

2024
2025

Atelier n°2

RSX112 - SECURITE DES RESEAUX
STEPHANE LARCHER



Atelier n°2

Table des matières

Phase 1 : Préparation de l'Environnement	3
1.1 Configuration des Conteneurs	3
1.2 Script de Monitoring Réseau	4
Phase 2 : Analyse de TLS/SSL	6
2.1 Configuration Multi-Versions TLS	6
2.2 Analyse du Handshake TLS.....	6
2.3 Exploitation de Heartbleed (Environnement Contrôlé).....	8
Phase 3 : Analyse SSH	10
3.1 Configuration SSH Multi-Niveaux	10
3.2 Analyse du Handshake SSH	10
3.3 Tests d'Attaques SSH.....	11
Phase 4 : Analyse IPsec	12
4.1 Configuration IPsec/IKEv2.....	12
4.2 Analyse du Trafic IPsec.....	12
4.3 Attaques sur IPsec.....	13
Phase 5 : Scénarios d'Attaque Intégrés	14
5.1 Scénario : Downgrade Attack.....	14
5.2 Scénario : MITM sur SSH	14
Phase 6 : Défense et Durcissement	16
6.1 Script de Durcissement Automatique.....	16
6.2 Monitoring et Alertes.....	17
Génération du Rapport Final	18
Good to know.....	19
1. TLS/SSL	19
2. SSH	19
3. IPsec	19
4. Défense en Profondeur	19

Phase 1 : Préparation de l'Environnement

1.1 Configuration des Conteneurs

Sur Attack-Station (CT1X3) :

1. Connexion initiale

```
ssh pkilab@10.1X.0.4
```

Mot de passe : pkilab2024

2. Installation des outils d'analyse

```
sudo apk update
```

```
sudo apk add --no-cache \  
    wireshark-cli tcpdump tshark \  
    nmap masscan \  
    openssl openssh-client \  
    curl wget netcat-openbsd \  
    python3 py3-pip \  
    git make gcc musl-dev \  
    john hydra
```

3. Installation des outils Python

```
pip3 install --user scapy paramiko pwntools cryptography
```

4. Création de la structure de travail

```
mkdir -p ~/proto-lab/{captures,exploits,scripts,reports}  
cd ~/proto-lab
```

✓ **Point de contrôle** : Vérifiez que tous les outils sont installés :

```
which nmap tcpdump openssl python3
```

Tous doivent retourner un chemin

Sur Target-Server (CT1X2) :

1. Connexion SSH depuis Attack-Station

```
ssh pkilab@10.1X.0.3
```

2. Installation des services

```
sudo apk add --no-cache \  
    nginx apache2 \  
    
```

```
openssh-server \  
vsftpd \  
strongswan \  
openssl
```

```
# 3. Création des répertoires de configuration  
mkdir -p ~/services/{tls-modern,tls-legacy,ssh-configs}
```

1.2 Script de Monitoring Réseau

Sur Attack-Station :

```
# 1. Créer le script netmon.sh  
cd ~/proto-lab/scripts  
nano netmon.sh  
# Copier le contenu du script depuis l'annexe A.1  
  
# 2. Rendre exécutable  
chmod +x netmon.sh  
  
# 3. Test du script  
./netmon.sh  
# Sélectionner option 1 pour capturer HTTPS/TLS
```

Résultat attendu :

```
=== Network Protocol Monitor ===  
Groupe X - [Date]  
  
Select protocol to monitor:  
1) HTTPS/TLS  
2) SSH  
3) IPsec  
4) All protocols  
5) Analyze capture  
6) Quit  
#? 1  
[*] Capturing tls on port 443 for 30 seconds...  
[*] Packets captured: 0
```


Phase 2 : Analyse de TLS/SSL

2.1 Configuration Multi-Versions TLS

Sur Target-Server :

```
# 1. Configuration moderne (TLS 1.3)
cd ~/services/tls-modern
nano nginx-tls13.conf
# Copier la configuration depuis l'annexe B.1

# 2. Configuration vulnérable
cd ~/services/tls-legacy
nano nginx-weak.conf
# Copier la configuration depuis l'annexe B.2

# 3. Démarrage des services
# Service moderne sur port 8443
sudo nginx -c ~/services/tls-modern/nginx-tls13.conf

# Service vulnérable sur port 8080
sudo nginx -c ~/services/tls-legacy/nginx-weak.conf

# 4. Vérification
sudo netstat -tlnp | grep nginx
```

2.2 Analyse du Handshake TLS

Sur Attack-Station :

```
# 1. Créer le script d'analyse
cd ~/proto-lab/scripts
nano tls-analyzer.sh
# Copier depuis l'annexe A.2
chmod +x tls-analyzer.sh

# 2. Analyser le serveur moderne
./tls-analyzer.sh 10.1X.0.3 8443
```

Résultat attendu pour serveur moderne :

=== TLS Protocol Analyzer ===
Target: 10.1X.0.3:8443

[1] Supported TLS versions:

ssl3: X NOT SUPPORTED
tls1: X NOT SUPPORTED
tls1_1: X NOT SUPPORTED
tls1_2: X NOT SUPPORTED
tls1_3: ✓ SUPPORTED

[2] Cipher suites:

TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256
TLS_AES_128_GCM_SHA256

[3] Certificate details:

Subject: CN=web-server.groupeX.local
Issuer: CN=Groupe X Root CA
Not Before: [Date]
Not After: [Date+365]

[4] Known vulnerabilities:

Heartbleed: ✓ Safe
POODLE: ✓ Safe

3. Analyser le serveur vulnérable :

./tls-analyzer.sh 10.1X.0.3 8080

Résultat attendu pour serveur vulnérable :

[1] Supported TLS versions:

ssl3: ✓ SUPPORTED # DANGER!
tls1: ✓ SUPPORTED # DANGER!
tls1_1: ✓ SUPPORTED # DANGER!

[2] Cipher suites:

EXPORT_RSA_RC4_40_MD5 # TRÈS FAIBLE!
DES_CBC_SHA # OBSOLÈTE!
RC4_MD5 # CASSÉ!

[4] Known vulnerabilities:

Heartbleed: ⚠ Possible (à vérifier)

POODLE: ⚠ VULNERABLE!

2.3 Exploitation de Heartbleed (Environnement Contrôlé)

Sur Lab-SecProto (CT1X1) :

1. Créer le serveur vulnérable

```
cd ~/vuln-lab/heartbleed
```

```
nano server.py
```

Copier depuis l'annexe C.1

2. Lancer le serveur

```
python3 server.py &
```

Output: [*] Vulnerable HTTPS server listening on port 4433...

Sur Attack-Station :

1. Créer l'exploit

```
cd ~/proto-lab/exploits
```

```
nano heartbleed_exploit.py
```

Copier depuis l'annexe C.2

2. Exécuter l'exploit

```
python3 heartbleed_exploit.py 10.1X.0.2 4433
```

Résultat attendu :

```
[*] Connecting to 10.1X.0.2:4433
```

```
[*] Sending Client Hello
```

```
[*] Sending malicious heartbeat
```

```
[*] Receiving leaked memory
```

```
[+] Received 16384 bytes
```

```
[+] Memory dump:
```

```
180302000301400...
```

```
[+] Found strings in memory:
```

```
- SECRET_KEY=Gr0up3X_S3cr3t_D4t4_1337
```

- HTTP/1.1 200 OK
- Vulnerable Server

⚠ Point pédagogique important :

- Heartbleed permet de lire jusqu'à 64KB de mémoire du serveur
- Les données sensibles (clés, sessions, mots de passe) peuvent être exposées
- Patch critique : mise à jour OpenSSL > 1.0.1g

Phase 3 : Analyse SSH

3.1 Configuration SSH Multi-Niveaux

Sur Target-Server :

```
# 1. Configuration sécurisée
cd ~/services/ssh-configs
nano sshd_secure.conf
# Copier depuis l'annexe B.3

# 2. Configuration vulnérable
nano sshd_weak.conf
# Copier depuis l'annexe B.4

# 3. Lancer les services
# SSH sécurisé sur port 2222
sudo /usr/sbin/sshd -f ~/services/ssh-configs/sshd_secure.conf -p
2222

# SSH vulnérable sur port 2200
sudo /usr/sbin/sshd -f ~/services/ssh-configs/sshd_weak.conf -p 2200
```

3.2 Analyse du Handshake SSH

Sur Attack-Station :

```
# 1. Créer le script d'analyse
cd ~/proto-lab/scripts
nano ssh-analyzer.sh
# Copier depuis l'annexe A.3
chmod +x ssh-analyzer.sh

# 2. Analyser le serveur sécurisé
./ssh-analyzer.sh 10.1X.0.3 2222
```

Résultat attendu (serveur sécurisé) :

```
=== SSH Protocol Analyzer ===
Target: 10.1X.0.3:2222
```

[1] SSH Banner:
SSH-2.0-OpenSSH_8.x

[2] Supported algorithms:
kex: [curve25519-sha256,curve25519-sha256@libssh.org](https://cve.mitre.org/cve/2019/11555/)
cipher: [chacha20-poly1305@openssh.com,aes256-gcm@openssh.com](https://cve.mitre.org/cve/2019/11555/)
mac: [hmac-sha2-512-etm@openssh.com](https://cve.mitre.org/cve/2019/11555/)

[4] Authentication methods:
Authentications that can continue: publickey

3.3 Tests d'Attaques SSH

```
# 1. Créer le script de test
cd ~/proto-lab/scripts
nano ssh-security-test.sh
# Copier depuis l'annexe A.4
chmod +x ssh-security-test.sh

# 2. Tester le serveur vulnérable
./ssh-security-test.sh 10.1X.0.3 2200
```

Résultat attendu :

```
=== SSH Security Testing ===

[2] Testing password strength (limited)...
WARNING: This is for educational purposes only!
[SUCCESS] host: 10.1X.0.3 login: root password: alpine

[3] User enumeration via timing:
Testing user 'root': 125ms
Testing user 'admin': 89ms
Testing user 'pkilab': 134ms
Testing user 'nonexistent...': 15ms # Plus rapide = n'existe pas

[5] Checking for deprecated algorithms:
Testing 3des: ⚠️ ENABLED
Testing arcfour: ⚠️ ENABLED
```

Phase 4 : Analyse IPsec

4.1 Configuration IPsec/IKEv2

Sur Target-Server :

```
# 1. Configuration IPsec
cd ~/services
nano ipsec.conf
# Copier depuis l'annexe B.5

nano ipsec.secrets
# Copier depuis l'annexe B.6

# 2. Copier les fichiers
sudo cp ipsec.conf /etc/ipsec.conf
sudo cp ipsec.secrets /etc/ipsec.secrets
sudo chmod 600 /etc/ipsec.secrets

# 3. Démarrer strongSwan
sudo ipsec start
sudo ipsec status
```

4.2 Analyse du Trafic IPsec

Sur Attack-Station :

```
# 1. Créer le script d'analyse
cd ~/proto-lab/scripts
nano ipsec-analyzer.sh
# Copier depuis l'annexe A.5
chmod +x ipsec-analyzer.sh

# 2. Lancer l'analyse
./ipsec-analyzer.sh
# Laisser capturer pendant 30 secondes
# Dans un autre terminal, essayer d'établir une connexion IPsec
```

4.3 Attaques sur IPsec

```
# 1. Créer le script d'attaque
cd ~/proto-lab/exploits
nano ike-aggressive-attack.py
# Copier depuis l'annexe C.3

# 2. Exécuter le test
python3 ike-aggressive-attack.py 10.1X.0.3
```

Résultat attendu :

```
[*] Testing Aggressive Mode on 10.1X.0.3:500
[-] No response - Aggressive Mode likely disabled
```

```
[*] Mitigation:
- Disable Aggressive Mode ✓
- Use Main Mode with certificates
- Implement IKEv2 instead
```

Phase 5 : Scénarios d'Attaque Intégrés

5.1 Scénario : Downgrade Attack

```
# 1. Créer le script de démonstration
cd ~/proto-lab/scripts
nano downgrade-attack-demo.sh
# Copier depuis l'annexe A.6
chmod +x downgrade-attack-demo.sh

# 2. Exécuter la démonstration
./downgrade-attack-demo.sh
```

Résultat attendu :

```
=== Protocol Downgrade Attack Demonstration ===
```

```
[1] Setting up multi-protocol server...
[*] Downgrade test server on port 8444
```

```
[2] Normal connection attempt:
TLS 1.3 requested
```

```
[3] Simulating downgrade attack:
[MITM] Intercepting connection...
[MITM] Stripping TLS 1.3 capability...
[MITM] Forcing TLS 1.0...
Connection downgraded!
```

```
[5] Mitigations:
✓ Disable old protocol versions
✓ Implement TLS_FALLBACK_SCSV
✓ Use HSTS preload
✓ Monitor for protocol anomalies
```

5.2 Scénario : MITM sur SSH

```
# 1. Créer le script MITM
cd ~/proto-lab/exploits
nano ssh-mitm-demo.py
# Copier depuis l'annexe C.4
```

```
# 2. Lancer la démonstration
python3 ssh-mitm-demo.py
```

Dans un autre terminal :

```
# Tenter une connexion
ssh -p 2299 test@localhost
```

Résultat attendu :

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@      WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!      @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle
attack)!

*** WARNING: MITM DEMONSTRATION ***
*** This shows why host key verification is critical ***
*** Never ignore SSH host key warnings! ***
```

Phase 6 : Défense et Durcissement

6.1 Script de Durcissement Automatique

```
# 1. Créer le script
cd ~/proto-lab/scripts
nano harden-protocols.sh
# Copier depuis l'annexe A.7
chmod +x harden-protocols.sh

# 2. Exécuter le durcissement
./harden-protocols.sh
```

Résultat attendu :

```
=== Protocol Hardening Script ===
Applying security best practices...
```

```
[1] Hardening TLS/SSL...
```

- ✓ TLS 1.2+ only
- ✓ Strong ciphers only
- ✓ HSTS enabled
- ✓ Session tickets disabled

```
[2] Hardening SSH...
```

- ✓ Key-based auth only
- ✓ Modern algorithms only
- ✓ Root login disabled
- ✓ Strict session limits

```
[3] Hardening IPsec...
```

- ✓ IKEv2 only
- ✓ AES-256-GCM only
- ✓ Certificate auth
- ✓ PFS enabled

```
[4] Security checklist:
```

- [] Remove all weak ciphers
- [] Disable deprecated protocols
- [] Enable logging and monitoring

```
[ ] Regular security updates
```

```
[*] Hardening complete!
```

```
[*] Configuration files saved in /tmp/
```

6.2 Monitoring et Alertes

```
# 1. Créer le script de monitoring
```

```
cd ~/proto-lab/scripts
```

```
nano protocol-monitor.sh
```

```
# Copier depuis l'annexe A.8
```

```
chmod +x protocol-monitor.sh
```

```
# 2. Lancer le monitoring
```

```
./protocol-monitor.sh
```

Résultat en temps réel :

```
=== Protocol Security Monitor ===
```

```
Monitoring for suspicious activity...
```

```
Starting monitors...
```

```
Monitors running in background.
```

```
Logs: /home/pkilab/proto-lab/logs/security.log
```

```
Press 's' for stats, 'q' to quit
```

```
[!] ALERT: Weak TLS version detected - SSLv3 connection attempt
```

```
[!] SSH: Failed login attempt - Failed password for root
```

```
[!] IPsec: Aggressive mode detected - potential hash disclosure
```

Génération du Rapport Final

```
# Créer le script de rapport
cd ~/proto-lab/scripts
nano generate-report.sh
# Copier depuis l'annexe A.9
chmod +x generate-report.sh

# Générer le rapport
./generate-report.sh
```

Good to know

1. TLS/SSL

- Ne jamais utiliser SSLv3, TLS 1.0/1.1
- TLS 1.2 minimum, préférer TLS 1.3
- Utiliser uniquement des cipher suites AEAD
- Implémenter HSTS et OCSP Stapling

2. SSH

- Éviter l'authentification par mot de passe
- Utiliser Ed25519 ou RSA 4096 bits
- Toujours vérifier les empreintes de clés
- Implémenter fail2ban contre le bruteforce

3. IPsec

- Ne jamais utiliser le mode agressif IKE
- IKEv2 avec DH Group 14 minimum
- AES-256-GCM pour ESP
- Utiliser des certificats plutôt que PSK

4. Défense en Profondeur

- Monitoring continu des protocoles
- Logs centralisés et alertes
- Mises à jour régulières
- Tests de sécurité périodiques