Rapport de Tests de Vulnérabilité en Cybersécurité

Madame, Monsieur,

Nous vous remercions de la confiance que vous nous avez témoignée en nous confiant cette mission de test de vulnérabilité de votre entité établie en France face aux cyber-risques.

Le présent compte-rendu a pour but de partager avec vous nos principales conclusions et d’échanger avec vous sur d’éventuelles actions d’amélioration.

Les conclusions auxquelles nous aboutissons relèvent d’une démarche analytique et d’une observation globale des cybermenaces actuelles en l’état actuel de l’art. Elles ne traduisent pas de certitudes mais des éventualités destinées à vous aider dans la prise de vos décisions stratégiques.

Table des matières

[***1***](#_inv8uuyxl2k9) ***Introduction 4***

[***2***](#_f2a91jnkbhed) ***Synthèse des tests de vulnérabilité 5***

[**2.1**](#_h78uftvi9vg8) **Analyse des ports et des enregistrements DNS 5**

[**2.2**](#_sqqbako9zexy) **Analyse de la configuration du serveur Internet 5**

[***3***](#_2zdozlet23oo) ***Données détaillées 7***

[**3.1**](#_qxheg3uky1aj) **Statut des ports 7**

[**3.2**](#_jlnaw7cwj297) **Enregistrement DNS 7**

[**3.3**](#_cv9qc9vnx5rm) **Configuration du site Internet 8**

[***4***](#_unk0usmjgv3q) ***Bonnes pratiques en cas d’indicent de sécurité 11***

[**4.1**](#_19y8rapf0jbi) **En cas de suspicion d’attaque par ransomware : 11**

[**4.2**](#_p85zs3fmhw5k) **En cas de suspicion de piratage de compte en ligne 12**

[**4.3**](#_ua9a0orywwic) **En cas de suspicion de virus 13**

# Introduction

L'audit de cybersécurité réalisé dans l'entreprise aaaa a pour objectif principal d’évaluer la conformité de ses systèmes, politiques et pratiques de sécurité avec les exigences de la norme **ISO/IEC 27001**. Cette norme internationale spécifie les meilleures pratiques pour la gestion de la sécurité de l’information, et vise à assurer la confidentialité, l'intégrité et la disponibilité des informations au sein de l’organisation.

L’audit a été mené dans le cadre de la stratégie de gestion des risques de l'entreprise et a pour but de :

- Identifier les écarts potentiels par rapport aux exigences de la norme ISO 27001.

- Évaluer l'efficacité du Système de Management de la Sécurité de l'Information (SMSI) en place.

- Proposer des recommandations pour corriger les éventuelles non-conformités et améliorer les processus de sécurité.

L’audit s’est concentré sur les principaux domaines couverts par l’ISO 27001, à savoir :

- La politique de sécurité de l’information,

- La gestion des risques,

- La gestion des actifs,

- Les contrôles d’accès,

- La sécurité physique et environnementale,

- La gestion des incidents de sécurité,

- Les plans de continuité d’activité.

Les méthodologies utilisées incluent une analyse documentaire et des entretiens avec les parties prenantes clés. Les résultats de cet audit permettront à aaaa de prendre des décisions éclairées concernant les améliorations à apporter à son SMSI, en vue de renforcer la sécurité globale de ses informations sensibles et d'obtenir ou de maintenir la certification ISO 27001.

# Synthèse des tests de vulnérabilité

Les tests se sont déroulés le 17/09/2025 sur le site Internet suivant : testphp.vulnweb.com.

L’analyse de résolution DNS a indiqué l’adresse IP du serveur sur lesquels ont été effectués les tests : \*IP\_ADDRESS\*

## Analyse des ports et des enregistrements DNS

L’objectif de cette analyse est de vérifier la cohérence entre le statut des ports et les enregistrements DNS.

| Test | Description | Analyse | Recommandation |
| --- | --- | --- | --- |
| SMTP/POP3/IMAP | Vérifie la non-utilisation des ports non sécurisés | ✅ Ports non sécurisés fermés | - |
| SMTPS/POP3S/IMAPS | Vérifie l'utilisation de ports sécurisés avec certificat valide | ✅ Utilisation un service de mail externe, avec ports fermés | - |
| SPF | Vérification de l'enregistrement SPF | ✅ Configuration SPF correcte | - |
| DKIM | Vérification de l'enregistrement DKIM | ❌ Problème de configuration DKIM détecté. | Ajouter un enregistrement DKIM |
| DMARC | Vérification de l'enregistrement DMARC | ❌ Problème de configuration DMARC détecté. | Ajouter un enregistrement DMARC avec l'option p=reject |

## Analyse de la configuration du serveur Internet

La configuration du site internet présente les vulnérabilités suivantes :

| Test | Description | Risque | Num Cas détectés | Recommendation |
| --- | --- | --- | --- | --- |
| Missing Anti-clickjacking Header | The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options. | Medium | 44 | Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app. If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive. |
| Content Security Policy (CSP) Header Not Set | Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files. | Medium | 48 | Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header. |
| Absence of Anti-CSRF Tokens | No Anti-CSRF tokens were found in a HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf. CSRF attacks are effective in a number of situations, including: \* The victim has an active session on the target site. \* The victim is authenticated via HTTP auth on the target site. \* The victim is on the same local network as the target site. CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy. | Medium | 40 | Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard. Phase: Implementation Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script. Phase: Architecture and Design Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330). Note that this can be bypassed using XSS. Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation. Note that this can be bypassed using XSS. Use the ESAPI Session Management control. This control includes a component for CSRF. Do not use the GET method for any request that triggers a state change. Phase: Implementation Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons. |
| Cross Site Scripting (Reflected) | Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology. When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Applications utilizing browser object instances which load content from the file system may execute code under the local machine zone allowing for system compromise. There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based. Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, or visit a malicious web page containing a web form, which when posted to the vulnerable site, will mount the attack. Using a malicious form will oftentimes take place when the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by using an embedded client, such as Adobe Flash. Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time. Examples of an attacker's favorite targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not required to interact with any additional site/link (e.g. an attacker site or a malicious link sent via email), just simply view the web page containing the code. | High | 19 | Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI Encoding module, and Apache Wicket. Phases: Implementation; Architecture and Design Understand the context in which your data will be used and the encoding that will be expected. This is especially important when transmitting data between different components, or when generating outputs that can contain multiple encodings at the same time, such as web pages or multi-part mail messages. Study all expected communication protocols and data representations to determine the required encoding strategies. For any data that will be output to another web page, especially any data that was received from external inputs, use the appropriate encoding on all non-alphanumeric characters. Consult the XSS Prevention Cheat Sheet for more details on the types of encoding and escaping that are needed. Phase: Architecture and Design For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server. If available, use structured mechanisms that automatically enforce the separation between data and code. These mechanisms may be able to provide the relevant quoting, encoding, and validation automatically, instead of relying on the developer to provide this capability at every point where output is generated. Phase: Implementation For every web page that is generated, use and specify a character encoding such as ISO-8859-1 or UTF-8. When an encoding is not specified, the web browser may choose a different encoding by guessing which encoding is actually being used by the web page. This can cause the web browser to treat certain sequences as special, opening up the client to subtle XSS attacks. See CWE-116 for more mitigations related to encoding/escaping. To help mitigate XSS attacks against the user's session cookie, set the session cookie to be HttpOnly. In browsers that support the HttpOnly feature (such as more recent versions of Internet Explorer and Firefox), this attribute can prevent the user's session cookie from being accessible to malicious client-side scripts that use document.cookie. This is not a complete solution, since HttpOnly is not supported by all browsers. More importantly, XMLHTTPRequest and other powerful browser technologies provide read access to HTTP headers, including the Set-Cookie header in which the HttpOnly flag is set. Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use an allow list of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a deny list). However, deny lists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. As an example of business rule logic, "boat" may be syntactically valid because it only contains alphanumeric characters, but it is not valid if you are expecting colors such as "red" or "blue." Ensure that you perform input validation at well-defined interfaces within the application. This will help protect the application even if a component is reused or moved elsewhere. |
| SQL Injection - MySQL | SQL injection may be possible. | High | 9 | Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side. If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?' If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries. If database Stored Procedures can be used, use them. Do \*not\* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality! Do not create dynamic SQL queries using simple string concatenation. Escape all data received from the client. Apply an 'allow list' of allowed characters, or a 'deny list' of disallowed characters in user input. Apply the principle of least privilege by using the least privileged database user possible. In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact. Grant the minimum database access that is necessary for the application. |
| SQL Injection - MySQL (Time Based) | SQL injection may be possible. | High | 2 | Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side. If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?' If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries. If database Stored Procedures can be used, use them. Do \*not\* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality! Do not create dynamic SQL queries using simple string concatenation. Escape all data received from the client. Apply an 'allow list' of allowed characters, or a 'deny list' of disallowed characters in user input. Apply the principle of least privilege by using the least privileged database user possible. In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact. Grant the minimum database access that is necessary for the application. |
| XSLT Injection | Injection using XSL transformations may be possible, and may allow an attacker to read system information, read and write files, or execute arbitrary code. | Medium | 2 | Sanitize and analyze every user input coming from any client-side. |

# Données détaillées

## Statut des ports

Le tableau ci-dessous donne la liste des ports les plus utilisés :

| Numéro du port | Service | Statut |
| --- | --- | --- |
| 80 | HTTP | Ouvert |
| 443 | HTTPS | Ouvert |
| 21 | FTP | Fermé |
| 22 | SSH | Fermé |
| 23 | Telnet | Fermé |
| 25 | SMTP | Fermé |
| 53 | DNS | Fermé |
| 110 | POP3 | Fermé |
| 143 | IMAP | Fermé |
| 445 | SMB | Fermé |
| 465 | SMTPS | Fermé |
| 993 | IMAPS | Fermé |
| 995 | POP3S | Fermé |
| 1433 | MSSQL | Fermé |
| 1521 | Oracle DB | Fermé |
| 3306 | MySQL | Fermé |
| 3389 | RDP | Fermé |
| 5432 | PostgreSQL | Fermé |
| 5900 | VNC | Fermé |
| 8080 | HTTP Proxy | Fermé |
| 8443 | HTTPS Proxy | Fermé |
| 27017 | MongoDB | Fermé |

## Enregistrement DNS

Le tableau ci-dessous donne la liste des enregistrements DNS :

| Type d'enregistrement | Valeur |
| --- | --- |
| A | 164.132.235.17 |
| AAAA | 2001:41d0:301::23 |
| MX | 5 mx2.mail.ovh.net. |
| MX | 1 mx1.mail.ovh.net. |
| MX | 100 mx3.mail.ovh.net. |
| NS | ns104.ovh.net. |
| NS | dns104.ovh.net. |
| TXT | 1|www.cybersafer.fr |
| SOA | dns104.ovh.net. tech.ovh.net. 2025072600 86400 3600 3600000 60 |
| SPF | v=spf1 include:spf.mailjet.com include:mx.ovh.com -all |
| DKIM | Aucune valeur |
| DMARC | Aucune valeur |
| DNSSEC | Activé |
| PTR | cluster023.hosting.ovh.net |

## Configuration du site Internet

Les cas détectés sont les suivants :

| Test | Cas détectés |
| --- | --- |
| Missing Anti-clickjacking Header | 1. URL : http://testphp.vulnweb.com/index.php  Paramètre : x-frame-options 2. URL : http://testphp.vulnweb.com  Paramètre : x-frame-options 3. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/  Paramètre : x-frame-options 4. URL : http://testphp.vulnweb.com/categories.php  Paramètre : x-frame-options 5. URL : http://testphp.vulnweb.com/disclaimer.php  Paramètre : x-frame-options 6. URL : http://testphp.vulnweb.com/artists.php  Paramètre : x-frame-options 7. URL : http://testphp.vulnweb.com/cart.php  Paramètre : x-frame-options 8. URL : http://testphp.vulnweb.com/guestbook.php  Paramètre : x-frame-options 9. URL : http://testphp.vulnweb.com/login.php  Paramètre : x-frame-options 10. URL : http://testphp.vulnweb.com/AJAX/index.php  Paramètre : x-frame-options 11. URL : http://testphp.vulnweb.com/hpp/  Paramètre : x-frame-options 12. URL : http://testphp.vulnweb.com/guestbook.php  Paramètre : x-frame-options 13. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/Details/color-printer/3/  Paramètre : x-frame-options 14. URL : http://testphp.vulnweb.com/artists.php?artist=3  Paramètre : x-frame-options 15. URL : http://testphp.vulnweb.com/listproducts.php?cat=1  Paramètre : x-frame-options 16. URL : http://testphp.vulnweb.com/listproducts.php?cat=3  Paramètre : x-frame-options 17. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/Details/web-camera-a4tech/2/  Paramètre : x-frame-options 18. URL : http://testphp.vulnweb.com/artists.php?artist=1  Paramètre : x-frame-options 19. URL : http://testphp.vulnweb.com/listproducts.php?cat=4  Paramètre : x-frame-options 20. URL : http://testphp.vulnweb.com/search.php?test=query  Paramètre : x-frame-options 21. URL : http://testphp.vulnweb.com/listproducts.php?cat=2  Paramètre : x-frame-options 22. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/Details/network-attached-storage-dlink/1/  Paramètre : x-frame-options 23. URL : http://testphp.vulnweb.com/artists.php?artist=2  Paramètre : x-frame-options 24. URL : http://testphp.vulnweb.com/signup.php  Paramètre : x-frame-options 25. URL : http://testphp.vulnweb.com/listproducts.php?artist=1  Paramètre : x-frame-options 26. URL : http://testphp.vulnweb.com/hpp/?pp=12  Paramètre : x-frame-options 27. URL : http://testphp.vulnweb.com/product.php?pic=6  Paramètre : x-frame-options 28. URL : http://testphp.vulnweb.com/listproducts.php?artist=2  Paramètre : x-frame-options 29. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/BuyProduct-1/  Paramètre : x-frame-options 30. URL : http://testphp.vulnweb.com/listproducts.php?artist=3  Paramètre : x-frame-options 31. URL : http://testphp.vulnweb.com/product.php?pic=2  Paramètre : x-frame-options 32. URL : http://testphp.vulnweb.com/product.php?pic=1  Paramètre : x-frame-options 33. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/BuyProduct-3/  Paramètre : x-frame-options 34. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/RateProduct-3.html  Paramètre : x-frame-options 35. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/BuyProduct-2/  Paramètre : x-frame-options 36. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/RateProduct-2.html  Paramètre : x-frame-options 37. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/RateProduct-1.html  Paramètre : x-frame-options 38. URL : http://testphp.vulnweb.com/product.php?pic=4  Paramètre : x-frame-options 39. URL : http://testphp.vulnweb.com/product.php?pic=3  Paramètre : x-frame-options 40. URL : http://testphp.vulnweb.com/product.php?pic=5  Paramètre : x-frame-options 41. URL : http://testphp.vulnweb.com/product.php?pic=7  Paramètre : x-frame-options 42. URL : http://testphp.vulnweb.com/secured/newuser.php  Paramètre : x-frame-options 43. URL : http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12  Paramètre : x-frame-options 44. URL : http://testphp.vulnweb.com/cart.php  Paramètre : x-frame-options ============================================================ |
| Content Security Policy (CSP) Header Not Set | 1. URL : http://testphp.vulnweb.com/sitemap.xml  Paramètre :  2. URL : http://testphp.vulnweb.com/robots.txt  Paramètre :  3. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/  Paramètre :  4. URL : http://testphp.vulnweb.com  Paramètre :  5. URL : http://testphp.vulnweb.com/categories.php  Paramètre :  6. URL : http://testphp.vulnweb.com/index.php  Paramètre :  7. URL : http://testphp.vulnweb.com/disclaimer.php  Paramètre :  8. URL : http://testphp.vulnweb.com/artists.php  Paramètre :  9. URL : http://testphp.vulnweb.com/privacy.php  Paramètre :  10. URL : http://testphp.vulnweb.com/high  Paramètre :  11. URL : http://testphp.vulnweb.com/guestbook.php  Paramètre :  12. URL : http://testphp.vulnweb.com/cart.php  Paramètre :  13. URL : http://testphp.vulnweb.com/hpp/  Paramètre :  14. URL : http://testphp.vulnweb.com/AJAX/index.php  Paramètre :  15. URL : http://testphp.vulnweb.com/login.php  Paramètre :  16. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/Details/color-printer/3/  Paramètre :  17. URL : http://testphp.vulnweb.com/artists.php?artist=3  Paramètre :  18. URL : http://testphp.vulnweb.com/guestbook.php  Paramètre :  19. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/Details/web-camera-a4tech/2/  Paramètre :  20. URL : http://testphp.vulnweb.com/listproducts.php?cat=1  Paramètre :  21. URL : http://testphp.vulnweb.com/listproducts.php?cat=3  Paramètre :  22. URL : http://testphp.vulnweb.com/artists.php?artist=1  Paramètre :  23. URL : http://testphp.vulnweb.com/listproducts.php?cat=4  Paramètre :  24. URL : http://testphp.vulnweb.com/search.php?test=query  Paramètre :  25. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/Details/network-attached-storage-dlink/1/  Paramètre :  26. URL : http://testphp.vulnweb.com/listproducts.php?cat=2  Paramètre :  27. URL : http://testphp.vulnweb.com/artists.php?artist=2  Paramètre :  28. URL : http://testphp.vulnweb.com/signup.php  Paramètre :  29. URL : http://testphp.vulnweb.com/listproducts.php?artist=1  Paramètre :  30. URL : http://testphp.vulnweb.com/hpp/?pp=12  Paramètre :  31. URL : http://testphp.vulnweb.com/product.php?pic=6  Paramètre :  32. URL : http://testphp.vulnweb.com/listproducts.php?artist=2  Paramètre :  33. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/BuyProduct-1/  Paramètre :  34. URL : http://testphp.vulnweb.com/listproducts.php?artist=3  Paramètre :  35. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/BuyProduct-3/  Paramètre :  36. URL : http://testphp.vulnweb.com/product.php?pic=2  Paramètre :  37. URL : http://testphp.vulnweb.com/product.php?pic=1  Paramètre :  38. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/RateProduct-2.html  Paramètre :  39. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/BuyProduct-2/  Paramètre :  40. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/RateProduct-3.html  Paramètre :  41. URL : http://testphp.vulnweb.com/Mod\_Rewrite\_Shop/RateProduct-1.html  Paramètre :  42. URL : http://testphp.vulnweb.com/product.php?pic=3  Paramètre :  43. URL : http://testphp.vulnweb.com/product.php?pic=4  Paramètre :  44. URL : http://testphp.vulnweb.com/product.php?pic=5  Paramètre :  45. URL : http://testphp.vulnweb.com/product.php?pic=7  Paramètre :  46. URL : http://testphp.vulnweb.com/secured/newuser.php  Paramètre :  47. URL : http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12  Paramètre :  48. URL : http://testphp.vulnweb.com/cart.php  Paramètre :  ============================================================ |
| Absence of Anti-CSRF Tokens | 1. URL : http://testphp.vulnweb.com  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 2. URL : http://testphp.vulnweb.com/index.php  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 3. URL : http://testphp.vulnweb.com/categories.php  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 4. URL : http://testphp.vulnweb.com/disclaimer.php  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 5. URL : http://testphp.vulnweb.com/artists.php  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 6. URL : http://testphp.vulnweb.com/cart.php  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 7. URL : http://testphp.vulnweb.com/guestbook.php  Paramètre :   Preuve : <form action="" method="post" name="faddentry"> 8. URL : http://testphp.vulnweb.com/guestbook.php  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 9. URL : http://testphp.vulnweb.com/login.php  Paramètre :   Preuve : <form name="loginform" method="post" action="userinfo.php"> 10. URL : http://testphp.vulnweb.com/login.php  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 11. URL : http://testphp.vulnweb.com/guestbook.php  Paramètre :   Preuve : <form action="" method="post" name="faddentry"> 12. URL : http://testphp.vulnweb.com/artists.php?artist=3  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 13. URL : http://testphp.vulnweb.com/listproducts.php?cat=1  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 14. URL : http://testphp.vulnweb.com/guestbook.php  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 15. URL : http://testphp.vulnweb.com/listproducts.php?cat=3  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 16. URL : http://testphp.vulnweb.com/artists.php?artist=1  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 17. URL : http://testphp.vulnweb.com/listproducts.php?cat=4  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 18. URL : http://testphp.vulnweb.com/search.php?test=query  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 19. URL : http://testphp.vulnweb.com/listproducts.php?cat=2  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 20. URL : http://testphp.vulnweb.com/artists.php?artist=2  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 21. URL : http://testphp.vulnweb.com/signup.php  Paramètre :   Preuve : <form name="form1" method="post" action="/secured/newuser.php"> 22. URL : http://testphp.vulnweb.com/signup.php  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 23. URL : http://testphp.vulnweb.com/listproducts.php?artist=1  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 24. URL : http://testphp.vulnweb.com/product.php?pic=6  Paramètre :   Preuve : <form name='f\_addcart' method='POST' action='cart.php'> 25. URL : http://testphp.vulnweb.com/product.php?pic=6  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 26. URL : http://testphp.vulnweb.com/listproducts.php?artist=2  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 27. URL : http://testphp.vulnweb.com/listproducts.php?artist=3  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 28. URL : http://testphp.vulnweb.com/product.php?pic=1  Paramètre :   Preuve : <form name='f\_addcart' method='POST' action='cart.php'> 29. URL : http://testphp.vulnweb.com/product.php?pic=1  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 30. URL : http://testphp.vulnweb.com/product.php?pic=2  Paramètre :   Preuve : <form name='f\_addcart' method='POST' action='cart.php'> 31. URL : http://testphp.vulnweb.com/product.php?pic=2  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 32. URL : http://testphp.vulnweb.com/product.php?pic=3  Paramètre :   Preuve : <form name='f\_addcart' method='POST' action='cart.php'> 33. URL : http://testphp.vulnweb.com/product.php?pic=3  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 34. URL : http://testphp.vulnweb.com/product.php?pic=4  Paramètre :   Preuve : <form name='f\_addcart' method='POST' action='cart.php'> 35. URL : http://testphp.vulnweb.com/product.php?pic=4  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 36. URL : http://testphp.vulnweb.com/product.php?pic=5  Paramètre :   Preuve : <form name='f\_addcart' method='POST' action='cart.php'> 37. URL : http://testphp.vulnweb.com/product.php?pic=5  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 38. URL : http://testphp.vulnweb.com/product.php?pic=7  Paramètre :   Preuve : <form name='f\_addcart' method='POST' action='cart.php'> 39. URL : http://testphp.vulnweb.com/product.php?pic=7  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> 40. URL : http://testphp.vulnweb.com/cart.php  Paramètre :   Preuve : <form action="search.php?test=query" method="post"> ============================================================ |
| Cross Site Scripting (Reflected) | 1. URL : http://testphp.vulnweb.com/secured/newuser.php  Paramètre : uuname  Payload : </li><scrIpt>alert(1);</scRipt><li>  Preuve : </li><scrIpt>alert(1);</scRipt><li> 2. URL : http://testphp.vulnweb.com/guestbook.php  Paramètre : name  Payload : </strong><scrIpt>alert(1);</scRipt><strong>  Preuve : </strong><scrIpt>alert(1);</scRipt><strong> 3. URL : http://testphp.vulnweb.com/search.php?test=%27%22%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E  Paramètre : test  Payload : '"<scrIpt>alert(1);</scRipt>  Preuve : '"<scrIpt>alert(1);</scRipt> 4. URL : http://testphp.vulnweb.com/guestbook.php  Paramètre : text  Payload : </td><scrIpt>alert(1);</scRipt><td>  Preuve : </td><scrIpt>alert(1);</scRipt><td> 5. URL : http://testphp.vulnweb.com/search.php?test=query  Paramètre : searchFor  Payload : </h2><scrIpt>alert(1);</scRipt><h2>  Preuve : </h2><scrIpt>alert(1);</scRipt><h2> 6. URL : http://testphp.vulnweb.com/userinfo.php  Paramètre : uname  Payload : '"<scrIpt>alert(1);</scRipt>  Preuve : '"<scrIpt>alert(1);</scRipt> 7. URL : http://testphp.vulnweb.com/userinfo.php  Paramètre : pass  Payload : '"<scrIpt>alert(1);</scRipt>  Preuve : '"<scrIpt>alert(1);</scRipt> 8. URL : http://testphp.vulnweb.com/secured/newuser.php  Paramètre : urname  Payload : </li><scrIpt>alert(1);</scRipt><li>  Preuve : </li><scrIpt>alert(1);</scRipt><li> 9. URL : http://testphp.vulnweb.com/secured/newuser.php  Paramètre : ucc  Payload : </li><scrIpt>alert(1);</scRipt><li>  Preuve : </li><scrIpt>alert(1);</scRipt><li> 10. URL : http://testphp.vulnweb.com/artists.php?artist=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E  Paramètre : artist  Payload : <scrIpt>alert(1);</scRipt>  Preuve : <scrIpt>alert(1);</scRipt> 11. URL : http://testphp.vulnweb.com/hpp/params.php?p=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E&pp=12  Paramètre : p  Payload : <scrIpt>alert(1);</scRipt>  Preuve : <scrIpt>alert(1);</scRipt> 12. URL : http://testphp.vulnweb.com/secured/newuser.php  Paramètre : uemail  Payload : </li><scrIpt>alert(1);</scRipt><li>  Preuve : </li><scrIpt>alert(1);</scRipt><li> 13. URL : http://testphp.vulnweb.com/listproducts.php?artist=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E  Paramètre : artist  Payload : <scrIpt>alert(1);</scRipt>  Preuve : <scrIpt>alert(1);</scRipt> 14. URL : http://testphp.vulnweb.com/product.php?pic=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E  Paramètre : pic  Payload : <scrIpt>alert(1);</scRipt>  Preuve : <scrIpt>alert(1);</scRipt> 15. URL : http://testphp.vulnweb.com/listproducts.php?cat=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E  Paramètre : cat  Payload : <scrIpt>alert(1);</scRipt>  Preuve : <scrIpt>alert(1);</scRipt> 16. URL : http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E  Paramètre : pp  Payload : <scrIpt>alert(1);</scRipt>  Preuve : <scrIpt>alert(1);</scRipt> 17. URL : http://testphp.vulnweb.com/secured/newuser.php  Paramètre : uphone  Payload : </li><scrIpt>alert(1);</scRipt><li>  Preuve : </li><scrIpt>alert(1);</scRipt><li> 18. URL : http://testphp.vulnweb.com/hpp/?pp=%22+onMouseOver%3D%22alert%281%29%3B  Paramètre : pp  Payload : " onMouseOver="alert(1);  Preuve : " onMouseOver="alert(1); 19. URL : http://testphp.vulnweb.com/secured/newuser.php  Paramètre : uaddress  Payload : </li><scrIpt>alert(1);</scRipt><li>  Preuve : </li><scrIpt>alert(1);</scRipt><li> ============================================================ |
| SQL Injection - MySQL | 1. URL : http://testphp.vulnweb.com/userinfo.php  Paramètre : uname  Payload : '  Preuve : You have an error in your SQL syntax 2. URL : http://testphp.vulnweb.com/search.php?test=%27  Paramètre : test  Payload : '  Preuve : You have an error in your SQL syntax 3. URL : http://testphp.vulnweb.com/secured/newuser.php  Paramètre : uuname  Payload : '  Preuve : You have an error in your SQL syntax 4. URL : http://testphp.vulnweb.com/userinfo.php  Paramètre : pass  Payload : '  Preuve : You have an error in your SQL syntax 5. URL : http://testphp.vulnweb.com/search.php?test=query  Paramètre : searchFor  Payload : ZAP'  Preuve : You have an error in your SQL syntax 6. URL : http://testphp.vulnweb.com/artists.php?artist=%27  Paramètre : artist  Payload : '  Preuve : You have an error in your SQL syntax 7. URL : http://testphp.vulnweb.com/listproducts.php?cat=%27  Paramètre : cat  Payload : '  Preuve : You have an error in your SQL syntax 8. URL : http://testphp.vulnweb.com/product.php?pic=%27  Paramètre : pic  Payload : '  Preuve : You have an error in your SQL syntax 9. URL : http://testphp.vulnweb.com/listproducts.php?artist=%27  Paramètre : artist  Payload : '  Preuve : You have an error in your SQL syntax ============================================================ |
| SQL Injection - MySQL (Time Based) | 1. URL : http://testphp.vulnweb.com/secured/newuser.php  Paramètre : uuname  Payload : ZAP' / sleep(15) / ' 2. URL : http://testphp.vulnweb.com/userinfo.php  Paramètre : uname  Payload : ZAP' / sleep(15) / ' ============================================================ |
| XSLT Injection | 1. URL : http://testphp.vulnweb.com/showimage.php?file=%3Cxsl%3Avalue-of+select%3D%22document%28%27http%3A%2F%2Ftestphp.vulnweb.com%3A22%27%29%22%2F%3E&size=160  Paramètre : file  Payload : <xsl:value-of select="document('http://testphp.vulnweb.com:22')"/>  Preuve : failed to open stream 2. URL : http://testphp.vulnweb.com/showimage.php?file=%3Cxsl%3Avalue-of+select%3D%22document%28%27http%3A%2F%2Ftestphp.vulnweb.com%3A22%27%29%22%2F%3E  Paramètre : file  Payload : <xsl:value-of select="document('http://testphp.vulnweb.com:22')"/>  Preuve : failed to open stream ============================================================ |

# Bonnes pratiques en cas d’indicent de sécurité

## En cas de suspicion d’attaque par ransomware :

En cas de suspicion d'attaque par ransomware, il est essentiel d'agir rapidement pour minimiser les dommages. Voici les bonnes pratiques à adopter :

**1. Isoler immédiatement les systèmes affectés**

* **Déconnecter du réseau** : Coupez l'accès à Internet et au réseau pour empêcher la propagation du ransomware sur d'autres machines.
* **Débrancher les périphériques externes** : Retirez les clés USB, disques externes et autres périphériques connectés.

**2. Bloquer les comptes compromis**

* Désactivez les comptes d'utilisateurs susceptibles d’être compromis pour limiter la propagation via des droits d'accès.

**3. Informer les équipes IT et la direction**

* Prévenez immédiatement votre équipe de sécurité informatique et, si nécessaire, votre direction pour activer le plan de réponse à un incident.

**4. Éviter de redémarrer les systèmes**

* Ne redémarrez pas les systèmes affectés, car cela pourrait aggraver la situation ou empêcher une analyse plus approfondie.

**5. Identifier et analyser l’attaque**

* Utilisez des outils de détection pour identifier le ransomware en cause.
* Collectez des preuves sans interagir directement avec les fichiers infectés (captures d'écran, logs, etc.).

**6. Utiliser des sauvegardes sûres**

* Si vous avez des sauvegardes récentes et non compromises, utilisez-les pour restaurer vos systèmes une fois que l'infection est sous contrôle.

**7. Ne pas payer la rançon**

* Les autorités recommandent **de ne pas payer la rançon**, car cela n'assure pas toujours la récupération des données et finance les activités criminelles.

**8. Contacter les autorités**

* Informez les autorités compétentes (police, agences de cybersécurité locales, etc.) pour signaler l'incident.

**9. Mettre à jour et renforcer la sécurité**

* Assurez-vous que vos systèmes sont à jour en termes de correctifs de sécurité, et installez des solutions de sécurité fiables (antivirus, pare-feu, etc.).
* Changez les mots de passe après l’incident.

**10. Former et sensibiliser les utilisateurs**

* Sensibilisez vos équipes aux techniques d'attaque (phishing, etc.) et formez-les sur les bonnes pratiques de sécurité pour éviter une récidive.

Ces étapes permettent de limiter les dommages, contenir l'attaque et éviter sa propagation tout en facilitant la récupération des données.

## En cas de suspicion de piratage de compte en ligne

En cas de suspicion de piratage d'un compte en ligne, il est important d'agir rapidement pour sécuriser vos informations et limiter les conséquences. Voici les bonnes pratiques à suivre :

**1. Changer immédiatement le mot de passe**

* Accédez au compte et changez le mot de passe **immédiatement**. Choisissez un mot de passe fort, unique, composé de lettres majuscules/minuscules, de chiffres et de caractères spéciaux.
* Si vous ne pouvez pas accéder au compte, utilisez la fonction "mot de passe oublié" pour réinitialiser votre mot de passe.

**2. Déconnecter les sessions ouvertes**

* Si le service le permet, déconnectez toutes les sessions ouvertes à partir de votre compte (souvent disponible dans les paramètres de sécurité).

**3. Activer la double authentification (2FA)**

* Si ce n'est pas déjà fait, activez l'authentification à deux facteurs (2FA). Cela ajoutera une couche de sécurité supplémentaire en nécessitant un code supplémentaire pour se connecter.

**4. Vérifier les activités récentes**

* Consultez l’historique des connexions et des activités récentes sur votre compte pour détecter des accès non autorisés.
* Si vous trouvez des actions suspectes (modification des informations, connexions depuis des emplacements inhabituels), signalez-les immédiatement.

**5. Vérifier et sécuriser les informations de récupération**

* Vérifiez que les adresses e-mail, numéros de téléphone ou autres informations de récupération associés à votre compte n’ont pas été modifiés.
* Si besoin, mettez à jour ces informations pour assurer que vous êtes le seul à pouvoir récupérer votre compte.

**6. Vérifier les permissions et les applications tierces**

* Consultez la liste des appareils connectés et des applications tierces ayant accès à votre compte. Révoquez les accès des appareils ou applications que vous ne reconnaissez pas.

**7. Avertir vos contacts**

* Si le compte piraté est un compte de messagerie ou de réseau social, informez vos contacts de ne pas ouvrir de liens ou de fichiers reçus de votre part récemment.

**8. Surveiller d'autres comptes**

* Si vous avez utilisé le même mot de passe pour d'autres comptes (même si c'est déconseillé), changez-le immédiatement sur ces comptes.
* Surveillez vos comptes bancaires ou d'autres services sensibles pour détecter toute activité inhabituelle.

**9. Mettre à jour et sécuriser vos appareils**

* Il est possible que votre appareil soit infecté par un logiciel malveillant qui a permis l'accès non autorisé à votre compte. Mettez à jour votre antivirus et exécutez une analyse complète de votre appareil pour détecter et supprimer tout malware.

Adopter ces pratiques permet de contenir la situation rapidement, de protéger vos autres comptes et de restaurer la sécurité de votre compte compromis.

## En cas de suspicion de virus

En cas de suspicion de virus sur un appareil (ordinateur, smartphone, etc.), il est essentiel de réagir rapidement pour minimiser les risques de perte de données ou de compromission. Voici les bonnes pratiques à adopter :

**1. Isoler l'appareil infecté**

* **Déconnecter de l'Internet** : Coupez l'accès à Internet (Wi-Fi, câble Ethernet) pour empêcher le virus de se propager ou d’envoyer des données à des serveurs distants.
* **Déconnecter les périphériques externes** : Débranchez les clés USB, disques externes ou autres appareils connectés pour éviter la propagation du virus.

**2. Ne pas exécuter de programmes inconnus**

* Évitez d'ouvrir des fichiers ou de lancer des programmes qui semblent suspects ou qui ne sont pas familiers, car ils pourraient aggraver l'infection.

**3. Passer un scan antivirus complet**

* **Mettre à jour l'antivirus** : Avant de lancer un scan, assurez-vous que votre logiciel antivirus est à jour avec les dernières définitions de virus.
* **Effectuer un scan complet** : Lancez un scan complet de l’appareil pour détecter et éliminer les virus ou logiciels malveillants.

**4. Utiliser des outils spécifiques pour la désinfection**

* Si l'antivirus ne peut pas supprimer le virus, vous pouvez essayer des outils spécialisés comme Malwarebytes ou AdwCleaner pour détecter et enlever les logiciels malveillants plus résistants.

**5. Mettre à jour le système d'exploitation et les logiciels**

* Une fois le virus supprimé, assurez-vous que votre système d’exploitation, vos navigateurs et autres logiciels sont à jour avec les derniers correctifs de sécurité.

**6. Changer les mots de passe**

* Si vous pensez que des données sensibles (comme vos identifiants de connexion) ont pu être compromises, changez immédiatement les mots de passe de vos comptes depuis un appareil sûr.

**7. Analyser les e-mails et les fichiers récents**

* Identifiez la source potentielle du virus. Si vous avez récemment ouvert un e-mail ou téléchargé un fichier suspect, supprimez-les et scannez votre appareil.

**8. Supprimer les fichiers suspects**

* Si vous identifiez des fichiers ou programmes suspects, supprimez-les immédiatement. Faites-le en mode sans échec si nécessaire pour éviter que le virus ne se réactive.

**9. Restaurer depuis une sauvegarde**

* Si l’infection est grave et ne peut pas être résolue, vous pouvez envisager de restaurer l’appareil à partir d’une sauvegarde propre effectuée avant l’infection. Assurez-vous que la sauvegarde est exempte de virus.

**10. Réinitialiser l’appareil si nécessaire**

* Si aucune des méthodes ci-dessus ne fonctionne et que le virus persiste, envisagez de réinitialiser l’appareil ou de réinstaller le système d’exploitation. Assurez-vous d’avoir sauvegardé vos données avant cette opération.

Ces mesures aident à détecter, contenir et éliminer les virus, tout en renforçant la sécurité de l'appareil pour éviter de futures infections.