Rapport de Sécurité ShadowTrace

1. Authentication Request Identified

Risque: Informational

URL: http://localhost:8080/login.php

Description: The given request has been identified as an authentication request. The 'Other Info' field contains a set of key=value lines which identify any relevant fields. If the request is in a context which has an Authentication Method set to "Auto-Detect" then this rule will change the authentication to match the request identified.

Solution: This is an informational alert rather than a vulnerability and so there is nothing to fix.

2. Server Leaks Version Information via "Server" HTTP Response Header Field

Risque: Low

URL: http://localhost:8080/robots.txt

Description: The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.

Solution: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

3. Content Security Policy (CSP) Header Not Set

Risque: Medium

URL: http://localhost:8080/sitemap.xml

Description: 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.

Solution: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

4. Session Management Response Identified

Risque: Informational

URL: http://localhost:8080/

Description: The given response has been identified as containing a session management token. The 'Other Info' field contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect" then this rule will change the session management to use the tokens identified.

Solution: This is an informational alert rather than a vulnerability and so there is nothing to fix.

5. Session Management Response Identified

Risque: Informational

URL: http://localhost:8080/

Description: The given response has been identified as containing a session management token. The 'Other Info' field contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect" then this rule will change the session management to use the tokens identified.

Solution: This is an informational alert rather than a vulnerability and so there is nothing to fix.

6. Session Management Response Identified

Risque: Informational

URL: http://localhost:8080/

Description: The given response has been identified as containing a session management token. The 'Other Info' field contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect" then this rule will change the session management to use the tokens identified.

Solution: This is an informational alert rather than a vulnerability and so there is nothing to fix.

7. Server Leaks Version Information via "Server" HTTP Response Header Field

Risque: Low

URL: http://localhost:8080/dvwa/images/login_logo.png

Description: The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.

Solution: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

8. Cookie No HttpOnly Flag

Risque: Low

URL: http://localhost:8080/

Description: A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.

Solution: Ensure that the HttpOnly flag is set for all cookies.

9. Server Leaks Version Information via "Server" HTTP Response Header

Field

Risque: Low

URL: http://localhost:8080/dvwa/images/RandomStorm.png

Description: The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.

Solution: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

10. Missing Anti-clickjacking Header

Risque: Medium

URL: http://localhost:8080/login.php

Description: The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.

Solution: 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.

11. Server Leaks Version Information via "Server" HTTP Response Header Field

Risque: Low

URL: http://localhost:8080/dvwa/css/login.css

Description: The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.

Solution: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

12. X-Content-Type-Options Header Missing

Risque: Low

URL: http://localhost:8080/dvwa/images/login_logo.png

Description: The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Solution: Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

13. X-Content-Type-Options Header Missing

Risque: Low

URL: http://localhost:8080/dvwa/images/RandomStorm.png

Description: The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Solution: Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

14. Content Security Policy (CSP) Header Not Set

Risque: Medium

URL: http://localhost:8080/login.php

Description: 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.

Solution: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

15. X-Content-Type-Options Header Missing

Risque: Low

URL: http://localhost:8080/dvwa/css/login.css

Description: The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Solution: Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

16. X-Content-Type-Options Header Missing

Risque: Low

URL: http://localhost:8080/robots.txt

Description: The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Solution: Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

17. Cookie No HttpOnly Flag

Risque: Low

URL: http://localhost:8080/

Description: A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.

Solution: Ensure that the HttpOnly flag is set for all cookies.

18. Server Leaks Version Information via "Server" HTTP Response Header Field

Risque: Low

URL: http://localhost:8080/login.php

Description: The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.

Solution: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

19. Server Leaks Version Information via "Server" HTTP Response Header Field

Risque: Low

URL: http://localhost:8080/sitemap.xml

Description: The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers

identifying other vulnerabilities your web/application server is subject to.

Solution: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

20. Server Leaks Version Information via "Server" HTTP Response Header Field

Risque: Low

URL: http://localhost:8080/login.php

Description: The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.

Solution: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

21. Cookie without SameSite Attribute

Risque: Low

URL: http://localhost:8080/

Description: A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.

Solution: Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.

22. X-Content-Type-Options Header Missing

Risque: Low

URL: http://localhost:8080/login.php

Description: The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Solution: Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

23. Cookie without SameSite Attribute

Risque: Low

URL: http://localhost:8080/

Description: A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute

is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.

Solution: Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.

24. Server Leaks Version Information via "Server" HTTP Response Header Field

Risque: Low

URL: http://localhost:8080/

Description: The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.

Solution: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

25. Session Management Response Identified

Risque: Informational

URL: http://localhost:8080/

Description: The given response has been identified as containing a session management token. The 'Other Info' field contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect" then this rule will change the session management to use the tokens identified.

Solution: This is an informational alert rather than a vulnerability and so there is nothing to fix.

26. Session Management Response Identified

Risque: Informational

URL: http://localhost:8080/

Description: The given response has been identified as containing a session management token. The 'Other Info' field contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect" then this rule will change the session management to use the tokens identified.

Solution: This is an informational alert rather than a vulnerability and so there is nothing to fix.