3.3 Utilisation de composants avec des vulnérabilités connues

Référence	REC3	Criticité	MOYENNE
Profil identifié	APP	Priorité	P3
Périmètre	https://ifprofs.org		
Recommandations	Mettre à jour les composants logiciels vulnérables.		

Description générique

Certains composants logiciels utilisés par l'application ont des vulnérabilités connues. Selon les vulnérabilités, un attaquant peut prendre la main sur l'application, effectuer une attaque par déni de service, exécuter du code sur le PC des utilisateurs de l'application voire prendre le contrôle du serveur. Un processus de gestion des correctifs doit être mis en place pour :

- Supprimer les dépendances inutilisées, les fonctionnalités, les composants, les fichiers et la documentation inutiles.
- Inventorier en continu les versions des composants côté client et côté serveur (par exemple, frameworks, bibliothèques) et leurs dépendances à l'aide d'outils tels que versions, DependencyCheck, retire.js, etc. Surveiller régulièrement les sources telles que CVE et NVD pour des vulnérabilités dans les composants. Utilisez des outils d'analyse de composition logicielle pour automatiser le processus. Abonnez-vous aux alertes par e-mail pour les vulnérabilités de sécurité liées aux composants que vous utilisez.
- N'obtenir des composants que de sources officielles via des liens sécurisés. Préférez-les packages signés pour réduire le risque d'inclure un composant malveillant modifié.
- Surveiller les bibliothèques et les composants qui ne sont pas maintenus ou ne créent pas de correctifs de sécurité pour les anciennes versions. Si l'application de correctifs n'est pas possible, envisager de déployer un correctif virtuel pour surveiller, détecter ou protéger contre le problème découvert.

État constaté

Plusieurs sites du périmètre utilisent des composants non mis à jour contenant des vulnérabilités connues.

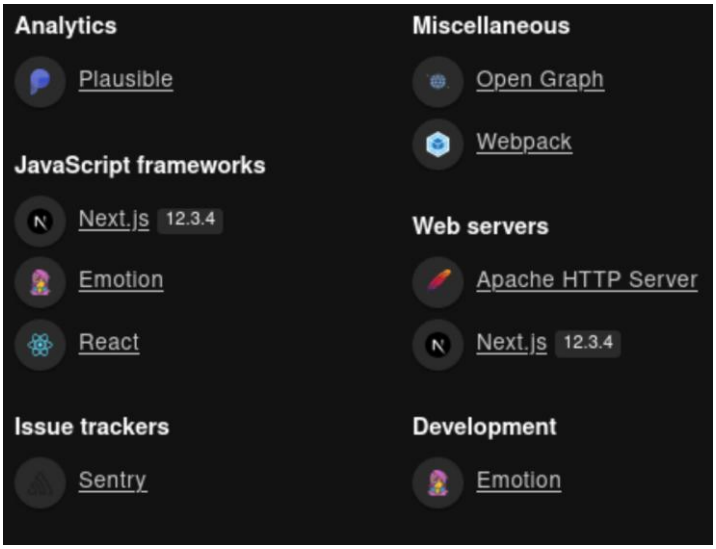

Figure 19 : Next.js 12.3.4 vulnérable sur https://ifprofs.org.

Vercel » Next.js » 12.3.4 for node.js

Vulnerabilities (8) Metasploit Modules

Version names

- Vercel Next.js 12.3.4 for Node.js
- cpe:2.3:a:vercel:next.js:12.3.4:*:*:*:*:node.js:*:*
- cpe:/a:vercel:next.js:12.3.4::~~~node.js~~

Product information

- https://vercel.com/ Vendor
- https://github.com/vercel/next.js Product
- https://github.com/vercel/next.js/tags Change Log

Vulnerabilities by types/categories

Year	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	File Inclusion	CSRF	XXE	SSRF	Open Redirect	Input Validation
2023	0	0	0	0	0	0	0	0	0	0	0
2024	0	0	0	0	0	0	0	0	0	0	0
2025	0	0	0	0	0	0	0	0	1	0	0
Total									1		

Vulnerabilities by impact types

Year	Code Execution	Bypass	Privilege Escalation	Denial of Service	Information Leak
2023	0	0	0	1	0
2024	0	0	0	1	0
2025	0	0	0	0	0
Total				2	

Figure 20 - Huit vulnérabilités connues sour Next.js 12.3.4.

Vulnerability	Vulnerable Version
H Server-side Request Forgery (SSRF)	<14.2.32 >=15.0.0 <15.4.2-canary.43 >=15.4.3 <15.4.7
M Use of Cache Containing Sensitive Information	<14.2.31 >=15.0.0 <15.4.2-canary.19 >=15.4.3 <15.4.5
L Missing Source Correlation of Multiple Independent Data	<14.2.31 >=15.0.0 <15.4.2-canary.19 >=15.4.3 <15.4.5
H HTTP Request Smuggling	>=15.0.4-canary.51 <15.1.8
M Interpretation Conflict	>=15.3.0 <15.3.3
L Missing Origin Validation in WebSockets	>=13.0.0 <14.2.30 >=15.0.0-rc.0 <15.2.2
M Race Condition	<14.2.24 >=15.0.0 <15.1.6
M Information Exposure	>=12.3.5 <12.3.6 >=13.5.9 <13.5.10 >=14.2.25 <14.2.26 >=15.2.3 <15.2.4
C Improper Authorization	>=11.1.4 <12.3.5 >=13.0.0 <13.5.9 >=14.0.0 <14.2.25 >=15.0.0-rc.0 <15.2.3 >=15.3.0-canary.0 <15.3.0-canary.12
M Allocation of Resources Without Limits or Throttling	>=13.0.0 <13.5.8 >=14.0.0 <14.2.21 >=15.0.0 <15.1.2
H Missing Authorization	>=9.5.5 <13.5.8 >=14.0.0 <14.2.15 >=15.0.0-canary.0 <15.0.0-canary.177

Figure 21 : Vulnérabilités Next.js.

Composant	Version	Périmètre	Nombre de vulnérabilités connues
Next.js	12.3.4	https://ifprofs.org	8

Liens utiles

- https://www.owasp.org/index.php/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities

3.6 Fuites d'informations techniques

Référence	REC6	Criticité	MOYENNE
Profil identifié	SYS	Priorité	P2
Périmètre	*Ensemble du périmètre.*		
Recommandations	Limiter au maximum les fuites d'informations techniques (Désactivation de bannières, masquage des versions de logiciels, etc).		

Description générique

A l'instar des fuites d'informations personnelles qui concernent des données à caractères personnelles, les fuites d'informations techniques sont des fuites d'informations relatives aux technologies utilisées sur le périmètre, à des secrets de développement, etc.

Non prévues, les fuites d'informations techniques sont souvent le résultat d'un manque d'expurgation de commentaires dans le code source contenant des informations sensibles, de mauvaises configurations de l'application ou des serveurs sous-jacents.

- Les développeurs laissent souvent des commentaires dans le code HTML et/ou les scripts pour faciliter le débogage ou l'intégration durant la phase de préproduction. Même s'il n'y a pas de problèmes à commenter le code, ces commentaires doivent absolument être expurgés du code source applicatif avant la mise en production.
- Une mauvaise configuration d'une application web et/ou des serveurs web sous-jacents peut donner accès à un attaquant à des informations sensibles contenues dans des messages d'erreur verbeux (« stacktraces »), dans le code source de l'application, dans les en-têtes des réponses des serveurs web, des cookies, etc. Il n'est ainsi pas rare d'obtenir la version précise des services, des applications utilisées, du pare-feu applicatif, des adresses IP et des arborescences de fichiers internes, des requêtes SQL, etc.

Ces informations peuvent aider un attaquant à mener une attaque plus sophistiquée, ou trouver des composants logiciels vulnérables. Il convient donc de limiter ces fuites d'informations au maximum.

État constaté

Plusieurs fuites d'informations techniques ont été relevées sur l'ensemble du périmètre audité. Dans les en-têtes de réponses HTTP, les scripts & librairies JavaScript, les plugins WordPress ainsi qu'une API GraphQL exposée sans authentification avec l'introspection d'activée.

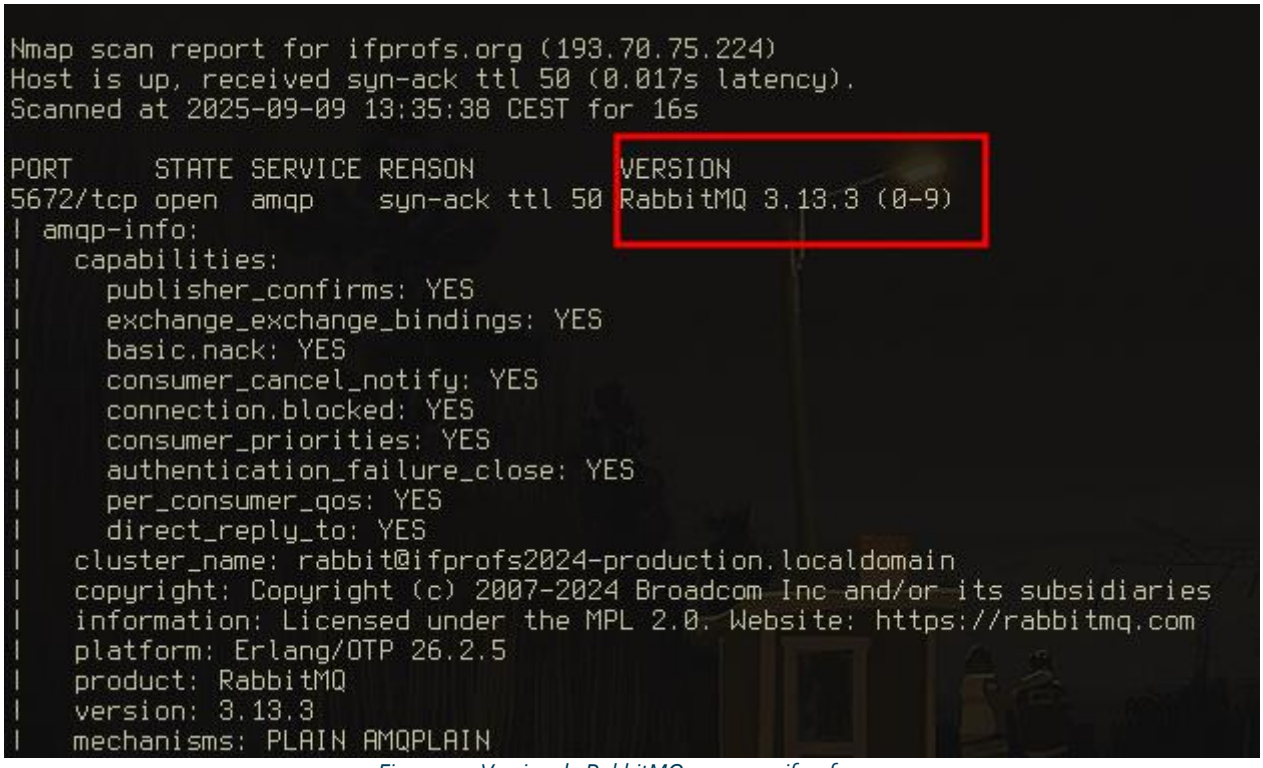

Figure 35 : Version du RabbitMQ 3.13.3 sur ifprofs.org.

Figure 37 : SourceMaps accessibles sur https://ifprofs.org permettant d'obtenir le code JavaScript non minifié/obfusqué.

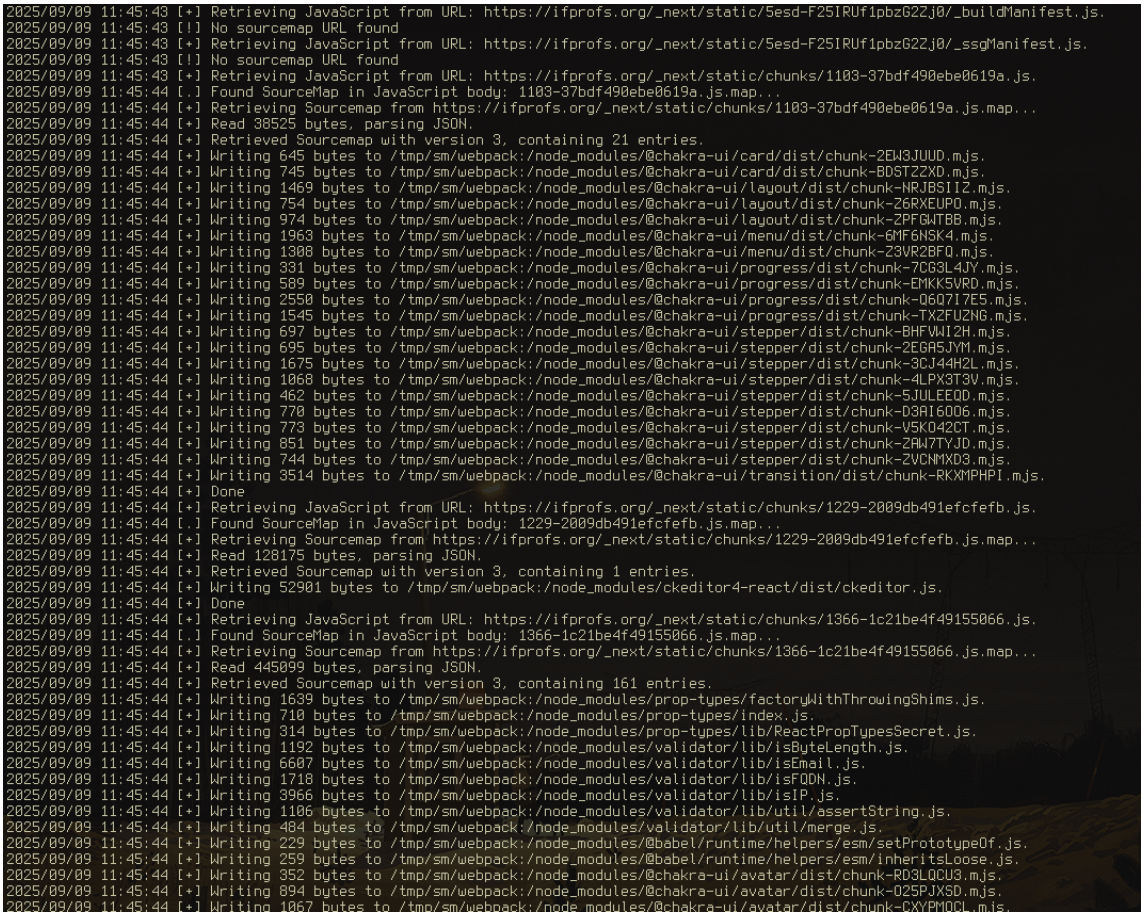

Figure 38 : Récupération de l'ensemble des SourceMaps sur https://ifprofs.org.

Liens utiles

- https://www.owasp.org/index.php/Top_10_2007-Information_Leakage_and_Improper_Error_Handling
- https://www.netsparker.com/blog/web-security/information-disclosure-issues-attacks/
- https://www.acunetix.com/blog/articles/configure-web-server-disclose-identity/
- https://www.hacksplaining.com/prevention/information-leakage

3.9 En-tête(s) de sécurité HTTP manquant(s)

Référence	REC9	Criticité	MINEURE
Profil identifié	SYS	Priorité	P4
Périmètre	*Ensemble du périmètre.*		
Recommandations	Mettre en place les en-têtes de sécurité HTTP manquants.		

Description générique

Il existe des entêtes au protocole HTTP permettant d'augmenter le niveau de sécurité de diverses façons.

Voici ci-dessous une liste de certains de ces entêtes :

- HTTP Strict Transport Security renforce votre implémentation de TLS en obligeant l'agent utilisateur à appliquer l'utilisation du protocole HTTPS. Valeur recommandée : "strict-transport-security: max-age=31536000; includeSubDomains".
- Content-Security-Policy est une mesure efficace pour protéger son site contre les attaques XSS. En mettant en liste blanche les sources de contenu approuvées, il est possible d'empêcher le navigateur de charger des actifs malveillants.
- X-Frame-Options indique au navigateur si vous souhaitez autoriser le cadrage ou non de votre site. En empêchant un navigateur de cadrer votre site, vous pouvez vous défendre contre des attaques telles que le détournement de clic. Valeur recommandée "X-Frame-Options : SAMEORIGIN".
- X-Content-Type-Options empêche un navigateur d'essayer de capturer le type de contenu et l'oblige à s'en tenir au type de contenu déclaré. La seule valeur valide pour cet en-tête est "X-Content-Type-Options: nosniff".
- Referrer-Policy permet à un site de contrôler les informations collectées par les navigateurs sur la navigation des utilisateurs.
- Permissions-Policy permet à un site de contrôler les fonctionnalités et les API pouvant être utilisées par le navigateur comme la caméra, par exemple.

État constaté

Les auditeurs ont constaté que plusieurs en-têtes de sécurité HTTP, sur l'ensemble des sites du périmètre, étaient mal configurés ou n'étaient pas implémentés.

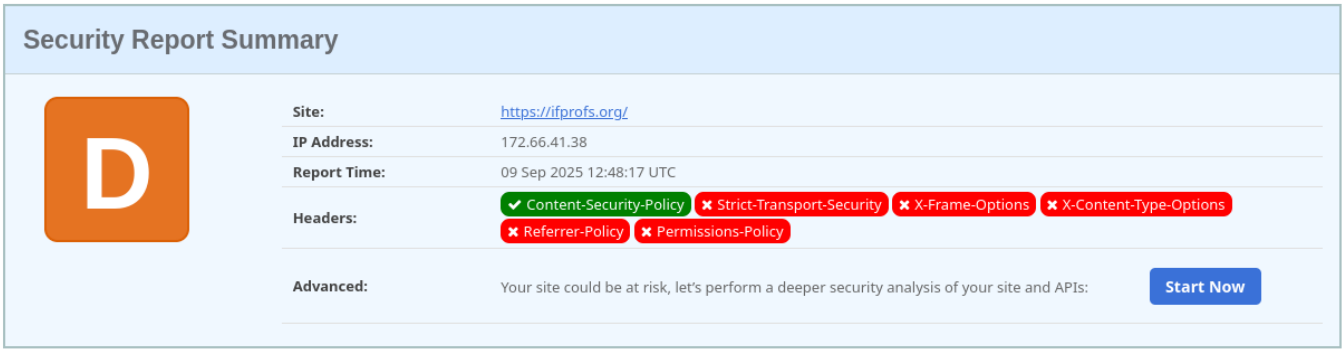

Figure 57 : En-têtes de sécurité HTTP manquants sur https://ifprofs.org.

L'entête de sécurité HTTP XSS-Protection déprécié et pouvant avoir des effets de bord impactant la sécurité a été également retrouvé sur le périmètre sur de nombreux sites :

Liens utiles

- https://www.keycdn.com/blog/http-security-headers
- https://unbonhacker.com/posts/security-headers/
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Permissions_Policy
- https://owasp.org/www-project-secure-headers/

3.10 Vulnérabilités SSL / TLS

Référence	REC10	Criticité	MINEURE
Profil identifié	SYS	Priorité	P4
Périmètre			
Recommandations	Désactiver les protocoles et les algorithmes de chiffrements faibles. Revoir la configuration de(s) serveur(s).		

Description générique

Les transmissions de données HTTP doivent être sécurisées par un tunnel SSL/TLS donnant un trafic HTTPS sécurisé assurant la confidentialité, mais aussi l'authentification. Certaines mauvaises configurations du serveur web permettent l'utilisation de protocoles et algorithmes cryptographiques faibles remettant en question cette confidentialité.

De plus, plusieurs vulnérabilités liées à des implémentations faibles permettent aujourd'hui des attaques théoriques qu'il convient de remonter. Il est possible de citer quelques vulnérabilités habituellement rencontrées :

- BEAST (Browser Exploit Against SSL/TLS) impacte SSL 3.0 et TLS 1.0. Un utilisateur malveillant peut déchiffrer les données chiffrées échangées entre un serveur web et le navigateur d'un client grâce à une vulnérabilité dans l'implémentation du mode CBC dans TLS 1.0. L'attaque se produit du coté client avec une attaque de type l'homme du milieu (Man In The Middle) qui consiste à injecter des paquets spécialement fabriqués dans le flux TLS. Cela permet à un attaquant de déchiffrer les communications en comparant le texte en clair injecté avec le même contenu, chiffré. Pour empêcher cette attaque, il est conseillé d'utiliser à minima TLS 1.2.
- SWEET32 profitant d'une faiblesse dans le mode CBC autorisant des attaques par collision. En collectant un maximum de 32 GB de données chiffrées, il est possible de trouver la clé privée et de déchiffrer le contenu.
- LUCKY13 est une attaque cryptographique basée sur le temps contre les implémentations de la couche protocolaire TLS. Cette vulnérabilité est aujourd'hui théoriquement exploitable mais improbable de nos jours.

État constaté

La configuration du SSL/TLS sur certains sites du périmètre contient encore des options considérées comme obsolètes :

```
Testing all IPv4 addresses (port 443): 172.66.41.38 172.66.42.218
--------------------------------------------------------------------------------------------------
 Start 2025-09-02 12:48:14        -->> 172.66.41.38:443 (ifprofs.org) <<--

 Further IP addresses:   172.66.42.218 2606:4700:3108::ac42:2926 2606:4700:3108::ac42:2ada
 rDNS (172.66.41.38):    --
 Service detected:       HTTP

 Testing protocols via sockets except NPN+ALPN

 SSLv2      not offered (OK)
 SSLv3      not offered (OK)
 TLS 1      offered (deprecated)
 TLS 1.1    offered (deprecated)
 TLS 1.2    offered (OK)
 TLS 1.3    offered (OK): final
 NPN/SPDY   h2, http/1.1 (advertised)
 ALPN/HTTP2 h2, http/1.1 (offered)

 Testing cipher categories

 NULL ciphers (no encryption)                  not offered (OK)
 Anonymous NULL Ciphers (no authentication)    not offered (OK)
 Export ciphers (w/o ADH+NULL)                 not offered (OK)
 LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export)  not offered (OK)
 Triple DES Ciphers / IDEA                     offered
 Obsoleted CBC ciphers (AES, ARIA etc.)        offered
 Strong encryption (AEAD ciphers) with no FS   offered (OK)
 Forward Secrecy strong encryption (AEAD ciphers)  offered (OK)
```

Figure 65: TLS 1.0, 1.1 et chiffrements obsolètes sur https://ifprofs.org.

Recommandations détaillées

Seulement les protocoles TLS 1.2 et 1.3 doivent être proposés et les suites cryptographiques suivantes sont recommandées par l'ANSSI selon la version de TLS :

Code TLS	Suite cryptographique
0x1302	TLS_AES_256_GCM_SHA384
0x1301	TLS_AES_128_GCM_SHA256
0x1304	TLS_AES_128_CCM_SHA256
0x1303	TLS_CHACHA20_POLY1305_SHA256

Figure 66 - Suites TLS 1.3 recommandées.

Code TLS	Suite cryptographique
0xC02C	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
0xC02B	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
0xC0AD	TLS_ECDHE_ECDSA_WITH_AES_256_CCM
0xC0AC	TLS_ECDHE_ECDSA_WITH_AES_128_CCM
0xCCA9	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256

Figure 67 - Suites TLS 1.2 recommandées avec un serveur disposant d'un certificat avec clé publique ECDSA.

Code TLS	Suite cryptographique
0xC030	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
0xC02F	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
0xCCA8	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256

Figure 68 - Suites TLS 1.2 recommandées avec un serveur disposant d'un certificat avec clé publique

Liens utiles

- https://mozilla.github.io/server-side-tls/ssl-config-generator/
- https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.md
- https://www.owasp.org/index.php/Testing_for_Weak_SSL/TLS_Ciphers,_Insufficient_Transport_Layer_Protection_(OTG-CRYPST-001)
- https://cyber.gouv.fr/sites/default/files/2017/07/anssi-guide-recommandations_de_securite_relatives_a_tls-v1.2.pdf

4 Annexes A – Ressources utiles

- Fonctionnalité de déconnexion défaillante ou inexistante
  - https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/06-Session_Management_Testing/06-Testing_for_Logout_Functionality
  - https://wiki.owasp.org/index.php/Logout

- Injection de code HTML/JavaScript (XSS)
  - https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
  - https://security.stackexchange.com/questions/143923/whats-the-difference-between-escaping-filtering-validating-and-sanitizing
  - https://www.wordfence.com/learn/how-to-prevent-cross-site-scripting-attacks/#functions-to-escape-and-sanitize-your-data

- Utilisation de composants avec des vulnérabilités connues
  - https://www.owasp.org/index.php/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities

- Énumération d'utilisateurs
  - https://www.owasp.org/index.php/Testing_for_User_Enumeration_and_Guessable_User_Account_(OWASP-AT-002)
  - https://www.hashbangcode.com/article/drupal-9-preventing-enumeration-attacks
  - https://owasp.org/www-community/attacks/Brute_force_attack

- Absence d'EDR ou d'antivirus sur les serveurs
  - N/A

- Fuites d'informations techniques
  - https://www.owasp.org/index.php/Top_10_2007-Information_Leakage_and_Improper_Error_Handling
  - https://www.netsparker.com/blog/web-security/information-disclosure-issues-attacks/
  - https://www.acunetix.com/blog/articles/configure-web-server-disclose-identity/
  - https://www.hacksplaining.com/prevention/information-leakage

- Injection CSV
  - https://owasp.org/www-community/attacks/CSV_Injection

- Exposition de services d'administration sur Internet
  - https://cyber.gouv.fr/publications/recommandations-relatives-ladministration-securisee-des-si

- En-tête(s) de sécurité HTTP manquant(s)
  - https://www.keycdn.com/blog/http-security-headers
  - https://unbonhacker.com/posts/security-headers/
  - https://developer.mozilla.org/en-US/docs/Web/HTTP/Permissions_Policy

- https://owasp.org/www-project-secure-headers/

- Vulnérabilités SSL / TLS
  - https://mozilla.github.io/server-side-tls/ssl-config-generator/
  - https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.md
  - https://www.owasp.org/index.php/Testing_for_Weak_SSL/TLS_Ciphers,_Insufficient_Transport_Layer_Protection_(OTG-CRYPST-001)
  - https://cyber.gouv.fr/sites/default/files/2017/07/anssi-guide-recommandations_de_securite_relatives_a_tls-v1.2.pdf

5 Annexes B – Reconnaissance

Cartographie du périmètre

Une cartographie de la surface d'audit externe est réalisée via `nmap` :

ifprofs.org
IP: 172.66.41.38
Number of open ports: 13

Expand Host

Port	Service	Product	Version	Links
80	**http**	Cloudflare http proxy		http (IP) http (Hostname) https (IP) https (Hostname)
443	**http**	Cloudflare http proxy		https (IP) https (Hostname)
2052	**http**	Cloudflare http proxy		http (IP) http (Hostname) https (IP) https (Hostname)
2053	**http**	nginx		http (IP) http (Hostname) https (IP) https (Hostname)
2082	**http**	Cloudflare http proxy		http (IP) http (Hostname) https (IP) https (Hostname)
2083	**http**	nginx		http (IP) http (Hostname) https (IP) https (Hostname)
2086	**http**	Cloudflare http proxy		http (IP) http (Hostname) https (IP) https (Hostname)
2087	**http**	nginx		http (IP) http (Hostname) https (IP) https (Hostname)
2095	**http**	Cloudflare http proxy		http (IP) http (Hostname) https (IP) https (Hostname)
2096	**http**	nginx		http (IP) http (Hostname) https (IP) https (Hostname)
8080	**http**	Cloudflare http proxy		http (IP) http (Hostname) https (IP) https (Hostname)
8443	**http**	Cloudflare http proxy		https (IP) https (Hostname)
8880	**http**	Cloudflare http proxy		http (IP) http (Hostname) https (IP) https (Hostname)

Figure 70 : Résultats du scan de ports TCP (2).

A noter que sur l'hôte ifprogs.fr de très nombreux sont ouverts. La pertinence de ce nombre est à évaluer en fonction des besoins de production.