dans les paquets, contrairement à l'analyse numéro 3 qui est vide d'info, qui est simplement des échanges de communications non suspectueux. Le système d'opération source semble également être Windows vu l'architecture des fichiers dans l'info de quelques paquets. Des jpg et exe files sont trouvés un peu partout dans les échanges. En voici un exemple d'un paquet avec "de l'info inséré".

```
j·{··|·· ······E·  
·\v1···· ·_C·P··G  
{···Lu·· ··,272 n  
othing2s ee.jpg··  
10/15/20 03 11:3  
3 AM  
6,506 R EA
```

Conseils

- Reportez-vous aux fichiers vus en classe pour comparer les résultats à ce fichier.
- Pouvez-vous identifier le système d'exploitation source ?
- Y a-t-il des tendances qui diffèrent de l'analyse précédente ?
- Quelque chose d'intéressant à propos des charges utiles ?

Exercice 5 (5p) – Analyse d'une capture en Wireshark II

Via l'outil Wireshark, ouvrez le fichier "Exercise5.pcap"

- Assurez-vous qu'il n'y a pas de filtre d'affichage appliqué.
- Analysez la capture et référez-vous à la matière du cours afin de déterminer ce qu'y a d'étrange dans les communications capturées.

Même adresse IP 2 fois et les adresses MAC sont anormales. Nous pouvons déduire que quelqu'un utilise du spoofing (usurpation).

- Documentez votre hypothèse via la matière du cours.

Avec la matière vue en classe, nous pouvons déduire que cela
192.168.11.13 is at 11:22:33:44:55:66

Est spoofed.

Conseils

- Reportez-vous aux fichiers vus en classe pour comparer les résultats à ce fichier.

Exercice 6 (10p) - Analyse d'une capture en Wireshark III

Via l'outil Wireshark, ouvrez le fichier "Exercise6.pcap" et répondez aux questions suivantes :

- Identifiez le trafic correspondant à une attaque de type « brute force attack » (BRUTE FORCE = plusieurs mots de passe pour le même nom d'utilisateur). Afficher une capture d'écran de celui-ci. (5p)

TIME	SOURCE	DESTINATION	PROTOCOL	LENGTH	INFO
1 0.000000	192.168.126.133	192.168.126.130	TCP	74	59784 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1355917686 TSecr=0 WS=128
2 0.000299	192.168.126.130	192.168.126.133	TCP	74	80 → 59784 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=892830 TSecr=1355917686 WS=32
3 0.000324	192.168.126.133	192.168.126.130	TCP	66	59784 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1355917686 TSecr=892830
4 0.000453	192.168.126.133	192.168.126.130	HTTP	651	GET /dwa/login.php HTTP/1.1
5 0.000716	192.168.126.130	192.168.126.133	TCP	66	80 → 59784 [ACK] Seq=1 Ack=586 Win=6976 Len=0 TSval=892830 TSecr=1355917686
6 0.000796	192.168.126.130	192.168.126.133	HTTP	2094	HTTP/1.1 200 OK (text/html)
7 0.000796	192.168.126.130	192.168.126.133	TCP	66	80 → 59784 [FIN, ACK] Seq=2029 Ack=586 Win=6976 Len=0 TSval=892831 TSecr=1355917686
8 0.000723	192.168.126.133	192.168.126.130	TCP	66	59784 → 80 [ACK] Seq=586 Ack=2029 Win=63488 Len=0 TSval=1355917694 TSecr=892831
9 0.000847	192.168.126.133	192.168.126.130	TCP	66	59784 → 80 [FIN, ACK] Seq=586 Ack=2030 Win=64128 Len=0 TSval=1355917694 TSecr=892831
10 0.000455	192.168.126.130	192.168.126.133	TCP	66	80 → 59784 [ACK] Seq=2030 Ack=587 Win=6976 Len=0 TSval=892831 TSecr=1355917694
11 7.532381	192.168.126.133	192.168.126.130	TCP	74	49560 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1355925218 TSecr=0 WS=128
12 7.532597	192.168.126.130	192.168.126.133	TCP	74	80 → 49560 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=892784 TSecr=1355925218 WS=32
13 7.532610	192.168.126.133	192.168.126.130	TCP	66	49560 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1355925218 TSecr=892784
14 7.532787	192.168.126.133	192.168.126.130	HTTP	792	POST /dwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
15 7.532946	192.168.126.130	192.168.126.133	TCP	66	80 → 49560 [ACK] Seq=1 Ack=727 Win=7264 Len=0 TSval=892784 TSecr=1355925219
16 7.540673	192.168.126.130	192.168.126.133	HTTP	420	HTTP/1.1 302 Found
17 7.540674	192.168.126.130	192.168.126.133	TCP	66	80 → 49560 [FIN, ACK] Seq=355 Ack=727 Win=7264 Len=0 TSval=892784 TSecr=1355925219
18 7.540710	192.168.126.133	192.168.126.130	TCP	66	49560 → 80 [ACK] Seq=727 Ack=355 Win=64128 Len=0 TSval=1355925226 TSecr=892784
19 7.541339	192.168.126.133	192.168.126.130	TCP	66	49560 → 80 [FIN, ACK] Seq=727 Ack=356 Win=64128 Len=0 TSval=1355925227 TSecr=892784
20 7.541460	192.168.126.130	192.168.126.133	TCP	66	80 → 49560 [ACK] Seq=356 Ack=728 Win=7264 Len=0 TSval=892785 TSecr=1355925227
21 7.650298	192.168.126.133	192.168.126.130	TCP	74	49570 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1355925336 TSecr=0 WS=128
22 7.650540	192.168.126.130	192.168.126.133	TCP	74	80 → 49570 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=892796 TSecr=1355925336 WS=32
23 7.650565	192.168.126.133	192.168.126.130	TCP	66	49570 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1355925336 TSecr=892796
24 7.650730	192.168.126.133	192.168.126.130	HTTP	796	POST /dwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
25 7.650982	192.168.126.130	192.168.126.133	TCP	66	80 → 49570 [ACK] Seq=1 Ack=731 Win=7264 Len=0 TSval=892796 TSecr=1355925337
26 7.658872	192.168.126.130	192.168.126.133	HTTP	420	HTTP/1.1 302 Found
27 7.658872	192.168.126.130	192.168.126.133	TCP	66	80 → 49570 [FIN, ACK] Seq=355 Ack=731 Win=7264 Len=0 TSval=892796 TSecr=1355925337
28 7.658898	192.168.126.133	192.168.126.130	TCP	66	49570 → 80 [ACK] Seq=731 Ack=355 Win=64128 Len=0 TSval=1355925345 TSecr=892796
29 7.659240	192.168.126.133	192.168.126.130	TCP	66	49570 → 80 [FIN, ACK] Seq=731 Ack=356 Win=64128 Len=0 TSval=1355925345 TSecr=892796
30 7.659432	192.168.126.130	192.168.126.133	TCP	66	80 → 49570 [ACK] Seq=356 Ack=732 Win=7264 Len=0 TSval=892796 TSecr=1355925345
31 7.772548	192.168.126.133	192.168.126.130	TCP	74	49582 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1355925458 TSecr=0 WS=128
32 7.772804	192.168.126.130	192.168.126.133	TCP	74	80 → 49582 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=892808 TSecr=1355925458 WS=32
33 7.772831	192.168.126.133	192.168.126.130	TCP	66	49582 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1355925459 TSecr=892808
34 7.773007	192.168.126.133	192.168.126.130	HTTP	794	POST /dwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
35 7.773266	192.168.126.130	192.168.126.133	TCP	66	80 → 49582 [ACK] Seq=1 Ack=729 Win=7264 Len=0 TSval=892808 TSecr=1355925459
36 7.781368	192.168.126.130	192.168.126.133	HTTP	420	HTTP/1.1 302 Found

```

} Connection: close
}
} username=user&password=Etudiant!&Login=LoginHTTP/1.1 302 Found
} Date: Thu, 01 Dec 2022 04:57:16 GMT
} Server: Apache/2.2.8 (Ubuntu) DAV/2
} X-Powered-By: PHP/5.2.4-2ubuntu5.10
} Expires: Thu, 19 Nov 1981 08:52:00 GMT
} Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
} Pragma: no-cache
} Location: login.php
} Content-Length: 0
} Connection: close
}

```

- Identifiez le dernier mot de passe qui a été essayé ? Afficher une capture d'écran de celui-ci. (5p)

Cr450rocks

```

username=user&password=cr450rocks&Login=LoginHTTP/1.1 302 Found
Date: Thu, 01 Dec 2022 04:57:17 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Expires: Thu, 19 Nov 1981 08:52:00 GMT

```

Exercice 7 (35p) Tunnellisation ICMP et C&C (35p)

- 7.1 Installez une nouvelle machine virtuelle [Ubuntu 20.04 Server](#) . Lors de l'installation, sélectionnez l'option serveur SSH. Veuillez conserver cette machine, car nous l'utiliserons pour un autre devoir. (5p)


```

Ubuntu 20.04.5 LTS cr450 tty1

cr450 login: perfre
Password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-137-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue 07 Feb 2023 11:13:37 AM UTC

System load:  0.02               Processes:           236
Usage of /:   24.3% of 9.75GB    Users logged in:    0
Memory usage: 8%                IPv4 address for ens33: 192.168.64.132
Swap usage:   0%

26 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Feb  7 10:04:11 UTC 2023 on tty1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

perfre@cr450:~$ _

```

- Utilisez <https://github.com/krabelize/icmptdoor> pour créer un canal de commande et de contrôle entre votre serveur Ubuntu (client) et Security Onion (serveur c&c). (5p)
 - o Exécutez les commandes avec
 - `sudo python3 scrip_name.py...`
 - o Si vous recevez une erreur indiquant que Scapy n'est pas installé, installez-le à l'aide de la commande
 - `sudo apt install python3-scapy`

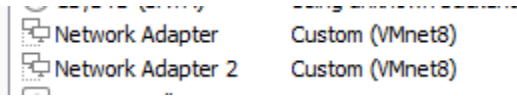
```

student@cr450: ~
student@cr450:~$ sudo git clone https://github.com/krabelize/icmptdoor
[sudo] password for student:
Cloning into 'icmptdoor'...
remote: Enumerating objects: 209, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 209 (delta 2), reused 0 (delta 0), pack-reused 203
Receiving objects: 100% (209/209), 26.61 MiB | 49.19 MiB/s, done.
Resolving deltas: 100% (109/109), done.
student@cr450:~$ sudo apt install python3-scapy
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-scan-plugin libfwupdplugin1 libxmlb1 ubuntu-advantage-desktop-daemon
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ipython3 python3-backcall python3-decorator python3-ipython
  python3-ipython-genutils python3-jedi python3-parso python3-pickleshare
  python3-prompt-toolkit python3-pygments python3-traitlets python3-wcwidth
Suggested packages:
  python3-docutils python3-pylint python3-setuptools python3-typing python3-venv

```

Depuis votre machine Security Onion

- Assurez-vous que la deuxième carte réseau est dans le même sous-réseau que la première interface.



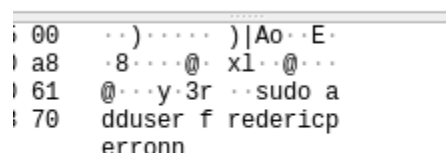
- Démarrez Wireshark en tant que sudo et commencez à capturer le trafic à partir de votre interface ens33.

```
student@cr450: ~/icmpdoor
student@cr450:~$ cd icmpdoor
student@cr450:~/icmpdoor$ sudo python3 icmp-cnc.py -i ens33 -d 192.168.64.132
[sudo] password for student:
[+]ICMP C2 started!
shell: 
```

```
[sudo] password for student:
[+]ICMP C2 started!
shell: sudo adduser fredericperronn
sudo adduser fredericperronn
shell: 
```

1900	647.967337238	192.168.64.131	192.168.64.132	ICMP	70	Echo (ping) request	id=0x3372, seq=0/0, ttl=64
1908	647.967337238	192.168.64.132	192.168.64.131	ICMP	70	Echo (ping) reply	id=0x3372, seq=0/0, ttl=64

- Depuis votre serveur C&C exécutez diverses commandes intégrées Linux (ex : pwd , cat /etc /passwd, ifconfig...) afin d'obtenir les configurations de la machine Ubuntu.
 - o Via l'outil Wireshark, identifiez les commandes et les réponses (5p)

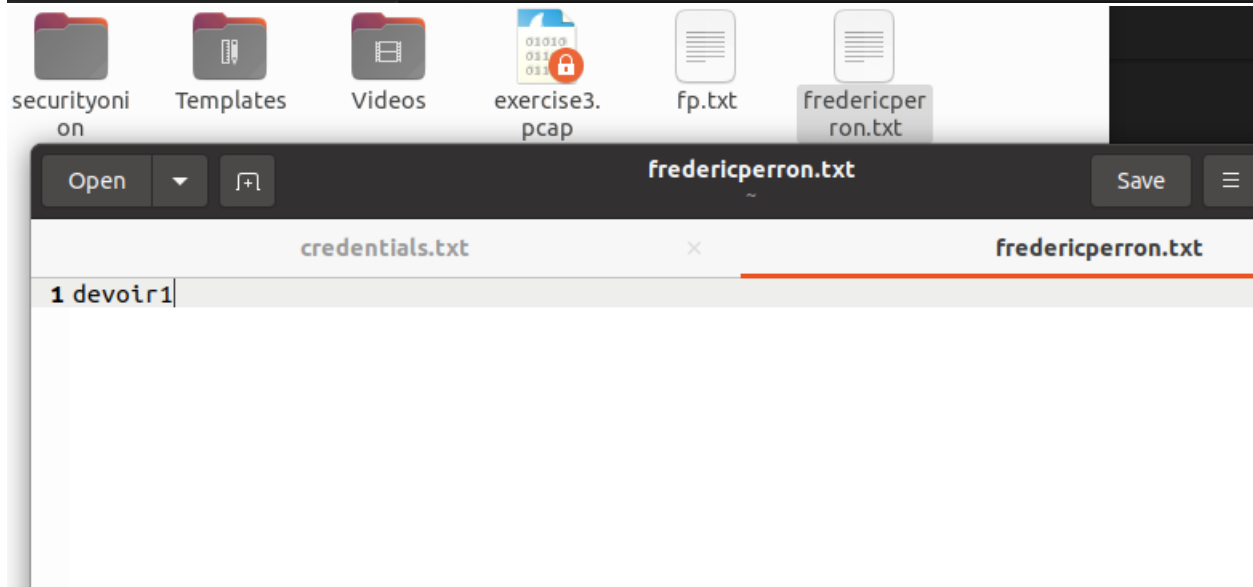


- Regardez les trames ICMP dans Wireshark. Expliquez en quoi ce trafic diffère d'un trafic ICMP normal (5p)

Évidemment les trames démontrent des données insérées dans les paquets ou des payloads. Les bytes sont également un peu plus volumineux qu'à l'habitude ce qui diffère de trafic normal avec vraisemblablement toujours le même nombre de byte et un équilibre entre les paquets.

- Créez un nouveau fichier sur le client first_lastname.txt ayant le contenu <Devoir1>. Afficher le contenu du fichier sur le terminal (5p)
 - Indice : la commande echo peut être utile dans ce cas

```
student@cr450:~/icmpdoor$ cd
student@cr450:~$ echo 'devoir1' > fredericperron.txt
student@cr450:~$
```



The screenshot shows a file manager window with a sidebar containing icons for 'securityoni on', 'Templates', 'Videos', 'exercice3.pcap', 'fp.txt', and 'fredericperron.txt'. The main window displays the file 'fredericperron.txt' with a single line of text: '1 devoir1'.

- Créez un nouvel utilisateur sur le serveur Ubuntu (votre nom_prenom). Via des captures d'écran, démontrez l'ensemble des étapes effectuées (5p)

```

perfre@cr450:~$ sudo adduser fredericperron
Adding user `fredericperron' ...
Adding new group `fredericperron' (1003) ...
Adding new user `fredericperron' (1003) with group `fredericperron' ...
Creating home directory `/home/fredericperron' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for fredericperron
Enter the new value, or press ENTER for the default
    Full Name []: fredericperron
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
perfre@cr450:~$ _

```

- Identifier et analyser le trafic C&C dans Security Onion (Alertes, Kibana ...) (5p)

Count	source.ip
26,136	192.168.64.131
836	192.168.64.1
820	192.168.10.128
598	192.168.64.132
331	fe80::555a:e04d:f838:e

9	ET P2P BitTorrent peer sync	suricata	high
9	ET MALWARE Zbot POST Request to C2	suricata	high
9	ET MALWARE Generic - POST To .php w/Extended ASCII Characters (Likely Zeus Derivative)	suricata	high

Nous pouvons constater en analysant les alertes que des malwares sont présents et requièrent une attention immédiate vu leur `high` risk.

Annexe A – Création d'un utilisateur dans votre OS et dans Security Onion

- Ajoutez un nouvel utilisateur dans le système d'exploitation:

`sudo adduser <UserNameHere>`

Le nom d'utilisateur doit être une combinaison de votre PrenomNom.

Ex: `sudo adduser landrei`

- Ajoutez le nouveau user dans les sudoers

`sudo usermod -aG sudo <new_user>`

Ex: `sudo usermod -aG sudo landrei`

- Ajoutez un nouvel utilisateur pour Security Onion

`sudo so-user-add PrenomNom@cr450.com`

Ex: `sudo so-user-add landrei@cr450.com`

- Une fois l'utilisateur ajouté, fermez la session active et connectez-vous avec le nouvel utilisateur.
- Une fois que vous avez vérifié, via la commande « `sudo so-status` » que tous les services ont démarrés, prenez un snapshot de votre machine virtuelle.