Valnara Security Scan Report

Scan Details:

Target URL: http://testphp.vulnweb.com/
Scan Type: Ajax Spider Scan
Scan Depth: 5
Start Time: 2025-04-04 01:11:14
End Time: 2025-04-04 01:11:20

Risk Summary:

Medium Risk Vulnerabilities: 48
Low Risk Vulnerabilities: 58
Informational Risk Vulnerabilities: 20

Detailed Vulnerabilities:

Missing Anti-clickjacking Header
Risk Level: Medium
URL: http://testphp.vulnweb.com/
Remediation: 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.

Charset Mismatch (Header Versus Meta Content-Type Charset)
Risk Level: Informational
URL: http://testphp.vulnweb.com/
Remediation: Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Absence of Anti-CSRF Tokens
Risk Level: Medium
URL: http://testphp.vulnweb.com/
Remediation: 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.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/
Remediation: 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.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/sitemap.xml
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Missing Anti-clickjacking Header
Risk Level: Medium
URL: http://testphp.vulnweb.com/hpp/
Remediation: 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
Risk Level: Medium
URL: http://testphp.vulnweb.com/robots.txt
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Missing Anti-clickjacking Header
Risk Level: Medium

URL: http://testphp.vulnweb.com/AJAX/index.php
Remediation: 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.

Missing Anti-clickjacking Header
Risk Level: Medium
URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/
Remediation: 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.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/userinfo.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/style.css
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/sitemap.xml
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Missing Anti-clickjacking Header
Risk Level: Medium
URL: http://testphp.vulnweb.com/categories.php
Remediation: 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
Risk Level: Medium
URL: http://testphp.vulnweb.com/high
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Missing Anti-clickjacking Header
Risk Level: Medium
URL: http://testphp.vulnweb.com/cart.php

Remediation: 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.

Missing Anti-clickjacking Header

Risk Level: Medium
URL: http://testphp.vulnweb.com/disclaimer.php
Remediation: 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.

Missing Anti-clickjacking Header

Risk Level: Medium
URL: http://testphp.vulnweb.com/guestbook.php
Remediation: 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.

Missing Anti-clickjacking Header

Risk Level: Medium
URL: http://testphp.vulnweb.com/login.php
Remediation: 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.

Missing Anti-clickjacking Header

Risk Level: Medium
URL: http://testphp.vulnweb.com/artists.php
Remediation: 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.

Missing Anti-clickjacking Header

Risk Level: Medium
URL: http://testphp.vulnweb.com/index.php
Remediation: 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

Risk Level: Medium
URL: http://testphp.vulnweb.com/hpp/
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/robots.txt
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/userinfo.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Charset Mismatch (Header Versus Meta Content-Type Charset)
Risk Level: Informational
URL: http://testphp.vulnweb.com/AJAX/index.php
Remediation: Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/style.css
Remediation: 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.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/high
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/images/logo.gif
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Charset Mismatch (Header Versus Meta Content-Type Charset)
Risk Level: Informational
URL: http://testphp.vulnweb.com/cart.php

Remediation: Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

Charset Mismatch (Header Versus Meta Content-Type Charset)
Risk Level: Informational
URL: http://testphp.vulnweb.com/disclaimer.php
Remediation: Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

Charset Mismatch (Header Versus Meta Content-Type Charset)
Risk Level: Informational
URL: http://testphp.vulnweb.com/categories.php
Remediation: Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

Charset Mismatch (Header Versus Meta Content-Type Charset)
Risk Level: Informational
URL: http://testphp.vulnweb.com/artists.php
Remediation: Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

Charset Mismatch (Header Versus Meta Content-Type Charset)
Risk Level: Informational
URL: http://testphp.vulnweb.com/guestbook.php
Remediation: Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

Charset Mismatch (Header Versus Meta Content-Type Charset)
Risk Level: Informational
URL: http://testphp.vulnweb.com/login.php
Remediation: Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/privacy.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/hpp/
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Charset Mismatch (Header Versus Meta Content-Type Charset)
Risk Level: Informational
URL: http://testphp.vulnweb.com/index.php
Remediation: Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/AJAX/index.php

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

Missing Anti-clickjacking Header
Risk Level: Medium
URL: http://testphp.vulnweb.com/artists.php?artist=1
Remediation: 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.

Missing Anti-clickjacking Header
Risk Level: Medium
URL: http://testphp.vulnweb.com/search.php?test=query
Remediation: 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.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Missing Anti-clickjacking Header
Risk Level: Medium
URL: http://testphp.vulnweb.com/artists.php?artist=3
Remediation: 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
Risk Level: Medium
URL: http://testphp.vulnweb.com/disclaimer.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/artists.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/categories.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/guestbook.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/cart.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/login.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/images/logo.gif
Remediation: 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.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/privacy.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/index.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/hpp/
Remediation: 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.

Modern Web Application
Risk Level: Informational
URL: http://testphp.vulnweb.com/AJAX/index.php
Remediation: This is an informational alert and so no changes are required.

Charset Mismatch (Header Versus Meta Content-Type Charset)
Risk Level: Informational

URL: http://testphp.vulnweb.com/artists.php?artist=1
Remediation: Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/
Remediation: 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.

Charset Mismatch (Header Versus Meta Content-Type Charset)
Risk Level: Informational
URL: http://testphp.vulnweb.com/search.php?test=query
Remediation: Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

Charset Mismatch (Header Versus Meta Content-Type Charset)
Risk Level: Informational
URL: http://testphp.vulnweb.com/artists.php?artist=3
Remediation: Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

Absence of Anti-CSRF Tokens
Risk Level: Medium
URL: http://testphp.vulnweb.com/disclaimer.php
Remediation: 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.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/privacy.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/hpp/
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Absence of Anti-CSRF Tokens

Risk Level: Medium

URL: http://testphp.vulnweb.com/artists.php

Remediation: 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.

Absence of Anti-CSRF Tokens

Risk Level: Medium

URL: http://testphp.vulnweb.com/login.php

Remediation: 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.

Missing Anti-clickjacking Header

Risk Level: Medium

URL: http://testphp.vulnweb.com/artists.php?artist=2

Remediation: 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.

Absence of Anti-CSRF Tokens

Risk Level: Medium

URL: http://testphp.vulnweb.com/index.php

Remediation: 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.

Absence of Anti-CSRF Tokens

Risk Level: Medium
URL: http://testphp.vulnweb.com/cart.php
Remediation: 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.

Absence of Anti-CSRF Tokens

Risk Level: Medium
URL: http://testphp.vulnweb.com/guestbook.php
Remediation: 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.

Absence of Anti-CSRF Tokens

Risk Level: Medium
URL: http://testphp.vulnweb.com/categories.php
Remediation: 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.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/AJAX/index.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/artists.php?artist=1
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/search.php?test=query
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/artists.php?artist=3
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/disclaimer.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Absence of Anti-CSRF Tokens
Risk Level: Medium
URL: http://testphp.vulnweb.com/login.php
Remediation: 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.

Absence of Anti-CSRF Tokens

Risk Level: Medium
URL: http://testphp.vulnweb.com/guestbook.php
Remediation: 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.

Modern Web Application

Risk Level: Informational
URL: http://testphp.vulnweb.com/artists.php
Remediation: This is an informational alert and so no changes are required.

X-Content-Type-Options Header Missing

Risk Level: Low
URL: http://testphp.vulnweb.com/AJAX/index.php
Remediation: 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.

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

Risk Level: Low
URL: http://testphp.vulnweb.com/Flash/add.swf
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

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

Risk Level: Low
URL: http://testphp.vulnweb.com/cart.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Charset Mismatch (Header Versus Meta Content-Type Charset)

Risk Level: Informational
URL: http://testphp.vulnweb.com/artists.php?artist=2
Remediation: Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

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

Risk Level: Low
URL: http://testphp.vulnweb.com/index.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

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

Risk Level: Low
URL: http://testphp.vulnweb.com/categories.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Absence of Anti-CSRF Tokens

Risk Level: Medium
URL: http://testphp.vulnweb.com/artists.php?artist=1
Remediation: 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.

Absence of Anti-CSRF Tokens

Risk Level: Medium
URL: http://testphp.vulnweb.com/search.php?test=query
Remediation: 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.

Absence of Anti-CSRF Tokens

Risk Level: Medium
URL: http://testphp.vulnweb.com/artists.php?artist=3
Remediation: 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.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/disclaimer.php
Remediation: 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.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/artists.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Content Security Policy (CSP) Header Not Set
Risk Level: Medium
URL: http://testphp.vulnweb.com/artists.php?artist=2
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/AJAX/index.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/cart.php
Remediation: 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.

X-Content-Type-Options Header Missing
Risk Level: Low

URL: http://testphp.vulnweb.com/index.php
Remediation: 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.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/Flash/add.swf
Remediation: 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.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/login.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/categories.php
Remediation: 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.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/guestbook.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Modern Web Application
Risk Level: Informational
URL: http://testphp.vulnweb.com/artists.php?artist=1
Remediation: This is an informational alert and so no changes are required.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/search.php?test=query
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/disclaimer.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Modern Web Application
Risk Level: Informational

URL: http://testphp.vulnweb.com/artists.php?artist=3
Remediation: This is an informational alert and so no changes are required.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/artists.php
Remediation: 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.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/index.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/cart.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/categories.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Absence of Anti-CSRF Tokens
Risk Level: Medium
URL: http://testphp.vulnweb.com/artists.php?artist=2
Remediation: 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.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/login.php
Remediation: 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.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/guestbook.php
Remediation: 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.

User Controllable HTML Element Attribute (Potential XSS)
Risk Level: Informational
URL: http://testphp.vulnweb.com/search.php?test=query
Remediation: Validate all input and sanitize output it before writing to any HTML attributes.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/artists.php?artist=1
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/artists.php?artist=3
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/artists.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/login.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/guestbook.php
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Modern Web Application
Risk Level: Informational
URL: http://testphp.vulnweb.com/artists.php?artist=2
Remediation: This is an informational alert and so no changes are required.

User Controllable HTML Element Attribute (Potential XSS)
Risk Level: Informational
URL: http://testphp.vulnweb.com/search.php?test=query
Remediation: Validate all input and sanitize output it before writing to any HTML attributes.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/artists.php?artist=3
Remediation: 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.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/artists.php?artist=1
Remediation: 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.

Server Leaks Version Information via "Server" HTTP Response Header Field
Risk Level: Low
URL: http://testphp.vulnweb.com/artists.php?artist=2
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/search.php?test=query
Remediation: 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.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/artists.php?artist=3
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/artists.php?artist=1
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/search.php?test=query
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

X-Content-Type-Options Header Missing
Risk Level: Low
URL: http://testphp.vulnweb.com/artists.php?artist=2
Remediation: 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.

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Risk Level: Low
URL: http://testphp.vulnweb.com/artists.php?artist=2
Remediation: Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.