

Exfiltration de Données Sans Connectivité Réseau via Codes QR, Une Menace Sous-Estimée

NICOLAS, Alain

Responsable pédagogique
UTT :
M. Alain CORPEL

Branche : Mastère Spécialisé EC

Semestre : Printemps 2023

RÉSUMÉ

La prévention des fuites de données est cruciale pour la sécurité des systèmes d'information . Cette thèse souligne le risque sous-estimé posé par l'exfiltration de données via des codes QR, qui sont de plus en plus couramment utilisés dans de nombreuses applications. Bien que l'approche actuelle des Entreprises minimise le risque en raison de la faible bande passante atteinte par ces techniques d'exfiltration, la thèse démontre que des débits de l'ordre de dizaines de kilobits par seconde sont atteignables. Pour atténuer ce risque, il est recommandé d'instaurer des politiques d'accès plus strictes aux webcams et d'explorer les solutions du marché pour surveiller l'affichage des codes QR sur les écrans des collaborateurs et des appareils mobiles de l'entreprise.

Entreprise : BNP PARIBAS

Lieu : PARIS

Responsable : M François THEZE

• Mots clés

- exfiltration
- donnée/ data
- code QR/ code QR
- canal caché -discret / covert channel
- air-gap

Remerciements

Tour d'abord, je tiens à remercier l'Université de Technologie de Troyes (UTT) pour m'avoir accueilli au sein du Mastère Spécialisé, et en particulier à toute l'équipe pédagogique pour la richesse de leurs enseignements et le partage de leur expérience professionnelle.

Mes remerciements les plus sincères vont à notre directeur de formation, M. Reza EL GALAI, pour sa proximité et son soutien constants, ainsi qu'à M. Alain CORPEL pour son suivi de stage.

Je remercie également BNP PARIBAS pour avoir pris en charge cette formation et pour m'avoir permis de me former pendant mon temps de travail.

Un grand merci à Dominique FLEURY et à Jean-Christophe CUQ, respectivement responsable du département IT PS et RSSI ITPS, qui m'ont fait confiance en validant mon intégration avant le début de cette formation.

Je tiens à exprimer ma gratitude envers Francis THEZE qui m'a suggéré l'idée de travailler sur l'exfiltration de données via des canaux discrets. Apprendre à ses côtés est un réel plaisir, et j'apprécie grandement sa capacité de synthèse et sa faculté à me recentrer lorsque je m'égare dans le vaste domaine de la cybersécurité que je découvre. Merci également pour sa relecture attentive.

Je tiens à remercier chaleureusement Pascal RAUX pour ses conseils précieux et pour m'avoir laissé la liberté de me consacrer à cette thèse pendant mon stage. J'ai hâte de travailler sur les sujets appSec* passionnantes que nous avons identifiés une fois cette thèse soutenue.

Un remerciement particulier à ma fille Maëlys qui m'a accompagné dans la création de la vidéo de démonstration de faisabilité.

Enfin, remerciements spéciaux à mon épouse, Sandrine, pour sa compréhension, son soutien et ses encouragements constants, ainsi qu'à mes amis pour leur patience et leur compréhension pendant cette période intense.

Glossaire

Les termes présentes dans le texte sont suffixés par un caractère *. Vous pouvez vous référer au glossaire suivant pour obtenir leur définition.

Termes	Définitions
AppSec	ou sécurité des applications, fait référence aux mesures et procédures intégrées dans le cycle de développement logiciel (SDLC) pour protéger une application contre les menaces et vulnérabilités de sécurité. Elle englobe les méthodes et outils utilisés pour identifier, corriger et protéger les applications contre les attaques de sécurité. Cela inclut des pratiques comme l'évaluation du code source, la détection et la réparation des vulnérabilités, et l'intégration de mesures de sécurité tout au long du cycle de développement. L'objectif est de garantir que les applications fonctionnent de manière sécurisée et résistante aux attaques, tout en protégeant les données et les fonctionnalités sensibles.
bit	Le bit est atomique, la plus petite unité de stockage informatique qui prend la valeur 0 ou 1
bit per second	Le bps est une unité de mesure informatique usuellement utilisée dans les mesures de débits de liaisons informatiques.
Byte	Un octet soit 8 bits
Code QR	Un code QR (en anglais QR code), en forme longue quick response code (« code à réponse rapide »), est un type de code-barres à deux dimensions constitué de modules-carrés noirs disposés dans un carré à fond blanc. Ces points définissent l'information que contient le code. Ce dessin, lisible par machine, peut être visualisé sur l'écran d'un appareil mobile ou imprimé sur papier.
Critères Communs	Les critères communs (CC) sont un ensemble de normes (ISO 15408) internationalement reconnu dont l'objectif est d'évaluer de façon impartiale la sécurité des systèmes et des logiciels informatiques. Également dénommés Common Criteria, ce référentiel est né d'un partenariat entre le Canada, les États-Unis et l'Europe.
Data Loss Prevention	Une approche de la cybersécurité visant à protéger les données sensibles et confidentielles d'une organisation, en prévenant leur divulgation, leur vol ou leur perte non autorisée.
Insider	se réfère généralement à une personne ou une entité qui a accès à des informations sensibles ou confidentielles au sein d'une organisation. Cela peut inclure des employés, des contractuels, des partenaires commerciaux, ou toute autre personne qui a un accès légitime à ces informations. La menace interne, ou menace "insider", est un risque particulier où ces personnes peuvent abuser de leur accès de manière malveillante.
Ko	Kilo-octet
machine learning	Ou apprentissage automatique en français, est un domaine de l'informatique qui donne aux ordinateurs la capacité d'apprendre à partir de données sans être explicitement programmés pour le faire. Dans le machine learning, un algorithme informatique améliore sa performance à accomplir une tâche spécifique au fur et à mesure qu'il est exposé à plus de données liées à cette tâche. Par exemple, un algorithme de machine learning pourrait être utilisé pour apprendre à identifier les e-mails de spam en analysant des exemples d'e-mails de spam et d'e-mails non-spam. Plus l'algorithme voit d'exemples, plus il peut apprendre et mieux il peut accomplir sa tâche.

Moniteur de référence	<p>fait référence à un système ou une application qui surveille les activités d'un hôte, d'un réseau ou d'un système air-gap pour détecter d'éventuelles menaces ou activités suspectes. Voici ce que signifient ces termes :</p> <p>Moniteur de référence basé sur l'hôte : le moniteur est installé directement sur un hôte spécifique (par exemple, un ordinateur ou un serveur) et surveille les activités de cet hôte. Il peut surveiller des éléments tels que les processus en cours d'exécution, les fichiers système, l'utilisation du réseau, etc. Un exemple pourrait être un logiciel antivirus installé sur un ordinateur.</p> <p>Moniteur de référence basé sur le réseau : le moniteur est utilisé pour surveiller l'activité sur un réseau entier. Il peut surveiller le trafic réseau, les tentatives de connexion, les changements dans la configuration du réseau, etc. Un exemple de ceci serait un système de détection d'intrusion réseau (NIDS).</p> <p>Moniteur de référence basé sur l'air-gap : Un système "air-gap" est un système qui est physiquement isolé de tous les autres réseaux ou systèmes pour fournir un niveau de sécurité maximal. Un moniteur de référence pour un tel système devrait être conçu pour surveiller les activités physiques (par exemple, les tentatives d'accès au système) ainsi que les activités du système lui-même. Dans ce contexte, cela pourrait impliquer des mesures de surveillance physique (comme les caméras de surveillance) ainsi que des logiciels de surveillance des activités du système.</p>
Proof Of Concept	Le proof of concept (POC ou « preuve de concept », pour la définition de POC en français) sert à démontrer la faisabilité d'un produit, d'une méthode ou d'une idée. En effet, pour que les parties prenantes et investisseurs donnent suite à votre projet en toute confiance, vous devez prouver le bien-fondé de votre idée.
Rançongiciel	(ou ransomware en anglais) Type de logiciel malveillant qui chiffre les fichiers d'un système informatique et demande une rançon pour les déchiffrer. Les cybercriminels utilisent des rançongiciels pour bloquer l'accès aux fichiers d'un utilisateur et exigent un paiement en échange d'une clé de déchiffrement. Les rançongiciels peuvent se propager par e-mail, sites web infectés ou vulnérabilités logicielles. Les attaques de rançongiciels représentent une menace majeure en cybersécurité, avec des conséquences financières et opérationnelles significatives pour les individus et les organisations.
Trusted Computer System Evaluation Criteria	Les Trusted Computer System Evaluation Criteria, expression abrégée en TCSEC, sont un ensemble de critères énoncés par le département de la Défense des États-Unis qui permettent d'évaluer la fiabilité de systèmes informatiques centralisés. On parle parfois de l'Orange Book (livre orange), d'après la couverture du livre. Il s'agit d'un cahier des charges définissant quatre degrés de sécurité, de A à D ; A étant le niveau le plus sécurisé. En général, le niveau C2 est suffisant.
Tunneling	Le tunneling est une technique de cybersécurité qui permet d'encapsuler et de transmettre des données sécurisées à travers un réseau non sécurisé. Il crée un "tunnel" virtuel qui garantit la confidentialité et l'intégrité des données lorsqu'elles sont transférées d'un point à un autre. Cette méthode est couramment utilisée pour établir des connexions sécurisées à distance, comme les tunnels VPN ou SSH.

Liste des abréviations

Les abréviations présentes dans le texte sont suffixées par un caractère [#]. Vous pouvez vous référer à la liste ci-dessous pour obtenir sa signification.

Abréviation	Signification acronyme
appSec	Application Security
bps	bit per second
CC	Critères Communs
CCPA	California Consumer Privacy Act
DLP	Data (Loss Leak) Prevention
POC	Proof Of Concept
SMC	Small Message Criterion
RSSI	Responsables de la Sécurité des Systèmes d'Information
RGPD	le Règlement général sur la protection des données
TCSEC	Trusted Computer System Evaluation Criteria

Index des figures

Figure 1: Canaux matériels DLP et hors DLP.....	7
Figure 2 : Classification usuelle des méthodes d'exfiltration (selon Giani, 2006).....	8
Figure 3: Modèle général du canal caché « Covert channel » (Carrara, 2016).....	10
Figure 4: Synthèse Acoustic and Light Covert Channel (Carrara, 2016).....	11
Figure 5: Modèles de masques (Assaad, 2019).....	14
Figure 6: Encodage et décodage d'un code QR (Tiwari, 2016).....	14
Figure 7: Structure d'un code QR (Tiwari, 2016).....	15
Figure 8: Aperçu des différentes versions de code QR (Tiwari,2016).....	17
Figure 9: Visuel des 5 types de code QRs.....	18
Figure 10: Schéma de principe de l'expérimentation proposée.....	22
Figure 11: Implémentation matérielle mise en place.....	25
Figure 12: Exemple de code QR « data » produit et structure des données encodées.....	26
Figure 13: Exemple de code QR « ack » produit et structure des données encodées.....	27
Figure 14: Diagramme de séquence du déroulement de l'exfiltration développée.....	28
Figure 15: Exemple d'implémentation de contre mesure contre l'affichage de code QR.....	33

Index des tableaux

Tableau 1: Niveau de correction d'erreur et % de correction d'erreur.....	16
Tableau 2: Même information encodée dans les 4 niveaux de correction d'erreur. (admin, 2011)....	16
Tableau 3: Capacité en datas de la version 40 du code QR en fonction du type de data et du niveau de correction d'erreur (Tiwari, 2016).....	17
Tableau 4: Caractérisation du canal proposé selon la matrice de Carrara.....	24
Tableau 5: Configurations matérielles et logicielles.....	24
Tableau 6: Débits moyen obtenus.....	30

Table des matières

Remerciements	1
Glossaire.....	1
Liste des abréviations.....	1
1. Introduction.....	1
1.1. Importance de la Donnée.....	1
1.2. L'exfiltration de la donnée.....	2
1.3. Objectif de cette thèse professionnelle.....	2
2. Contexte et état de l'art.....	3
2.1. Méthodes d'exfiltration de données.....	3
a) Méthodes courantes.....	3
Utilisation des protocoles de communication légitimes	3
Tunneling :.....	3
Phishing :.....	3
Pharming :.....	4
Social Engineering :.....	4
Shoulder surfing :.....	4
Privilege escalation :.....	4
Botnets :.....	4
Rootkits :.....	4
Spyware :.....	4
Utilisation de périphériques externes :.....	4
Le cas de la Stéganographie :.....	5
b) Méthodes plus atypiques : les canaux cachés.....	6
Signaux Acoustiques :.....	6
Signaux Électromagnétiques :.....	6
Signaux Thermiques :.....	6
Signaux Optiques :.....	6
Signaux Vibratoires :.....	6
c) Proposition de classification.....	7
d) Risques et canal caché.....	10
2.2. La technologie code QR.....	13
a) Histoire et cas d'utilisations.....	13
b) Fonctionnement du code QR.....	13
encodage.....	13
.....	14
décodage.....	14
c) Structure d'un code QR.....	15
d) Correction d'erreur.....	16
e) Tailles, version et capacité.....	17
f) Les types de code QR.....	18
2.3. Risques de cybersécurité potentiels liés à l'utilisation malveillante de la technologie des codes QR.....	19
a) Les différents types d'attaques.....	19
b) Techniques d'exfiltration de données par l'utilisation du code QR.....	20
3. Démonstration de faisabilité.....	21
3.1. Présentation du scénario et du contexte de la démonstration.....	21
3.2. Analyse du modèle selon l'approche de Carrara.....	22
3.3. Description détaillée de la mise en œuvre du Proof of Concept en Python.....	23

a) Configuration matérielle et logicielle.....	23
b) Programme 1 : qrEncoder_OnDeviceEmission_WithSeq.py.....	24
c) Programme 2 : qrAfficher_OnDeviceEmission_WithQrAck.py.....	24
d) Programme 3 :qrDecoder_OnDeviceReception_WithQrAck.py.....	25
3.4. Résultats obtenus, les difficultés rencontrées et les solutions trouvées.....	27
a) Journal de développement (abrégé).....	27
b) Présentation des résultats de l'expérimentation : les débits obtenus.....	28
c) Analyse des résultats et discussion.....	28
4. Détections et contres-mesures.....	29
4.1. Surveillance du poste local.....	29
a) Accès à l'utilisation de la webcam par les applications.....	29
b) Détection et analyse des codes QR affichés sur les écrans des collaborateurs.....	30
5. Conclusion.....	32

1. Introduction

1.1. Importance de la Donnée

Pour la grande majorité des entreprises de taille significative, les données constituent un **actif majeur**.

La protection des données d'entreprise est d'une importance capitale dans le contexte actuel de la digitalisation. Elle doit permettre de garantir la confidentialité, l'intégrité et la disponibilité des informations sensibles appartenant à l'entreprise, ainsi que la conformité aux lois et réglementations en vigueur.

La protection des données d'entreprise est essentielle pour plusieurs raisons :

- ⇒ La confidentialité des informations : Les entreprises collectent et stockent souvent des données sensibles telles que des informations financières, des données clients, des secrets commerciaux et des propriétés intellectuelles telles que du code informatique. La protection de ces données confidentielles est essentielle pour éviter les fuites d'informations et les atteintes à la réputation de l'entreprise.
- ⇒ La conformité réglementaire : Les réglementations telles que le RGPD[#] en Europe et le CCPA[#] localement en Californie imposent des obligations légales en matière de protection des données. Le non-respect de ces réglementations peut entraîner des amendes financières et des dommages à la réputation de l'entreprise.
- ⇒ La gestion des risques : Les cyberattaques et les violations de données peuvent avoir des conséquences financières désastreuses pour une entreprise. Les coûts associés à la remédiation d'une violation de données, tels que les enquêtes forensiques, les notifications aux personnes concernées et les mesures correctives, peuvent être extrêmement élevés. Une bonne protection des données aide à réduire les risques et les impacts financiers potentiels.
- ⇒ La confiance des Clients et partenaires : Les clients et les partenaires commerciaux attachent de plus en plus d'importance à la protection de leurs données personnelles. Une entreprise qui démontre un engagement envers la sécurité des données peut gagner la confiance de ses clients et de ses partenaires, renforçant ainsi sa réputation et sa compétitivité.
- ⇒ L'avantage concurrentiel : Dans un paysage commercial de plus en plus concurrentiel, une solide posture en matière de protection des données peut constituer un avantage concurrentiel pour une entreprise. Les clients sont de plus en plus conscients des risques liés à la protection des données et préfèrent souvent faire affaire avec des entreprises qui accordent la priorité à la sécurité de leurs informations.

1.2. L'exfiltration de la donnée

Cette augmentation de la quantité et de la valeur des données a également ouvert la porte à de nouveaux risques et menaces.

Les entreprises sont confrontées à une multitude de menaces potentielles telles que les cyberattaques, les vols de données, les rançongiciels* et les violations de la vie privée.

Les techniques d'exfiltration de données sont diverses et en constante évolution, allant des attaques de type "man-in-the-middle" aux logiciels malveillants sophistiqués, en passant par l'utilisation abusive de services cloud.

Cependant, toutes ces techniques ont un point commun : elles exploitent généralement la connectivité réseau des systèmes pour transférer les données.

Mais qu'en est-il si cette connectivité réseau n'était pas nécessaire pour opérer une exfiltration de données ?

1.3. Objectif de cette thèse professionnelle

Dans le contexte actuel de l'augmentation de l'adoption de la pratique du télétravail par les employeurs et leurs salariés, cette thèse se propose d'explorer un scénario mettant à l'épreuve les mécanismes traditionnels de prévention des pertes de données (DLP[#]).

Imaginez un collaborateur en fin de mission ou mal-intentionné, un "insider*", qui cherche à exfiltrer des données sensibles, que ce soit du code informatique, des documents sensibles, ou d'autres informations de valeur. Ce collaborateur travaille à distance, en télé-travail, à l'abri du regard direct de ses collègues ou supérieurs, ce qui lui donne plus de liberté pour mener à bien son acte malveillant.

Il dispose de deux ordinateurs portables : le premier est un ordinateur portable dont l'administration est managée par l'Entreprise qui l'emploie et représente la source des informations qu'il cherche à exfiltrer et le second, un ordinateur personnel, qui lui permet de recevoir et de stocker les données exfiltrées. L'objectif du collaborateur est de contourner les systèmes de DLP en place, qui sont généralement conçus pour surveiller et contrôler les transferts de données via les réseaux.

Ici, l'objet vecteur d'information est le code QR, si familier, pour transférer les données sans faire appel à la connectivité réseau.

Cette méthodologie présente une menace significative, car elle permet de contourner les protections DLP.

En développant et en réalisant une démonstration de faisabilité (POC[#]), nous cherchons à comprendre cette menace et à proposer des contre-mesures pour la mitiger.

À travers un état de l'art puis une analyse de cette menace basée sur ma démonstration de faisabilité, ma contribution est de mettre en lumière cette technique d'exfiltration de données, de la

démontrer techniquement, évaluer le débit et de proposer des contre-mesures pour aider les entreprises à se protéger contre elle.

2. Contexte et état de l'art

2.1. Méthodes d'exfiltration de données

L'exfiltration de données se produit lorsqu'il y a une copie, un transfert ou une récupération non autorisés de données d'un serveur ou de l'ordinateur d'un individu (Rouse M., 2013).

Les organisations possédant des données de grande valeur sont particulièrement exposées à ce type d'attaques, qu'elles soient le fait d'acteurs extérieurs ou d'initiés de confiance.

Les menaces internes constituent l'un des facteurs majeurs contribuant à l'exfiltration de données, qu'elles soient causées par des erreurs accidentelles ou des actes malveillants. Les menaces d'initiés malveillants se réfèrent à des individus en position de confiance au sein d'une organisation, qui cherchent délibérément à extraire des données dans le but de nuire à l'entreprise dans leur propre intérêt ou dans celui d'autres personnes.

L'exfiltration de données est une préoccupation majeure pour les organisations de nos jours. Selon une récente étude de McAfee, 61 % des professionnels de la sécurité ont fait face à une violation de données dans leur entreprise actuelle (Sheridan K., 2019). Cette statistique souligne l'importance croissante de prendre des mesures préventives et des mesures de protection pour contrer ces menaces internes et minimiser les risques liés à l'exfiltration de données.

a) Méthodes courantes

Utilisation des protocoles de communication légitimes :

Cette méthode consiste à utiliser des protocoles de communication légitimes, tels que HTTP, DNS, FTP, SSH pour exfiltrer des données en encapsulant les informations sensibles dans les requêtes ou les réponses de ces protocoles. (Cidon, 2018).

Tunneling :

Les attaquants utilisent des techniques de tunneling pour transférer les données hors du réseau, en établissant des tunnels authentifiés et chiffrés tels que des tunnels VPN ou SSH, afin de masquer le trafic de données et de contourner les systèmes de détection. (Brown, 2016)

Phishing :

Technique d'ingénierie sociale où des attaquants se font passer pour une entité légitime afin de tromper les utilisateurs et les inciter à divulguer des informations sensibles, telles que des identifiants de connexion ou des informations financières. (Jakobsson, 2019)

Pharming :

Attaque visant à rediriger le trafic d'un site web légitime vers une fausse version contrôlée par un attaquant. Cela permet de collecter des informations confidentielles telles que les identifiants de connexion. (Cohen, 2017)

Social Engineering :

Méthode utilisée pour manipuler les individus afin d'obtenir des informations confidentielles ou de les persuader d'effectuer des actions indésirables. Cela peut inclure des techniques de manipulation psychologique pour inciter les personnes à divulguer des informations sensibles. (Hadnagy, 2018)

Shoulder surfing :

Technique où un attaquant observe discrètement l'écran d'un utilisateur pour collecter des informations sensibles, telles que des mots de passe ou des informations confidentielles affichées à l'écran. (Roush, 2018)

Privilege escalation :

Méthode consistant à obtenir des priviléges plus élevés sur un système ou un réseau que ceux initialement attribués, ce qui permet d'accéder à des informations sensibles ou de contourner les restrictions de sécurité. (Sepehrdad, 2020)

Botnets :

Réseau d'ordinateurs infectés par des logiciels malveillants et contrôlés à distance par un attaquant. Les botnets peuvent être utilisés pour exfiltrer des données sensibles en utilisant les machines compromises comme points de sortie. (Antonakakis, 2017)

Rootkits :

Logiciels malveillants conçus pour accéder ou contrôler un système de manière furtive, en masquant leur présence et en modifiant les fonctionnalités du système d'exploitation. Les rootkits peuvent être utilisés pour exfiltrer des données de manière invisible. (Choo, 2021)

Spyware :

Logiciels malveillants conçus pour collecter des informations sur un utilisateur ou un système sans le consentement de l'utilisateur. Les données collectées peuvent être exfiltrées à des fins malveillantes. (Eckert, 2020)

Utilisation de périphériques externes :

Les attaquants exploitent des périphériques externes tels que des clés USB, des disques durs externes ou des cartes mémoire pour copier et transporter les données sensibles hors du réseau. (Conti, 2010)

Le cas de la Stéganographie :

Elle constitue une technique intéressante car elle permet de dissimuler des informations sensibles au sein d'autres médias ou supports de communication, tels que des images, des fichiers audio ou des vidéos, sans éveiller les soupçons. Cela peut être utilisé pour exfiltrer des données de manière furtive, en les cachant au sein d'un média apparemment anodin. La stéganographie est une technique qui permet de cacher des informations sensibles au sein d'autres données, telles que des images, des fichiers audio ou des vidéos, sans éveiller les soupçons. Elle peut être utilisée à la fois comme méthode d'exfiltration et de protection de l'information.

Exfiltration de l'information :

Lorsqu'il s'agit d'exfiltrer des données, la stéganographie permet de dissimuler des informations confidentielles au sein de médias apparemment innocents. Par exemple, des fichiers sensibles peuvent être dissimulés dans des images ou des documents apparemment normaux. Cela permet aux attaquants de contourner les mécanismes de détection et de passer inaperçus lors de la transmission des données hors du réseau. Les destinataires autorisés peuvent extraire les données cachées en utilisant une clé ou une méthode spécifique.

Protection de l'information :

La stéganographie peut également être utilisée comme moyen de protection de l'information. En cachant les données sensibles au sein de supports apparemment anodins, elle rend difficile la détection et la compréhension de ces données par des personnes non autorisées. Cela peut être utile pour protéger des informations confidentielles lorsqu'elles sont stockées ou transmises. Par exemple, en utilisant la stéganographie, il est possible de dissimuler des messages ou des fichiers importants dans des médias publics tels que des images sur Internet. (Johnson, 1998)

b) Méthodes plus atypiques : les canaux cachés

Ici, ce sont des canaux de communication dissimulés qui permettent à des utilisateurs ou à des programmes de transférer des données de manière secrète et non autorisée. Les « covert channels » peuvent être utilisés pour exfiltrer des données sans être détectés. (Vasilomanolakis, 2020)

Ici, ces sont les éléments matériels, physiques qui à travers leur fonctionnement produisent des ondes, des vibrations, des changements d'états qui sont utilisés pour encoder et transporter l'information. Ce sont des techniques plus sophistiquées mettant en relation différents capteurs.

Signaux Acoustiques :

Cette méthode exploite les variations acoustiques pour exfiltrer des données, en utilisant par exemple les haut-parleurs et les microphones d'un système pour transmettre les informations sous forme de signaux sonores inaudibles. (Asonov, 2004)

Signaux Électromagnétiques :

Les attaquants peuvent utiliser des techniques d'émission électromagnétique pour exfiltrer des données, en captant les signaux électromagnétiques émis par les dispositifs électroniques lors du traitement des données sensibles. (Kuhn, 2013)

Signaux Thermiques :

Cette méthode exploite les variations de température générées par les composants électroniques lors du traitement de données sensibles pour transmettre les informations à travers des variations thermiques détectables. (Genkin, 2014)

Signaux Optiques :

Les attaquants peuvent utiliser des techniques optiques pour exfiltrer des données, en exploitant par exemple les variations de luminosité d'un écran ou les clignotements d'un voyant LED pour transmettre des informations discrètes. (Backes, 2010)

Signaux Vibratoires :

Cette méthode se base sur la capture des vibrations émises par les appareils lors du traitement des données, en utilisant des capteurs sensibles aux vibrations pour extraire les informations transférées (Xu, 2012)

Comme les systèmes DLP traditionnels ont évolué pour fermer les canaux utilisés par les initiés responsables des fuites de données, l'imagination humaine a trouvé d'autres moyens d'exfiltrer les données. Sur la figure 1, nous pouvons voir les canaux hors DLP encore facilement utilisables.

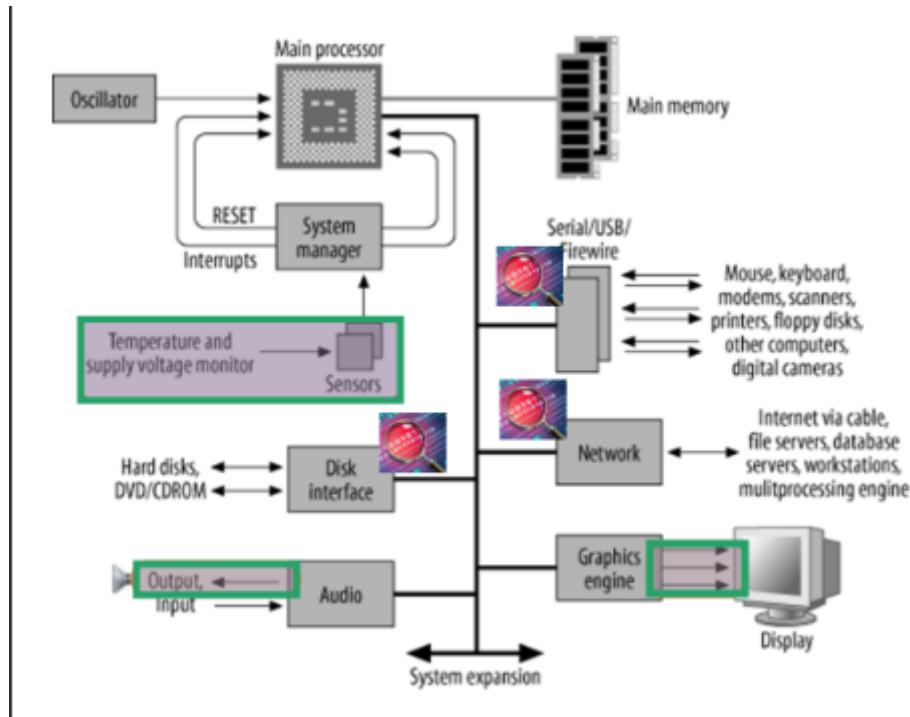


Figure 1: Canaux matériels DLP et hors DLP

c) Proposition de classification

A travers ce panorama des différentes méthodes d'exfiltrations qui se veut le plus exhaustif possible, nous voyons que les possibilités sont énormes et extrêmement variées et que bien souvent une combinaison des techniques est utilisée pour éviter les systèmes de détection DLP.

Plusieurs travaux de classification existent dans la littérature de recherche de cybersécurité.

Les travaux menés par Giani, (2006), propose une discussion sur les critères de gravité des conséquences de l'exfiltration, le profil du protagoniste qui déplace les données (Insider, outsider, hacker, Organisation criminelle) et de la méthode de transfert des données (physique, réseau ou cognitive)

Network	Usually benign	Conventional : Custom	<i>HTTP</i> <i>FTP</i> <i>SMTP</i> <i>SSH</i> <i>Instant messenger</i> : <i>Oracle</i> <i>MySQL</i> <i>Specially software</i>
	Known malicious		<i>Rootkits</i> <i>Botnets</i> <i>Spyware</i> <i>Covert Channels</i> <i>Phishing</i> <i>Pharming</i> <i>MITM</i>
		Attack	<i>Exploits</i> <i>DNS poisoning</i> <i>Directory traversal</i> <i>Privilege escalation</i>
Physical	Usually benign		<i>Printing devices</i> <i>CD, DVD</i> <i>Disk</i> <i>USB</i> <i>Digital Media Players</i>
	Known malicious		<i>Laptop theft</i>
Cognitive			<i>Social engineering</i> <i>Shoulder surfing</i>

Figure 2 : Classification usuelle des méthodes d'exfiltration (selon Giani, 2006)

J'ai recherché plus précisément dans la littérature traitant de l'exfiltration par canal discret sans connectivité réseau. Les travaux menés par Carrara (2016) propose une analyse multi-critères spécifiquement pour les canaux cachés, discrets (Covert Channel) en mode air-gap selon les critères suivants :

- ***Modèle de « bruit de canal » : bruyant ou sans bruit***

La réception de données via un canal caché peut être bruyante ou sans bruit en fonction de divers facteurs environnementaux. Il est important de connaître ou d'estimer le modèle de bruit du canal pour catégoriser et mesurer efficacement le canal caché. De plus, le modèle de bruit peut influencer les techniques de dissimulation de l'information pouvant être utilisées, car la présence ou l'absence de bruit dans le canal peut limiter certaines techniques.

- ***Modèle de couverture de canal : stéganographique ou ouvert***

Un canal caché stéganographique utilise une source de couverture probabiliste pour établir la communication, tandis qu'un canal caché ouvert utilise soit aucune source de couverture, soit une source de couverture complètement déterministe. La principale différence entre ces deux classes de canaux réside dans le fait que la source de couverture suit une distribution aléatoire dans le cas des canaux cachés stéganographiques, tandis qu'elle suit une valeur déterministe dans le cas des canaux cachés ouverts. Par conséquent, les techniques de dissimulation de l'information basées sur la stéganographie ne conviennent pas aux canaux cachés ouverts.

- ***Modèle de contrôle du canal : partagé ou intermédiaire***

Dans le modèle partagé, les données sont directement, simultanément accessibles à l'attaquant et au récepteur, tandis que dans le modèle intermédiaire, les symboles passent par l'attaquant avant d'atteindre le récepteur. Une métaphore utilisée pour illustrer cette distinction est le scénario où les prisonniers communiquent par tapotement sur un radiateur partagé.

- ***Type de modulation : Déetectable, indéetectable ou canal caché sécurisé indéetectable***

Lors de la conception d'un canal caché, il est possible de choisir entre un canal détectable ou un canal indétectable. Le choix du canal indétectable tient compte des capacités d'audit du moniteur de référence du système, tandis que le canal détectable ne fournit aucune défense contre la détection. De plus, le concepteur peut opter pour un canal indétectable sécurisé, qui dépend d'un secret partagé entre l'émetteur et le récepteur, ou pour un canal qui ne l'exige pas. Cependant, l'utilisation d'un canal caché sécurisé indétectable nécessite que Alice et Bob soient en mesure de pré-partager un secret d'une manière quelconque.

- ***Type de modulation : Stockage, synchronisation ou hybride***

Les canaux cachés peuvent être catégorisés en canaux de stockage, de synchronisation ou hybrides, en fonction du temps nécessaire pour transmettre leurs symboles. La capacité d'un canal caché dépend du support utilisé pour communiquer les symboles. Les canaux de stockage cachés ont leur capacité mesurée en bits par utilisation du canal ou en bits par unité de temps, tandis que les canaux de synchronisation cachés ont leur capacité mesurée en bits par unité de temps.

- ***Mode de modulation : Full Duplex, Half Duplex ou Simplex***

Les canaux cachés peuvent offrir une communication bidirectionnelle ou unidirectionnelle. La communication bidirectionnelle peut être en mode full duplex, permettant des échanges simultanés

dans les deux sens, ou en mode half duplex, où les échanges se font dans un seul sens à la fois. Le mode de communication a des implications sur le débit du canal, car les canaux en mode duplex permettent au récepteur d'indiquer quand une retransmission est nécessaire, tandis que les canaux simplex nécessitent l'application de schémas de correction d'erreurs pour assurer une communication sans erreur.

- ***Exploitation cachée : Invasive, semi-invasive ou non invasive***

Les canaux cachés sont souvent accompagnés d'une méthode d'exploitation cachée pour permettre la communication. La classification basée sur l'exploitation cachée permet de mieux comprendre comment le canal est activé et aide à développer des défenses appropriées. Les exploitations cachées peuvent être invasives (nécessitant une modification matérielle), semi-invasives (nécessitant une modification logicielle) ou non invasives (ne nécessitant aucune modification matérielle ou logicielle). Cette classification permet d'étudier la capacité des canaux indépendamment des exploits spécifiques utilisés.

- ***Moniteur de référence *: basé sur l'hôte, réseau ou air-gap***

Les canaux cachés peuvent être classés en fonction des moniteurs de référence qu'ils contournent. Certains mécanismes de canaux cachés peuvent contourner les politiques de sécurité de plusieurs moniteurs de référence. Un modèle de canal caché, plus général, est proposé pour comprendre la dissimulation d'information dans les systèmes de communication numériques

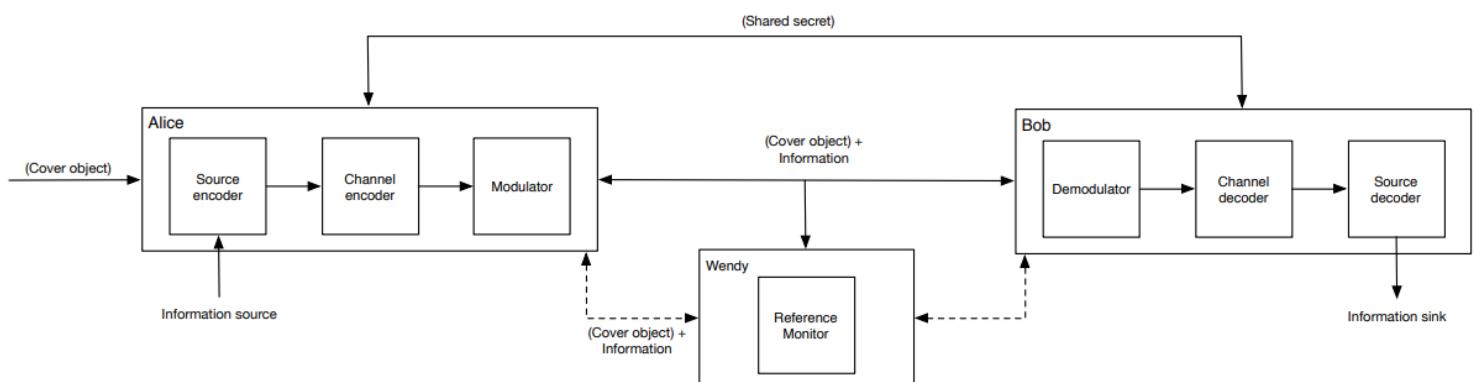


Figure 3: Modèle général du canal caché « Covert channel » (Carrara, 2016)

d) Risques et canal caché

Dans le domaine de la cybersécurité, les canaux cachés sont classifiés en fonction de leur bande passante. Le TCSEC* les classe en trois catégories : les canaux à haute bande passante (plus de 100 bits per second (bps*)), les canaux à basse bande passante (entre 1 et 100 bps) et les canaux acceptables (moins de 1 bps). Le TCSEC recommande de supprimer les canaux à haute bande passante et d'auditer ceux à basse bande passante s'ils ne peuvent pas être supprimés. (Latham, 1986)

Le risque est directement associé au débit permis par la technique d'exfiltration.

Dans la figure suivante, Carrera décrit les caractéristiques et limitations pour le canal « acoustique » et le canal « Lumière »

Covert -(channel)	Acoustic				Light		
Modulator Requirements	Speaker	CPU	Screen	Screen	LEDs	LEDs	Infrared transceiver
Demodulator Requirements	Microphone	Microphone	ALS	Camera	ALS	Camera	Infrared transceiver
Order of Data Rate	Kilobits per second	Data rates not provided in [134]	Bits per second	Hundreds of bits per second	Hundreds of bits per second	Hundreds of bits per second	Megabits per second
Order of Distance	Tens of meters	Tenths of a meter	Meters	Meters	Meters	Meters	Meters
Channel Limitations	<ul style="list-style-type: none"> • Relatively large ambient noise • Relatively large signal delay • Relatively large Doppler effect • Reverberations • Limited range (CPU modulator) • Limited transmission power (CPU modulator) 		<ul style="list-style-type: none"> • Relatively large ambient noise (e.g., sun, room lighting) • Signal does not travel through opaque objects • Limited deployment of hardware in some cases (e.g., Infrared) 				

Figure 4: Synthèse Acoustic and Light Covert Channel (Carrara, 2016)

En revanche, le CC* adopte une approche plus pragmatique en laissant la définition de la sécurité aux exigences spécifiques de chaque produit. Il se concentre sur l'analyse des canaux cachés lors de la certification des produits. Les critères de traitement des canaux cachés sont basés sur la quantité d'informations compromises et la capacité d'exploitation des vulnérabilités du système.(Common Criteria, 2005)

Il est important de distinguer les systèmes à source continue, qui produisent des informations sensibles à un débit élevé, des systèmes à source fixe, qui ont besoin de garder une quantité spécifique d'informations confidentielles. Pour les systèmes à source continue, la capacité du canal est mesurée en fonction de la bande passante, tandis que pour les systèmes à source fixe, le critère SMC est utilisé pour évaluer les risques.

Cependant, il n'existe pas de métrique universellement acceptée pour évaluer les canaux cachés dans les systèmes à source fixe, ni de méthode largement reconnue pour déterminer leur capacité lorsqu'ils sont détectés par audit.(Moskowitz, 1994)

2.2. La technologie code QR

a) Histoire et cas d'utilisations

Les codes Quick Response (QR) ont été créés par Denso Wave, une filiale de Toyota, en 1994 et étaient initialement utilisés pour suivre l'inventaire lors de la fabrication de pièces de véhicules. Ils peuvent être décrits comme un code-barres matriciel bidimensionnel ayant la capacité de stocker des données (Tiwari, 2016).

De nos jours, les codes QR sont de plus en plus populaires car leur précision, leur vitesse de lecture et leur commodité se sont révélées supérieures à celles des codes-barres linéaires unidimensionnels (Narayanan, 2012).

Les codes QR peuvent contenir beaucoup plus de données et peuvent être facilement scannés avec un smartphone. Cela a incité les entreprises à les intégrer dans leurs stratégies de marketing, où ils peuvent être utilisés pour promouvoir l'entreprise et fournir au client des informations supplémentaires sur un produit ou un service.

Les applications des codes QR sont nombreuses et sont utilisées dans plusieurs industries car elles peuvent facilement encoder une URL, un numéro de téléphone, une adresse e-mail, du texte, et même donner accès à un réseau Wi-Fi.

b) Fonctionnement du code QR

Le système de codes QR est composé d'une procédure d'encodage et de décodage. La procédure d'encodage consiste à convertir du texte en utilisant l'un des quatre modes disponibles : alphanumérique, numérique, octets ou kanji, en une séquence de bits (qui sont des 1 et des 0).

encodage

Lorsque le texte est encodé en utilisant l'un des quatre modes, une séquence de bits est construite qui est divisée en mots de code de données de 8 bits de long. Des mots de code de correction d'erreur sont ensuite générés, qui sont correctement ordonnés aux côtés des mots de code de données et sont ensuite placés de manière spécifique dans la matrice 2D du code QR.

Ensuite, le masquage de données est utilisé pour alterner la couleur de chaque module de noir à blanc ou vice versa pour permettre aux scanners de lire facilement le code QR (les modules sont les motifs carrés noirs et blancs qui composent le code QR). Il existe 8 modèles de masque qui peuvent être choisis (Garg, 2015).

Les pixels d'information de version et de format sont ensuite ajoutés au code QR dans les zones qui ne sont pas occupées par les modules de données.

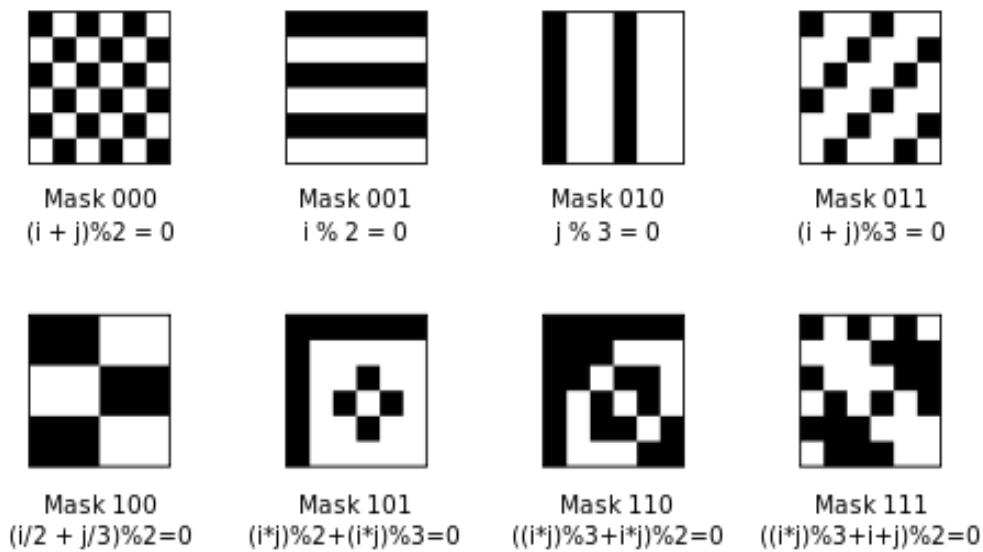


Figure 5: Modèles de masques (Assaad, 2019)

décodage

Lorsque le code QR est scanné, leur arrangement se traduit de nouveau en la forme originale des données, c'est le processus de décodage (Scott, 2020). Cela se produit lorsque le scanner analyse les modules blancs et noirs comme un tableau 2D composé de bits de 1 et de 0. Les informations de version et de format sont déterminées, puis le motif de masquage est libéré. Les mots de code de correction d'erreur et de données sont restaurés et toute erreur est corrigée à l'aide des mots de code de correction d'erreur. Enfin, les données sont décodées en utilisant le mode qui a été utilisé au départ pour encoder les données.

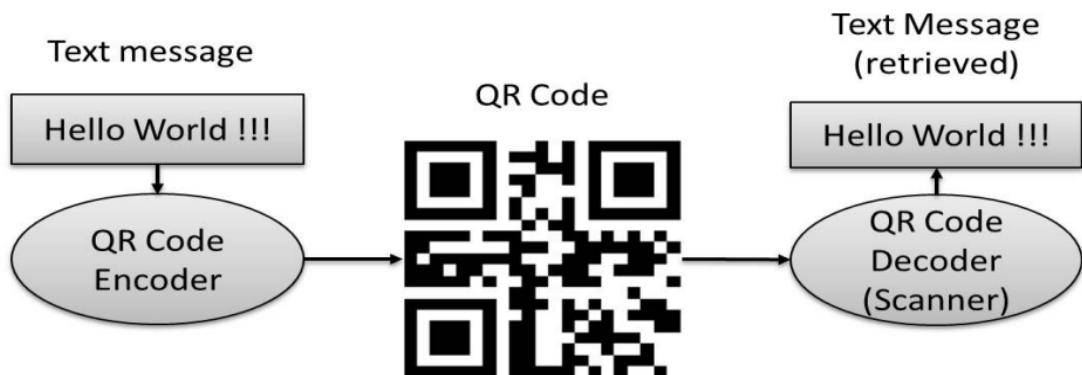


Figure 6: Encodage et décodage d'un code QR (Tiwari, 2016)

c) Structure d'un code QR

La structure d'un code QR comprend :

- Un motif de synchronisation : Permet au décodeur de déterminer la largeur d'un module de données unitaire.
- Une zone de repos : Il s'agit de l'espace vide qui entoure le code QR et qui permet de l'identifier par rapport à son environnement.
- Un motif d'alignement : Ce motif permet de décoder le code QR sous tous les angles, en corrigeant la distorsion du code QR. Les codes QR de version 1 n'ont pas de motif d'alignement.
- Un motif de position : Il y a 3 motifs de position situés sur les codes QR qui aident les décodeurs à indiquer la bonne direction du code QR.
- Données et correction d'erreur : Stocke les informations du code QR et les mots de code de correction d'erreur.
- Un motif de version : Ce motif identifie la version du code QR.
- Format : Ce motif contient des informations sur la correction d'erreur et le masque de données.
- Séparateurs : Il s'agit d'une zone d'espace blanc qui sépare le motif de position de la zone d'encodage. La zone d'encodage étant la zone qui contient les informations de format, la version, **et les données**.

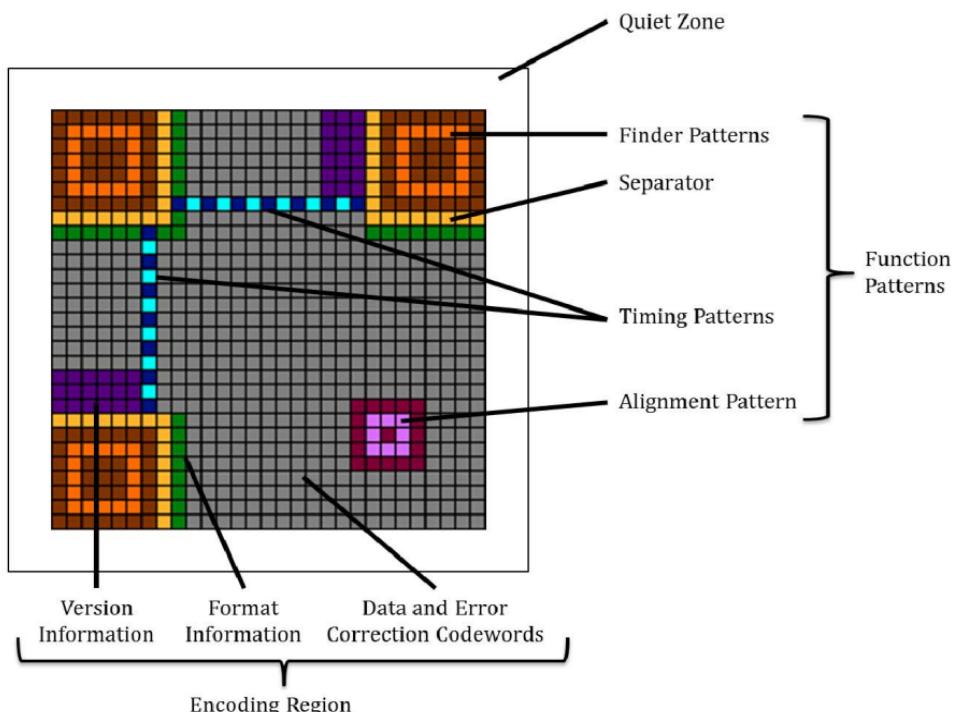


Figure 7: Structure d'un code QR (Tiwari, 2016)

d) Correction d'erreur

Les codes QR contiennent une fonction de correction d'erreur qui leur permet d'être lus même si le code QR est endommagé ou sale.

Les bits de données que nous encodons sont utilisés pour créer des mots de code de correction d'erreur. Cela est réalisé en utilisant les codes de Reed-Solomon, qui sont des codes algébriques qui effectuent une correction d'erreur vers l'avant. Des blocs de données numériques sont pris par un encodeur Reed-Solomon et ajoutés ensemble avec des bits supplémentaires.

Chaque bloc est traité par le décodeur Reed-Solomon et une tentative est faite pour récupérer les données originales et corriger les erreurs trouvées.

La catégorie et le nombre d'erreurs qu'il tente de corriger sont basés sur les propriétés du code de Reed-Solomon utilisé (Anon, 2012). Il existe 4 niveaux différents de correction d'erreur que les codes QR peuvent utiliser :

S No.	Error-Correction Level	Approximate Amount of Correction
1.	L	7%
2.	M	15%
3.	Q	25%
4.	H	30%

Tableau 1: Niveau de correction d'erreur et % de correction d'erreur

Chaque niveau ajoute une quantité différente de données de sauvegarde en fonction du niveau de dommages que le code QR est censé subir.

Plus le pourcentage de correction d'erreur utilisé est élevé, plus le code QR devient grand car davantage de rangées et de colonnes sont ajoutées pour soutenir l'ajout des données de sauvegarde (admin, 2011).

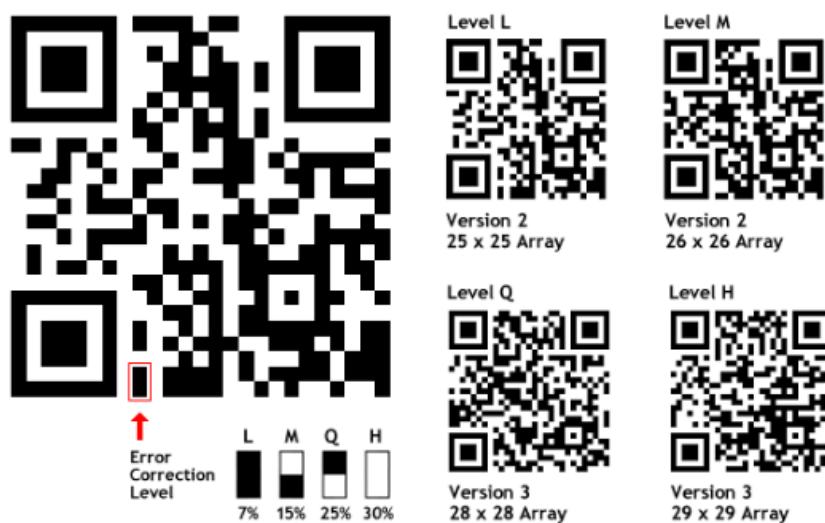


Tableau 2: Même information encodée dans les 4 niveaux de correction d'erreur. (admin, 2011)

e) Tailles, version et capacité

La taille d'un code QR varie car il existe différentes versions qui peuvent être créées. Ces versions vont de 1 à 40.

Chacune de ces versions a un nombre contrasté de modules. Les versions 1 se composent de 21 rangées de pixels par 21 colonnes de pixels, tandis que le plus grand code QR possible est la version 40 qui se compose de 177 rangées de pixels par 177 colonnes de pixels.

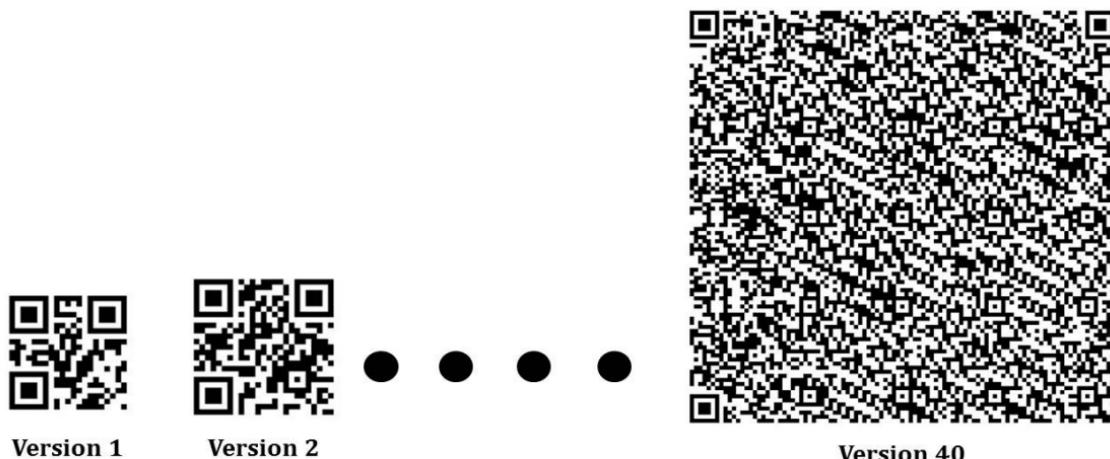


Figure 8: Aperçu des différentes versions de code QR (Tiwari, 2016)

Avec le plus grand code QR possible composé de 177x177 modules, cela signifie qu'il peut comporter jusqu'à 31 329 carrés qui peuvent encoder 3 Ko de données. Cela donne à chaque mode un nombre maximal de caractères possibles qui peuvent être encodés.

Une augmentation du niveau de correction d'erreur réduit la quantité de données utiles qui peuvent être encodées dans le code QR. Voici un tableau récapitulatif pour la version 40, celle que j'ai utilisée dans mon POC.

Version	Modules	ECC Level	Data Bits (mixed)	Numeric	Alpha-numeric	Binary	Kanji
40	177x177	L	23,648	7,089	4,296	2,953	1,817
		M	18,672	5,596	3,391	2,331	1,435
		Q	13,328	3,993	2,420	1,663	1,024
		H	10,208	3,057	1,852	1,273	784

Tableau 3: Capacité en datas de la version 40 du code QR en fonction du type de data et du niveau de correction d'erreur (Tiwari, 2016)

f) Les types de code QR

Il existe 5 types de codes QR qui peuvent être construits (Mathuria, 2017).

code QR Modèle 1 & Modèle 2 : Le code QR original est le modèle 1, avec sa plus grande version étant de 73 x 73 modules. Alors que le modèle 2 est la version mise à jour du modèle 1 qui est ce qui est utilisé aujourd'hui avec sa plus grande version possédant 177 x 177 modules.

Micro code QR : Ce code QR n'a qu'un seul motif de position, ce qui le rend nettement plus petit que les autres codes QR avec sa version maximale n'étant que de 17x17 modules, ce qui le rend approprié pour être imprimé sur de petits espaces.

SQRC : Un code de réponse rapide sécurisé apparaît comme un code QR normal mais possède la fonctionnalité de stocker des informations confidentielles de manière sécurisée à partir de n'importe quel appareil utilisé pour scanner le code QR.

rMQR : Le code répond au besoin d'imprimer dans des espaces étroits où le code QR conventionnel ne pouvait pas être imprimé, et de stocker plus d'informations que possible avec le Micro code QR.

Frame QR : Ce type de code QR a une zone de canevas au centre qui peut être utilisée à des fins de marketing et de promotion car des dessins et des données peuvent être placés dans cette zone (Anon., 2019).



e 9: Visuel des 5 types de code QRs

Figur

2.3. Risques de cybersécurité potentiels liés à l'utilisation malveillante de la technologie des codes QR.

a) Les différents types d'attaques

Les codes QR sont largement utilisés dans divers domaines, tels que le marketing, la publicité, le suivi des produits, les billets d'événements, les cartes de visite électroniques, les paiements mobiles, etc. Leur popularité est due à leur capacité à stocker des informations de manière compacte et à leur facilité d'utilisation grâce aux smartphones et aux applications de lecture de code QR.

L'ère du Covid-19 lui aura permis un regain d'utilisation en 2020. Les règles sanitaires préconisant de toucher le moins de surface possible afin de limiter la propagation du coronavirus, le code QR se révèle comme un moyen pratique de donner accès à des données « à distance » (Hue, 2020)

Les smartphones actuels permettent de flasher facilement les codes QR soit avec une application dédiée soit avec la lecture directement intégrée dans la caméra.

L'augmentation des usages liés à cette technologie attire les cybercriminels qui remplacent des codes légitimes par des codes malveillants (Marchand , 2020) permettant d'opérer les types d'attaques suivants :

- Phishing : Les codes QR malveillants peuvent être utilisés pour diriger les utilisateurs vers de faux sites web conçus pour voler des informations sensibles. Les attaquants peuvent inciter les utilisateurs à fournir leurs identifiants de connexion, leurs informations bancaires ou d'autres données personnelles via des pages web frauduleuses.
- Malware : Les codes QR peuvent être utilisés comme vecteurs de distribution de malwares. En scannant un code QR compromis, un utilisateur peut involontairement télécharger et installer un logiciel malveillant sur son appareil. Ce logiciel peut permettre à l'attaquant de prendre le contrôle de l'appareil, d'accéder à des données sensibles ou de surveiller les activités de l'utilisateur.
- Attaques de paiement : Les codes QR peuvent être manipulés pour effectuer des paiements frauduleux. Les attaquants peuvent remplacer un code QR légitime par un code malveillant, conduisant ainsi à un transfert d'argent non autorisé depuis le compte de l'utilisateur vers celui de l'attaquant.
- Ingénierie sociale : Les codes QR peuvent être utilisés dans des attaques d'ingénierie sociale pour tromper les utilisateurs et les inciter à effectuer des actions indésirables. Par exemple, un code QR malveillant peut diriger les utilisateurs vers une fausse promotion ou une fausse loterie, les incitant à partager des informations personnelles ou à effectuer des paiements.
- Vulnérabilités des lecteurs de codes QR : Les lecteurs de codes QR peuvent présenter des vulnérabilités de sécurité, notamment des failles de lecture ou des erreurs de traitement. Les attaquants peuvent exploiter ces vulnérabilités pour exécuter du code malveillant sur les appareils des utilisateurs ou pour collecter des informations sensibles.

b) Techniques d'exfiltration de données par l'utilisation du code QR

Les codes QR peuvent contenir des informations sensibles telles que des informations d'identification personnelle, des numéros de carte de crédit ou des données confidentielles. Si ces codes QR sont interceptés ou exploités de manière malveillante, ces informations peuvent être divulguées et utilisées à des fins nuisibles.

Cependant, ce scénario se situe dans une utilisation qui est proche de l'usage habituel du code QR.

Mais si maintenant, nous nous plaçons dans l'optique d'utiliser le code QR comme un vecteur transporteur d'information et que nous réfléchissons à comment exfiltrer un maximum de données et ce dans le scénario d'un collaborateur qui veut exfiltrer des types de fichiers divers et variés sans se faire détecter par les solutions DLP en place (filtrage proxy, blocage fonctionnalité USB MassStorage, filtrage e-mails et sorties imprimantes) .

Comme nous l'avons vu, la capacité maximale de donnée que l'on peut insérer dans un code QR est environ 4 Ko*. Il apparaît donc évident qu'il va falloir générer une multitude de codes QR pour déplacer ces données. On pourrait utiliser l'image d'un film ou d'une vidéo de code QR comme un film est une séquence d'images ordonnancée.

Un article traitant de l'utilisation de la stéganographie de messages chiffrés à l'intérieur de code QR valides (Alajmi, 2020) décrit l'utilisation du code comme conteneur de donnée permettant de dissimuler de l'information utile (charge utile) dans une autre information.

Ici, le sujet n'est pas vraiment un sujet de stéganographie car nous ne dissimulons pas la charge utile dans un autre message, nous sommes à la frontière. Le cas d'usage n'exige pas de se dissimuler car nous sommes dans un contexte de télétravail, totalement libre de faire ce que l'on veut.

La recherche sur la façon de séparer les données des systèmes sécurisés peut inspirer des systèmes de protection contre les fuites de données pour empêcher les vulnérabilités révélées d'être exploitées. Le hacker va rechercher des moyens de séparer les données pour trouver les failles.

La plupart de ces exploits reposent sur des logiciels malveillants qui compromettent initialement la machine cible.

Devant l'absence de travaux sur une méthode clairement décrite d'exfiltration de données au moyen du conteneur code QR, j'ai mis en place une méthodologie pour construire un Proof-of-concept que nous allons maintenant aborder.

3. Démonstration de faisabilité

3.1. Présentation du scénario et du contexte de la démonstration

« En entreprise, je suis affecté à un service IT de développement informatique délivrant des solutions de paiements. Les solutions de paiements sont aujourd’hui un secteur hyper-concurrentiel où l’innovation bat son plein. Je vais quitter l’entreprise mais je souhaite absolument emporter le code sur lequel j’ai passé des heures et quelques documents internes fort intéressants. J’ai conscience que ce n’est pas autorisé mais si je me débrouille bien, personne ne le saura !».

Le protagoniste principal de l’attaque qui va manipuler et exfiltrer les données est un collaborateur de l’entreprise, disposant d’un matériel normal et d’un accès normal non-augmentés tous deux (non administrateur), souhaitant contourner le système DLP.

Il va exécuter l’exfiltration en situation de télé-travail : il dispose ainsi de toute liberté pour installer ses ordinateurs portables face à face et disposer d’une webcam usb pour une captation plus aisée des codes QR « datas ».

Pour que cette démonstration soit pertinente et soit le reflet de la menace présentée comme sous-estimée, je me suis fixé comme impératif

- de ne pas modifier la configuration standard du matériel fourni aux collaborateurs
- la démonstration doit fonctionner sur les PC windows ainsi que sur les Macintosh.

Les PC portables Entreprise (Mac ou Windows) disposent d’une webCam intégrée.

Ils sont la source des informations qui vont être exfiltrées.

La méthode d’exfiltration doit être accessible à n’importe quel collaborateur qui dispose d’un environnement logiciel standard pour ce rôle : runtime python et IDE (ex PyCharm) pour exécuter le code développé. Le logiciel PyCharm est disponible au téléchargement sur le magasin de logiciel Entreprise.

Du côté poste récepteur des informations, celui-ci est un pc linux non administré/non managé et dont je suis administrateur. Pour des raisons de commodités et d’expérimentation, j’utilise une webcam externe connectée par câble USB pour capter les codes QR sur l’écran Source de la donnée.

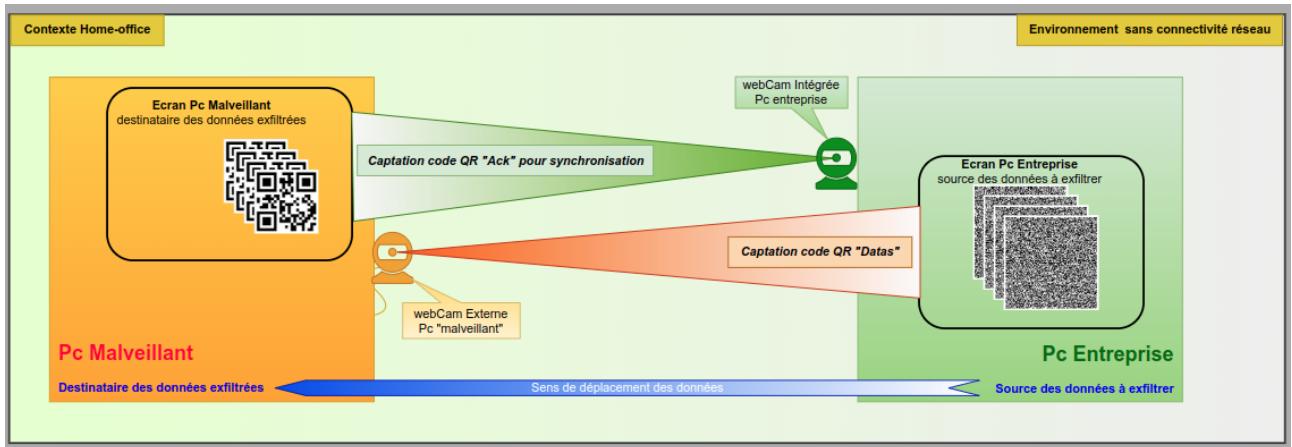


Figure 10: Schéma de principe de l'expérimentation proposée

3.2. Analyse du modèle selon l'approche de Carrara

Afin de mieux caractériser le type de canal caché que je suis en train de mettre en place, j'ai trouvé intéressant de renseigner sa proposition de matrice.

POC MS EC exfiltration par code QR sans utilisation de la connectivité réseau	
Critère	Catégorisation
bruit du canal ?	génére des écritures fichiers .png sur le disque et affichage en séquence sur l'écran mais dans un contexte de télé-travail
Couverture canal ?	pas d'utilisation de stéganographie, utilisation d'encodage base_45 puis fragmentation de la donnée
contrôle du canal ?	contrôle du canal intermédiaire par les programmes pour la synchronisation, envoi en séquence ordonnée
Déetectable ?	l'affichage des codes QR rend visible l'exécution mais dans un contexte télé-travail, l'exécution n'est pas détectable
Stockage, synchronisation ou hybride ?	modulation synchronisée et mesure en bits par unité de temps
sens de communication ?	la version du poc présentée est une communication bi-directionnelle dans un seul sens à la fois : half-duplex
mode exploitation ?	non-invasive
moniteur de référence contourné ?	basé sur l'hôte

Tableau 4: Caractérisation du canal proposé selon la matrice de Carrara.

3.3. Description détaillée de la mise en œuvre du Proof of Concept en Python

Le code est disponible sous https://github.com/AL1NICOLAS/These_MS_EC

Trois programmes utilisant le langage python constituent la solution permettant de démontrer qu'il est possible d'exfiltrer des données de toutes natures depuis un appareil source vers un appareil réceptrice sans utiliser la connectivité réseau en utilisant le conteneur code QR comme vecteur de données.

a) Configuration matérielle et logicielle

PC	PC Entreprise Source des données exfiltrées		PC Malveillant Destinataire des données exfiltrées
	PC Windows Entreprise	PC Macintosh	PC Linux
Configuration matérielle :	Puce Intel I7-8665U Ram 32Go SSD 500 Go WebCam intégrée HP HD luxvisions	Puce M1 pro Ram 16Go SSD 500 Go WebCam intégrée HD FaceTime	Puce Intel® Core™ i7-10610U × 8 Ram 32Go SSD 1 To WebCam externe : Trust- Tyro Full HD - 1920*1080 - Auto-focus - Max FrameRate : 30 fps - sur trépied
Configuration logicielle	OS Windows 10 Entreprise 21H2 PyCharm 2019.2 (Community Edition) Python 3.10	OS MacOS Ventura 13.4.1 PyCharm 2023.1 (Community Edition) Python 3.10	OS Ubuntu 23.04 PyCharm 2023.1 (Community Edition) Python 3.10

Tableau 5: Configurations matérielles et logicielles

b) Implémentation matérielle réalisée



Figure 11: Implémentation matérielle mise en place.

c) Programme 1 : qrEncoder_OnDeviceEmission_WithSeq.py

Exécution de ce programme sur l'appareil « Entreprise » source des datas à exfiltrer.

Ce programme encode des datas à exfiltrer sous forme d'une collection de code QRs. Le collaborateur malveillant rassemble l'ensemble des documents dans le répertoire "`./in_files_to_exfiltrate`". Après avoir constitué une archive qui ensuite est compressée "`payload.tar.gz`", le binaire du zip est encodé en base_45. Le payload base_45 est découpé en partie de 4270 caractères afin de générer autant de code QRs que nécessaire. Chaque code QR généré est flanqué en début de data de l'information du numéro de la séquence / le nombre total de code QRs produits.

Ex : 1/6 suivi d'un séparateur `%%::%%` suivies des données base_45

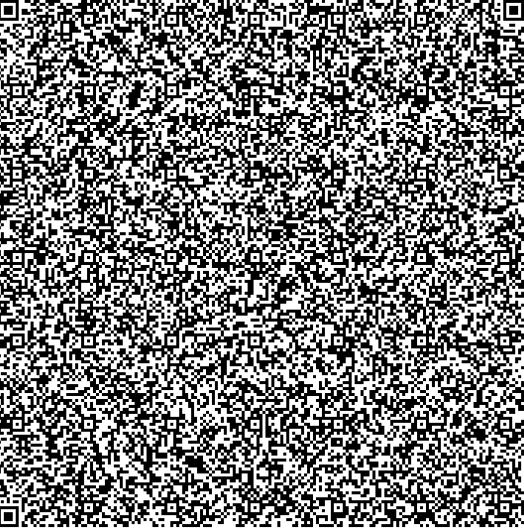
Code QR « data »	Données encodées
	<p>1/6%%::%%K:3V01MCCWZL2H0B\$CP+8TVDBPE3WE0IEL2UO71-ICWRU%5JAPLT87SC9APL*F:-NE3R.L9F OPHJN0DOSD.7N.PMBY4:BIBE9BP4AU2*E5OWGXOP\$71E90//G:YP4 ZUN4KAFU/XFFGWCDWD.NU7TSUDUNXBXJ7ELNCEVE5QPOEZ8H* V8P8G*-R424:6USCOL2WR32-VIJQTC%G- F98GNH9LLDKR36NB2U8I2AS+4S-O3AI0%4A76K5NIL6C/ S3TK0TE0Q*IT9G.CC*PMG32/B7+CDUBBDJQ/ 4WF5KC5W3+D7+JL+9/ 04-G+JP88N7CO1S70JBK%8Q79- B9EV9.3MW54Q080VLVVCL79*G18W9OESS\$ILCW OG:JSD+1L6E*90*ZKKMGZWHUS8RB8O60:8NFGW %9172IN94Y2IZ22S29LCW/-V/+IA5W/- V0LG+:SCDFE7J:EIKHUX*A7IU*FS0/ H+S1O7FMDW9850XUNHD1DH064V74H3RL*LG+LK+LU34N36MZB 42F9959M9\$3473QY:2:Q2CIC\$IKFYE 84VD2K6UYP3QM1...</p>

Figure 12: Exemple de code QR « data » produit et structure des données encodées

d) Programme 2 : qrAfficher_OnDeviceEmission_WithQrAck.py

Exécution de ce programme sur l'appareil « Entreprise » source des datas à exfiltrer.

Ce programme lit et affiche la série de Codes QR .png produits par le **Programme 1**.

Le balayage de la collection de code QRs se fait par ordre alphanumérique croissant sur le nom du .png suffixé du numéro de séquence est déclenché par la captation (lecture et décodage) d'un code QR "ACK" généré et affiché par le **Programme 3** décodeur sur l'appareil malveillant destinatrice des données.

L'index de la webcam de l'appareil « Entreprise » est à positionner dans cv2.VideoCapture(?)

le Code QR « ack » contient simplement le rang de la séquence et le nombre total de séquence :

Ex : 116/116

Ainsi le **Programme 2** et le **Programme 3** se synchronise jusqu'à l'affichage et la captation du dernier code QR du run d'exfiltration sans rupture de séquence.

Code QR « data »	Données encodées
	116/116

Figure 13: Exemple de code QR « ack » produit et structure des données encodées

e) Programme 3 :qrDecoder_OnDeviceReception_WithQrAck.py

Exécution de ce programme sur l'appareil « MALVEILLANT » destinataire des datas à exfiltrer.

Ce programme est lancé en même temps que le **Programme 2** Afficheur et permet de lire et décoder les Codes QR affichés par l'appareil « Entreprise » Source et contenant les données à exfiltrer.

L'index de la webcam de l'appareil « MALVEILLANT » récepteur est à positionner dans cv2.VideoCapture(?).

Lorsque que l'un des Codes QR « data » est lu et décodé, la data en est extraite et mise en liste puis un code QR « ack » d'acquittement contenant en data la séquence acquittée est généré et affiché dans le but d'être scanné, analysé et décodé par le **Programme 2** Afficheur sur l'appareil « Entreprise » source qui se synchronise et affiche le code QR suivant.

Lorsque tous les codes QRs sont reçus, le **Programme 2** se termine, les datas sont concaténées et décodées de la base_45 pour restituer le payload binaire tar.gz. La sortie de ce **Programme 3** affiche l'emplacement sur le disque où trouver le l'archive reconstituée :

ex : Fin du traitement de décodage et ré-assemblage. ==> Emplacement de l'archive tar.gz sur le disque : "file:///home/user/PycharmProjects/These_MS_EC/decodedOut/datasLeakOut.tar.gz"

Diagramme POC: Exfiltration datas via code QR avec synchronisation par code QR ACK

Connectivité réseau non-utilisée

Contexte : Télé-travail

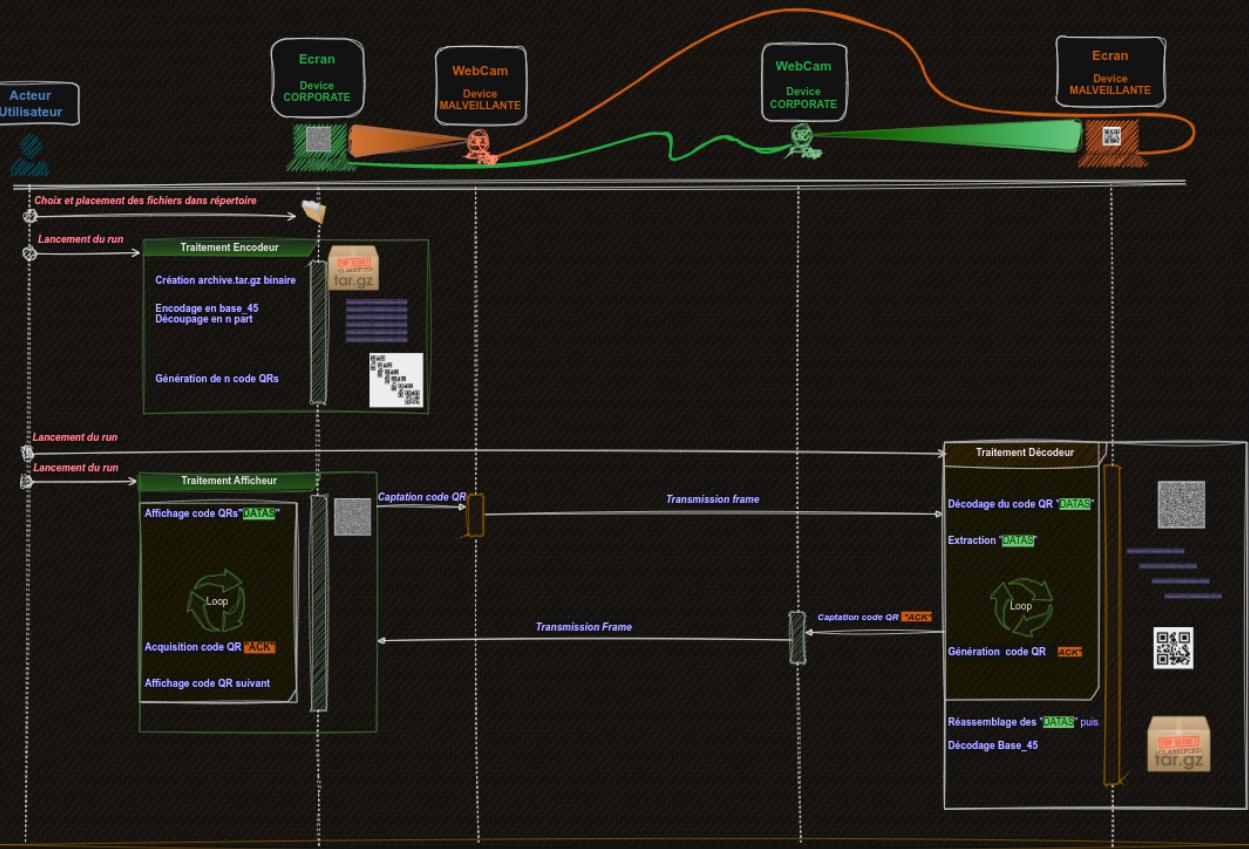
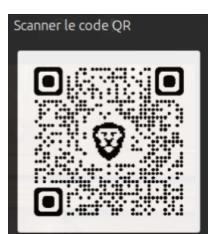


Figure 14: Diagramme de séquence du déroulement de l'exfiltration développée

La vidéo de démonstration est disponible en suivant le lien ci-après : <https://youtu.be/-LtkA9NyXDE> ou en scannant le code QR suivant :



3.4. Résultats obtenus, les difficultés rencontrées et les solutions trouvées

a) Journal de développement (abrégé)

J'ai commencé par écrire le programme encodeur, après la découverte des spécificités et normes du code QR, j'ai testé la librairie qrcode puis je l'ai remplacé par la librairie

« segno » pour avoir de meilleurs temps de traitements d'encodage du QR code (Voir Annexe 1)

J'avais commencé par utiliser l'encodage base_64 et je me suis rendu compte que je ne pouvais pas utiliser la capacité maximale de stockage de la version 40 (voir tableau 3) de 4296 caractères car base_64 inclus entre-autres les minuscules qui ne sont pas acceptées pour le mode alphanumérique de la norme Code QR. Le passage en encodage base_45 plus adapté au code QR m'a permis de passer la limite du mode bytes 2953 caractères à 4296 caractères encodés par code QR version 40.

J'ai affiné différents paramètres comme la taille du code QR, le niveau de correction d'erreur que j'ai finalement positionné au plus bas (ECC Level L) pour accroître la capacité de stockage par code QR. En effet, ici le code QR n'est pas imprimé mais affiché et ne risque pas de subir des altérations physiques qui pourraient rendre difficile sa lecture.

Ensuite, j'ai développé le programme affichage mais sans intégrer la gestion des séquences. Je l'ai intégrée un peu plus tard dans le projet quand j'ai eu besoin de synchroniser l'ensemble afin d'être dans la capacité de récupérer l'ensemble de mes séquences et ce dans l'ordre et d'identifier la dernière séquence afin de terminer l'affichage en automatique.

Cet ajout de séquence m'a fait porter l'évolution dans le programme encodeur en premier lieu.

Et ensuite, je me suis attelé à développer le programme décodeur qui était le plus sophistiqué à élaborer car en plus de lire les codes QR « datas » et de stocker les datas, il devait également générer à la volée les codes QR « de type ack » permettant de communiquer avec le programme afficheur pour acquitter la bonne réception du code QR et déclencher l'affichage du suivant.

Il m'a fallu quelques heures d'optimisations, d'affinage pour obtenir un système très fiable et rapide pour un canal caché (Voir encadré bleu Figure 4 mentionnant des débits de l'ordre de centaines de bits/secondes)

Un bug était qu'aléatoirement, selon des conditions de lumière, d'orientation et de vitesse d'exécution, le programme décodeur lisait des bar-codes au lieu de bien lire le code QR affiché, une fois détecté, il m'a été facile de le fixer.

Cette expérience de développement informatique m'a permis de m'aguerrir sur le langage python, l'utilisation de l'IDE PyCharm et la manipulation de git/ github.

b) Présentation des résultats de l'expérimentation : les débits obtenus

Performances observées	Poste Entreprise Windows	Poste Entreprise MacOS
Débit en kbytes par seconde	42,1 kbps	63,1 kpps

Tableau 6: Débits moyen obtenus

c) Analyse des résultats et discussion

Rappelons tout d'abord qu'aucune connectivité réseau n'est utilisée durant l'exécution de ce modèle d'exfiltration.

Durant l'ensemble de mes tests de performances, je n'ai constaté aucune rupture d'intégrité une fois l'archive reconstituée en fin d'exécution du programme décodeur.

Le fait de créer une archive au départ de l'encodage permet confortablement de mélanger toutes natures de fichiers (binaires, exécutables, base de données, documents suites bureautiques standards, codes, pdf ...), des formats largement utilisés. Une fois transférés, nous retrouvons directement les fichiers informatiques sans devoir faire quelconque manipulations sinon un double-clic.

Une comparaison des hashs (sha512sum) calculés sur l'archive source et l'archive reconstituée permettait de s'assurer le respect de l'intégrité durant ce déplacement de données.

Ce traitement est parallélisable en théorie afin de multiplier les débits. Cela nous obligerait cependant à ajouter une ou plusieurs webcam sur le pc entreprise afin de lire les code QR « ack ». Nous ne serions plus dans le cadre strict que je me suis imposé, celui de ne pas modifier la configuration matérielle / logicielle du poste collaborateur.

La preuve de démonstration est donc faite. Elle permet une fuite de donnée avec un débit certain permettant une fuite conséquente d'informations, donc d'informations sensibles, confidentielles ou secrètes en plus de notre propriété intellectuelle.

J'ai eu occasion de présenter une démonstration à deux reprises à différents référents sur les sujets DLP au sein de l'entreprise. La démonstration rappelle la réalité de la menace et nous allons poursuivre ensemble afin d'explorer les options de mitigation du risque de ce canal mis en lumière.

4. Détections et contres-mesures

Naturellement, le réflexe maintenant est de s'attacher à définir des contre-mesures.

L'End-point Management (EM) semble être le moniteur de référence* concerné par la détection et du blocage de ce modèle d'exfiltration.

La connectivité réseau n'est pas sollicitée par la méthode, il est donc inutile d'explorer cette direction.

L'accès à la webCam et l'analyse de l'écran sont des pistes explorables

4.1. Surveillance du poste local

a) Accès à l'utilisation de la webcam par les applications

Contrôler l'accès et bloquer l'utilisation de la webcam intégrée sur les ordinateurs Windows ou Mac est possible pour des raisons de confidentialité et de sécurité.

Ces systèmes proposent déjà des possibilités dans les menus de paramétrages.

Windows :

Paramètres de confidentialité : Vous pouvez désactiver l'accès à la caméra pour toutes les applications dans les paramètres de confidentialité de Windows. Allez dans "Paramètres" -> "Confidentialité" -> "Caméra" et désactivez l'option "Autoriser les applications à accéder à votre caméra".

Gestionnaire de périphériques : Vous pouvez désactiver complètement la caméra en allant dans le gestionnaire de périphériques, en trouvant votre webcam sous "Périphériques d'images", en faisant un clic droit dessus et en choisissant "Désactiver".

MacOS :

Préférences système : Sur les versions récentes de MacOS, vous pouvez contrôler quelles applications ont accès à votre caméra dans les Préférences Système. Allez dans "Préférences Système" -> "Sécurité et confidentialité" -> "Confidentialité" -> "Caméra" et décochez les applications auxquelles vous ne voulez pas donner accès.

Terminal : Il n'y a pas d'option intégrée pour désactiver complètement la webcam sur un système MacOs, mais cela peut être fait en utilisant le Terminal pour décharger le pilote de la webcam. Cependant, cela nécessite un certain niveau de connaissances techniques et peut avoir d'autres conséquences.

Avec l'adoption généralisée du télétravail, les applications de visiophonie (Teams, Skype,...) nécessitent l'accès à la webcam.

Il est totalement inenvisageable de désactiver totalement la caméra du système mais par contre il est possible d'affiner les politiques de sécurité des moniteurs de référence afin de confiner l'accès et l'utilisation de la webcam qu'à cette famille d'applications.

Les autres besoins plus spécifiques de certains collaborateurs pourrait être gérés par la mise en place d'un workflow de validation de demande d'accès à la webCam.

b) Détection et analyse des codes QR affichés sur les écrans des collaborateurs

Sans affichage sur l'écran, notre modèle ne peut aboutir à une fuite de donnée.
Il serait donc nécessaire de surveiller et analyser les écrans de façon continue ?

Le cadre juridique en France établit des limites claires sur la surveillance des employés par leurs employeurs, en particulier en ce qui concerne le respect de la vie privée des employés. Selon la loi française, les employeurs peuvent mettre en place des mesures de surveillance pour des raisons de sécurité informatique, mais ces mesures doivent respecter certains principes :

- Proportionnalité : Les mesures de surveillance doivent être proportionnées à l'objectif visé. Cela signifie qu'elles doivent être nécessaires et appropriées pour atteindre l'objectif de sécurité, sans imposer de restrictions inutiles à la vie privée des employés.
- Transparence : Les employés doivent être informés de toute mesure de surveillance en place, y compris de la nature de la surveillance, de l'objectif visé et de leurs droits en matière de protection des données.
- Droit d'accès : Les employés ont le droit d'accéder aux informations collectées à leur sujet et, dans certains cas, de demander la rectification ou la suppression de ces informations.

En ce qui concerne l'utilisation d'un logiciel qui surveille et analyse en permanence l'écran d'un ordinateur, cela pourrait être considéré comme une intrusion excessive dans la vie privée de l'employé, surtout si l'employé utilise l'ordinateur pour des tâches non professionnelles.

Ici, disposer d'une solution capable d'analyser la multitude des écrans des collaborateurs par la détection et le décodage des code QR afin de qualifier la data et le risque constituerait une contre-mesure idéale.

La donnée pourrait être typée, évaluée en utilisant des techniques simples et plus évoluées intégrant du Machine Learning* :

- Si la donnée décodée est une url : des éditeurs sont déjà leaders dans l'évaluation de la réputation des noms de domaines par exemple.
- Si la taille de donnée décodée dépasse un seuil (50 octets par exemple), une alerte levée permettrait d'instruire la nature des données. En face de données illisibles ou encodées en base_45, base_64 ou autres : la fuite de donnée pourrait être détectée et un blocage par apposition d'une pastille de masquage sur l'écran rendrait le code QR illisible.

Il existe aujourd'hui une société américaine qui propose une solution de ce type avec inscription d'alerte dans le journal d'événements Windows et l'apposition de masques au niveau des pixels affichant les modules du code QR.

Une fonctionnalité screen-Safe qui renforce les mesures DLP* déjà en place.

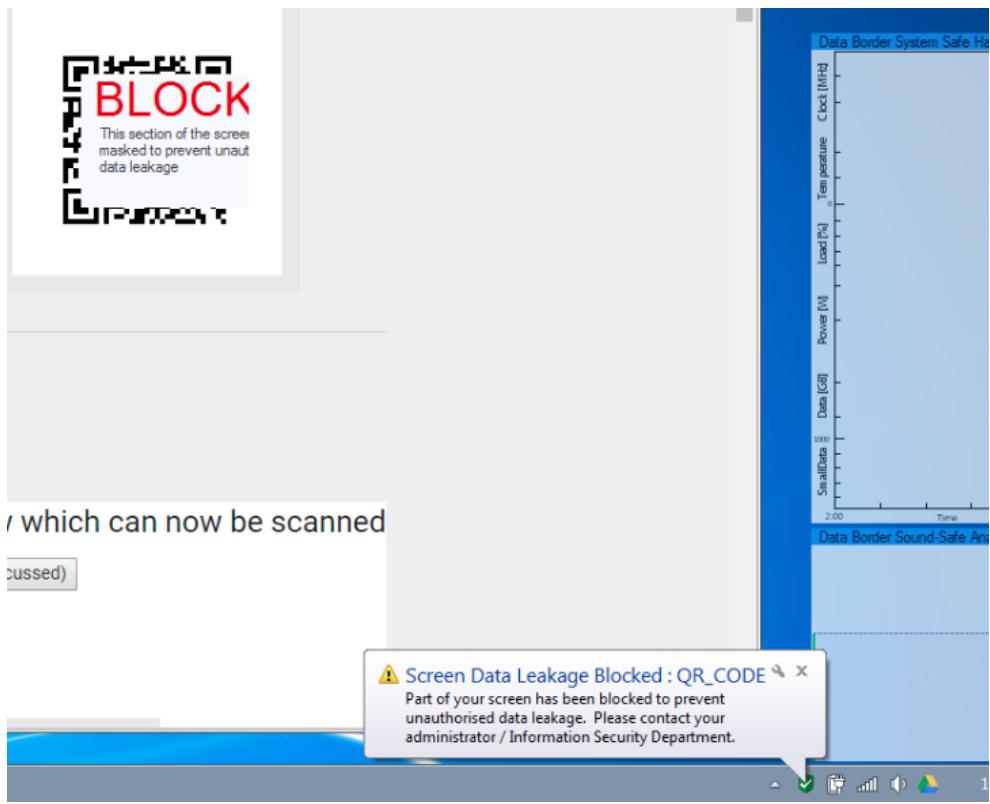


Figure 15: Exemple d'implémentation de contre mesure contre l'affichage de code QR

J'ai pris contact avec cette société mais je n'ai pas eu de retours de leur part.

5. Conclusion

La prévention des fuites de données, qu'elles soient fortuites ou malveillantes, revêt une importance primordiale. Les RSSI disposent d'un éventail de solutions de prévention des pertes de données DLP pour minimiser considérablement ces risques.

L'existence des canaux de communication dissimulés ou discrets est bien documentée dans la littérature académique, démontrant l'ingéniosité infinie des chercheurs et des pirates informatiques.

Cependant, une évaluation des risques basée sur la faible bande passante offerte par ces techniques a entraîné une sous-estimation de ceux-ci, même dans les entreprises critiques, comme les banques ou d'autres secteurs stratégiques.

Les travaux menés dans le cadre de cette thèse professionnelle ont démontré la faisabilité d'atteindre des débits de l'ordre de dizaines de kilobits par seconde, ce qui confirme que le risque associé à l'exfiltration de données sans utilisation de la connectivité, par le biais du code QR, est sous-estimé.

Il est envisageable de mettre en place des politiques d'accès plus strictes aux webcams intégrées. Des utilisations spécifiques de la caméra pourraient être soumises à une procédure de validation, impliquant un flux de travail de gestion d'accès à plusieurs niveaux, basé sur le poste et l'identifiant de l'utilisateur.

Une autre piste à explorer est l'analyse des fonctionnalités proposées par l'industrie pour surveiller l'affichage des codes QR sur les écrans des collaborateurs.

Nous pourrions requérir dans les futurs appels d'offre pour le choix de solutions de sécurité EndPoint / EndManagement une nouvelle fonctionnalité de sécurité : la capacité de détection d'exfiltration par code QR et ce idéalement en temps réel.

Cette approche pourrait non seulement aider à contrer l'exfiltration de données via les codes QR, mais aussi à atténuer d'autres menaces associées à ces codes QR.

Enfin, il convient de noter que les écrans des smartphones de l'entreprise représentent également un risque d'exposition des données de l'entreprise. Par conséquent, les solutions mises en place pour atténuer le risque associé aux codes QR pourraient également bénéficier à ces dispositifs.

Bibliographie :

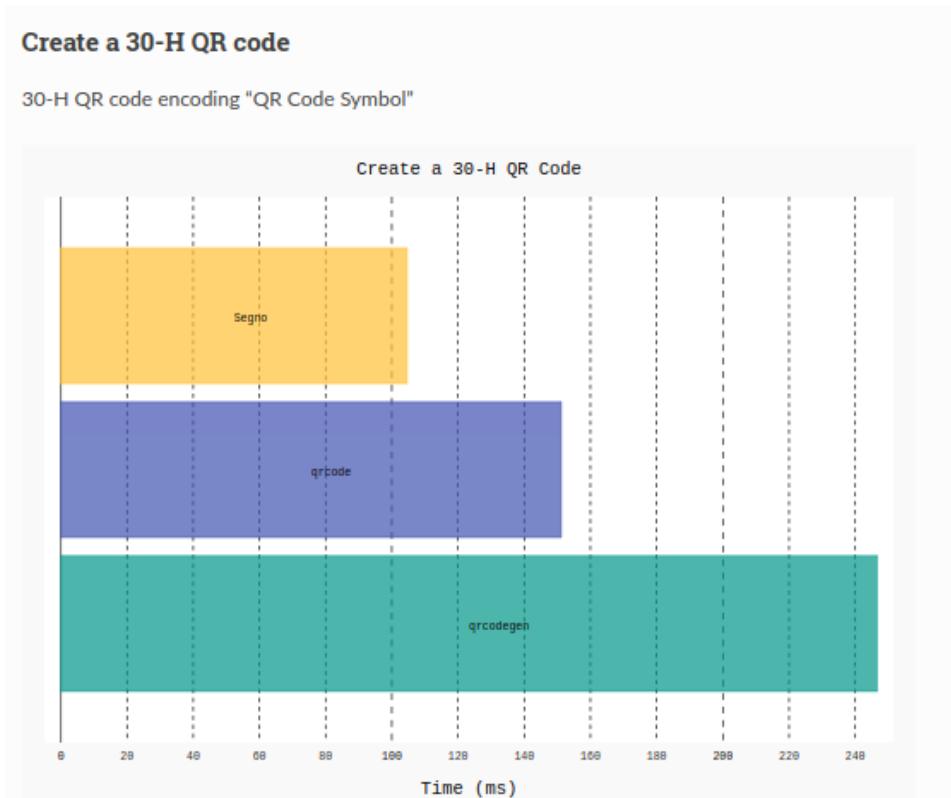
- Admin, (2011). QR Code Error Correction. [Online] Available at: <https://blog.qrstuff.com/2011/12/14/qr-code-error-correction>
- Alajmi, M., Elashry I., El-Sayed H. S. and Farag Allah O. S, (2020). "Steganography of Encrypted Messages Inside Valid QR Codes," in IEEE Access, vol. 8.
- Anon., (2012). Reed-Solomon Codes. [Online] Available at: <https://www.techopedia.com/definition/25798/reed-solomon-codes-work>
- Anon., (2019). Types of QR Code. [Online] Available at: <https://www.codeQR.com/en/codes/>
- Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., ... & Levchenko, K. (2017). « Understanding the Mirai Botnet. In Proceedings of the 26th USENIX Security Symposium (pp. 1093-1110) »
- Asonov, D., & Agrawal, R. (2004). Keyboard acoustic emanations revisited.
- Assaad, A., (2019). What's inside The QR code?. [Online] Available at: <https://medium.com/analytics-vidhya/whats-inside-the-qr-code-bf8a465378fd>
- Backes, M., Dürmuth, M., & Pinkal, M. (2010). Compromising reflections or how to read LCD monitors around the corner.
- Brown, T. W., & Xu, H. (2016) « An Analysis of Data Exfiltration Techniques over Encrypted Channels.»
- Carrara, (2016) « Air-Gap Covert Channels »
- Choo, K. K. R., & Liu, Q. (Eds.). (2021). « Rootkits and Bootkits: Reversing Modern Malware and Next Generation Threats. Springer »
- Cidon, I., Gavish, S., & Markovitch, S. (2018) « Exfiltration channels for data loss: a comprehensive taxonomy and analysis »
- Cohen, Y., Herzberg, A., & Karnin, E. D. (2017). « Pharming Attacks and Countermeasures: A Survey. ACM Computing Surveys (CSUR), 49(4), 6 »
- Common Criteria (2005), Part 3: Security assurance components, August URL <https://www.commoncriteriaportal.org/files/ccfiles/ccpart3v2.3.pdf>
- Conti, M., & Sobers, R. (2010). «Investigating data exfiltration techniques on USB devices. IEEE Security & Privacy »
- Donald C Latham. (1986) Department of Defense trusted computer system evaluation criteria. Department of Defense
- Eckert, C., & Sönmez, T. (2020) « Cybersecurity: The Insights You Need from Harvard Business Review. Harvard Business Press. »
- Garg, G., (2015). How QR Codes work: Everything you need to know and more.. [Online] Available at: <https://scanova.io/blog/blog/2015/02/19/how-qr-codes-work/>
- Genkin, D., Shamir, A., & Tromer, E. (2014). RSA key extraction via low-bandwidth acoustic cryptanalysis.
- Giani, Berk and Cybenko, (2006) «Data Exfiltration and Covert Channels »
- Hadnagy, C. (2018) « Social Engineering: The Science of Human Hacking. Wiley Publishing »
- Hue B., (2020) « QR codes, SMS, dépistage : gare aux nouvelles arnaques qui surfent sur le Covid-19 », sur [www.rtl.fr](http://www rtl fr)
- Jakobsson, M., & Myers, S. (2019). « Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft. Wiley Publishing. »

- Johnson, N. F., & Jajodia, S. (1998) « Exploring steganography : seeing the unseen»
- Kuhn, M. G. (2013). Compromising electromagnetic emanations of wired and wireless keyboards.
- Marchand L., (2020) « Le retour en grâce du QR Code en cinq questions» sur www.lesechos.fr
- Mathuria, M., (2017). A Review on QR Codes. [Online] Available at: https://www.researchgate.net/profile/Manish-Mathuria/publication/316177848_A_Review_on_QR_Code/links/5d4832e592851cd046a2d5df/A-Review-on-QR-Code.pdf
- Moskowitz I.S. and Kang M.H., (1994) Covert channels-here to stay? In Computer Assurance. COMPASS '94 Safety, Reliability, Fault Tolerance, Concurrency and Real Time, Security. Proceedings of the Ninth Annual Conference on, pages 235–243. IEEE, Jun 1994.
- Narayanan, S., (2012). QR Codes and Security Solutions. [Online] Available at: https://www.ijcst.org/Volume3/Issue7/p13_3_7.pdf
- Rouse M., (2013) « Data Exfiltration » sur www.techopedia.com Available at : <https://www.techopedia.com/definition/14682/data-exfiltration>
- Roush, W. (2018). Computers & Security, 72, 105-118 » « A Study of Shoulder Surfing on Mobile Devices.
- Scott, (2020). How Do QR Codes Work ? QR Code Technical Basics.. [Online] Available at: <https://www.sproutqr.com/blog/how-do-qr-codes>
- Sepehrdad, P., Ghafarian, M., & Dehghanianha, (2020). A« Systematic Literature Review on Windows Privilege Escalation. Computers & Security, 97, 101945»
- Sheridan K., (2019) « Database Leaks, Network Traffic Top Data Exfiltration Methods» sur www.darkreading.com
- Tiwari, S., (2016). An Introduction to QR Code Technology. [Online] Available at: <https://ieeexplore.ieee.org/document/7966807>
- Vasilomanolakis, E., Daubert, J., Frincu, M., & Castillo, C. (2020) « A Survey on Covert Channels and Countermeasures in Intrusion Detection Systems. ACM Computing Surveys (CSUR), 53(5), 1-37»
- Xu, H., Zhuang, Y., & Liu, K. (2012). An experimental study of air-gap covert channels in RF-based gesture recognition.

Annexes

Index des Annexes

Annexe 1: Comparaison de 3 librairies code QR python sur la fonction création.....	34
Annexe 2: lien github contenant le code de la démonstration de faisabilité.....	35



**Annexe 1: Comparaison de 3 librairies code QR
python sur la fonction création**

Annexe 2: lien github contenant le code de la démonstration de faisabilité

https://github.com/AL1NICOLAS/These_MS_EC/tree/main/sourceCode