
INTRUSION DETECTION SYSTEM

Contents

1	Introduction	3
2	Intrusion	3
2.1	Scan TCP connect ou SYN	4
2.2	Scan UDP	5
2.3	Scan Xmas ou Null	5
3	Detection d'intrusion: IDS, IPS	6
4	Modèle de détection	7
4.1	Modèle	7
4.2	Data set	8
4.3	Résultats	9
5	Mesure et prévention	9
6	Limites et analyse	10
6.1	Fragmentation	10
6.2	Flooding	10
6.3	Encryption	11
7	Conclusion	11
A	Annexe	13

1 Introduction

Se définissant comme tout type d'activité non autorisée pouvant causer dommage et porter atteinte à un système, une intrusion résulte de la pénétration non autorisée dans un système, constituant ainsi une menace potentielle pour la confidentialité, l'intégrité ou la disponibilité des informations.

Constituant généralement la première phase d'une cyber-attaque, l'intrusion fait partie des principaux enjeux de la sécurité informatique, soulevant ainsi la question suivante : « Comment détecter une intrusion ? ».

Répondant à cette question, les IDS sont, de nos jours, l'une des principales solutions mises en œuvre afin de détecter toute activité anormale éventuelle à l'échelle d'un réseau ou d'un système.

Permettant ainsi d'automatiser mais également d'optimiser la détection d'intrusion, leur utilité ainsi que leur fonctionnement sont présentés dans le cadre de ce travail, soulevant ainsi leur importance mais aussi leur enjeu à l'échelle de la sécurité informatique.

2 Intrusion

Constituant souvent la première étape d'une intrusion - et pouvant donc se présenter comme un exemple même d'intrusion-, le scan des ports permet en effet de déterminer les services ou applications actifs sur un système cible, mais également s'il est actif ou non.

Utilisant les protocoles de communication établis - TCP, UDP – afin d'identifier les ports ouverts sur ce système – ou les points sur un ordinateur où l'échange d'informations entre plusieurs programmes et Internet vers des appareils ou d'autres ordinateurs a lieu –, le scan de ports permet également d'établir la possibilité d'une connexion, afin d'obtenir des informations précises sur le serveur Web qui écoute sur ce port.

Le numéro de ports étant codé sur 16 bits, il est par ailleurs possible de dénombrer environ 65 536 ports sur un ordinateur.

Numéro de port	Service
20	Transfert de données FTP
21	Contrôle FTP
22	SSH
23	Telnet
25	SMTP
53	DNS
80	http
137-139	NetBIOS
443	HTTPS
445	SMB
1433	MSSQL
3306	MySQL
3389	RDP
5800	VNC au dessus de http
5900	VNC

Table 1: Exemples de ports présents sur un système.

Le but étant par ailleurs de solliciter une réponse par l'envoi de paquets à la machine cible afin de montrer ou non la présence d'une application ou d'un service, les différents états d'un port peuvent être caractérisés comme suit: *open*, *closed*, *filtered* - dans le cas de la présence d'un pare-feu -, *unfiltered*, *open/filtered*, *closed/filtered*.

Par ailleurs, afin de mieux expliciter cet exemple d'intrusion, plusieurs exemples sont donnés ci-dessous, reposant sur l'utilisation de l'outil Nmap ainsi que de ses différentes options.

2.1 Scan TCP connect ou SYN

Etant l'un des types de scan les plus répandus ainsi que les plus utilisés, le scan TCP Connect tente d'effectuer une connexion en trois étapes complètes - suivant le protocole TCP - sur chacun des ports indiqués, afin de déterminer leur état.

En effet, exigeant une communication synchronisée entre émetteur et récepteur, le protocole TCP suit le protocole de communication suivant :

1. SI récepteur se met en ouverture passive, prêt à recevoir des demandes de connexion
2. SI émetteur devenue une ouverture active, envoie un segment de contrôle appelé SYN récepteur.
3. SI récepteur répond par un segment de contrôle SYN-ACK
4. SI émetteur confirme par un segment ACK

Reposant ainsi sur le protocole TCP, le scan TCP connect cherche en effet à établir une connexion suivant ces 3 étapes – afin de déterminer si le système cible est actif ou qu'un port est ouvert, afin de permettre une éventuelle connexion, tel que le montre la fig 1.

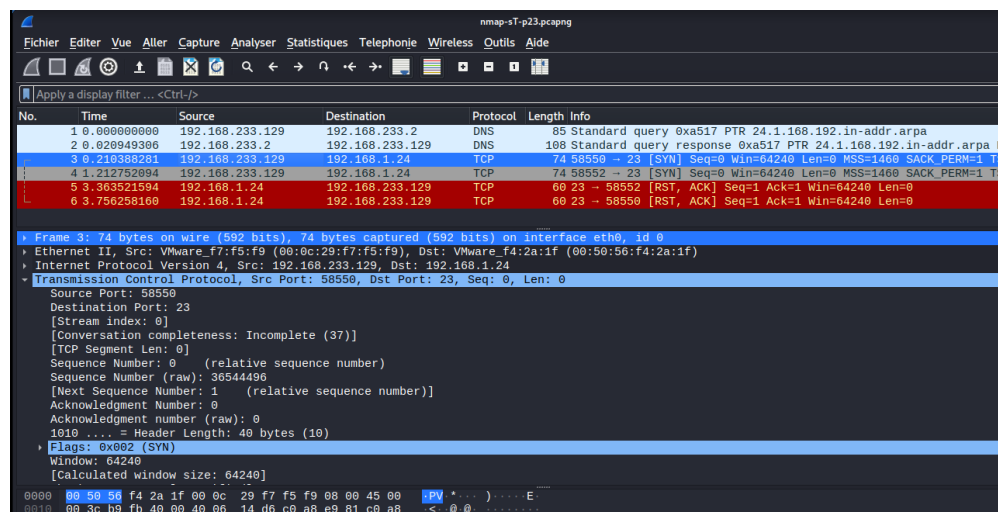


Figure 1: Capture Wireshark d'un scan nmap TCP Connect sur le port 23.

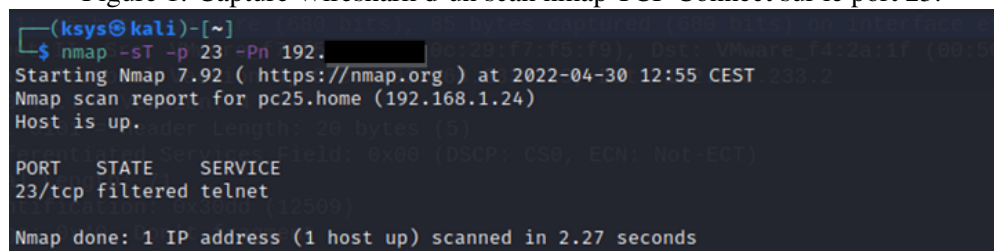


Figure 2: Scan nmap TCP Connect sur le port 23.

Telnet étant un protocole permettant de communiquer avec un serveur distant en échangeant des lignes de texte, la figure 1 présente en effet la capture Wireshark d'un scan TCP Connect réalisé avec Nmap sur le port 23, permettant ainsi d'observer l'échange de paquets SYN et ACK - relatifs aux deux premières étapes du protocole TCP - et d'établir l'état du port - tel que le montre la figure 2 -.

Etant également une variante de ce type de scan et étant sans doute le scan le plus répandu – car plus rapide que le précédent -, le scan SYN tente d'effectuer une connexion en deux étapes, ne cherchant à réaliser que les deux premières étapes du protocole TCP.

2.2 Scan UDP

Par ailleurs, les différents ports pouvant utiliser différents protocoles, un scan UDP est aussi possible, mettant en œuvre le protocole de communication UDP, afin de déterminer l'état du système et d'établir une éventuelle connexion.

Cependant, étant orienté sans connexion, le protocole UDP ne garantit pas l'arrivée des paquets envoyé au récepteur. En effet, envoyant simplement des paquets au destinataire, une communication UDP ne déclenche pas nécessairement une réponse du récepteur – et donc pas nécessairement la confirmation de la réception d'un paquet –, rendant difficile de déterminer si un port UDP est ouvert ou encore filtré, tel que le montre la figure 3, présentant une capture Wireshark d'un scan UDP sur le port 69 – ou TFTP, étant un protocole simplifié de transfert de fichiers –.

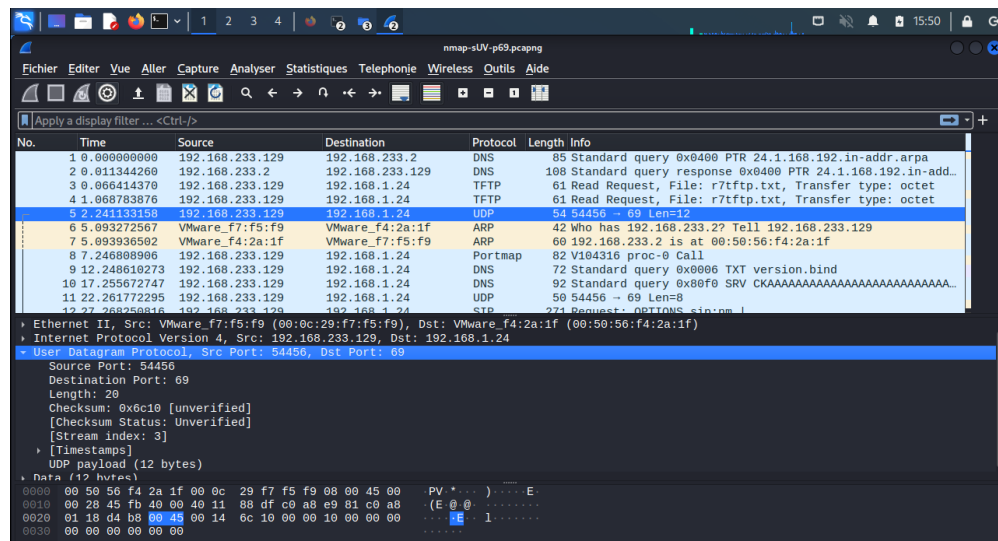


Figure 3: Capture Wireshark d'un scan nmap UDP sur le port 69.

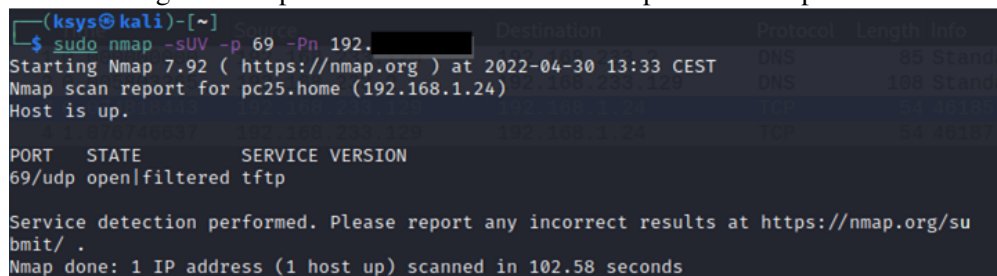


Figure 4: Scan nmap UDP sur le port 69.

2.3 Scan Xmas ou Null

Enfin, reposant également sur les étapes du protocole TCP, un scan Xmas est également possible, utilisant un type particulier de paquets.

En effet, reposant sur le fait qu'un port fermé recevant un paquet dans lequel un drapeau SYN, ACK, ou RST n'est pas positionné doit répondre par un paquet RST – ou qu'un port fermé recevant ce même type de paquet doit l'ignorer –, le scan Xmas utilise des paquets dont les drapeaux FIN - permettant d'interrompre la communication s'il est positionné à 1 –, PSH - impliquant l'utilisation de la méthode PUSH –, et URG - impliquant de traiter le paquet de façon urgente – sont positionnés – et non pas les drapeaux SYN ou ACK –, permettant de déclencher ce type de réponse afin de déterminer si le port est ouvert ou non.

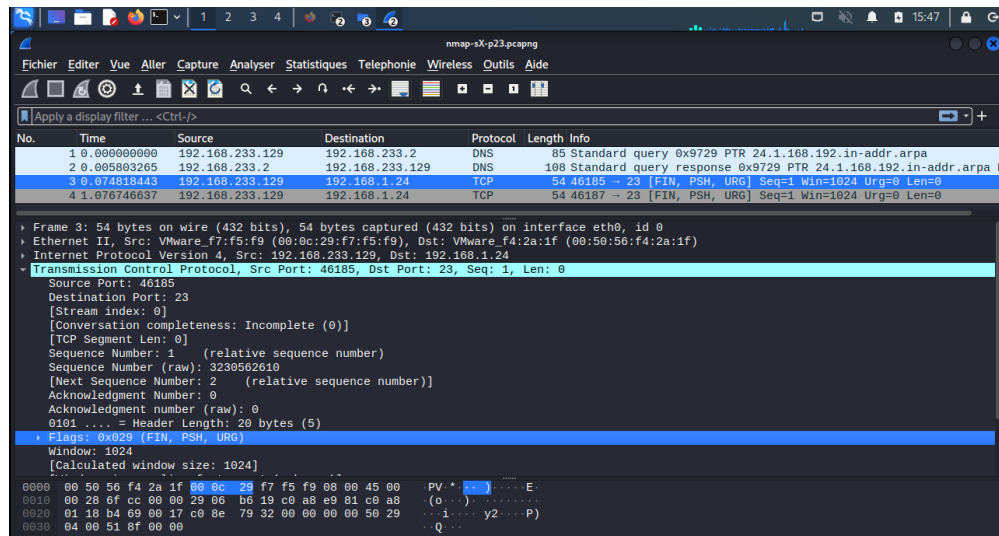


Figure 5: Capture Wireshark d'un scan nmap Xmas sur le port 23.

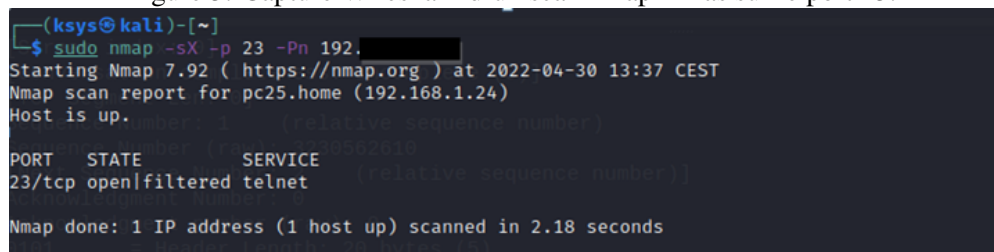


Figure 6: Scan nmap Xmas sur le port 23.

Ainsi, permettant de déterminer l'état des ports d'un système et donc d'établir une possible connexion avec un système cible, le scan des ports ainsi détaillé permet de fournir un exemple d'intrusion, ainsi que la nécessité de les détecter.

3 Détection d'intrusion: IDS, IPS

Faisant partie intégrante du domaine de la détection d'intrusion étant définie comme étant « l'ensemble des pratiques et des mécanismes utilisés qui permettent de détecter des problèmes pouvant conduire à des violations de la politique de sécurité » (1), un système de détection (IDS) – ou Intrusion Detection System – est un système logiciel ou matériel identifiant les actions malveillantes sur un système informatique afin de permettre le maintien de sa sécurité.

L'objectif d'un IDS étant d'identifier différents types de trafic réseau malveillant et d'utilisation de l'ordinateur ne pouvant pas être identifiés par un pare-feu traditionnel -, les systèmes de détection d'intrusion peuvent se distinguer en deux types que sont les IDS Réseaux (N-IDS, Network Based Intrusion Detection System) et les IDS Hôtes (H-IDS, Host Based Intrusion Detection System), selon leur localisation (sur l'infrastructure réseau dans un système hôte) et leur champ d'action.

IDS Réseaux (N-IDS, Network Based Intrusion Detection System): sondes similaires aux sniffers qui autorisent la surveillance du trafic afin d'identifier les activités suspectes, capable d'analyser le trafic réseau entrant.

IDS Hôtes (H-IDS, Host Based Intrusion Detection System): sont fonctions des systèmes d'exploitation et des systèmes surveillés, peuvent déterminer les processus, utilisateurs et données à l'origine d'un problème et analyser des logs, capable de surveiller les fichiers sensibles du système d'exploitation.

Visant ainsi à détecter les données qui pourraient éventuellement conduire à toute forme d'incident ou d'intrusion, un système de détection d'intrusions collecte dans un premier temps les informations et événements relatifs à un système, avant de les analyser. La collecte d'informations peut se faire à deux niveaux.

Les informations peuvent être obtenues au niveau de la machine hôte (généralement par le biais de son système d'exploitation) en réalisant des fonctions d'audit des événements et en les enregistrant – au niveau du noyau et des applications de l'utilisateur –, permettant ainsi d'observer directement le comportement d'un système et les événements qui y surviennent.

les informations peuvent également être collectées au niveau du réseau, le système de détection d'intrusion étant alors un point de contrôle obligé par lequel transitent toutes les données qui sont alors enregistrées dans le système, lit simplement les données au moment où elles lui parviennent de manière indépendante des autres systèmes connectés au réseau, permettant ainsi de détecter certaines attaques qui ne sont pas visibles par les systèmes de détection installés sur les machines hôtes.

Succédant à la collecte des informations réalisées, une étape d'analyse des données est nécessaire afin d'identifier les comportements suspects ou pouvant relever d'une intrusion.

Pour ce faire, deux méthodes d'analyse existent.

La première méthode d'analyse, étant basée sur des signatures, consiste à examiner les paquets réseau - ou l'ensemble des activités-, afin d'essayer de les comparer à une base de données de signature, déclenchant une alarme si une correspondance est trouvée entre celle fournie et celles déjà identifiées dans la base de données.

La seconde, se basant sur des techniques d'apprentissage automatique, de méthodes statistiques ou encore sur la connaissance, consiste à créer un modèle normal du comportement d'un système informatique, afin de comparer et d'évaluer chaque comportement, et de parvenir à distinguer les comportements anormaux ou les intrusions éventuelles au système.

Finalement, succédant aux phases de collecte et d'analyse des informations, les résultats obtenus sont par la suite restitués à l'utilisateur, afin de permettre la mise en œuvre de mesures de prévention ou encore de solution, afin de prévenir l'intrusion.

4 Modèle de détection

S'intéressant plus particulièrement à cette deuxième famille d'IDS – ou aux *Anomaly-based intrusion detection system* (AIDS), cherchant ainsi à créer un modèle normal du comportement d'un système afin de détecter les comportements anormaux ou les intrusions éventuelles, à l'aide d'un modèle d'apprentissage – un modèle de classification est présenté dans ce rapport, utilisant la donnée du KDDCup99, afin d'illustrer le principe de fonctionnement d'un IDS.

4.1 Modèle

Dans un premier temps, constituant une première étape de préparation de la donnée, les targets – ou attaques – identifiées sont regroupées selon cinq classes définies que sont normal, u2R, r2l, probe, tels que décrit dans le paragraphe suivant.

Succédant à cette étape d'identification des classes - ou targets – correspondant au problème de classification établi, une étape de normalisation est par la suite mise en œuvre, utilisant la méthode de Min-Max scaling telle que $x' = \frac{x - \min}{\max - \min}$, afin de réduire l'écart entre les données ainsi que leur duplication. (optionnel)

La donnée étant ainsi préparée, les features – ou caractéristiques principales - sont par la suite extraites utilisant les méthodes Random Forest ou PCA.

Enfin, utilisant la donnée ainsi établie, un modèle de classification est par la suite utilisé afin de déterminer le type de connexion, s'il s'agit ou non d'une éventuelle intrusion.

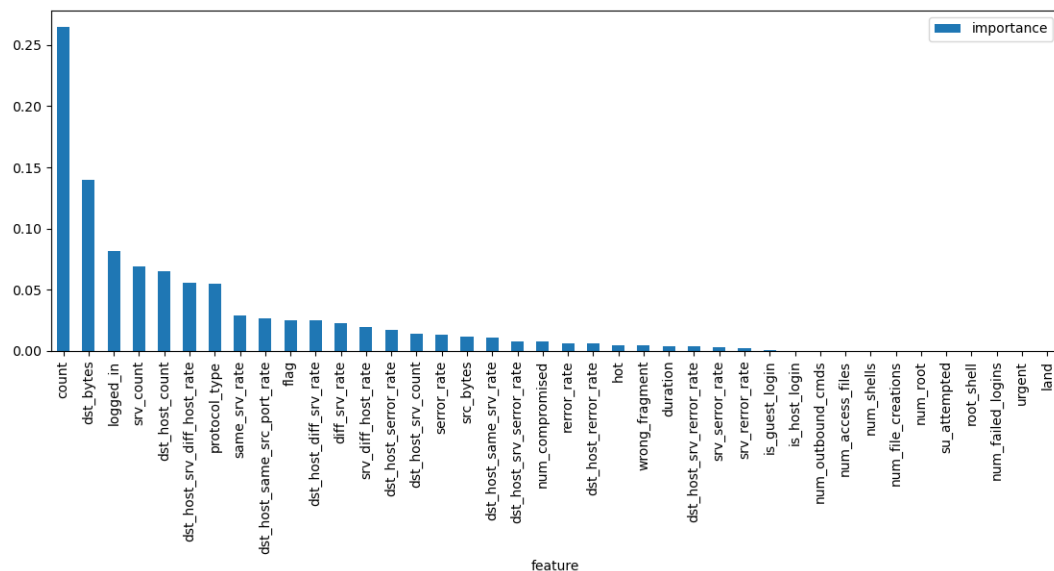
4.2 Data set

Afin de tester et d'évaluer le modèle établi, la donnée de la KDDCup99 est utilisée, une donnée présentant un ensemble standard de données à auditer, comprenant une grande variété d'intrusions simulées dans un environnement de réseau militaire. Ce data set utilise par ailleurs 41 features pour décrire une connexion, ainsi que 23 types d'attaques afin de classifier les différents types de connexion - tels que décrit dans l'annexe -.

Par ailleurs, utilisant le Random Forest Classifier – reposant sur l'utilisation des arbres de décision -, afin de déterminer l'importance ou l'impact sur la classification de chacun des features présentés dans la data set – tel que présenté dans la fig.7 décrivant l'importance de chacun des features -, il est possible d'observer qu'un nombre restreint de features peut être utilisé pour la classification, sans pour autant prendre en compte la totalité d'entre eux.

Les features count - décrivant le nombre de connexions au même hôte que la connexion actuelle au cours des deux dernières secondes - ; dst_bytes - décrivant le nombre de data bytes envoyés de la source à la destination - ; logged_in - égal à 1 si la connexion est réussie, 0 autrement - ; srv_count - décrivant le nombre de connexions au même service que la connexion actuelle au cours des deux dernières secondes - ; protocol_type - décrivant le type de protocole utilisé par la communication - ; semblent en effet être les plus importants du point de vue de la classification - apparaissant également cohérents avec l'exemple d'intrusion fourni qu'est le scan des ports, puisque relatifs aux principales caractéristiques d'une connexion -, d'où leur extraction pour l'application du modèle.

Figure 7: Extraction des features.



Par ailleurs, afin de simplifier le problème proposé, les 23 types d'attaques proposés dans le data set - que sont back, buffer_overflow, ftp_write, guess_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap.perl, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster- sont regroupés selon les 4 classes suivantes, permettant ainsi de définir les labels pour la classification :

Denial-of-Service (DoS) : attaque ayant pour but de bloquer ou de restreindre les services délivrés par le réseau informatique aux utilisateurs, ou encore de refuser l'accès légitime à une machine.

Probing attack : attaque ayant pour objectif l'acquisition d'informations sur le réseau ou le système informatique dans le but apparent de contourner ses contrôles de sécurité

User-to-Root (U2R) : classe d'exploit dans laquelle l'attaquant, commençant avec un accès à un compte d'utilisateur normal sur le système (pouvant être obtenu en reniflant des mots de passe, une attaque par dictionnaire ou une ingénierie sociale), est capable d'exploiter certaines vulnérabilités pour obtenir un accès root au système.

Remote-to-Local (R2L) : attaque lors de laquelle un attaquant qui a la capacité d'envoyer des paquets à une machine sur un réseau mais qui n'a pas de compte sur cette machine exploite une vulnérabilité pour obtenir un accès local en tant qu'utilisateur de cette machine.

A l'échelle du data set, il est par ailleurs possible d'observer que les connexions normales représentent 19.6910% du data set, les connexions de type dos 79.2391%, u2r 0.0105%, r2l 0.2279%, et probe 0.8313%, exigeant ainsi un échantillonnage de chacune de ces classes afin d'établir un data set équilibré pour la classification.

Ainsi, utilisant les 15 features extraits par les méthodes Random Forest – ou éventuellement PCA – ainsi que sur les 5 classes déduites du data set, le modèle réalisé permet en effet de détecter le type d'une connexion, s'il s'agit ou non d'une intrusion – correspondant à l'un des 4 types d'attaques établis -.

4.3 Résultats

Utilisant la donnée extraite du data set – tel que décrit dans le paragraphe précédent – plusieurs algorithmes de classification sont par ailleurs testés afin d'être comparés.

Obtenant les résultats décrit dans la Table 2., les algorithmes Naive Bayes, Logistic regression, SVM et Decision Tree sont en effet testés, permettant en particulier d'obtenir une accuracy de 0.9766 et un kappa score de 0.9677 pour ce dernier modèle de classification.

	Accuracy	Kappa score
Naive Bayes	0.8327	0.7714
Logistic regression	0.8696	0.8223
SVM	0.9121	0.8795
Decision Tree	0.9839	0.9773

Table 2: Résultats obtenus pour une extraction des features.

	Accuracy	Kappa score
Naive Bayes	0.8110	0.7404
Logistic regression	0.9124	0.8796
SVM	0.9420	0.9189
Decision Tree	0.9766	0.9677

Table 3: Résultats obtenus pour la méthode PCA.

Ainsi , reposant sur l'utilisation du KDDCup 99 afin d'extraire les features utilisés mais aussi les classes utilisés pour la classification, le modèle mis en œuvre permet en effet de classer différentes connexions selon leur type, permettant ainsi distinguer les connexions normales des éventuelles intrusions.

5 Mesure et prévention

Une fois l'intrusion détectée par le système de détection, une réponse ou la mise en œuvre d'une solution ou mesure de prévention peut-être envisagée.

En effet, un système de prévention d'intrusion (IPS), permettant tout comme les IDS de détecter une intrusion, permet également de les prévenir.

Parmi, les réponses ou solutions envisageables succédant la détection d'une intrusion, il est possible de distinguer les réponses actives des réponses passives.

Alors qu'une réponse passive ne consiste qu'à présenter toute les informations récoltées et ayant permis la détection (à l'administrateur ou le responsable de la sécurité réseau), les réponses actives peuvent se distinguer en différents types d'actions que sont entreprendre directement une action contre l'intrus, restructurer l'architecture du réseau (isoler le système attaqué, modifier les paramètres d'environnement ayant permis à l'intrusion d'avoir lieu...etc) pour stopper la propagation éventuelle de l'attaque, ou encore surveiller le système attaqué afin de collecter des informations pour tenter de comprendre l'origine de l'intrusion et en identifier l'auteur ou encore la démarche. L'obtention de ces informations relatives à l'intrusion pouvant par ailleurs permettre d'améliorer l'IDS en question.

6 Limites et analyse

Tout comme tous les autres systèmes, les IDS peuvent être la cible d'attaques qui exploitent leurs vulnérabilités, et en particulier de techniques d'évasion consistant à faire accepter des paquets, que l'IDS aurait pu rejeter au système cible.

En effet, en plus d'éventuellement tirer profit des faux positifs – en confectionnant des paquets malveillants afin de générer un très grand nombre de faux positifs et rendre difficile la différenciation entre le trafic provenant de l'attaque et celui des faux positifs -, différentes méthodes ou types d'attaques existent afin de détourner un IDS, et donc la détection éventuelle d'une intrusion.

Parmi ces méthodes, il est en effet possible de citer la coordination des attaques à faible bande passante – permettant de complexifier l'analyse réseau pour l'IDS -, les attaques par proxy ou usurpation – permettant de masquer la source de l'attaque et rendre sa détection davantage difficile -, les dénis de service ou encore la saturation de la mémoire – permettant de rendre l'IDS incapable d'analyser correctement les événements, les IDS utilisant de la mémoire volatile afin de stocker les états des événements ainsi que les données utiles à la détection -, mais également la *fragmentation*, l'*obfuscation* ou encore le *flooding*, abordées dans les paragraphes suivants.

6.1 Fragmentation

La fragmentation consistant à diviser un paquet en paquets plus petits pour ensuite être réassemblés par le destinataire au niveau de la couche IP avant de les transmettre à la couche Application, il est en effet possible de contourner la détection d'une attaque par un IDS, en envoyant des paquets fragmentés afin de passer inaperçu.

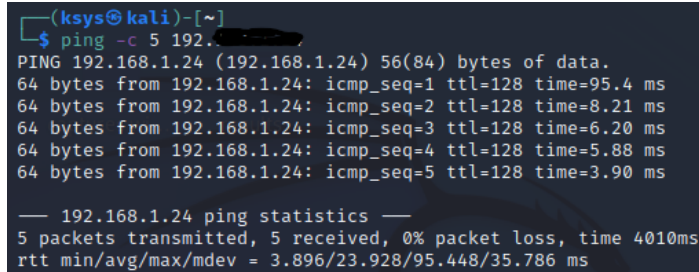
Un IDS pouvant disposer d'un système de réassemblage des paquets, la fragmentation peut en effet être utilisée comme vecteur d'attaque ou de détournement, en particulier lorsque les temps d'arrêts (timeouts) diffèrent entre le l'IDS et l'hôte destinataire différent.

Par exemple, il est possible de supposer que si le timeout d'un fragment est de t_i au niveau de l'IDS et t_c sur le système cible, avec $t_i < t_c$, les paquets envoyés par un attaquant peuvent en effet échapper à la détection, en attendant un temps t_f avec $t_i < t_f < t_c$ avant l'envoi du paquet suivant, permettant ainsi d'arrêter le réassemblage des paquets chez l'IDS qui le supprimera, tout en permettant le réassemblage par le système cible.

De même, pouvant être mis en œuvre avec l'option `-f` de `nmap`, d'autres attaques reposant sur la fragmentation de paquets peuvent être mises en œuvre, afin d'échapper à la détection en masquant les attaques, telles que le chevauchement de la fragmentation, l'écrasement et les délais d'attente.

6.2 Flooding

Consistant à envoyer une grande quantité de données inutiles dans un réseau afin de le rendre inutilisable en saturant par exemple sa bande passante ou en rendant indisponible les machine du réseau, une attaque de type flooding peut être mise en œuvre afin de détourner un IDS, permettant ainsi de submerger l'IDS, afin de masquer toute activité anormale et rendre par conséquent difficile la détection de paquets malveillants dans cette énorme quantité de trafic.



```
(ksys@kali)-[~]
$ ping -c 5 192.168.1.24
PING 192.168.1.24 (192.168.1.24) 56(84) bytes of data.
64 bytes from 192.168.1.24: icmp_seq=1 ttl=128 time=95.4 ms
64 bytes from 192.168.1.24: icmp_seq=2 ttl=128 time=8.21 ms
64 bytes from 192.168.1.24: icmp_seq=3 ttl=128 time=6.20 ms
64 bytes from 192.168.1.24: icmp_seq=4 ttl=128 time=5.88 ms
64 bytes from 192.168.1.24: icmp_seq=5 ttl=128 time=3.90 ms

— 192.168.1.24 ping statistics —
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 3.896/23.928/95.448/35.786 ms
```

Figure 8: Utilisation de la commande ping.

Plusieurs attaques de types flooding peuvent se distinguer. Reposant sur l'ICMP (Internet Control Message Protocol) - étant un protocole de niveau 3, permettant le contrôle des erreurs de transmission n'étant pas gérées par le protocole IP -, une attaque ping flooding peut être mise en œuvre en utilisant la commande ping, utilisant en effet une requête ICMP Request à laquelle doit être retournée une réponse Reply afin de tester l'accessibilité à un système au travers d'un réseau IP.

En effet, cherchant à provoquer la surcharge d'un ordinateur cible avec des paquets « Echo Request » ICMP, le ping flooding consiste en l'envoi de paquets Echo Request par vague machine de la victime par l'attaquant, auxquels cette dernière répond par des paquets Echo Reply.

Reposant sur ce même principe, une attaque SYN flooding peut être mise en œuvre au travers de l'utilisation de l'outil nmap et du protocole TCP, reposant une nouvelle fois sur l'envoi de paquets SYN et ACK.

6.3 Encryption

Offrant un certain nombre de services de sécurité, tels que la confidentialité, l'intégrité et la confidentialité des données, le chiffrement peut également être utilisée comme technique d'évasion, afin d'éviter la détection par un IDS d'un paquet malveillant.

Utilisant cette technique, il est en effet possible pour l'attaquant de chiffrer la charge utile du paquet servant d'attaque, ne la rendant ainsi déchiffrable que pour le destinataire et non pour l'IDS.

L'IDS ne pouvant détecter une intrusion éventuelle s'il n'interprète pas le trafic chiffré, l'examen des paquets, l'examen du trafic chiffré rend ainsi difficile la détection des attaques par les détecteurs, et peut ainsi être utilisé comme technique d'évasion afin d'éviter la détection.

7 Conclusion

Cherchant à répondre à cette question de l'intrusion et des enjeux liés à leur détection - soulevée au travers de l'exemple des scan des ports -, la notion d'IDS a été abordée dans le cadre de ce projet, présentant leur rôle et leurs différents types, ou encore leur mise en œuvre afin de répondre à ces enjeux.

Reposant ainsi sur cette notion, un modèle permettant de détecter une intrusion a également été présenté, classant un type de connexion selon 5 catégories - soit une connexion de type normale et 4 types d'attaques - avec une accuracy de 0.9766 et un kappa score de 0.9677.

Enfin, succédant à la détection de l'intrusion par le système de détection, les mesures possibles de prévention ont également été présentées, ainsi que les limites et risques d'une telle mise en œuvre, ou les possibles détournements d'un tel système, permettant ainsi de répondre à cette question "Comment détecter une intrusion ?" ainsi qu'à ses enjeux.

References

- [1] *Cybersécurité Analyser les risques, mettre en œuvre les solutions*, 6ème ed., Solange Ghernouati, ed. Dunod
- [2] *Hacking Interdit*, 7ème édition, Micro Application, Alexandre J.Gomez Urbina
- [3] *Les bases du Hacking*, Patrick Engebretson, ed. Pearson
- [4] *Hacking The Art of exploitation*, Jon Erickson, ed.
- [5] *Wireshark para novatos*, Anmol K Sachan (documentation Kali)
- [6] *Wireshark Analysis of various Nmap scans*, Aditya Srivastava (documentation Kali)
- [7] *Survey of intrusion detection systems : technique, datasets and challenges*, Ansam Khraisat, Iqbal Gondal, Peter Vamplew and Joarder Kamruzzaman, 2019
- [8] *A detailed analysis of the KDD Cup 99 data set*, Mahbod Tavallae, Ebrahim Bagheri, Wei Lu, 2009
- [9] *Intensive Preprocessing of KDD Cup 99 for Network Intrusion Classification Using Machine Learning Techniques*, Ibrahim Obeidat, Nabhan Hamadneh, Mouhammd Alkasassbeh, mohammad Almseidin,
- [10] *A Statistical Analysis on KDD Cup'99 Dataset for the Network Intrusion Detection System*, Satish Kumar, Sunanda, Sakshi Arora, 2020
- [11] *Intrusion Detection using Machine Learning Techniques : An experimental Comparison*, Kathryn-Ann Tait, Jan Sher Khan. Fehaid Alqahtani, Awais

A Annexe

<i>feature name</i>	<i>description</i>	<i>type</i>
duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	discrete
service	network service on the destination, e.g., http, telnet, etc.	discrete
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
flag	normal or error status of the connection	discrete
land	1 if connection is from/to the same host/port; 0 otherwise	discrete
wrong_fragment	number of "wrong" fragments	continuous
urgent	number of urgent packets	continuous

Table 1: Basic features of individual TCP connections.

<i>feature name</i>	<i>description</i>	<i>type</i>
hot	number of "hot" indicators	continuous
num_failed_logins	number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	discrete
num_compromised	number of "compromised" conditions	continuous
root_shell	1 if root shell is obtained; 0 otherwise	discrete
su_attempted	1 if "su root" command attempted; 0 otherwise	discrete
num_root	number of "root" accesses	continuous
num_file_creations	number of file creation operations	continuous
num_shells	number of shell prompts	continuous
num_access_files	number of operations on access control files	continuous
num_outbound_cmds	number of outbound commands in an ftp session	continuous
is_hot_login	1 if the login belongs to the "hot" list; 0 otherwise	discrete
is_guest_login	1 if the login is a "guest" login; 0 otherwise	discrete

Table 2: Content features within a connection suggested by domain knowledge.

<i>feature name</i>	<i>description</i>	<i>type</i>
count	number of connections to the same host as the current connection in the past two seconds	continuous
	<i>Note: The following features refer to these same-host connections.</i>	
error_rate	% of connections that have "SYN" errors	continuous
rerror_rate	% of connections that have "REJ" errors	continuous
same_srv_rate	% of connections to the same service	continuous
diff_srv_rate	% of connections to different services	continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
	<i>Note: The following features refer to these same-service connections.</i>	
srv_error_rate	% of connections that have "SYN" errors	continuous
srv_rerror_rate	% of connections that have "REJ" errors	continuous
srv_diff_host_rate	% of connections to different hosts	continuous

Table 3: Traffic features computed using a two-second time window.

Figure 9: Features du data set KDDCup99