Website Vulnerability Scanner Result

Export | Schedule periodic scan | Report incorrect result

✔ http://sandbox.end0tknr.net/

Summary

Overall risk level: Medium

Risk ratings:

High:	0
Medium:	3
Low:	9
Info:	33

Scan information:

Start time:	2022-02-28 00:15:45 UTC+02
Finish time:	2022-02-28 00:18:17 UTC+02
Scan duration:	2 min, 32 sec
Tests performed:	45/45
Scan status:	Finished

Findings

Communication is not secure CONFIRMED

URL	Evidence
http://sandbox.end0tknr.net/	Communication is made over unsecure, unencrypted HTTP.

Details

Risk description:
The communication between the web browser and the server is done using the HTTP protocol, which transmits data unencrypted over the network. Thus, an attacker who manages to intercept the communication at the network level, is able to read and modify the data transmitted (including passwords, secret tokens, credit card information and other sensitive data).

Recommendation:
We recommend you to reconfigure the web server to use HTTPS - which encrypts the communication between the web browser and the server.

Classification:
CWE : CWE-311
OWASP Top 10 - 2013 : A6 - Sensitive Data Exposure
OWASP Top 10 - 2017 : A3 - Sensitive Data Exposure

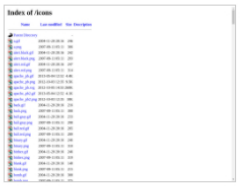
Vulnerabilities found for server-side software

Risk Level	CVSS	CVE	Summary	Exploit	Affected software
●	5.8	CVE-2021-3712	ASN.1 strings are represented internally within OpenSSL as an ASN1_STRING structure which contains a buffer holding the string data and a field holding the buffer length. This contrasts with normal C strings which are repesented as a buffer for the string data which is terminated with a NUL (0) byte. Although not a strict requirement, ASN.1 strings that are parsed using OpenSSL's own "d2i" functions (and other similar parsing functions) as well as any string whose value has been set with the ASN1_STRING_set() function will additionally NUL terminate the byte array in the ASN1_STRING structure. However, it is possible for applications to directly construct valid ASN1_STRING structures which do not NUL terminate the byte array by directly setting the "data" and "length" fields in the ASN1_STRING array. This can also happen by using the ASN1_STRING_set0() function. Numerous OpenSSL functions that print ASN.1 data have been found to assume that the ASN1_STRING byte array will be NUL terminated, even though this is not guaranteed for strings that have been directly constructed. Where an application requests an ASN.1 structure to be printed, and where that ASN.1 structure contains ASN1_STRINGs that have been directly constructed by the application without NUL terminating the "data" field, then a read buffer overrun can occur. The same thing can also occur during name constraints processing of certificates (for example if a certificate has been directly constructed by the application instead of loading it via the OpenSSL parsing functions, and the certificate contains non NUL terminated ASN1_STRING structures). It can also occur in the X509_get1_email(), X509_REQ_get1_email() and X509_get1_ocsp() functions. If a malicious actor can cause an application to directly construct an ASN1_STRING and then process it through one of the affected OpenSSL functions then this issue could be hit. This might result in a crash (causing a Denial of Service attack). It could also result in the disclosure of private memory contents (such as private keys, or sensitive plaintext). Fixed in OpenSSL 1.1.1l (Affected 1.1.1-1.1.1k). Fixed in OpenSSL 1.0.2za (Affected 1.0.2-1.0.2y).	N/A	OpenSSL 1.0.2k

			(Affected 1.0.2-1.0.2y).		
	5	CVE-2017-3735	While parsing an IPAddressFamily extension in an X.509 certificate, it is possible to do a one-byte overread. This would result in an incorrect text display of the certificate. This bug has been present since 2006 and is present in all versions of OpenSSL before 1.0.2m and 1.1.0g.	N/A	OpenSSL 1.0.2k
	5	CVE-2018-0732	During key agreement in a TLS handshake using a DH(E) based ciphersuite a malicious server can send a very large prime value to the client. This will cause the client to spend an unreasonably long period of time generating a key for this prime resulting in a hang until the client has finished. This could be exploited in a Denial Of Service attack. Fixed in OpenSSL 1.1.0i-dev (Affected 1.1.0-1.1.0h). Fixed in OpenSSL 1.0.2p-dev (Affected 1.0.2-1.0.2o).	N/A	OpenSSL 1.0.2k
	5	CVE-2019-1551	There is an overflow bug in the x64_64 Montgomery squaring procedure used in exponentiation with 512-bit moduli. No EC algorithms are affected. Analysis suggests that attacks against 2-prime RSA1024, 3-prime RSA1536, and DSA1024 as a result of this defect would be very difficult to perform and are not believed likely. Attacks against DH512 are considered just feasible. However, for an attack the target would have to re-use the DH512 private key, which is not recommended anyway. Also applications directly using the low level API BN_mod_exp may be affected if they use BN_FLG_CONSTTIME. Fixed in OpenSSL 1.1.1e (Affected 1.1.1-1.1.1d). Fixed in OpenSSL 1.0.2u (Affected 1.0.2-1.0.2t).	N/A	OpenSSL 1.0.2k
	5	CVE-2021-23840	Calls to EVP_CipherUpdate, EVP_EncryptUpdate and EVP_DecryptUpdate may overflow the output length argument in some cases where the input length is close to the maximum permissable length for an integer on the platform. In such cases the return value from the function call will be 1 (indicating success), but the output length value will be negative. This could cause applications to behave incorrectly or crash. OpenSSL versions 1.1.1i and below are affected by this issue. Users of these versions should upgrade to OpenSSL 1.1.1j. OpenSSL versions 1.0.2x and below are affected by this issue. However OpenSSL 1.0.2 is out of support and no longer receiving public updates. Premium support customers of OpenSSL 1.0.2 should upgrade to 1.0.2y. Other users should upgrade to 1.1.1j. Fixed in OpenSSL 1.1.1j (Affected 1.1.1-1.1.1i). Fixed in OpenSSL 1.0.2y (Affected 1.0.2-1.0.2x).	N/A	OpenSSL 1.0.2k

› Details

Directory listing is enabled CONFIRMED

URL	Screenshot
http://sandbox.end0tknr.net/icons/	
http://sandbox.end0tknr.net/icons/#{set}($a='aabtasb')#{if}(1==1)cok$a#{else}NO#end/	
http://sandbox.end0tknr.net/icons/#{set}($a='xvsovdh')#{if}(1==1)zvy$a#{else}NO#end/	
http://sandbox.end0tknr.net/icons/small/	
http://sandbox.end0tknr.net/icons/small/#{set}($a='ilnagal')#{if}(1==1)jpn$a#{else}NO#end/	Directory Listing

Details

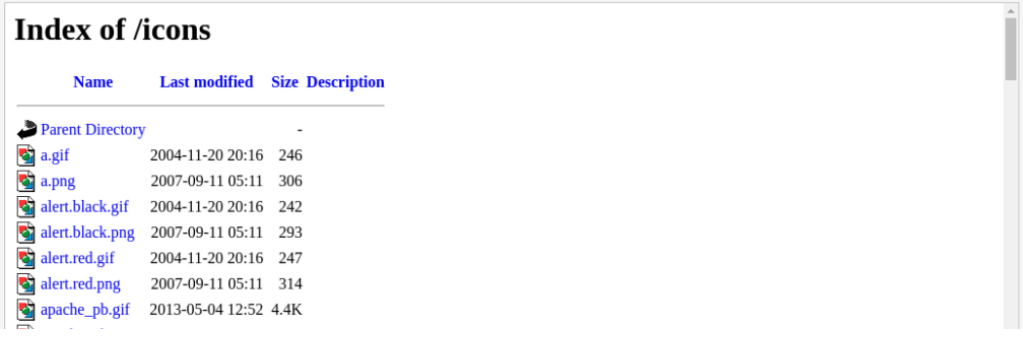
Risk description:
An attacker can see the entire structure of files and subdirectories from the affected URL. It is often the case that sensitive files are "hidden" among public files in that location and attackers can use this vulnerability to access them.

Recommendation:
We recommend reconfiguring the web server in order to deny directory listing. Furthermore, you should verify that there are no sensitive files at the mentioned URLs.

More information about this issue:
http://projects.webappsec.org/w/page/13246922/Directory%20Indexing.

Classification:
CWE : CWE-548
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Screenshot:



Index of /icons

Name	Last modified	Size	Description
Parent Directory		-	
a.gif	2004-11-20 20:16	246	
a.png	2007-09-11 05:11	306	
alert.black.gif	2004-11-20 20:16	242	
alert.black.png	2007-09-11 05:11	293	
alert.red.gif	2004-11-20 20:16	247	
alert.red.png	2007-09-11 05:11	314	
apache_pb.gif	2013-05-04 12:52	4.4K	

Figure 1. Directory Listing

Server software and technology found

Software / Version	Category	Screenshot
OpenSSL 1.0.2k	Web server extensions	 Website Screenshot
Apache 2.4.52	Web servers	

Details

Risk description:
An attacker could use this information to mount specific attacks against the identified software type and version.

Recommendation:
We recommend you to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

More information about this issue:
https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server.html.

Classification:
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Screenshot:

Figure 2. Website Screenshot

Missing security header: Content-Security-Policy CONFIRMED

URL	Evidence
http://sandbox.end0tknr.net/	Response headers do not include the HTTP Content-Security-Policy security header

Details

Risk description:
The Content-Security-Policy (CSP) header activates a protection mechanism implemented in web browsers which prevents exploitation of Cross-Site Scripting vulnerabilities (XSS). If the target application is vulnerable to XSS, lack of this header makes it easily exploitable by attackers.

Recommendation:
Configure the Content-Security-Header to be sent with each HTTP response in order to apply the specific policies needed by the application.

Read more about CSP:
https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy

Classification:
CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Missing security header: X-Frame-Options CONFIRMED

URL	Evidence
http://sandbox.end0tknr.net/	Response headers do not include the HTTP X-Frame-Options security header

Details

Risk description:
Because the `X-Frame-Options` header is not sent by the server, an attacker could embed this website into an iframe of a third party website. By manipulating the display attributes of the iframe, the attacker could trick the user into performing mouse clicks in the application, thus performing activities without user's consent (ex: delete user, subscribe to newsletter, etc). This is called a Clickjacking attack and it is described in detail here:
https://owasp.org/www-community/attacks/Clickjacking

Recommendation:
We recommend you to add the `X-Frame-Options` HTTP header with the values `DENY` or `SAMEORIGIN` to every page that you want to be protected against Clickjacking attacks.

More information about this issue:
https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html

Classification:
CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Missing security header: X-XSS-Protection CONFIRMED

URL	Evidence
http://sandbox.end0tknr.net/	Response headers do not include the HTTP X-XSS-Protection security header

Details

Risk description:
The `X-XSS-Protection` HTTP header instructs the browser to stop loading web pages when they detect reflected Cross-Site Scripting (XSS) attacks. Lack of this header exposes application users to XSS attacks in case the web application contains such vulnerability.

Recommendation:
We recommend setting the X-XSS-Protection header to `X-XSS-Protection: 1; mode=block` .

More information about this issue:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection

Classification:

CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Missing security header: X-Content-Type-Options CONFIRMED

URL	Evidence
http://sandbox.end0tknr.net/	Response headers do not include the X-Content-Type-Options HTTP security header

Details

Risk description:
The HTTP header `X-Content-Type-Options` is addressed to the Internet Explorer browser and prevents it from reinterpreting the content of a web page (MIME-sniffing) and thus overriding the value of the Content-Type header). Lack of this header could lead to attacks such as Cross-Site Scripting or phishing.

Recommendation:
We recommend setting the X-Content-Type-Options header such as `X-Content-Type-Options: nosniff` .

More information about this issue:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options.

Classification:
CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Missing security header: Referrer-Policy CONFIRMED

URL	Evidence
http://sandbox.end0tknr.net/	Response headers do not include the Referrer-Policy HTTP security header as well as the <meta> tag with name 'referrer' is not present in the response.

Details

Risk description:
The Referrer-Policy HTTP header controls how much referrer information the browser will send with each request originated from the current web application.
For instance, if a user visits the web page "http://example.com/pricing/" and it clicks on a link from that page going to e.g. "https://www.google.com", the browser will send to Google the full originating URL in the `Referer` header, assuming the Referrer-Policy header is not set. The originating URL could be considered sensitive information and it could be used for user tracking.

Recommendation:
The Referrer-Policy header should be configured on the server side to avoid user tracking and inadvertent information leakage. The value `no-referrer` of this header instructs the browser to omit the Referer header entirely.

Read more:
https://developer.mozilla.org/en-US/docs/Web/Security/Referer_header:_privacy_and_security_concerns

Classification:
CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Internal Server Error Found CONFIRMED

Method	URL	Parameters	Evidence
TRACK	http://sandbox.end0tknr.net/	GET: canary=dmpupkjlfc	Response has 501 internal server error status_code

Details

Risk description:
The website does not handle or incorrectly handles an exceptional condition. An attacker may use the contents of error messages to help launch another, more focused attack. For example, an attempt to exploit a path traversal weakness (CWE-22) might yield the full pathname of the installed application.

Recommendation:
Ensure that error messages only contain minimal details that are useful to the intended audience, and nobody else. The messages need to strike the balance between being too cryptic and not being cryptic enough. They should not necessarily reveal the methods that were used to determine the error. Such detailed information can be used to refine the original attack to increase the chances of success. If errors must be tracked in some detail, capture them in log messages - but consider what could occur if the log messages can be viewed by attackers. Avoid recording highly sensitive information such as passwords in any form. Avoid inconsistent messaging that might accidentally tip off an attacker about internal state, such as whether a username is valid or not.

Classification:
CWE : CWE-209
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

HTTP Debug methods enabled CONFIRMED

Method	URL	Summary
TRACE	http://sandbox.end0tknr.net/?canary=dmpupkjlfc	We injected a random query parameter inside a HTTP TRACE request. The server responded with a 200 OK HTTP status code and we found the random value reflected in the body of the response.

Details

Risk description:
The webserver responded with a 200 OK HTTP status when a TRACE/TRACK HTTP request was sent. These methods were intended for debugging purposes. When a server receives such a request, it replies with the exact contents of the original request.
This is most often harmless. The only risk it might present nowadays is revealing HTTP headers that have been appended by intermediate proxy servers on the way to the destination. This can present a danger if any of those headers contain sensitive information like authentication information, secret keys.

Recommendation:
Generally it is good practice to disable unused functionality to minimize your attack surface. We recommend that you disable unused HTTP methods, or even better, allow only the onesthat you know are used. This can be done using your webserver configuration.

References:
https://httpd.apache.org/docs/2.4/mod/core.html#traceenable
https://httpd.apache.org/docs/2.4/mod/mod_authz_core.html#reqmethod
https://docs.microsoft.com/en-us/iis/manage/configuring-security/use-request-filtering#filter-by-verbs
https://nginx.org/en/docs/http/ngx_http_core_module.html#limit_except

Classification:
CWE : CWE-16
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Exposure of Sensitive Information

Method	URL	Parameters	Evidence
GET	http://sandbox.end0tknr.net/icons/		Email Address: mike@hyperreal.org kevinh@kevcom.com

Details

Risk description:
This application does not properly prevent a person's private, personal information from being accessed by actors who either (1) are not explicitly authorized to access the information or (2) do not have the implicit consent of the person about whom the information is collected.

Recommendation:
Compartmentalize the application to have "safe" areas where trust boundaries can be unambiguously drawn. Do not allow sensitive data to go outside of the trust boundary and always be careful when interfacing with a compartment outside of the safe area.

Website is accessible.

Nothing was found for client access policies.

Nothing was found for robots.txt file.

Security.txt file is missing CONFIRMED

URL
Missing: http://sandbox.end0tknr.net/.well-known/security.txt

Details

Risk description:
We have detected that the server is missing the security.txt file. There is no particular risk in not creating a valid Security.txt file for your server. However

We have detected that the server is missing the security.txt file. There is no particular risk in not creating a valid Security.txt file for your server. However, this file is important because it offers a designated channel for reporting vulnerabilities and security issues.

Recommendation:
We recommend you to implement the security.txt file according to the standard, in order to allow researchers or users report any security issues they find, improving the defensive mechanisms of your server.

More information about the security.txt standard:
https://securitytxt.org/

Classification:
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Nothing was found for outdated JavaScript libraries.

Nothing was found for CORS misconfiguration.

Nothing was found for use of untrusted certificates.

Spider results

Method	URL	Parameters
GET	http://sandbox.end0tknr.net/	
GET	http://sandbox.end0tknr.net/icons	
GET	http://sandbox.end0tknr.net/icons/	
GET	http://sandbox.end0tknr.net/icons/small	

Nothing was found for passwords submitted unencrypted.

Nothing was found for Cross-Site Scripting.

Nothing was found for SQL Injection.

Nothing was found for Local File Inclusion.

Nothing was found for OS Command Injection.

Nothing was found for error messages.

Nothing was found for debug messages.

Nothing was found for code comments.

Nothing was found for missing HTTP header - Strict-Transport-Security.

Nothing was found for domain too loose set for cookies.

- Nothing was found for mixed content between HTTP and HTTPS.
- Nothing was found for cross domain file inclusion.
- Nothing was found for HttpOnly flag of cookie.
- Nothing was found for Secure flag of cookie.
- Nothing was found for login interfaces.
- Nothing was found for secure password submission.
- Nothing was found for Server Side Request Forgery.
- Nothing was found for Open Redirect.
- Nothing was found for PHP Code Injection.
- Nothing was found for JavaScript Code Injection.
- Nothing was found for Ruby Code Injection.
- Nothing was found for Python Code Injection.
- Nothing was found for Perl Code Injection.
- Nothing was found for Remote Code Execution through Log4j.
- Nothing was found for Server Side Template Injection.

Scan coverage information

List of tests performed (45/45)

- Checking for website accessibility...
- Checking for secure communication...
- Checking for website technologies...
- Checking for vulnerabilities of server-side software...
- Checking for client access policies...
- Checking for robots.txt file...
- Checking for absence of the security.txt file...
- Checking for outdated JavaScript libraries...
- Checking for CORS misconfiguration...
- Checking for use of untrusted certificates...
- Checking for missing HTTP header - Content Security Policy...
- Checking for missing HTTP header - X-Frame-Options...
- Checking for missing HTTP header - X-XSS-Protection...
- Checking for missing HTTP header - X-Content-Type-Options...
- Checking for missing HTTP header - Referrer...

- Checking for internal error code...
- Checking for enabled HTTP debug methods...
- Spidering target...
- Checking for directory listing...
- Checking for sensitive data...
- Checking for passwords submitted unencrypted...
- Checking for Cross-Site Scripting...
- Checking for SQL Injection...
- Checking for Local File Inclusion...
- Checking for OS Command Injection...
- Checking for error messages...
- Checking for debug messages...
- Checking for code comments...
- Checking for missing HTTP header - Strict-Transport-Security...
- Checking for domain too loose set for cookies...
- Checking for mixed content between HTTP and HTTPS...
- Checking for cross domain file inclusion...
- Checking for HttpOnly flag of cookie...
- Checking for Secure flag of cookie...
- Checking for login interfaces...
- Checking for secure password submission...
- Checking for Server Side Request Forgery...
- Checking for Open Redirect...
- Checking for PHP Code Injection...
- Checking for JavaScript Code Injection...
- Checking for Ruby Code Injection...
- Checking for Python Code Injection...
- Checking for Perl Code Injection...
- Checking for Remote Code Execution through Log4j...
- Checking for Server Side Template Injection...

Scan parameters

Website URL:	http://sandbox.end0tknr.net/
Scan type:	Ptt_engine
Authentication:	False
Fingerprint Website:	True
Server Software Vulnerabilities:	True
Robots.txt:	True
JavaScript libraries:	True
SSL/TLS Certificates:	True
Client access policies:	True
HTTP Debug Methods:	True
Security.txt file missing:	True
CORS Misconfiguration:	True
Resource Discovery:	False
Approach:	Classic
Depth:	10
Request Speed:	10000
XSS:	True
SQL Injection:	True
Local File Inclusion:	True
OS Command Injection:	True
Server Side Request Forgery:	True
Open Redirect:	True
Broken Authentication:	True
PHP Code Injection:	True
Server-Side JavaScript Code Injection:	True
Ruby Code Injection:	True
Python Code Injection:	True
Perl Code Injection:	True
Log4j Remote Code Execution:	True
Server-Side Template Injection:	True
Security Headers:	True
Cookie Security:	True
Directory Listing:	True
Secure Communication:	True
Weak Password Submission Method:	True
Commented code/Error codes:	True
Clear Text Submission of Credentials:	True
Verify Domain Sources:	True
Mixed Encryptions Content:	True
Sensitive Data Crawl:	True
Find Login Interfaces:	True

Scan stats

Unique Injection Points Detected:	4
URLs spidered:	5
Total number of HTTP requests:	423