CRASHTEST SECURITY

WEB VULNERABILITY SCANNING REPORT

DVWA

17 SEP 20 12:17 CEST
https://dvwa.devstack.crashtest.cloud

1 Overview

1.1 Vulnerability Overview

Based on our testing, we identified **57** vulnerabilities.

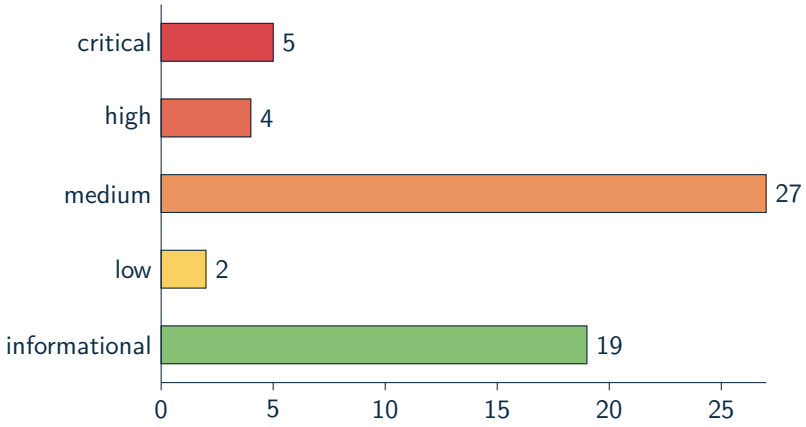


Figure 1.1: Total number of vulnerabilities for "DVWA"

STATE	DESCRIPTION	BASE SCORE
CRITICAL	These findings are very critical whilst posing an immediate threat. Fixing these issues should be the highest priority, regardless of any other issues.	**9 - 10**
HIGH	Findings in this category pose an immediate threat and should be fixed immediately.	**7 - 8.9**
MEDIUM	Medium findings may cause serious harm in combination with other security vulnerabilites. These findings should be considered during project planning and be fixed within short time.	**4 - 6.9**
LOW	Low severity findings do not impose an immediate threat. Such findings should be reviewed for their specific impact on the application and be fixed accordingly.	**0.1 - 3.9**
INFO	Informational findings do not pose any threat but have solely informational purpose.	**0**

1.2 Scanner Overview

During the scan, the Crashtest Security Suite was looking for the following kinds of vulnerabilities and security issues:

- ✓ Server Version Fingerprinting
- ✓ Web Application Version Fingerprinting
- ✓ CVE Comparison
- ✓ Heartbleed
- ✓ ROBOT
- ✓ BREACH
- ✓ BEAST
- ✓ Old SSL/TLS Version
- ✓ SSL/TLS Cipher Order
- ✓ SSL/TLS Perfect Forward Secrecy
- ✓ SSL/TLS Session Resumption
- ✓ SSL/TLS secure algorithm
- ✓ SSL/TLS key size
- ✓ SSL/TLS trust chain
- ✓ SSL/TLS expiration date
- ✓ SSL/TLS revocation (CRL, OCSP)
- ✓ SSL/TLS OCSP stapling
- ✓ Security Headers
- ✓ Content-Security-Policy headers
- ✓ Portscan
- ✓ Boolean-based blind SQL Injection
- ✓ Time-based blind SQL Injection
- ✓ Error-based SQL Injection
- ✓ UNION query-based SQL Injection
- ✓ Stacked queries SQL Injection
- ✓ Out-of-band SQL Injection
- ✓ Reflected Cross-site scripting (XSS)
- ✓ Stored Cross-site scripting (XSS)
- ✓ Cross-Site Request Forgery (CSRF)
- ✓ File Inclusion
- ✓ Directory Fuzzer
- ✓ File Fuzzer
- ✓ Command Injection
- ✓ XML External Entity Processing (XXE)

1.2.1 Status for executed Scanners

SCANNER	PERCENTAGE	STATUS
Command Injection	100%	18 completed
Deserialization	100%	18 completed
Fuzzer	100%	1 completed
Portscan	100%	1 completed
Transport Layer Security (TLS/SSL)	100%	1 completed
Cross-Site Scripting (XSS)	100%	18 completed
XML External Entity (XXE)	100%	18 completed
Cross-Site Request Forgery (CSRF)	100%	18 completed
CVE	100%	1 completed
File Inclusion	100%	18 completed
Fingerprinting	100%	1 completed
SQL Injection	100%	18 completed
	100%	**131 completed**

1.3 Findings Checklist

1.3.1 COMMANDINJECTION

STATE	FINDING RESULT	NOTICED	FIXED
9.8	Found command injection in parameter "ip" with method "post" for URL "https://dvwa.devstack.crashtest.cloud/vulnerabilities/exec/", with payload "; echo crashtest-security$((12*12))"	☐	☐

1.3.2 FILEINCLUSION

STATE	FINDING RESULT	NOTICED	FIXED
9.8	Found file inclusion with method "get" for parameter "page" on "https://dvwa.devstack.crashtest.cloud/vulnerabilities/fi/." with payload "/etc/passwd"	☐	☐

1.3.3 SQLINJECTION

STATE	FINDING RESULT	NOTICED	FIXED
9.1	Found boolean-based blind sqlinjection for parameter username (GET) on https://dvwa.devstack.crashtest.cloud/vulnerabilities/brute/ with payload Login=Login&password=Crashtest123!&username=xyz' AND 7725=(SELECT (CASE WHEN (7725=7725) THEN 7725 ELSE (SELECT 3607 UNION SELECT 8444) END))– -	☐	☐

1.3.4 XXE

STATE	FINDING RESULT	NOTICED	FIXED
9.4	Found XXE in parameter "xml" with method "get" for URL "https://dvwa.devstack.crashtest.cloud/vulnerabilities/xxe/", with payload "<?xml version='1.0' encoding='utf-8'?><!DOCTYPE creds [<!ELEMENT user ANY ><!ELEMENT pass ANY ><!ENTITY user SYSTEM 'file:///etc/passwd'>]><creds><user>%26user;</user><pass>%26user;</pass></creds>"	☐	☐

1.3.5 FINGERPRINTING

STATE	FINDING RESULT	NOTICED	FIXED
5.3	The webserver is running Apache 2.4.7 (**34** connected CVE issues have been found. The most severe vulnerability has a CVSS score of high (7.5/10). See Appendix **APACHE 2.4.7 CVE FINDINGS** for a detailed list of the CVEs)	☐	☐

STATE	FINDING RESULT	NOTICED	FIXED
5.3	Found PHP running in version 5.5.9. (**203** connected CVE issues have been found. The most severe vulnerability has a CVSS score of critical (10/10). See Appendix **PHP 5.5.9 CVE FINDINGS** for a detailed list of the CVEs)	☐	☐

1.3.6 PORTSCAN

STATE	FINDING RESULT	NOTICED	FIXED
0.0	Found open port "8099/tcp"	☐	☐
0.0	Found open port "5222/tcp"	☐	☐
0.0	Found open port "8080/tcp"	☐	☐
0.0	Found open port "5901/tcp"	☐	☐
0.0	Found open port "5432/tcp"	☐	☐
0.0	Found open port "110/tcp"	☐	☐
0.0	Found open port "43/tcp"	☐	☐
0.0	Found open port "700/tcp"	☐	☐

STATE	FINDING RESULT	NOTICED	FIXED
0.0	Found open port "9200/tcp"	☐	☐
0.0	Found open port "443/tcp" with service name "Apache httpd"	☐	☐
0.0	Found open port "465/tcp"	☐	☐
0.0	Found open port "5900/tcp"	☐	☐
0.0	Found open port "995/tcp"	☐	☐
0.0	Found open port "587/tcp"	☐	☐
0.0	Found open port "993/tcp"	☐	☐
0.0	Found open port "143/tcp"	☐	☐
0.0	Found open port "8085/tcp"	☐	☐
0.0	Found open port "80/tcp" with service name "Apache httpd"	☐	☐

STATE	FINDING RESULT	NOTICED	FIXED
0.0	Found open port "3389/tcp"	☐	☐

1.3.7 XSS

STATE	FINDING RESULT	NOTICED	FIXED
6.1	Found possible XSS vulnerability on site dvwa.devstack.crashtest.cloud/vulnerabilities/xss_s/. The parameter 'btnSign' seems vulnerable for payload '<svg dddf7814-868e-4170-aac1-58ad9b90eee0 "ons>'	☐	☐
6.1	Found possible XSS vulnerability on site dvwa.devstack.crashtest.cloud/vulnerabilities/xss_s/. The parameter 'txtName' seems vulnerable for payload '<svg dddf7814-868e-4170-aac1-58ad9b90eee0 "ons>'	☐	☐
6.1	Found possible XSS vulnerability on site dvwa.devstack.crashtest.cloud/vulnerabilities/xss_r/. The parameter 'name' seems vulnerable for payload '<svg 2e10116c-7243-4bee-a867-e60201086244 "ons>'	☐	☐
6.1	The potentially vulnerable code was found on url 'dvwa.devstack.crashtest.cloud/ids_log.php'. An attacker may be able to inject JavaScript using the the code 'window.name' at line 42:188 and control its display using the code 'eval' at line 42:183	☐	☐

1.3.8 SSL/TLS

STATE	FINDING RESULT	NOTICED	FIXED
4.8	HSTS is not offered by the server.	☐	☐
2.2	OCSP_stapling is not offered by the server.	☐	☐
8.2	TLS 1.1 is offered by the server. This version of TLS is deprecated. You should use TLS 1.2 or TLS 1.3	☐	☐

STATE	FINDING RESULT	NOTICED	FIXED
8.2	TLS 1.0 is offered by the server. This version of TLS is deprecated. You should use TLS 1.2 or TLS 1.3	☐	☐
6.5	The X-Frame-Options header is not set for URL https://dvwa.devstack.crashtest.cloud.	☐	☐
4.3	The X-Content-Type-Options header is not set for URL https://dvwa.devstack.crashtest.cloud.	☐	☐
4.3	VULNERABLE – but also supports higher protocols TLSv1.1 TLSv1.2 (likely mitigated)	☐	☐
4.8	Valid for 100800 seconds (>daily)	☐	☐
4.3	The Referrer-Policy header is not set for URL https://dvwa.devstack.crashtest.cloud.	☐	☐
7.4	Cipher suits based on DES (Data Encryption Standard) and IDEA (International Data Encryption Algorithm) are not recommended for general use in TLS. In TLS 1.2 and TLS 1.3 DES and IDEA are removed. You should use TLS 1.2 or TLS 1.3.	☐	☐
3.7	The server is configured to use average ciphers like SEED + 128+256 Bit CBC ciphers (AES, CAMELLIA and ARIA) which are deprecated	☐	☐
4.3	BEAST TLS1: The BEAST attack leverages weakness in the cipher block chaining (CBC) which allows man in the middle attacks.	☐	☐
6.5	The Content-Security-Policy header is not set for URL https://dvwa.devstack.crashtest.cloud.	☐	☐

STATE	FINDING RESULT	NOTICED	FIXED
6.1	The X-XSS-Protection header is not set for URL https://dvwa.devstack.crashtest.cloud.	☐	☐
5.9	64 bit block ciphers are used which are vulnerable to SWEET32 birthday attack. 3DES ciphers should be disabled.	☐	☐
4.8	The cookie with the name PHPSESSID does not have the flag httponly set. This may leak sensitive information. This was found on URL https://dvwa.devstack.crashtest.cloud.	☐	☐
4.8	The cookie with the name PHPSESSID does not have the flag secure set. This may leak sensitive information. This was found on URL https://dvwa.devstack.crashtest.cloud.	☐	☐

1.3.9 FUZZER

STATE	FINDING RESULT	NOTICED	FIXED
5.3	Retrieved https://dvwa.devstack.crashtest.cloud/php.ini by using a GET request on the URL without prior knowledge.	☐	☐
5.3	Retrieved https://dvwa.devstack.crashtest.cloud/about.php by using a GET request on the URL without prior knowledge.	☐	☐
5.3	Retrieved https://dvwa.devstack.crashtest.cloud/instructions.php by using a GET request on the URL without prior knowledge.	☐	☐
5.3	Retrieved https://dvwa.devstack.crashtest.cloud/phpinfo.php by using a GET request on the URL without prior knowledge.	☐	☐
5.3	Retrieved https://dvwa.devstack.crashtest.cloud/docs/ by using a GET request on the URL without prior knowledge.	☐	☐

STATE	FINDING RESULT	NOTICED	FIXED
5.3	Retrieved https://dvwa.devstack.crashtest.cloud/setup.php by using a GET request on the URL without prior knowledge.	☐	☐
5.3	Retrieved https://dvwa.devstack.crashtest.cloud/CHANGELOG.md by using a GET request on the URL without prior knowledge.	☐	☐
5.3	Retrieved https://dvwa.devstack.crashtest.cloud/.git/ by using a GET request on the URL without prior knowledge.	☐	☐
5.3	Retrieved https://dvwa.devstack.crashtest.cloud/README.md by using a GET request on the URL without prior knowledge.	☐	☐
5.3	Retrieved https://dvwa.devstack.crashtest.cloud/COPYING by using a GET request on the URL without prior knowledge.	☐	☐
5.3	Retrieved https://dvwa.devstack.crashtest.cloud/config/ by using a GET request on the URL without prior knowledge.	☐	☐

Contents

2 Findings

2.1 SSL/TLS

2.1.1 What is this?

Transport Layer Security (TLS), more widely known by its predecessor Secure Sockets Layer (SSL), is a hybrid encryption protocol for secure data transmission over the Internet. It encrypts the communication between server and client. The most obvious part of it is HTTPS, with which providers can secure all communications between their servers and web browsers. This ensures that valuable information like usernames, passwords and credit card information cannot be stolen by someone analyzing the network traffic. The "S" in HTTPS stands for SSL. For secure connection with HTTPS a certificate is needed. Those certificates offer different levels of security and have a fixed start- and expiration-date. To ensure a secure connection, webservers must use well configured certificates. With some misconfigured certificates it is possible to bypass the encryption, others may be blocked by web browsers because they are outdated or unknown.

2.1.2 Missing HSTS

Severity

Base Score:	medium (4.8/10)
Impact:	low (2.5/10)
Exploitability:	low (2.2/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

The webserver does not offer HTTP Strict Transport Security (HSTS). HSTS enforces HTTPS connections, which prevents downgrade attacks to an insecure HTTP connection.

Finding

+ HSTS is not offered by the server.

How to fix

The webserver does not offer HTTP Strict Transport Security (HSTS). HSTS enforces HTTPS connections. This prevents downgrade attacks to an insecure HTTP connection. Depending on the used SSL certificate and the webserver certain configurations have to be changed. More details on how to enable HSTS can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/enable-hsts

2.1.3 OCSP Stapling

Severity

Base Score:	low (2.2/10)
Impact:	low (1.4/10)
Exploitability:	low (0.7/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

OCSP Stapling is disabled on your server. Therefore, your certificate authority might track which users visit your site.

Finding

+ OCSP_stapling is not offered by the server.

How to fix

OSCP stapling can be enabled in the servers configuration (apache/nginx). For Let's Encrypt Certificates OCSP stapling can be activated during the creation of the certificate by adding the "–staple-oscp" parameter. More details on how to fix this problem can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/certificate-revocation

2.1.4 SSL Protocol Version

Severity

Base Score:	high (8.2/10)
Impact:	medium (4.2/10)
Exploitability:	low (3.9/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

A SSL/TLS version offered by the server is outdated. The deprecated versions contain weak implementations that cannot be considered as secure anymore. Please use TLS 1.2 or TLS 1.3 instead.

Finding

+ TLS 1.1 is offered by the server. This version of TLS is deprecated. You should use TLS 1.2 or TLS 1.3
+ TLS 1.0 is offered by the server. This version of TLS is deprecated. You should use TLS 1.2 or TLS 1.3

How to fix

The webserver is using a deprecated SSL/TLS version and needs to be updated. The webserver needs to be configured to use strong and trusted certificates. In addition they need to be configured to use the newest version of SSL and TLS as well as strong cipher suites. More details on how to configure these certificates can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/disable-deprecated-ssl-protocol-versions

2.1.5 X-Frame-Options Header

Severity

Base Score:	medium (6.5/10)
Impact:	low (3.6/10)
Exploitability:	low (2.8/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

The X-Frame-Options header declares whether this site may be embedded as a frame into other websites. If this header is not configured correctly, your application can be embedded into third party websites which makes it vulnerable for clickjacking attacks.

Finding

+ The X-Frame-Options header is not set for URL https://dvwa.devstack.crashtest.cloud.

How to fix

Configure the X-Frame-Options header as 'deny' to prevent it to be embedded at all. The values 'sameorigin' or 'allow-from DOMAIN' can be used to allow it to be embedded on certain websites while forbidding embedding on other websites

Recommendations

https://wiki.crashtest-security.com/enable-security-headers

2.1.6 X-Content-Type-Options Header

Severity

Base Score:	medium (4.3/10)
Impact:	low (1.4/10)
Exploitability:	low (2.8/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

The X-Content-Type-Options prevents the browser from trying to detect MIME-types on downloaded files. This protects against attacks in cases where a malicious file is offered with an unsuspicious MIME-type.

Finding

+ The X-Content-Type-Options header is not set for URL https://dvwa.devstack.crashtest.cloud.

How to fix

Set the X-Content-Type-Options header to 'nosniff' in order to prevent the browser from detecting MIME-types based on file content.

Recommendations

https://wiki.crashtest-security.com/enable-security-headers

2.1.7 SSL BEAST

Severity

Base Score:	medium (4.3/10)
Impact:	low (2.9/10)
Exploitability:	high (8.6/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

The server is vulnerable for BEAST (Browser Exploit Against SSL/TLS) attacks. By using weaknesses in cipher block chaining, an attacker can use a Man-In-The-Middle attacks to decrypt and obtain authentication tokens.

Finding

+ VULNERABLE – but also supports higher protocols TLSv1.1 TLSv1.2 (likely mitigated)

How to fix

BEAST attacks can be prevented by ensuring, that neither SSLv3 nor TLSv1 are used. More details on how to fix this problem can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/prevent-ssl-beast

2.1.8 SSL Session

Severity

Base Score:	medium (4.8/10)
Impact:	low (2.5/10)
Exploitability:	low (2.2/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

The TLS session resumption functionality is misconfigured. This opens attackers the possibility to steal existing TLS sessions from other users.

Finding

+ Valid for 100800 seconds (>daily)

How to fix

Existing TLS sessions can be stolen, due to a TLS misconfiguration. TLS session resumption can be disabled to ensure this attack is not possible. Depending on the used webserver (Apache, Nginx, ...) the respective site configuration has to be extended by a short line which disables session tickets. More details on how the exact configuration change for your webserver can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/harden-tls-session-resumption

2.1.9 Referrer-Policy Header

Severity

Base Score:	medium (4.3/10)
Impact:	low (1.4/10)
Exploitability:	low (2.8/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

The Referrer-Policy header defines how much information about the referrer is sent, when the user clicks on a link. A misconfiguration or missing header may leak sensitive information to third party websites that are visited by the click on a link.

Finding

+ The Referrer-Policy header is not set for URL https://dvwa.devstack.crashtest.cloud.

How to fix

Set the Referrer-Policy header to a secure value such as 'strict-origin-when-cross-origin' to overwrite the Referer header with your domain instead of the full path when clicking on external links and keep the Referer for internal links, but only when the connection is not downgraded from HTTPS to HTTP.

Recommendations

https://wiki.crashtest-security.com/enable-security-headers

2.1.10 SSL Cipherlist 3DES IDEA

Severity

Base Score:	high (7.4/10)
Impact:	medium (5.2/10)
Exploitability:	low (2.2/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

The server is configured to support 3DES and IDEA Ciphers like "3DES:IDEA". This means, that an attacker can make use of an insecure SSL/TLS connection.

Finding

+ Cipher suits based on DES (Data Encryption Standard) and IDEA (International Data Encryption Algorithm) are not recommended for general use in TLS. In TLS 1.2 and TLS 1.3 DES and IDEA are removed. You should use TLS 1.2 or TLS 1.3.

How to fix

The list of supported HTTPS ciphers includes insecure ciphers. This means, that an attacker can make use of in insecure SSL/TLS connection. In the SSL/TLS configuration, the allowed ciphers and their order should be set to match secure values. More details on how to set these values can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/secure-tls-configuration

2.1.11 SSL Cipherlist AVERAGE

Severity

Base Score:	low (3.7/10)
Impact:	low (1.4/10)
Exploitability:	low (2.2/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

The server is configured to support average Ciphers like SEED + 128+256 Bit CBC ciphers (AES, CAMELLIA and ARIA). This means, that an attacker can make use of an insecure SSL/TLS connection.

Finding

+ The server is configured to use average ciphers like SEED + 128+256 Bit CBC ciphers (AES, CAMELLIA and ARIA) which are deprecated

How to fix

The list of supported HTTPS ciphers includes insecure ciphers. This means, that an attacker can make use of in insecure SSL/TLS connection. In the SSL/TLS configuration, the allowed ciphers and their order should be set to match secure values. More details on how to set these values can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/secure-tls-configuration

2.1.12 SSL Cipher Block Chaining TLS1

Severity

Base Score:	medium (4.3/10)
Impact:	low (2.9/10)
Exploitability:	high (8.6/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

The webserver is configured to allow connections encrypted with TLS V1 in Cipher Block Chaining Mode (CBC). Connections using this settings contain predictable information that allow an attacker to break the encryption using the BEAST attack.

Finding

+ BEAST TLS1: The BEAST attack leverages weakness in the cipher block chaining (CBC) which allows man in the middle attacks.

How to fix

The webserver needs to be configured to use strong and trusted certificates. In addition they need to be configured to use the newest version of SSL and TLS as well as strong cipher suites. More details on how to configure these certificates can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/secure-tls-configuration

2.1.13 Content-Security-Policy Header

Severity

Base Score:	medium (6.5/10)
Impact:	low (2.5/10)
Exploitability:	low (3.9/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

The Content-Security-Policy header tells the browser which domains are whitelisted to download further resources such as scripts, images or stylesheets from. This can prevent various XSS and other Cross-Site-Scripting attacks.

Finding

+ The Content-Security-Policy header is not set for URL https://dvwa.devstack.crashtest.cloud.

How to fix

Configure the Content-Security-Policy header in a way that it only allows loading resources from trusted resources such as 'self'. Do not include 'unsafe-eval' or 'unsafe-inline' in order to prevent direct injections into the website.

Recommendations

https://wiki.crashtest-security.com/enable-security-headers

2.1.14 X-XSS-Protection Header

Severity

Base Score:	medium (6.1/10)
Impact:	low (2.7/10)
Exploitability:	low (2.8/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

The X-XSS-Protection header tells the browser how it should handle detected XSS attacks. If this header is not configured correctly, XSS attacks may not be blocked despite being detected by the browser.

Finding

+ The X-XSS-Protection header is not set for URL https://dvwa.devstack.crashtest.cloud.

How to fix

Configure the X-XSS-Protection header as '1' or '1; mode=block' to make sure that XSS attacks detected by the browser are sanitized or blocked.

Recommendations

https://wiki.crashtest-security.com/enable-security-headers

2.1.15 SSL SWEET32

Severity

Base Score:	medium (5.9/10)
Impact:	low (3.6/10)
Exploitability:	low (2.2/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

The server uses short block sizes, which makes it vulnerable to hit the same hash for multiple inputs. By observing the data for a longer period of time, an attacker can recover secure HTTP cookies.

Finding

+ 64 bit block ciphers are used which are vulnerable to SWEET32 birthday attack. 3DES ciphers should be disabled.

How to fix

SWEET32 attacks can be prevented by using cipher suites with large block sizes. More details on what block sizes are considered large enough can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/prevent-ssl-sweet32

2.1.16 Non Httponly Cookies

Severity

Base Score:	medium (4.8/10)
Impact:	low (2.5/10)
Exploitability:	low (2.2/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

Cookies that are not marked as http-only can be read by local scripts. In case of an Cross-Site-Scripting (XSS) attack, an attacker is able to read these cookies.

Finding

+ The cookie with the name PHPSESSID does not have the flag httponly set. This may leak sensitive information. This was found on URL https://dvwa.devstack.crashtest.cloud.

How to fix

Cookies that are not marked as secure can be transferred via an unencrypted connection. A man-in-the-middle attack can be used to get the contents of these cookies. Cookies that are not marked as http-only can be read by local scripts. In case of an Cross-Site-Scripting (XSS) attack, an attacker is able to read these cookies. Depending on the cookie content, think of enabling both settings for all cookies. This is especially important for session cookies. More details on how to set these two settings can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/enable-secure-cookies

2.1.17 Insecure Cookies

Severity

Base Score:	medium (4.8/10)
Impact:	low (2.5/10)
Exploitability:	low (2.2/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

Cookies that are not marked as secure can be transferred via an unencrypted connection. A man-in-the-middle attack can be used to get the contents of these cookies.

Finding

+ The cookie with the name PHPSESSID does not have the flag secure set. This may leak sensitive information. This was found on URL https://dvwa.devstack.crashtest.cloud.

How to fix

Cookies that are not marked as secure can be transferred via an unencrypted connection. A man-in-the-middle attack can be used to get the contents of these cookies. Cookies that are not marked as http-only can be read by local scripts. In case of an Cross-Site-Scripting (XSS) attack, an attacker is able to read these cookies. Depending on the cookie content, think of enabling both settings for all cookies. This is especially important for session cookies. More details on how to set these two settings can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/enable-secure-cookies

2.2 FINGERPRINTING

2.2.1 What is this?

The responses a server sends to its client often contain more information then necessary. This surplus of information makes it possible to draw conclusions about the servers software or used programming languages. It could reveal the version of the web application and the libraries in use. The analysis of these information is called fingerprinting. Based on fingerprinting, an attacker can get valuable input to plan and carry out his attack. Without it, an attacker is attacking blindly. Whenever a special version of a server or a web application is vulnerable for an attack, crawlers search the web for traces of this version and start an attack if they found one. So it is likely that someone gets attacked just because they leak these information, and therefore show that your application or server is vulnerable.

2.2.2 Fingerprint Web Server

Severity

Base Score:	high (7.5/10)
Impact:	overwritten by CVE
Exploitability:	overwritten by CVE

All values are based on the Common Vulnerability Scoring System v3.

Description

The webserver publicly provides information about itself such as the name or version. This opens attackers the possibility to look for exploits specifically targetting the webserver in its exact version.

Finding

+ The webserver is running Apache 2.4.7 (**34** connected CVE issues have been found. The most severe vulnerability has a CVSS score of high (7.5/10). See Appendix **APACHE 2.4.7 CVE FINDINGS** for a detailed list of the CVEs)

How to fix

The amount of information a server is sharing can be set in its configuration files. Depending on the used webserver, the configuration file can be found on different locations (see Recommendations to find the exact location). In most cases it is sufficient to change one or two settings to avoid publishing unnecessary information. After saving the changes, it is recommended to restart or reload the webserver to activate the changes.

Recommendations

https://wiki.crashtest-security.com/server-version-fingerprinting

2.2.3 Fingerprint Web Application Framework

Severity

Base Score:	critical (10/10)
Impact:	overwritten by CVE
Exploitability:	overwritten by CVE

All values are based on the Common Vulnerability Scoring System v3.

Description

The installed web application framework(s) offer information about their version. This opens attackers the possibility to look for exploits specifically targeting the software running in its exact version.

Finding

+ Found PHP running in version 5.5.9. (**203** connected CVE issues have been found. The most severe vulnerability has a CVSS score of critical (10/10). See Appendix **PHP 5.5.9 CVE FINDINGS** for a detailed list of the CVEs)

How to fix

Depending on the used application there are multiple ways to remove version information. Some applications also share the information in multiple places, which makes it harder to remove it. Common places for version information are the filename of included libraries like "jquery.3.2.1.min.js" or the documentation within a file, where the version number is stated within the first lines. While some information is required to be left within these files as a part of the copyright, other information like the version number can be removed. Other places could be the footer of an application "powered by Wordpress 4.9.1" or meta-tags within the header of the website. Unlike servers, most web applications cannot remove these information via a config file and therefore need to be removed manually, by editing the specific templates and files. More details on how to fix this problem can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/prevent-web-application-framework-information-leakage

2.3 PORTSCAN

2.3.1 What is this?

A port is a kind of door on the server that can be used to connect to a specific service. For a webserver the port 80 and port 443, which are for HTTP/HTTPS, are most likely open to serve the website to the users. Other ports should be closed if they are not needed for any service. The portscanner tests the webserver with a SYN scan for a wide range of possibly open ports and reports them back. If there are any other open ports except of port 80 and port 443, they should be blocked by the firewall if they are not needed.

2.3.2 Portscanner

Severity

Base Score:	informational (0/10)
Impact:	informational (0/10)
Exploitability:	informational (0/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

Unneeded open ports on the webserver opens a large attack surface to a malicious user. This can be used to find unmaintained and possibly vulnerable network services that can be targeted.

Finding

+ Found open port "8099/tcp"
+ Found open port "5222/tcp"
+ Found open port "8080/tcp"
+ Found open port "5901/tcp"
+ Found open port "5432/tcp"
+ Found open port "110/tcp"
+ Found open port "43/tcp"
+ Found open port "700/tcp"
+ Found open port "9200/tcp"
+ Found open port "443/tcp" with service name "Apache httpd"
+ Found open port "465/tcp"
+ Found open port "5900/tcp"
+ Found open port "995/tcp"
+ Found open port "587/tcp"
+ Found open port "993/tcp"
+ Found open port "143/tcp"
+ Found open port "8085/tcp"
+ Found open port "80/tcp" with service name "Apache httpd"
+ Found open port "3389/tcp"

How to fix

Unnecessarily open ports can be closed by setting up a firewall and block connections to all ports except of those that are needed by the server. Furthermore services that are not needed should be uninstalled.

Recommendations

https://wiki.crashtest-security.com/insecure-network-services-open-port-scanner

2.4 FUZZER

2.4.1 What is this?

Fuzzing, or robustness testing, fuzzy testing or negative testing, is a software testing technique that uses random or pre-defined data as input of a program. The random data can be used to simulate the later use, in which not only plausible data must be processed. In this case, the Fuzzer is looking for publicly available default paths through which attackers could gain access to the system. Those default paths may leak sensitive information or grant access to functionality which modifies the application.

2.4.2 Sensitive Data Exposure

Severity

Base Score:	medium (5.3/10)
Impact:	low (1.4/10)
Exploitability:	low (3.9/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

The server grants access to a file or directory which might contain sensitive data. This can either leak sensitive data itself or allow an attacker to use the provided information to prepare a further attack.

Finding

+ Retrieved https://dvwa.devstack.crashtest.cloud/php.ini by using a GET request on the URL without prior knowledge.
+ Retrieved https://dvwa.devstack.crashtest.cloud/about.php by using a GET request on the URL without prior knowledge.
+ Retrieved https://dvwa.devstack.crashtest.cloud/instructions.php by using a GET request on the URL without prior knowledge.
+ Retrieved https://dvwa.devstack.crashtest.cloud/phpinfo.php by using a GET request on the URL without prior knowledge.
+ Retrieved https://dvwa.devstack.crashtest.cloud/docs/ by using a GET request on the URL without prior knowledge.
+ Retrieved https://dvwa.devstack.crashtest.cloud/setup.php by using a GET request on the URL without prior knowledge.
+ Retrieved https://dvwa.devstack.crashtest.cloud/CHANGELOG.md by using a GET request on the URL without prior knowledge.
+ Retrieved https://dvwa.devstack.crashtest.cloud/.git/ by using a GET request on the URL without prior knowledge.
+ Retrieved https://dvwa.devstack.crashtest.cloud/README.md by using a GET request on the URL without prior knowledge.
+ Retrieved https://dvwa.devstack.crashtest.cloud/COPYING by using a GET request on the URL without prior knowledge.
+ Retrieved https://dvwa.devstack.crashtest.cloud/config/ by using a GET request on the URL without prior knowledge.

How to fix

In some cases, it is completely OK to expose certain file paths as long as it is on purpose. While they can be exposed on purpose, others may be unwillingly exposed. These paths can either be protected by Basic Auth (htaccess) or be removed as they might not be needed on a production environment. More details on how to avoid exposing unnecessary information can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/sensitive-data-exposure

2.5 COMMANDINJECTION

2.5.1 What is this?

Command injection is a vulnerability which is caused if the web application executes data from an untrusted source without proper validation. With this vulnerability, an attacker can execute any available system command. This can lead to a entirely compromised system.

2.5.2 Command Injection

Severity

Base Score:	critical (9.8/10)
Impact:	medium (5.9/10)
Exploitability:	low (3.9/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

Command injection allows an attacker to execute arbitrary system commands.

Finding

+ Found command injection in parameter "ip" with method "post" for URL "https://dvwa.devstack.crashtest.cloud/vulnerabilities/exec/", with payload "; echo crashtest-security$((12*12))"

How to fix

Every user input has to be checked for malicious requests by the web application. Untrusted user input should not be passed to functions like "exec()" or "system()" without a sanity check.

Recommendations

https://wiki.crashtest-security.com/command-injection

2.6 FILEINCLUSION

2.6.1 What is this?

Local/remote file inclusion is a vulnerability which is caused by including files into the web application without validating which file is going to be included. The attacker attempts to include arbitrary files from the webserverś hard drive, to identify existing user accounts or passwords. In some cases it is possible to include files from a remote server, which is under control of the attacker. This vulnerability can lead to exposing sensitive files on the webserver and could also result in a remote code execution, which would entirely compromise the target machine.

2.6.2 Local File Inclusion

Severity

Base Score:	critical (9.8/10)
Impact:	medium (5.9/10)
Exploitability:	low (3.9/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

Local file inclusion allows an attacker to include arbitrary local files into the website

Finding

+ Found file inclusion with method ”get” for parameter ”page” on ”https://dvwa.devstack.crashtest.cloud/vulnerabilities/fi/.” with payload ”/etc/passwd”

How to fix

Every user input has to be checked for malicious requests by the web application. For example, the files which are allowed to be included (whitelisted) are written into an array. For every request the web application should check the whitelist if the required file is allowed for inclusion.

Recommendations

https://wiki.crashtest-security.com/file-inclusion

2.7 SQLINJECTION

2.7.1 What is this?

SQL injection refers to the exploitation of a SQL database vulnerability caused by the lack of masking or validation of meta-characters in user input. The attacker attempts to inject his own database commands through the application which has access to the database. As the request is not validated correctly, the inserted code changes the original SQL commands and therefore alters the results in favor of the attacker. With a successful attack, the attacker is able to spy on data, modify it or delete it altogether, and gain control over the server. For this to work, the attacker has different ways to breach the system. For example it is possible to find a way into the system via response time or error messages.

2.7.2 SQL Injection

Severity

Base Score:	critical (9.1/10)
Impact:	medium (5.2/10)
Exploitability:	low (3.9/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

Your application is vulnerable for an SQL injection. This allows an attacker to run SQL code in your database so that he may retrieve, change or delete data from your database.

Finding

+ Found boolean-based blind sqlinjection for parameter username (GET) on https://dvwa.devstack.crashtest.cloud/vulnerabilities/brute/ with payload Login=Login&password=Crashtest123!&username=xyz' AND 7725=(SELECT (CASE WHEN (7725=7725) THEN 7725 ELSE (SELECT 3607 UNION SELECT 8444) END))– -

How to fix

The simple answer is: Sanitize the users input before sending it to the database. Sanitizing includes escaping all potentially harmful characters to not let them effect the resulting SQL query. There are multiple ways to do so and most common frameworks also support ways to simplify this step. One possible solutions is, to use Object-relational mapping libraries to take care of the sanitizing. In case direct SQL queries are required, it is recommended to use so called "prepared statements". These are queries containing placeholders for the users input and while binding the input in the query, the users data will be escaped. More details on how to use these methods can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/sql-injections

2.8 XSS

2.8.1 What is this?

Cross-site scripting (XSS) refers to exploiting a computer security vulnerability in web applications by causing an attacker to infect web pages with client-side scripts that are invoked by other users. In 2007, XSS accounted for about 80% of the exploited vulnerabilities in web applications on cross-site scripting accounts. The impact of XSS can be between a small nuisance and a significant security risk, depending on the sensitivity of the data. With XSS, an attacker can for example bypass access controls, steal client data or place external content like advertisement, redirects or spam in an application. Cross-site scripting provides the foundation for a variety of other attacks, such as session hijacking or session fixation.

2.8.2 Cross-Site Scripting (XSS)

Severity

Base Score:	medium (6.1/10)
Impact:	low (2.7/10)
Exploitability:	low (2.8/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

Cross-Site Scripting (XSS) allows an attacker to send malicious code to a different user.

Finding

+ Found possible XSS vulnerability on site dvwa.devstack.crashtest.cloud/vulnerabilities/xss_s/. The parameter 'btnSign' seems vulnerable for payload '<svg dddf7814-868e-4170-aac1-58ad9b90eee0 "ons>'
+ Found possible XSS vulnerability on site dvwa.devstack.crashtest.cloud/vulnerabilities/xss_s/. The parameter 'txtName' seems vulnerable for payload '<svg dddf7814-868e-4170-aac1-58ad9b90eee0 "ons>'
+ Found possible XSS vulnerability on site dvwa.devstack.crashtest.cloud/vulnerabilities/xss_r/. The parameter 'name' seems vulnerable for payload '<svg 2e10116c-7243-4bee-a867-e60201086244 "ons>'

How to fix

XSS can be prevented by sanitizing the users input before saving to a database or returning it back to the user. In most cases the attacker injects JavaScript into the application. By escaping the "<script>" tags, this can be avoided. More details on how to fix this problem can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/cross-site-scripting

2.8.3 DOM based Cross-Site Scripting (XSS)

Severity

Base Score:	medium (6.1/10)
Impact:	low (2.7/10)
Exploitability:	low (2.8/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

DOM based Cross-Site Scripting (XSS) allows an attacker to send malicious code to a different user by modifying the DOM environment in the browser of the user.

Finding

+ The potentially vulnerable code was found on url 'dvwa.devstack.crashtest.cloud/ids_log.php'. An attacker may be able to inject JavaScript using the the code 'window.name' at line 42:188 and control its display using the code 'eval' at line 42:183

How to fix

DOM based XSS can be prevented by using save JavaScript properties like 'element.textContent' for untrusted user input. Furthermore DOM based XSS can be avoided by using JavaScript methods like 'innerText' or 'textContent' instead of 'innerHtml'. More details on how to fix this problem can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/cross-site-scripting

2.9 XXE

2.9.1 What is this?

XXE is a vulnerability that arises if web applications handle XML documents from an untrusted source without proper validation. In order to exploit this vulnerability an attacker extends the XML document with a document type definition (DTD) that includes a external entity. If the website passes the XML document to the XML parser the external entity will be called in some cases. This can lead to sensitive data exposure or even remote code execution.

2.9.2 XXE

Severity

Base Score:	critical (9.4/10)
Impact:	medium (5.5/10)
Exploitability:	low (3.9/10)

All values are based on the Common Vulnerability Scoring System v3.

Description

XXE allows an attacker to inject malicious XML documents into the website, which is then executed. This can lead to sensitive data disclosure or remote code execution.

Finding

+ Found XXE in parameter "xml" with method "get" for URL "https://dvwa.devstack.crashtest.cloud/vulnerabilities/xxe/", with payload "<?xml version='1.0' encoding='utf-8'?><!DOCTYPE creds [<!ELEMENT user ANY ><!ELEMENT pass ANY ><!ENTITY user SYSTEM 'file:///etc/passwd'>]><creds><user>%26user;</user><pass>%26user;</pass></creds>"

How to fix

If XML documents are communicated from an untrusted source the XML processor should be configured to disallow any declared document type definition (DTD) included in the XML document.

Recommendations

https://wiki.crashtest-security.com/xxe-processing

2.10 Appendix

2.10.1 APACHE 2.4.7 CVE FINDINGS

STATE	APACHE 2.4.7
5.0	**CVE-2013-6438:** The dav_xml_get_cdata function in main/util.c in the mod_dav module in the Apache HTTP Server before 2.4.8 does not properly remove whitespace characters from CDATA sections, which allows remote attackers to cause a denial of service (daemon crash) via a crafted DAV WRITE request.
5.0	**CVE-2014-0098:** The log_cookie function in mod_log_config.c in the mod_log_config module in the Apache HTTP Server before 2.4.8 allows remote attackers to cause a denial of service (segmentation fault and daemon crash) via a crafted cookie that is not properly handled during truncation.
4.3	**CVE-2014-0118:** The deflate_in_filter function in mod_deflate.c in the mod_deflate module in the Apache HTTP Server before 2.4.10, when request body decompression is enabled, allows remote attackers to cause a denial of service (resource consumption) via crafted request data that decompresses to a much larger size.
6.8	**CVE-2014-0226:** Race condition in the mod_status module in the Apache HTTP Server before 2.4.10 allows remote attackers to cause a denial of service (heap-based buffer overflow), or possibly obtain sensitive credential information or execute arbitrary code, via a crafted request that triggers improper scoreboard handling within the status_handler function in modules/generators/mod_status.c and the lua_ap_scoreboard_worker function in modules/lua/lua_request.c.
5.0	**CVE-2014-0231:** The mod_cgid module in the Apache HTTP Server before 2.4.10 does not have a timeout mechanism, which allows remote attackers to cause a denial of service (process hang) via a request to a CGI script that does not read from its stdin file descriptor.
5.0	**CVE-2015-0228:** The lua_websocket_read function in lua_request.c in the mod_lua module in the Apache HTTP Server through 2.4.12 allows remote attackers to cause a denial of service (child-process crash) by sending a crafted WebSocket Ping frame after a Lua script has called the wsupgrade function.
5.0	**CVE-2015-3183:** The chunked transfer coding implementation in the Apache HTTP Server before 2.4.14 does not properly parse chunk headers, which allows remote attackers to conduct HTTP request smuggling attacks via a crafted request, related to mishandling of large chunk-size values and invalid chunk-extension characters in modules/http/http_filters.c.
5.1	**CVE-2016-5387:** The Apache HTTP Server through 2.4.23 follows RFC 3875 section 4.1.18 and therefore does not protect applications from the presence of untrusted client data in the HTTP_PROXY environment variable, which might allow remote attackers to redirect an application's outbound HTTP traffic to an arbitrary proxy server via a crafted Proxy header in an HTTP request, aka an "httpoxy" issue. NOTE: the vendor states "This mitigation has been assigned the identifier CVE-2016-5387"; in other words, this is not a CVE ID for a vulnerability.
3.3	**CVE-2016-8612:** Apache HTTP Server mod_cluster before version httpd 2.4.23 is vulnerable to an Improper Input Validation in the protocol parsing logic in the load balancer resulting in a Segmentation Fault in the serving httpd process.

STATE	APACHE 2.4.7
6.4	**CVE-2017-9788:** In Apache httpd before 2.2.34 and 2.4.x before 2.4.27, the value placeholder in [Proxy-]Authorization headers of type 'Digest' was not initialized or reset before or between successive key=value assignments by mod_auth_digest. Providing an initial key with no '=' assignment could reflect the stale value of uninitialized pool memory used by the prior request, leading to leakage of potentially confidential information, and a segfault in other cases resulting in denial of service.
5.0	**CVE-2017-9798:** Apache httpd allows remote attackers to read secret data from process memory if the Limit directive can be set in a user's .htaccess file, or if httpd.conf has certain misconfigurations, aka Optionsbleed. This affects the Apache HTTP Server through 2.2.34 and 2.4.x through 2.4.27. The attacker sends an unauthenticated OPTIONS HTTP request when attempting to read secret data. This is a use-after-free issue and thus secret data is not always sent, and the specific data depends on many factors including configuration. Exploitation with .htaccess can be blocked with a patch to the ap_limit_section function in server/core.c.
4.3	**CVE-2018-1301:** A specially crafted request could have crashed the Apache HTTP Server prior to version 2.4.30, due to an out of bound access after a size limit is reached by reading the HTTP header. This vulnerability is considered very hard if not impossible to trigger in non-debug mode (both log and build level), so it is classified as low risk for common server usage.
4.3	**CVE-2018-1302:** When an HTTP/2 stream was destroyed after being handled, the Apache HTTP Server prior to version 2.4.30 could have written a NULL pointer potentially to an already freed memory. The memory pools maintained by the server make this vulnerability hard to trigger in usual configurations, the reporter and the team could not reproduce it outside debug builds, so it is classified as low risk.
5.0	**CVE-2018-1303:** A specially crafted HTTP request header could have crashed the Apache HTTP Server prior to version 2.4.30 due to an out of bound read while preparing data to be cached in shared memory. It could be used as a Denial of Service attack against users of mod_cache_socache. The vulnerability is considered as low risk since mod_cache_socache is not widely used, mod_cache_disk is not concerned by this vulnerability.
5.0	**CVE-2018-17189:** In Apache HTTP server versions 2.4.37 and prior, by sending request bodies in a slow loris way to plain resources, the h2 stream for that request unnecessarily occupied a server thread cleaning up that incoming data. This affects only HTTP/2 (mod_http2) connections.
4.3	**CVE-2016-4975:** Possible CRLF injection allowing HTTP response splitting attacks for sites which use mod_userdir. This issue was mitigated by changes made in 2.4.25 and 2.2.32 which prohibit CR or LF injection into the "Location" or other outbound header key or value. Fixed in Apache HTTP Server 2.4.25 (Affected 2.4.1-2.4.23). Fixed in Apache HTTP Server 2.2.32 (Affected 2.2.0-2.2.31).
7.5	**CVE-2017-7679:** In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, mod_mime can read one byte past the end of a buffer when sending a malicious Content-Type response header.
6.8	**CVE-2018-1312:** In Apache httpd 2.2.0 to 2.4.29, when generating an HTTP Digest authentication challenge, the nonce sent to prevent reply attacks was not correctly generated using a pseudo-random seed. In a cluster of servers using a common Digest authentication configuration, HTTP requests could be replayed across servers by an attacker without detection.

STATE	APACHE 2.4.7
4.3	**CVE-2014-8109:** mod_lua.c in the mod_lua module in the Apache HTTP Server 2.3.x and 2.4.x through 2.4.10 does not support an httpd configuration in which the same Lua authorization provider is used with different arguments within different contexts, which allows remote attackers to bypass intended access restrictions in opportunistic circumstances by leveraging multiple Require directives, as demonstrated by a configuration that specifies authorization for one group to access a certain directory, and authorization for a second group to access a second directory.
4.3	**CVE-2015-3185:** The ap_some_auth_required function in server/request.c in the Apache HTTP Server 2.4.x before 2.4.14 does not consider that a Require directive may be associated with an authorization setting rather than an authentication setting, which allows remote attackers to bypass intended access restrictions in opportunistic circumstances by leveraging the presence of a module that relies on the 2.2 API behavior.
5.0	**CVE-2016-0736:** In Apache HTTP Server versions 2.4.0 to 2.4.23, mod_session_crypto was encrypting its data/cookie using the configured ciphers with possibly either CBC or ECB modes of operation (AES256-CBC by default), hence no selectable or builtin authenticated encryption. This made it vulnerable to padding oracle attacks, particularly with CBC.
5.0	**CVE-2016-2161:** In Apache HTTP Server versions 2.4.0 to 2.4.23, malicious input to mod_auth_digest can cause the server to crash, and each instance continues to crash even for subsequently valid requests.
6.8	**CVE-2017-15715:** In Apache httpd 2.4.0 to 2.4.29, the expression specified in <FilesMatch> could match '$' to a newline character in a malicious filename, rather than matching only the end of the filename. This could be exploited in environments where uploads of some files are are externally blocked, but only by matching the trailing portion of the filename.
3.5	**CVE-2018-1283:** In Apache httpd 2.4.0 to 2.4.29, when mod_session is configured to forward its session data to CGI applications (SessionEnv on, not the default), a remote user may influence their content by using a "Session" header. This comes from the "HTTP_SESSION" variable name used by mod_session to forward its data to CGIs, since the prefix "HTTP_" is also used by the Apache HTTP Server to pass HTTP header fields, per CGI specifications.
5.0	**CVE-2018-17199:** In Apache HTTP Server 2.4 release 2.4.37 and prior, mod_session checks the session expiry time before decoding the session. This causes session expiry time to be ignored for mod_session_cookie sessions since the expiry time is loaded when the session is decoded.
6.0	**CVE-2019-0217:** In Apache HTTP Server 2.4 release 2.4.38 and prior, a race condition in mod_auth_digest when running in a threaded server could allow a user with valid credentials to authenticate using another username, bypassing configured access control restrictions.
5.0	**CVE-2019-0220:** A vulnerability was found in Apache HTTP Server 2.4.0 to 2.4.38. When the path component of a request URL contains multiple consecutive slashes ('/'), directives such as LocationMatch and RewriteRule must account for duplicates in regular expressions while other aspects of the servers processing will implicitly collapse them.

STATE	APACHE 2.4.7
4.3	**CVE-2019-10092:** In Apache HTTP Server 2.4.0-2.4.39, a limited cross-site scripting issue was reported affecting the mod_proxy error page. An attacker could cause the link on the error page to be malformed and instead point to a page of their choice. This would only be exploitable where a server was set up with proxying enabled but was misconfigured in such a way that the Proxy Error page was displayed.
5.8	**CVE-2019-10098:** In Apache HTTP server 2.4.0 to 2.4.39, Redirects configured with mod_rewrite that were intended to be self-referential might be fooled by encoded newlines and redirect instead to an unexpected URL within the request URL.
5.0	**CVE-2015-3184:** mod_authz_svn in Apache Subversion 1.7.x before 1.7.21 and 1.8.x before 1.8.14, when using Apache httpd 2.4.x, does not properly restrict anonymous access, which allows remote anonymous users to read hidden files via the path name.
5.0	**CVE-2016-8743:** Apache HTTP Server, in all releases prior to 2.2.32 and 2.4.25, was liberal in the whitespace accepted from requests and sent in response lines and headers. Accepting these different behaviors represented a security concern when httpd participates in any chain of proxies or interacts with back-end application servers, either through mod_proxy or using conventional CGI mechanisms, and may result in request smuggling, response splitting and cache pollution.
5.0	**CVE-2017-15710:** In Apache httpd 2.0.23 to 2.0.65, 2.2.0 to 2.2.34, and 2.4.0 to 2.4.29, mod_authnz_ldap, if configured with AuthLDAPCharsetConfig, uses the Accept-Language header value to lookup the right charset encoding when verifying the user's credentials. If the header value is not present in the charset conversion table, a fallback mechanism is used to truncate it to a two characters value to allow a quick retry (for example, 'en-US' is truncated to 'en'). A header value of less than two characters forces an out of bound write of one NUL byte to a memory location that is not part of the string. In the worst case, quite unlikely, the process would crash which could be used as a Denial of Service attack. In the more likely case, this memory is already reserved for future use and the issue has no effect at all.
5.0	**CVE-2014-3523:** Memory leak in the winnt_accept function in server/mpm/winnt/child.c in the WinNT MPM in the Apache HTTP Server 2.4.x before 2.4.10 on Windows, when the default AcceptFilter is enabled, allows remote attackers to cause a denial of service (memory consumption) via crafted requests.
4.3	**CVE-2014-0117:** The mod_proxy module in the Apache HTTP Server 2.4.x before 2.4.10, when a reverse proxy is enabled, allows remote attackers to cause a denial of service (child-process crash) via a crafted HTTP Connection header. Per vendor advisory http://httpd.apache.org/security/vulnerabilities_24.html "A flaw was found in mod_proxy in httpd versions 2.4.6 to 2.4.9."

2.10.2 PHP 5.5.9 CVE FINDINGS

STATE	PHP 5.5.9
4.3	**CVE-2014-0207:** The cdf_read_short_sector function in cdf.c in file before 5.19, as used in the Fileinfo component in PHP before 5.4.30 and 5.5.x before 5.5.14, allows remote attackers to cause a denial of service (assertion failure and application exit) via a crafted CDF file.

STATE	PHP 5.5.9
5.0	**CVE-2014-3478:** Buffer overflow in the mconvert function in softmagic.c in file before 5.19, as used in the Fileinfo component in PHP before 5.4.30 and 5.5.x before 5.5.14, allows remote attackers to cause a denial of service (application crash) via a crafted Pascal string in a FILE_PSTRING conversion.
4.3	**CVE-2014-3479:** The cdf_check_stream_offset function in cdf.c in file before 5.19, as used in the Fileinfo component in PHP before 5.4.30 and 5.5.x before 5.5.14, relies on incorrect sector-size data, which allows remote attackers to cause a denial of service (application crash) via a crafted stream offset in a CDF file.
4.3	**CVE-2014-3480:** The cdf_count_chain function in cdf.c in file before 5.19, as used in the Fileinfo component in PHP before 5.4.30 and 5.5.x before 5.5.14, does not properly validate sector-count data, which allows remote attackers to cause a denial of service (application crash) via a crafted CDF file.
4.3	**CVE-2014-3487:** The cdf_read_property_info function in file before 5.19, as used in the Fileinfo component in PHP before 5.4.30 and 5.5.x before 5.5.14, does not properly validate a stream offset, which allows remote attackers to cause a denial of service (application crash) via a crafted CDF file.
4.3	**CVE-2014-3587:** Integer overflow in the cdf_read_property_info function in cdf.c in file through 5.19, as used in the Fileinfo component in PHP before 5.4.32 and 5.5.x before 5.5.16, allows remote attackers to cause a denial of service (application crash) via a crafted CDF file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2012-1571.
5.0	**CVE-2014-9652:** The mconvert function in softmagic.c in file before 5.21, as used in the Fileinfo component in PHP before 5.4.37, 5.5.x before 5.5.21, and 5.6.x before 5.6.5, does not properly handle a certain string-length field during a copy of a truncated version of a Pascal string, which might allow remote attackers to cause a denial of service (out-of-bounds memory access and application crash) via a crafted file.
7.5	**CVE-2014-9653:** readelf.c in file before 5.22, as used in the Fileinfo component in PHP before 5.4.37, 5.5.x before 5.5.21, and 5.6.x before 5.6.5, does not consider that pread calls sometimes read only a subset of the available data, which allows remote attackers to cause a denial of service (uninitialized memory access) or possibly have unspecified other impact via a crafted ELF file.
4.3	**CVE-2014-9767:** Directory traversal vulnerability in the ZipArchive::extractTo function in ext/zip/php_zip.c in PHP before 5.4.45, 5.5.x before 5.5.29, and 5.6.x before 5.6.13 and ext/zip/ext_zip.cpp in HHVM before 3.12.1 allows remote attackers to create arbitrary empty directories via a crafted ZIP archive.
5.0	**CVE-2015-4024:** Algorithmic complexity vulnerability in the multipart_buffer_headers function in main/rfc1867.c in PHP before 5.4.41, 5.5.x before 5.5.25, and 5.6.x before 5.6.9 allows remote attackers to cause a denial of service (CPU consumption) via crafted form data that triggers an improper order-of-growth outcome.
7.5	**CVE-2016-4543:** The exif_process_IFD_in_JPEG function in ext/exif/exif.c in PHP before 5.5.35, 5.6.x before 5.6.21, and 7.x before 7.0.6 does not validate IFD sizes, which allows remote attackers to cause a denial of service (out-of-bounds read) or possibly have unspecified other impact via crafted header data.

STATE	PHP 5.5.9
5.1	**CVE-2016-5385:** PHP through 7.0.8 does not attempt to address RFC 3875 section 4.1.18 namespace conflicts and therefore does not protect applications from the presence of untrusted client data in the HTTP_PROXY environment variable, which might allow remote attackers to redirect an application's outbound HTTP traffic to an arbitrary proxy server via a crafted Proxy header in an HTTP request, as demonstrated by (1) an application that makes a getenv('HTTP_PROXY') call or (2) a CGI configuration of PHP, aka an "httpoxy" issue.
5.0	**CVE-2014-9709:** The GetCode_ function in gd_gif_in.c in GD 2.1.1 and earlier, as used in PHP before 5.5.21 and 5.6.x before 5.6.5, allows remote attackers to cause a denial of service (buffer over-read and application crash) via a crafted GIF image that is improperly handled by the gdImageCreateFromGif function.
5.0	**CVE-2015-8877:** The gdImageScaleTwoPass function in gd_interpolation.c in the GD Graphics Library (aka libgd) before 2.2.0, as used in PHP before 5.6.12, uses inconsistent allocate and free approaches, which allows remote attackers to cause a denial of service (memory consumption) via a crafted call, as demonstrated by a call to the PHP imagescale function.
6.4	**CVE-2016-5116:** gd_xbm.c in the GD Graphics Library (aka libgd) before 2.2.0, as used in certain custom PHP 5.5.x configurations, allows context-dependent attackers to obtain sensitive information from process memory or cause a denial of service (stack-based buffer under-read and application crash) via a long name.
6.8	**CVE-2019-6977:** gdImageColorMatch in gd_color_match.c in the GD Graphics Library (aka LibGD) 2.2.5, as used in the imagecolormatch function in PHP before 5.6.40, 7.x before 7.1.26, 7.2.x before 7.2.14, and 7.3.x before 7.3.1, has a heap-based buffer overflow. This can be exploited by an attacker who is able to trigger imagecolormatch calls with crafted image data.
5.0	**CVE-2017-16642:** In PHP before 5.6.32, 7.x before 7.0.25, and 7.1.x before 7.1.11, an error in the date extension's timelib_meridian handling of 'front of' and 'back of' directives could be used by attackers able to supply date strings to leak information from the interpreter, related to ext/date/lib/parse_date.c out-of-bounds reads affecting the php_parse_date function. NOTE: this is a different issue than CVE-2017-11145.
1.9	**CVE-2018-10545:** An issue was discovered in PHP before 5.6.35, 7.0.x before 7.0.29, 7.1.x before 7.1.16, and 7.2.x before 7.2.4. Dumpable FPM child processes allow bypassing opcache access controls because fpm_unix.c makes a PR_SET_DUMPABLE prctl call, allowing one user (in a multiuser environment) to obtain sensitive information from the process memory of a second user's PHP applications by running gcore on the PID of the PHP-FPM worker process.
5.0	**CVE-2018-10546:** An issue was discovered in PHP before 5.6.36, 7.0.x before 7.0.30, 7.1.x before 7.1.17, and 7.2.x before 7.2.5. An infinite loop exists in ext/iconv/iconv.c because the iconv stream filter does not reject invalid multibyte sequences.
4.3	**CVE-2018-10547:** An issue was discovered in ext/phar/phar_object.c in PHP before 5.6.36, 7.0.x before 7.0.30, 7.1.x before 7.1.17, and 7.2.x before 7.2.5. There is Reflected XSS on the PHAR 403 and 404 error pages via request data of a request for a .phar file. NOTE: this vulnerability exists because of an incomplete fix for CVE-2018-5712.
5.0	**CVE-2018-10548:** An issue was discovered in PHP before 5.6.36, 7.0.x before 7.0.30, 7.1.x before 7.1.17, and 7.2.x before 7.2.5. ext/ldap/ldap.c allows remote LDAP servers to cause a denial of service (NULL pointer dereference and application crash) because of mishandling of the ldap_get_dn return value.

STATE	PHP 5.5.9
6.8	**CVE-2018-10549:** An issue was discovered in PHP before 5.6.36, 7.0.x before 7.0.30, 7.1.x before 7.1.17, and 7.2.x before 7.2.5. exif_read_data in ext/exif/exif.c has an out-of-bounds read for crafted JPEG data because exif_iif_add_value mishandles the case of a MakerNote that lacks a final " character.
4.3	**CVE-2018-14851:** exif_process_IFD_in_MAKERNOTE in ext/exif/exif.c in PHP before 5.6.37, 7.0.x before 7.0.31, 7.1.x before 7.1.20, and 7.2.x before 7.2.8 allows remote attackers to cause a denial of service (out-of-bounds read and application crash) via a crafted JPEG file.
5.0	**CVE-2018-14883:** An issue was discovered in PHP before 5.6.37, 7.0.x before 7.0.31, 7.1.x before 7.1.20, and 7.2.x before 7.2.8. An Integer Overflow leads to a heap-based buffer over-read in exif_thumbnail_extract of exif.c.
5.0	**CVE-2018-15132:** An issue was discovered in ext/standard/link_win32.c in PHP before 5.6.37, 7.0.x before 7.0.31, 7.1.x before 7.1.20, and 7.2.x before 7.2.8. The linkinfo function on Windows doesn't implement the open_basedir check. This could be abused to find files on paths outside of the allowed directories.
4.3	**CVE-2018-17082:** The Apache2 component in PHP before 5.6.38, 7.0.x before 7.0.32, 7.1.x before 7.1.22, and 7.2.x before 7.2.10 allows XSS via the body of a "Transfer-Encoding: chunked" request, because the bucket brigade is mishandled in the php_handler function in sapi/apache2handler/sapi_apache2.c.
7.5	**CVE-2019-9020:** An issue was discovered in PHP before 5.6.40, 7.x before 7.1.26, 7.2.x before 7.2.14, and 7.3.x before 7.3.1. Invalid input to the function xmlrpc_decode() can lead to an invalid memory access (heap out of bounds read or read after free). This is related to xml_elem_parse_buf in ext/xmlrpc/libxmlrpc/xml_element.c.
7.5	**CVE-2019-9021:** An issue was discovered in PHP before 5.6.40, 7.x before 7.1.26, 7.2.x before 7.2.14, and 7.3.x before 7.3.1. A heap-based buffer over-read in PHAR reading functions in the PHAR extension may allow an attacker to read allocated or unallocated memory past the actual data when trying to parse the file name, a different vulnerability than CVE-2018-20783. This is related to phar_detect_phar_fname_ext in ext/phar/phar.c.
7.5	**CVE-2019-9023:** An issue was discovered in PHP before 5.6.40, 7.x before 7.1.26, 7.2.x before 7.2.14, and 7.3.x before 7.3.1. A number of heap-based buffer over-read instances are present in mbstring regular expression functions when supplied with invalid multibyte data. These occur in ext/mbstring/oniguruma/regcomp.c, ext/mbstring/oniguruma/regexec.c, ext/mbstring/oniguruma/regparse.c, ext/mbstring/oniguruma/enc/unicode.c, and ext/mbstring/oniguruma/src/utf32_be.c when a multibyte regular expression pattern contains invalid multibyte sequences.
5.0	**CVE-2019-9024:** An issue was discovered in PHP before 5.6.40, 7.x before 7.1.26, 7.2.x before 7.2.14, and 7.3.x before 7.3.1. xmlrpc_decode() can allow a hostile XMLRPC server to cause PHP to read memory outside of allocated areas in base64_decode_xmlrpc in ext/xmlrpc/libxmlrpc/base64.c.
5.0	**CVE-2019-9637:** An issue was discovered in PHP before 7.1.27, 7.2.x before 7.2.16, and 7.3.x before 7.3.3. Due to the way rename() across filesystems is implemented, it is possible that file being renamed is briefly available with wrong permissions while the rename is ongoing, thus enabling unauthorized users to access the data.

STATE	PHP 5.5.9
5.0	**CVE-2019-9638:** An issue was discovered in the EXIF component in PHP before 7.1.27, 7.2.x before 7.2.16, and 7.3.x before 7.3.3. There is an uninitialized read in exif_process_IFD_in_MAKERNOTE because of mishandling the maker_note->offset relationship to value_len.
5.0	**CVE-2019-9639:** An issue was discovered in the EXIF component in PHP before 7.1.27, 7.2.x before 7.2.16, and 7.3.x before 7.3.3. There is an uninitialized read in exif_process_IFD_in_MAKERNOTE because of mishandling the data_len variable.
7.5	**CVE-2019-9641:** An issue was discovered in the EXIF component in PHP before 7.1.27, 7.2.x before 7.2.16, and 7.3.x before 7.3.3. There is an uninitialized read in exif_process_IFD_in_TIFF.
7.5	**CVE-2015-2331:** Integer overflow in the _zip_cdir_new function in zip_dirent.c in libzip 0.11.2 and earlier, as used in the ZIP extension in PHP before 5.4.39, 5.5.x before 5.5.23, and 5.6.x before 5.6.7 and other products, allows remote attackers to cause a denial of service (application crash) or possibly execute arbitrary code via a ZIP archive that contains many entries, leading to a heap-based buffer overflow.
7.5	**CVE-2015-1351:** Use-after-free vulnerability in the _zend_shared_memdup function in zend_shared_alloc.c in the OPcache extension in PHP through 5.6.7 allows remote attackers to cause a denial of service or possibly have unspecified other impact via unknown vectors. CWE-416: Use After Free
3.6	**CVE-2014-5459:** The PEAR_REST class in REST.php in PEAR in PHP through 5.6.0 allows local users to write to arbitrary files via a symlink attack on a (1) rest.cachefile or (2) rest.cacheid file in /tmp/pear/cache/, related to the retrieveCacheFirst and useLocalCache functions.
4.6	**CVE-2013-6501:** The default soap.wsdl_cache_dir setting in (1) php.ini-production and (2) php.ini-development in PHP through 5.6.7 specifies the /tmp directory, which makes it easier for local users to conduct WSDL injection attacks by creating a file under /tmp with a predictable filename that is used by the get_sdl function in ext/soap/php_sdl.c.
5.0	**CVE-2014-0236:** file before 5.18, as used in the Fileinfo component in PHP before 5.6.0, allows remote attackers to cause a denial of service (NULL pointer dereference and application crash) via a zero root_storage value in a CDF file, related to cdf.c and readcdf.c. CWE-476: NULL Pointer Dereference
5.0	**CVE-2014-0237:** The cdf_unpack_summary_info function in cdf.c in the Fileinfo component in PHP before 5.4.29 and 5.5.x before 5.5.13 allows remote attackers to cause a denial of service (performance degradation) by triggering many file_printf calls.
5.0	**CVE-2014-0238:** The cdf_read_property_info function in cdf.c in the Fileinfo component in PHP before 5.4.29 and 5.5.x before 5.5.13 allows remote attackers to cause a denial of service (infinite loop or out-of-bounds memory access) via a vector that (1) has zero length or (2) is too long.

STATE	PHP 5.5.9
7.5	**CVE-2014-3515:** The SPL component in PHP before 5.4.30 and 5.5.x before 5.5.14 incorrectly anticipates that certain data structures will have the array data type after unserialization, which allows remote attackers to execute arbitrary code via a crafted string that triggers use of a Hashtable destructor, related to ”type confusion” issues in (1) ArrayObject and (2) SPLObject-Storage.
6.8	**CVE-2014-3597:** Multiple buffer overflows in the php_parserr function in ext/standard/dns.c in PHP before 5.4.32 and 5.5.x before 5.5.16 allow remote DNS servers to cause a denial of service (application crash) or possibly execute arbitrary code via a crafted DNS record, related to the dns_get_record function and the dn_expand function. NOTE: this issue exists because of an incomplete fix for CVE-2014-4049.
5.0	**CVE-2014-3668:** Buffer overflow in the date_from_ISO8601 function in the mkgmtime implementation in libxmlrpc/xmlrpc.c in the XMLRPC extension in PHP before 5.4.34, 5.5.x before 5.5.18, and 5.6.x before 5.6.2 allows remote attackers to cause a denial of service (application crash) via (1) a crafted first argument to the xmlrpc_set_type function or (2) a crafted argument to the xmlrpc_decode function, related to an out-of-bounds read operation.
7.5	**CVE-2014-3669:** Integer overflow in the object_custom function in ext/standard/-var_unserializer.c in PHP before 5.4.34, 5.5.x before 5.5.18, and 5.6.x before 5.6.2 allows remote attackers to cause a denial of service (application crash) or possibly execute arbitrary code via an argument to the unserialize function that triggers calculation of a large length value.
6.8	**CVE-2014-3670:** The exif_ifd_make_value function in exif.c in the EXIF extension in PHP before 5.4.34, 5.5.x before 5.5.18, and 5.6.x before 5.6.2 operates on floating-point arrays incorrectly, which allows remote attackers to cause a denial of service (heap memory corruption and application crash) or possibly execute arbitrary code via a crafted JPEG image with TIFF thumbnail data that is improperly handled by the exif_thumbnail function.
3.3	**CVE-2014-3981:** acinclude.m4, as used in the configure script in PHP 5.5.13 and earlier, allows local users to overwrite arbitrary files via a symlink attack on the /tmp/phpglibccheck file.
4.6	**CVE-2014-4670:** Use-after-free vulnerability in ext/spl/spl_dllist.c in the SPL component in PHP through 5.5.14 allows context-dependent attackers to cause a denial of service or possibly have unspecified other impact via crafted iterator usage within applications in certain web-hosting environments. CWE-416: Use After Free
4.6	**CVE-2014-4698:** Use-after-free vulnerability in ext/spl/spl_array.c in the SPL component in PHP through 5.5.14 allows context-dependent attackers to cause a denial of service or possibly have unspecified other impact via crafted ArrayIterator usage within applications in certain web-hosting environments. CWE-416: Use After Free

STATE	PHP 5.5.9
2.6	**CVE-2014-4721:** The phpinfo implementation in ext/standard/info.c in PHP before 5.4.30 and 5.5.x before 5.5.14 does not ensure use of the string data type for the PHP_AUTH_PW, PHP_AUTH_TYPE, PHP_AUTH_USER, and PHP_SELF variables, which might allow context-dependent attackers to obtain sensitive information from process memory by using the integer data type with crafted values, related to a "type confusion" vulnerability, as demonstrated by reading a private SSL key in an Apache HTTP Server web-hosting environment with mod_ssl and a PHP 5.3.x mod_php.
7.5	**CVE-2014-8142:** Use-after-free vulnerability in the process_nested_data function in ext/standard/var_unserializer.re in PHP before 5.4.36, 5.5.x before 5.5.20, and 5.6.x before 5.6.4 allows remote attackers to execute arbitrary code via a crafted unserialize call that leverages improper handling of duplicate keys within the serialized properties of an object, a different vulnerability than CVE-2004-1019. CWE-416: Use After Free
7.5	**CVE-2014-9425:** Double free vulnerability in the zend_ts_hash_graceful_destroy function in zend_ts_hash.c in the Zend Engine in PHP through 5.5.20 and 5.6.x through 5.6.4 allows remote attackers to cause a denial of service or possibly have unspecified other impact via unknown vectors. CWE-415: Double Free
7.5	**CVE-2014-9426:** ** DISPUTED ** The apprentice_load function in libmagic/apprentice.c in the Fileinfo component in PHP through 5.6.4 attempts to perform a free operation on a stack-based character array, which allows remote attackers to cause a denial of service (memory corruption or application crash) or possibly have unspecified other impact via unknown vectors. NOTE: this is disputed by the vendor because the standard erealloc behavior makes the free operation unreachable.
7.5	**CVE-2014-9705:** Heap-based buffer overflow in the enchant_broker_request_dict function in ext/enchant/enchant.c in PHP before 5.4.38, 5.5.x before 5.5.22, and 5.6.x before 5.6.6 allows remote attackers to execute arbitrary code via vectors that trigger creation of multiple dictionaries.
7.5	**CVE-2014-9912:** The get_icu_disp_value_src_php function in ext/intl/locale/locale_methods.c in PHP before 5.3.29, 5.4.x before 5.4.30, and 5.5.x before 5.5.14 does not properly restrict calls to the ICU uresbund.cpp component, which allows remote attackers to cause a denial of service (buffer overflow) or possibly have unspecified other impact via a locale_get_display_name call with a long first argument.
7.5	**CVE-2015-0231:** Use-after-free vulnerability in the process_nested_data function in ext/standard/var_unserializer.re in PHP before 5.4.37, 5.5.x before 5.5.21, and 5.6.x before 5.6.5 allows remote attackers to execute arbitrary code via a crafted unserialize call that leverages improper handling of duplicate numerical keys within the serialized properties of an object. NOTE: this vulnerability exists because of an incomplete fix for CVE-2014-8142. CWE-416: Use After Free
6.8	**CVE-2015-0232:** The exif_process_unicode function in ext/exif/exif.c in PHP before 5.4.37, 5.5.x before 5.5.21, and 5.6.x before 5.6.5 allows remote attackers to execute arbitrary code or cause a denial of service (uninitialized pointer free and application crash) via crafted EXIF data in a JPEG image. CWE-824: Access of Uninitialized Pointer

STATE	PHP 5.5.9
7.5	**CVE-2015-0273:** Multiple use-after-free vulnerabilities in ext/date/php_date.c in PHP before 5.4.38, 5.5.x before 5.5.22, and 5.6.x before 5.6.6 allow remote attackers to execute arbitrary code via crafted serialized input containing a (1) R or (2) r type specifier in (a) DateTimeZone data handled by the php_date_timezone_initialize_from_hash function or (b) DateTime data handled by the php_date_initialize_from_hash function. CWE-416: Use After Free
5.0	**CVE-2015-1352:** The build_tablename function in pgsql.c in the PostgreSQL (aka pgsql) extension in PHP through 5.6.7 does not validate token extraction for table names, which allows remote attackers to cause a denial of service (NULL pointer dereference and application crash) via a crafted name. CWE-476: NULL Pointer Dereference
5.0	**CVE-2015-2348:** The move_uploaded_file implementation in ext/standard/basic_functions.c in PHP before 5.4.39, 5.5.x before 5.5.23, and 5.6.x before 5.6.7 truncates a pathname upon encountering a 00 character, which allows remote attackers to bypass intended extension restrictions and create files with unexpected names via a crafted second argument. NOTE: this vulnerability exists because of an incomplete fix for CVE-2006-7243.
5.8	**CVE-2015-2783:** ext/phar/phar.c in PHP before 5.4.40, 5.5.x before 5.5.24, and 5.6.x before 5.6.8 allows remote attackers to obtain sensitive information from process memory or cause a denial of service (buffer over-read and application crash) via a crafted length value in conjunction with crafted serialized data in a phar archive, related to the phar_parse_metadata and phar_parse_pharfile functions.
7.5	**CVE-2015-2787:** Use-after-free vulnerability in the process_nested_data function in ext/standard/var_unserializer.re in PHP before 5.4.39, 5.5.x before 5.5.23, and 5.6.x before 5.6.7 allows remote attackers to execute arbitrary code via a crafted unserialize call that leverages use of the unset function within an __wakeup function, a related issue to CVE-2015-0231. CWE-416: Use After Free
7.5	**CVE-2015-3307:** The phar_parse_metadata function in ext/phar/phar.c in PHP before 5.4.40, 5.5.x before 5.5.24, and 5.6.x before 5.6.8 allows remote attackers to cause a denial of service (heap metadata corruption) or possibly have unspecified other impact via a crafted tar archive.
7.5	**CVE-2015-3329:** Multiple stack-based buffer overflows in the phar_set_inode function in phar_internal.h in PHP before 5.4.40, 5.5.x before 5.5.24, and 5.6.x before 5.6.8 allow remote attackers to execute arbitrary code via a crafted length value in a (1) tar, (2) phar, or (3) ZIP archive.
6.8	**CVE-2015-3330:** The php_handler function in sapi/apache2handler/sapi_apache2.c in PHP before 5.4.40, 5.5.x before 5.5.24, and 5.6.x before 5.6.8, when the Apache HTTP Server 2.4.x is used, allows remote attackers to cause a denial of service (application crash) or possibly execute arbitrary code via pipelined HTTP requests that result in a "deconfigured interpreter."
6.4	**CVE-2015-3411:** PHP before 5.4.40, 5.5.x before 5.5.24, and 5.6.x before 5.6.8 does not ensure that pathnames lack %00 sequences, which might allow remote attackers to read or write to arbitrary files via crafted input to an application that calls (1) a DOMDocument load method, (2) the xmlwriter_open_uri function, (3) the finfo_file function, or (4) the hash_hmac_file function, as demonstrated by a filename.xml attack that bypasses an intended configuration in which client users may read only .xml files.

STATE	PHP 5.5.9
5.0	**CVE-2015-3412:** PHP before 5.4.40, 5.5.x before 5.5.24, and 5.6.x before 5.6.8 does not ensure that pathnames lack %00 sequences, which might allow remote attackers to read arbitrary files via crafted input to an application that calls the stream_resolve_include_path function in ext/standard/streamsfuncs.c, as demonstrated by a filename.extension attack that bypasses an intended configuration in which client users may read files with only one specific extension.
5.0	**CVE-2015-4021:** The phar_parse_tarfile function in ext/phar/tar.c in PHP before 5.4.41, 5.5.x before 5.5.25, and 5.6.x before 5.6.9 does not verify that the first character of a filename is different from the character, which allows remote attackers to cause a denial of service (integer underflow and memory corruption) via a crafted entry in a tar archive.
7.5	**CVE-2015-4022:** Integer overflow in the ftp_genlist function in ext/ftp/ftp.c in PHP before 5.4.41, 5.5.x before 5.5.25, and 5.6.x before 5.6.9 allows remote FTP servers to execute arbitrary code via a long reply to a LIST command, leading to a heap-based buffer overflow.
7.5	**CVE-2015-4025:** PHP before 5.4.41, 5.5.x before 5.5.25, and 5.6.x before 5.6.9 truncates a pathname upon encountering a 00 character in certain situations, which allows remote attackers to bypass intended extension restrictions and access files or directories with unexpected names via a crafted argument to (1) set_include_path, (2) tempnam, (3) rmdir, or (4) readlink. NOTE: this vulnerability exists because of an incomplete fix for CVE-2006-7243.
7.5	**CVE-2015-4026:** The pcntl_exec implementation in PHP before 5.4.41, 5.5.x before 5.5.25, and 5.6.x before 5.6.9 truncates a pathname upon encountering a 00 character, which might allow remote attackers to bypass intended extension restrictions and execute files with unexpected names via a crafted first argument. NOTE: this vulnerability exists because of an incomplete fix for CVE-2006-7243.
7.5	**CVE-2015-4116:** Use-after-free vulnerability in the spl_ptr_heap_insert function in ext/spl/spl_heap.c in PHP before 5.5.27 and 5.6.x before 5.6.11 allows remote attackers to execute arbitrary code by triggering a failed SplMinHeap::compare operation. CWE 416: Use After Free
7.5	**CVE-2015-4147:** The SoapClient::__call method in ext/soap/soap.c in PHP before 5.4.39, 5.5.x before 5.5.23, and 5.6.x before 5.6.7 does not verify that __default_headers is an array, which allows remote attackers to execute arbitrary code by providing crafted serialized data with an unexpected data type, related to a "type confusion" issue.
5.0	**CVE-2015-4148:** The do_soap_call function in ext/soap/soap.c in PHP before 5.4.39, 5.5.x before 5.5.23, and 5.6.x before 5.6.7 does not verify that the uri property is a string, which allows remote attackers to obtain sensitive information by providing crafted serialized data with an int data type, related to a "type confusion" issue.
7.5	**CVE-2015-4598:** PHP before 5.4.42, 5.5.x before 5.5.26, and 5.6.x before 5.6.10 does not ensure that pathnames lack %00 sequences, which might allow remote attackers to read or write to arbitrary files via crafted input to an application that calls (1) a DOMDocument save method or (2) the GD imagepsloadfont function, as demonstrated by a filename.html attack that bypasses an intended configuration in which client users may write to only .html files.

STATE	PHP 5.5.9
10.0	**CVE-2015-4599:** The SoapFault::__toString method in ext/soap/soap.c in PHP before 5.4.40, 5.5.x before 5.5.24, and 5.6.x before 5.6.8 allows remote attackers to obtain sensitive information, cause a denial of service (application crash), or possibly execute arbitrary code via an unexpected data type, related to a "type confusion" issue. Access of Resource Using Incompatible Type ('Type Confusion')
10.0	**CVE-2015-4600:** The SoapClient implementation in PHP before 5.4.40, 5.5.x before 5.5.24, and 5.6.x before 5.6.8 allows remote attackers to cause a denial of service (application crash) or possibly execute arbitrary code via an unexpected data type, related to "type confusion" issues in the (1) SoapClient::__getLastRequest, (2) SoapClient::__getLastResponse, (3) SoapClient::__getLastRequestHeaders, (4) SoapClient::__getLastResponseHeaders, (5) SoapClient::__getCookies, and (6) SoapClient::__setCookie methods. Access of Resource Using Incompatible Type ('Type Confusion')
10.0	**CVE-2015-4601:** PHP before 5.6.7 might allow remote attackers to cause a denial of service (application crash) or possibly execute arbitrary code via an unexpected data type, related to "type confusion" issues in (1) ext/soap/php_encoding.c, (2) ext/soap/php_http.c, and (3) ext/soap/soap.c, a different issue than CVE-2015-4600. Access of Resource Using Incompatible Type ('Type Confusion')
10.0	**CVE-2015-4602:** The __PHP_Incomplete_Class function in ext/standard/incomplete_class.c in PHP before 5.4.40, 5.5.x before 5.5.24, and 5.6.x before 5.6.8 allows remote attackers to cause a denial of service (application crash) or possibly execute arbitrary code via an unexpected data type, related to a "type confusion" issue. Access of Resource Using Incompatible Type ('Type Confusion')
10.0	**CVE-2015-4603:** The exception::getTraceAsString function in Zend/zend_exceptions.c in PHP before 5.4.40, 5.5.x before 5.5.24, and 5.6.x before 5.6.8 allows remote attackers to execute arbitrary code via an unexpected data type, related to a "type confusion" issue. Access of Resource Using Incompatible Type ('Type Confusion')
5.0	**CVE-2015-4604:** The mget function in softmagic.c in file 5.x, as used in the Fileinfo component in PHP before 5.4.40, 5.5.x before 5.5.24, and 5.6.x before 5.6.8, does not properly maintain a certain pointer relationship, which allows remote attackers to cause a denial of service (application crash) or possibly execute arbitrary code via a crafted string that is mishandled by a "Python script text executable" rule.
5.0	**CVE-2015-4605:** The mcopy function in softmagic.c in file 5.x, as used in the Fileinfo component in PHP before 5.4.40, 5.5.x before 5.5.24, and 5.6.x before 5.6.8, does not properly restrict a certain offset value, which allows remote attackers to cause a denial of service (application crash) or possibly execute arbitrary code via a crafted string that is mishandled by a "Python script text executable" rule.
10.0	**CVE-2015-4642:** The escapeshellarg function in ext/standard/exec.c in PHP before 5.4.42, 5.5.x before 5.5.26, and 5.6.x before 5.6.10 on Windows allows remote attackers to execute arbitrary OS commands via a crafted string to an application that accepts command-line arguments for a call to the PHP system function.

STATE	PHP 5.5.9
7.5	**CVE-2015-4643:** Integer overflow in the ftp_genlist function in ext/ftp/ftp.c in PHP before 5.4.42, 5.5.x before 5.5.26, and 5.6.x before 5.6.10 allows remote FTP servers to execute arbitrary code via a long reply to a LIST command, leading to a heap-based buffer overflow. NOTE: this vulnerability exists because of an incomplete fix for CVE-2015-4022.
5.0	**CVE-2015-4644:** The php_pgsql_meta_data function in pgsql.c in the PostgreSQL (aka pgsql) extension in PHP before 5.4.42, 5.5.x before 5.5.26, and 5.6.x before 5.6.10 does not validate token extraction for table names, which might allow remote attackers to cause a denial of service (NULL pointer dereference and application crash) via a crafted name. NOTE: this vulnerability exists because of an incomplete fix for CVE-2015-1352. CWE-476: NULL Pointer Dereference
10.0	**CVE-2015-5589:** The phar_convert_to_other function in ext/phar/phar_object.c in PHP before 5.4.43, 5.5.x before 5.5.27, and 5.6.x before 5.6.11 does not validate a file pointer before a close operation, which allows remote attackers to cause a denial of service (segmentation fault) or possibly have unspecified other impact via a crafted TAR archive that is mishandled in a Phar::convertToData call.
7.5	**CVE-2015-5590:** Stack-based buffer overflow in the phar_fix_filepath function in ext/phar/phar.c in PHP before 5.4.43, 5.5.x before 5.5.27, and 5.6.x before 5.6.11 allows remote attackers to cause a denial of service or possibly have unspecified other impact via a large length value, as demonstrated by mishandling of an e-mail attachment by the imap PHP extension.
7.5	**CVE-2015-6832:** Use-after-free vulnerability in the SPL unserialize implementation in ext/spl/spl_array.c in PHP before 5.4.44, 5.5.x before 5.5.28, and 5.6.x before 5.6.12 allows remote attackers to execute arbitrary code via crafted serialized data that triggers misuse of an array field. CWE-416: Use After Free
5.0	**CVE-2015-6833:** Directory traversal vulnerability in the PharData class in PHP before 5.4.44, 5.5.x before 5.5.28, and 5.6.x before 5.6.12 allows remote attackers to write to arbitrary files via a .. (dot dot) in a ZIP archive entry that is mishandled during an extractTo call.
7.5	**CVE-2015-6834:** Multiple use-after-free vulnerabilities in PHP before 5.4.45, 5.5.x before 5.5.29, and 5.6.x before 5.6.13 allow remote attackers to execute arbitrary code via vectors related to (1) the Serializable interface, (2) the SplObjectStorage class, and (3) the SplDoublyLinkedList class, which are mishandled during unserialization. CWE-502: Deserialization of Untrusted Data
7.5	**CVE-2015-6835:** The session deserializer in PHP before 5.4.45, 5.5.x before 5.5.29, and 5.6.x before 5.6.13 mishandles multiple php_var_unserialize calls, which allow remote attackers to execute arbitrary code or cause a denial of service (use-after-free) via crafted session content. CWE-416: Use After Free
7.5	**CVE-2015-6836:** The SoapClient __call method in ext/soap/soap.c in PHP before 5.4.45, 5.5.x before 5.5.29, and 5.6.x before 5.6.13 does not properly manage headers, which allows remote attackers to execute arbitrary code via crafted serialized data that triggers a "type confusion" in the serialize_function_call function. CWE-843: Access of Resource Using Incompatible Type ('Type Confusion')

STATE	PHP 5.5.9
5.0	**CVE-2015-6837:** The xsl_ext_function_php function in ext/xsl/xsltprocessor.c in PHP before 5.4.45, 5.5.x before 5.5.29, and 5.6.x before 5.6.13, when libxml2 before 2.9.2 is used, does not consider the possibility of a NULL valuePop return value before proceeding with a free operation during initial error checking, which allows remote attackers to cause a denial of service (NULL pointer dereference and application crash) via a crafted XML document, a different vulnerability than CVE-2015-6838. CWE-476: NULL Pointer Dereference
5.0	**CVE-2015-6838:** The xsl_ext_function_php function in ext/xsl/xsltprocessor.c in PHP before 5.4.45, 5.5.x before 5.5.29, and 5.6.x before 5.6.13, when libxml2 before 2.9.2 is used, does not consider the possibility of a NULL valuePop return value before proceeding with a free operation after the principal argument loop, which allows remote attackers to cause a denial of service (NULL pointer dereference and application crash) via a crafted XML document, a different vulnerability than CVE-2015-6837. CWE-476: NULL Pointer Dereference
6.8	**CVE-2015-7803:** The phar_get_entry_data function in ext/phar/util.c in PHP before 5.5.30 and 5.6.x before 5.6.14 allows remote attackers to cause a denial of service (NULL pointer dereference and application crash) via a .phar file with a crafted TAR archive entry in which the Link indicator references a file that does not exist. CWE-476: NULL Pointer Dereference <a href="http://lists.apple.com/archives/security-announce/2015/Dec/msg00005.html">Per Advisory: The attack can lead to remote code execution
6.8	**CVE-2015-7804:** Off-by-one error in the phar_parse_zipfile function in ext/phar/zip.c in PHP before 5.5.30 and 5.6.x before 5.6.14 allows remote attackers to cause a denial of service (uninitialized pointer dereference and application crash) by including the / filename in a .zip PHAR archive. <a href="http://lists.apple.com/archives/security-announce/2015/Dec/msg00005.html">Per Advisory: The attack can lead to remote code execution
7.5	**CVE-2015-8835:** The make_http_soap_request function in ext/soap/php_http.c in PHP before 5.4.44, 5.5.x before 5.5.28, and 5.6.x before 5.6.12 does not properly retrieve keys, which allows remote attackers to cause a denial of service (NULL pointer dereference, type confusion, and application crash) or possibly execute arbitrary code via crafted serialized data representing a numerically indexed _cookies array, related to the SoapClient::__call method in ext/soap/soap.c. CWE-476: NULL Pointer Dereference
4.3	**CVE-2015-8838:** ext/mysqlnd/mysqlnd.c in PHP before 5.4.43, 5.5.x before 5.5.27, and 5.6.x before 5.6.11 uses a client SSL option to mean that SSL is optional, which allows man-in-the-middle attackers to spoof servers via a cleartext-downgrade attack, a related issue to CVE-2015-3152.
7.5	**CVE-2015-8865:** The file_check_mem function in funcs.c in file before 5.23, as used in the Fileinfo component in PHP before 5.5.34, 5.6.x before 5.6.20, and 7.x before 7.0.5, mishandles continuation-level jumps, which allows context-dependent attackers to cause a denial of service (buffer overflow and application crash) or possibly execute arbitrary code via a crafted magic file.
5.0	**CVE-2015-8873:** Stack consumption vulnerability in Zend/zend_exceptions.c in PHP before 5.4.44, 5.5.x before 5.5.28, and 5.6.x before 5.6.12 allows remote attackers to cause a denial of service (segmentation fault) via recursive method calls.

STATE	PHP 5.5.9
5.0	**CVE-2015-8874:** Stack consumption vulnerability in GD in PHP before 5.6.12 allows remote attackers to cause a denial of service via a crafted imagefilltoborder call.
5.0	**CVE-2015-8879:** The odbc_bindcols function in ext/odbc/php_odbc.c in PHP before 5.6.12 mishandles driver behavior for SQL_WVARCHAR columns, which allows remote attackers to cause a denial of service (application crash) in opportunistic circumstances by leveraging use of the odbc_fetch_array function to access a certain type of Microsoft SQL Server table.
10.0	**CVE-2015-8880:** Double free vulnerability in the format printer in PHP 7.x before 7.0.1 allows remote attackers to have an unspecified impact by triggering an error. CWE-415: Double Free
4.3	**CVE-2015-8935:** The sapi_header_op function in main/SAPI.c in PHP before 5.4.38, 5.5.x before 5.5.22, and 5.6.x before 5.6.6 supports deprecated line folding without considering browser compatibility, which allows remote attackers to conduct cross-site scripting (XSS) attacks against Internet Explorer by leveraging (1) %0A%20 or (2) %0D%0A%20 mishandling in the header function.
6.8	**CVE-2015-9253:** An issue was discovered in PHP 7.3.x before 7.3.0alpha3, 7.2.x before 7.2.8, and before 7.1.20. The php-fpm master process restarts a child process in an endless loop when using program execution functions (e.g., passthru, exec, shell_exec, or system) with a non-blocking STDIN stream, causing this master process to consume 100% of the CPU, and consume disk space with a large volume of error logs, as demonstrated by an attack by a customer of a shared-hosting facility.
5.0	**CVE-2016-10158:** The exif_convert_any_to_int function in ext/exif/exif.c in PHP before 5.6.30, 7.0.x before 7.0.15, and 7.1.x before 7.1.1 allows remote attackers to cause a denial of service (application crash) via crafted EXIF data that triggers an attempt to divide the minimum representable negative integer by -1.
5.0	**CVE-2016-10159:** Integer overflow in the phar_parse_pharfile function in ext/phar/phar.c in PHP before 5.6.30 and 7.0.x before 7.0.15 allows remote attackers to cause a denial of service (memory consumption or application crash) via a truncated manifest entry in a PHAR archive.
7.5	**CVE-2016-10160:** Off-by-one error in the phar_parse_pharfile function in ext/phar/phar.c in PHP before 5.6.30 and 7.0.x before 7.0.15 allows remote attackers to cause a denial of service (memory corruption) or possibly execute arbitrary code via a crafted PHAR archive with an alias mismatch.
5.0	**CVE-2016-10161:** The object_common1 function in ext/standard/var_unserializer.c in PHP before 5.6.30, 7.0.x before 7.0.15, and 7.1.x before 7.1.1 allows remote attackers to cause a denial of service (buffer over-read and application crash) via crafted serialized data that is mishandled in a finish_nested_data call.
5.0	**CVE-2016-10397:** In PHP before 5.6.28 and 7.x before 7.0.13, incorrect handling of various URI components in the URL parser could be used by attackers to bypass hostname-specific URL checks, as demonstrated by evil.example.com:80#@good.example.com/ and evil.example.com:80?@good.example.com/ inputs to the parse_url function (implemented in the php_url_parse_ex function in ext/standard/url.c).

STATE	PHP 5.5.9
5.0	**CVE-2016-10712:** In PHP before 5.5.32, 5.6.x before 5.6.18, and 7.x before 7.0.3, all of the return values of stream_get_meta_data can be controlled if the input can be controlled (e.g., during file uploads). For example, a "$uri = stream_get_meta_data(fopen($file, "r"))['uri']" call mishandles the case where $file is data:text/plain;uri=eviluri, – in other words, metadata can be set by an attacker.
6.4	**CVE-2016-1903:** The gdImageRotateInterpolated function in ext/gd/libgd/gd_interpolation.c in PHP before 5.5.31, 5.6.x before 5.6.17, and 7.x before 7.0.2 allows remote attackers to obtain sensitive information or cause a denial of service (out-of-bounds read and application crash) via a large bgd_color argument to the imagerotate function.
10.0	**CVE-2016-2554:** Stack-based buffer overflow in ext/phar/tar.c in PHP before 5.5.32, 5.6.x before 5.6.18, and 7.x before 7.0.3 allows remote attackers to cause a denial of service (application crash) or possibly have unspecified other impact via a crafted TAR archive.
7.5	**CVE-2016-3078:** Multiple integer overflows in php_zip.c in the zip extension in PHP before 7.0.6 allow remote attackers to cause a denial of service (heap-based buffer overflow and application crash) or possibly have unspecified other impact via a crafted call to (1) getFromIndex or (2) getFromName in the ZipArchive class.
7.5	**CVE-2016-3141:** Use-after-free vulnerability in wddx.c in the WDDX extension in PHP before 5.5.33 and 5.6.x before 5.6.19 allows remote attackers to cause a denial of service (memory corruption and application crash) or possibly have unspecified other impact by triggering a wddx_deserialize call on XML data containing a crafted var element.
6.4	**CVE-2016-3142:** The phar_parse_zipfile function in zip.c in the PHAR extension in PHP before 5.5.33 and 5.6.x before 5.6.19 allows remote attackers to obtain sensitive information from process memory or cause a denial of service (out-of-bounds read and application crash) by placing a PK0506 signature at an invalid location.
6.4	**CVE-2016-3185:** The make_http_soap_request function in ext/soap/php_http.c in PHP before 5.4.44, 5.5.x before 5.5.28, 5.6.x before 5.6.12, and 7.x before 7.0.4 allows remote attackers to obtain sensitive information from process memory or cause a denial of service (type confusion and application crash) via crafted serialized _cookies data, related to the SoapClient::__call method in ext/soap/soap.c.
5.0	**CVE-2016-4070:** ** DISPUTED ** Integer overflow in the php_raw_url_encode function in ext/standard/url.c in PHP before 5.5.34, 5.6.x before 5.6.20, and 7.x before 7.0.5 allows remote attackers to cause a denial of service (application crash) via a long string to the rawurlencode function. NOTE: the vendor says "Not sure if this qualifies as security issue (probably not)."
8.3	**CVE-2016-4342:** ext/phar/phar_object.c in PHP before 5.5.32, 5.6.x before 5.6.18, and 7.x before 7.0.3 mishandles zero-length uncompressed data, which allows remote attackers to cause a denial of service (heap memory corruption) or possibly have unspecified other impact via a crafted (1) TAR, (2) ZIP, or (3) PHAR archive.
6.8	**CVE-2016-4343:** The phar_make_dirstream function in ext/phar/dirstream.c in PHP before 5.6.18 and 7.x before 7.0.3 mishandles zero-size ./.@LongLink files, which allows remote attackers to cause a denial of service (uninitialized pointer dereference) or possibly have unspecified other impact via a crafted TAR archive. CWE-824: Access of Uninitialized Pointer

STATE	PHP 5.5.9
7.5	**CVE-2016-4344:** Integer overflow in the xml_utf8_encode function in ext/xml/xml.c in PHP before 7.0.4 allows remote attackers to cause a denial of service or possibly have unspecified other impact via a long argument to the utf8_encode function, leading to a heap-based buffer overflow.
7.5	**CVE-2016-4345:** Integer overflow in the php_filter_encode_url function in ext/filter/sanitizing_filters.c in PHP before 7.0.4 allows remote attackers to cause a denial of service or possibly have unspecified other impact via a long string, leading to a heap-based buffer overflow.
7.5	**CVE-2016-4346:** Integer overflow in the str_pad function in ext/standard/string.c in PHP before 7.0.4 allows remote attackers to cause a denial of service or possibly have unspecified other impact via a long string, leading to a heap-based buffer overflow.
7.5	**CVE-2016-4537:** The bcpowmod function in ext/bcmath/bcmath.c in PHP before 5.5.35, 5.6.x before 5.6.21, and 7.x before 7.0.6 accepts a negative integer for the scale argument, which allows remote attackers to cause a denial of service or possibly have unspecified other impact via a crafted call.
7.5	**CVE-2016-4538:** The bcpowmod function in ext/bcmath/bcmath.c in PHP before 5.5.35, 5.6.x before 5.6.21, and 7.x before 7.0.6 modifies certain data structures without considering whether they are copies of the _zero_, _one_, or _two_ global variable, which allows remote attackers to cause a denial of service or possibly have unspecified other impact via a crafted call.
7.5	**CVE-2016-4539:** The xml_parse_into_struct function in ext/xml/xml.c in PHP before 5.5.35, 5.6.x before 5.6.21, and 7.x before 7.0.6 allows remote attackers to cause a denial of service (buffer under-read and segmentation fault) or possibly have unspecified other impact via crafted XML data in the second argument, leading to a parser level of zero.
7.5	**CVE-2016-4540:** The grapheme_stripos function in ext/intl/grapheme/grapheme_string.c in PHP before 5.5.35, 5.6.x before 5.6.21, and 7.x before 7.0.6 allows remote attackers to cause a denial of service (out-of-bounds read) or possibly have unspecified other impact via a negative offset. CWE-125: Out-of-bounds Read
7.5	**CVE-2016-4541:** The grapheme_strpos function in ext/intl/grapheme/grapheme_string.c in PHP before 5.5.35, 5.6.x before 5.6.21, and 7.x before 7.0.6 allows remote attackers to cause a denial of service (out-of-bounds read) or possibly have unspecified other impact via a negative offset. CWE-125: Out-of-bounds Read
7.5	**CVE-2016-4542:** The exif_process_IFD_TAG function in ext/exif/exif.c in PHP before 5.5.35, 5.6.x before 5.6.21, and 7.x before 7.0.6 does not properly construct spprintf arguments, which allows remote attackers to cause a denial of service (out-of-bounds read) or possibly have unspecified other impact via crafted header data.
7.5	**CVE-2016-5093:** The get_icu_value_internal function in ext/intl/locale/locale_methods.c in PHP before 5.5.36, 5.6.x before 5.6.22, and 7.x before 7.0.7 does not ensure the presence of a " character, which allows remote attackers to cause a denial of service (out-of-bounds read) or possibly have unspecified other impact via a crafted locale_get_primary_language call.

STATE	PHP 5.5.9
7.5	**CVE-2016-5094:** Integer overflow in the php_html_entities function in ext/standard/html.c in PHP before 5.5.36 and 5.6.x before 5.6.22 allows remote attackers to cause a denial of service or possibly have unspecified other impact by triggering a large output string from the htmlspecialchars function.
7.5	**CVE-2016-5095:** Integer overflow in the php_escape_html_entities_ex function in ext/standard/html.c in PHP before 5.5.36 and 5.6.x before 5.6.22 allows remote attackers to cause a denial of service or possibly have unspecified other impact by triggering a large output string from a FILTER_SANITIZE_FULL_SPECIAL_CHARS filter_var call. NOTE: this vulnerability exists because of an incomplete fix for CVE-2016-5094.
7.5	**CVE-2016-5096:** Integer overflow in the fread function in ext/standard/file.c in PHP before 5.5.36 and 5.6.x before 5.6.22 allows remote attackers to cause a denial of service or possibly have unspecified other impact via a large integer in the second argument.
6.4	**CVE-2016-5114:** sapi/fpm/fpm/fpm_log.c in PHP before 5.5.31, 5.6.x before 5.6.17, and 7.x before 7.0.2 misinterprets the semantics of the snprintf return value, which allows attackers to obtain sensitive information from process memory or cause a denial of service (out-of-bounds read and buffer overflow) via a long string, as demonstrated by a long URI in a configuration with custom REQUEST_URI logging.
6.8	**CVE-2016-5399:** The bzread function in ext/bz2/bz2.c in PHP before 5.5.38, 5.6.x before 5.6.24, and 7.x before 7.0.9 allows remote attackers to cause a denial of service (out-of-bounds write) or execute arbitrary code via a crafted bz2 archive.
7.5	**CVE-2016-5768:** Double free vulnerability in the _php_mb_regex_ereg_replace_exec function in php_mbregex.c in the mbstring extension in PHP before 5.5.37, 5.6.x before 5.6.23, and 7.x before 7.0.8 allows remote attackers to execute arbitrary code or cause a denial of service (application crash) by leveraging a callback exception.
7.5	**CVE-2016-5769:** Multiple integer overflows in mcrypt.c in the mcrypt extension in PHP before 5.5.37, 5.6.x before 5.6.23, and 7.x before 7.0.8 allow remote attackers to cause a denial of service (heap-based buffer overflow and application crash) or possibly have unspecified other impact via a crafted length value, related to the (1) mcrypt_generic and (2) mdecrypt_generic functions.
7.5	**CVE-2016-5770:** Integer overflow in the SplFileObject::fread function in spl_directory.c in the SPL extension in PHP before 5.5.37 and 5.6.x before 5.6.23 allows remote attackers to cause a denial of service or possibly have unspecified other impact via a large integer argument, a related issue to CVE-2016-5096.
7.5	**CVE-2016-5771:** spl_array.c in the SPL extension in PHP before 5.5.37 and 5.6.x before 5.6.23 improperly interacts with the unserialize implementation and garbage collection, which allows remote attackers to execute arbitrary code or cause a denial of service (use-after-free and application crash) via crafted serialized data.
7.5	**CVE-2016-5773:** php_zip.c in the zip extension in PHP before 5.5.37, 5.6.x before 5.6.23, and 7.x before 7.0.8 improperly interacts with the unserialize implementation and garbage collection, which allows remote attackers to execute arbitrary code or cause a denial of service (use-after-free and application crash) via crafted serialized data containing a ZipArchive object.

STATE	PHP 5.5.9
7.5	**CVE-2016-6288:** The php_url_parse_ex function in ext/standard/url.c in PHP before 5.5.38 allows remote attackers to cause a denial of service (buffer over-read) or possibly have unspecified other impact via vectors involving the smart_str data type.
6.8	**CVE-2016-6289:** Integer overflow in the virtual_file_ex function in TSRM/tsrm_virtual_cwd.c in PHP before 5.5.38, 5.6.x before 5.6.24, and 7.x before 7.0.9 allows remote attackers to cause a denial of service (stack-based buffer overflow) or possibly have unspecified other impact via a crafted extract operation on a ZIP archive.
7.5	**CVE-2016-6290:** ext/session/session.c in PHP before 5.5.38, 5.6.x before 5.6.24, and 7.x before 7.0.9 does not properly maintain a certain hash data structure, which allows remote attackers to cause a denial of service (use-after-free) or possibly have unspecified other impact via vectors related to session deserialization.
7.5	**CVE-2016-6291:** The exif_process_IFD_in_MAKERNOTE function in ext/exif/exif.c in PHP before 5.5.38, 5.6.x before 5.6.24, and 7.x before 7.0.9 allows remote attackers to cause a denial of service (out-of-bounds array access and memory corruption), obtain sensitive information from process memory, or possibly have unspecified other impact via a crafted JPEG image.
4.3	**CVE-2016-6292:** The exif_process_user_comment function in ext/exif/exif.c in PHP before 5.5.38, 5.6.x before 5.6.24, and 7.x before 7.0.9 allows remote attackers to cause a denial of service (NULL pointer dereference and application crash) via a crafted JPEG image.
7.5	**CVE-2016-6294:** The locale_accept_from_http function in ext/intl/locale/locale_methods.c in PHP before 5.5.38, 5.6.x before 5.6.24, and 7.x before 7.0.9 does not properly restrict calls to the ICU uloc_acceptLanguageFromHTTP function, which allows remote attackers to cause a denial of service (out-of-bounds read) or possibly have unspecified other impact via a call with a long argument.
7.5	**CVE-2016-6295:** ext/snmp/snmp.c in PHP before 5.5.38, 5.6.x before 5.6.24, and 7.x before 7.0.9 improperly interacts with the unserialize implementation and garbage collection, which allows remote attackers to cause a denial of service (use-after-free and application crash) or possibly have unspecified other impact via crafted serialized data, a related issue to CVE-2016-5773.
7.5	**CVE-2016-6296:** Integer signedness error in the simplestring_addn function in simplestring.c in xmlrpc-epi through 0.54.2, as used in PHP before 5.5.38, 5.6.x before 5.6.24, and 7.x before 7.0.9, allows remote attackers to cause a denial of service (heap-based buffer overflow) or possibly have unspecified other impact via a long first argument to the PHP xmlrpc_encode_request function.
6.8	**CVE-2016-6297:** Integer overflow in the php_stream_zip_opener function in ext/zip/zip_stream.c in PHP before 5.5.38, 5.6.x before 5.6.24, and 7.x before 7.0.9 allows remote attackers to cause a denial of service (stack-based buffer overflow) or possibly have unspecified other impact via a crafted zip:// URL.
7.5	**CVE-2016-7124:** ext/standard/var_unserializer.c in PHP before 5.6.25 and 7.x before 7.0.10 mishandles certain invalid objects, which allows remote attackers to cause a denial of service or possibly have unspecified other impact via crafted serialized data that leads to a (1) __destruct call or (2) magic method call.

STATE	PHP 5.5.9
5.0	**CVE-2016-7125:** ext/session/session.c in PHP before 5.6.25 and 7.x before 7.0.10 skips invalid session names in a way that triggers incorrect parsing, which allows remote attackers to inject arbitrary-type session data by leveraging control of a session name, as demonstrated by object injection.
7.5	**CVE-2016-7126:** The imagetruecolortopalette function in ext/gd/gd.c in PHP before 5.6.25 and 7.x before 7.0.10 does not properly validate the number of colors, which allows remote attackers to cause a denial of service (select_colors allocation error and out-of-bounds write) or possibly have unspecified other impact via a large value in the third argument.
7.5	**CVE-2016-7127:** The imagegammacorrect function in ext/gd/gd.c in PHP before 5.6.25 and 7.x before 7.0.10 does not properly validate gamma values, which allows remote attackers to cause a denial of service (out-of-bounds write) or possibly have unspecified other impact by providing different signs for the second and third arguments.
5.0	**CVE-2016-7128:** The exif_process_IFD_in_TIFF function in ext/exif/exif.c in PHP before 5.6.25 and 7.x before 7.0.10 mishandles the case of a thumbnail offset that exceeds the file size, which allows remote attackers to obtain sensitive information from process memory via a crafted TIFF image.
7.5	**CVE-2016-7129:** The php_wddx_process_data function in ext/wddx/wddx.c in PHP before 5.6.25 and 7.x before 7.0.10 allows remote attackers to cause a denial of service (segmentation fault) or possibly have unspecified other impact via an invalid ISO 8601 time value, as demonstrated by a wddx_deserialize call that mishandles a dateTime element in a wddxPacket XML document.
5.0	**CVE-2016-7130:** The php_wddx_pop_element function in ext/wddx/wddx.c in PHP before 5.6.25 and 7.x before 7.0.10 allows remote attackers to cause a denial of service (NULL pointer dereference and application crash) or possibly have unspecified other impact via an invalid base64 binary value, as demonstrated by a wddx_deserialize call that mishandles a binary element in a wddxPacket XML document.
5.0	**CVE-2016-7131:** ext/wddx/wddx.c in PHP before 5.6.25 and 7.x before 7.0.10 allows remote attackers to cause a denial of service (NULL pointer dereference and application crash) or possibly have unspecified other impact via a malformed wddxPacket XML document that is mishandled in a wddx_deserialize call, as demonstrated by a tag that lacks a < (less than) character.
5.0	**CVE-2016-7132:** ext/wddx/wddx.c in PHP before 5.6.25 and 7.x before 7.0.10 allows remote attackers to cause a denial of service (NULL pointer dereference and application crash) or possibly have unspecified other impact via an invalid wddxPacket XML document that is mishandled in a wddx_deserialize call, as demonstrated by a stray element inside a boolean element, leading to incorrect pop processing.
7.5	**CVE-2016-7411:** ext/standard/var_unserializer.re in PHP before 5.6.26 mishandles object-deserialization failures, which allows remote attackers to cause a denial of service (memory corruption) or possibly have unspecified other impact via an unserialize call that references a partially constructed object.
6.8	**CVE-2016-7412:** ext/mysqlnd/mysqlnd_wireprotocol.c in PHP before 5.6.26 and 7.x before 7.0.11 does not verify that a BIT field has the UNSIGNED_FLAG flag, which allows remote MySQL servers to cause a denial of service (heap-based buffer overflow) or possibly have unspecified other impact via crafted field metadata.

STATE	PHP 5.5.9
7.5	**CVE-2016-7413:** Use-after-free vulnerability in the wddx_stack_destroy function in ext/wddx/wddx.c in PHP before 5.6.26 and 7.x before 7.0.11 allows remote attackers to cause a denial of service or possibly have unspecified other impact via a wddxPacket XML document that lacks an end-tag for a recordset field element, leading to mishandling in a wddx_deserialize call.
7.5	**CVE-2016-7414:** The ZIP signature-verification feature in PHP before 5.6.26 and 7.x before 7.0.11 does not ensure that the uncompressed_filesize field is large enough, which allows remote attackers to cause a denial of service (out-of-bounds memory access) or possibly have unspecified other impact via a crafted PHAR archive, related to ext/phar/util.c and ext/phar/zip.c.
5.0	**CVE-2016-7416:** ext/intl/msgformat/msgformat_format.c in PHP before 5.6.26 and 7.x before 7.0.11 does not properly restrict the locale length provided to the Locale class in the ICU library, which allows remote attackers to cause a denial of service (application crash) or possibly have unspecified other impact via a MessageFormatter::formatMessage call with a long first argument.
7.5	**CVE-2016-7417:** ext/spl/spl_array.c in PHP before 5.6.26 and 7.x before 7.0.11 proceeds with SplArray unserialization without validating a return value and data type, which allows remote attackers to cause a denial of service or possibly have unspecified other impact via crafted serialized data.
5.0	**CVE-2016-7418:** The php_wddx_push_element function in ext/wddx/wddx.c in PHP before 5.6.26 and 7.x before 7.0.11 allows remote attackers to cause a denial of service (invalid pointer access and out-of-bounds read) or possibly have unspecified other impact via an incorrect boolean element in a wddxPacket XML document, leading to mishandling in a wddx_deserialize call.
7.5	**CVE-2016-7480:** The SplObjectStorage unserialize implementation in ext/spl/spl_observer.c in PHP before 7.0.12 does not verify that a key is an object, which allows remote attackers to execute arbitrary code or cause a denial of service (uninitialized memory access) via crafted serialized data.
7.5	**CVE-2016-9137:** Use-after-free vulnerability in the CURLFile implementation in ext/curl/curl_file.c in PHP before 5.6.27 and 7.x before 7.0.12 allows remote attackers to cause a denial of service or possibly have unspecified other impact via crafted serialized data that is mishandled during __wakeup processing.
7.5	**CVE-2016-9138:** PHP through 5.6.27 and 7.x through 7.0.12 mishandles property modification during __wakeup processing, which allows remote attackers to cause a denial of service or possibly have unspecified other impact via crafted serialized data, as demonstrated by Exception::__toString with DateInterval::__wakeup.
5.0	**CVE-2016-9934:** ext/wddx/wddx.c in PHP before 5.6.28 and 7.x before 7.0.13 allows remote attackers to cause a denial of service (NULL pointer dereference) via crafted serialized data in a wddxPacket XML document, as demonstrated by a PDORow string.
7.5	**CVE-2016-9935:** The php_wddx_push_element function in ext/wddx/wddx.c in PHP before 5.6.29 and 7.x before 7.0.14 allows remote attackers to cause a denial of service (out-of-bounds read and memory corruption) or possibly have unspecified other impact via an empty boolean element in a wddxPacket XML document.

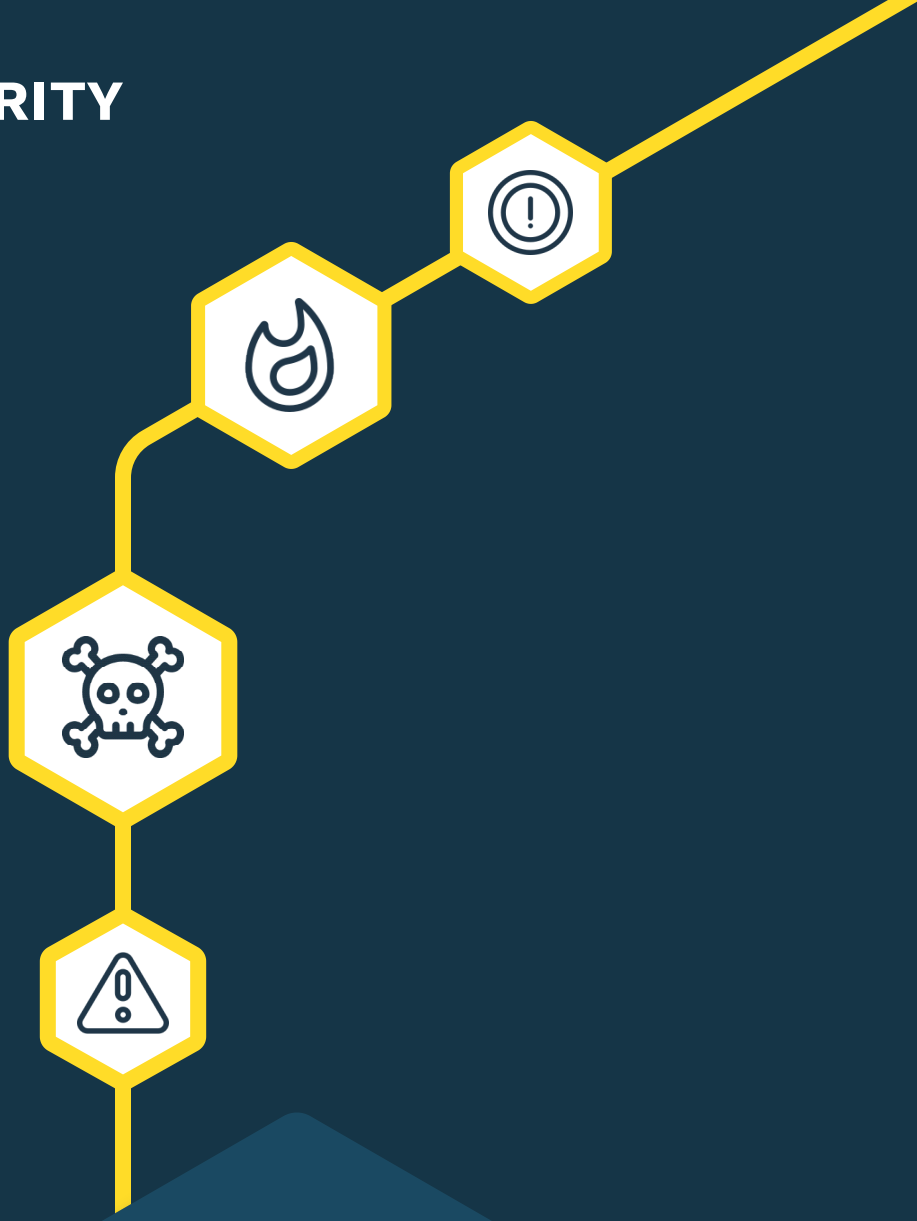
STATE	PHP 5.5.9
7.8	**CVE-2017-11142:** In PHP before 5.6.31, 7.x before 7.0.17, and 7.1.x before 7.1.3, remote attackers could cause a CPU consumption denial of service attack by injecting long form variables, related to main/php_variables.c.
5.0	**CVE-2017-11143:** In PHP before 5.6.31, an invalid free in the WDDX deserialization of boolean parameters could be used by attackers able to inject XML for deserialization to crash the PHP interpreter, related to an invalid free for an empty boolean element in ext/wddx/wddx.c.
5.0	**CVE-2017-11144:** In PHP before 5.6.31, 7.x before 7.0.21, and 7.1.x before 7.1.7, the openssl extension PEM sealing code did not check the return value of the OpenSSL sealing function, which could lead to a crash of the PHP interpreter, related to an interpretation conflict for a negative number in ext/openssl/openssl.c, and an OpenSSL documentation omission.
5.0	**CVE-2017-11145:** In PHP before 5.6.31, 7.x before 7.0.21, and 7.1.x before 7.1.7, an error in the date extension's timelib_meridian parsing code could be used by attackers able to supply date strings to leak information from the interpreter, related to ext/date/lib/parse_date.c out-of-bounds reads affecting the php_parse_date function. NOTE: the correct fix is in the e8b7698f5ee757ce2c8bd10a192a491a498f891c commit, not the bd77ac90d3bdf31ce2a5251ad92e9e75 gist.
6.4	**CVE-2017-11147:** In PHP before 5.6.30 and 7.x before 7.0.15, the PHAR archive handler could be used by attackers supplying malicious archive files to crash the PHP interpreter or potentially disclose information due to a buffer over-read in the phar_parse_pharfile function in ext/phar/phar.c.
6.8	**CVE-2017-11628:** In PHP before 5.6.31, 7.x before 7.0.21, and 7.1.x before 7.1.7, a stack-based buffer overflow in the zend_ini_do_op() function in Zend/zend_ini_parser.c could cause a denial of service or potentially allow executing code. NOTE: this is only relevant for PHP applications that accept untrusted input (instead of the system's php.ini file) for the parse_ini_string or parse_ini_file function, e.g., a web application for syntax validation of php.ini directives.
7.5	**CVE-2017-12933:** The finish_nested_data function in ext/standard/var_unserializer.re in PHP before 5.6.31, 7.0.x before 7.0.21, and 7.1.x before 7.1.7 is prone to a buffer over-read while unserializing untrusted data. Exploitation of this issue can have an unspecified impact on the integrity of PHP.
5.8	**CVE-2017-7272:** PHP through 7.1.11 enables potential SSRF in applications that accept an fsockopen or pfsockopen hostname argument with an expectation that the port number is constrained. Because a :port syntax is recognized, fsockopen will use the port number that is specified in the hostname argument, instead of the port number in the second argument of the function.
4.3	**CVE-2017-7890:** The GIF decoding function gdImageCreateFromGifCtx in gd_gif_in.c in the GD Graphics Library (aka libgd), as used in PHP before 5.6.31 and 7.x before 7.1.7, does not zero colorMap arrays before use. A specially crafted GIF image could use the uninitialized tables to read 700 bytes from the top of the stack, potentially disclosing sensitive information.

STATE	PHP 5.5.9
5.0	**CVE-2017-7963:** ** DISPUTED ** The GNU Multiple Precision Arithmetic Library (GMP) interfaces for PHP through 7.1.4 allow attackers to cause a denial of service (memory consumption and application crash) via operations on long strings. NOTE: the vendor disputes this, stating "There is no security issue here, because GMP safely aborts in case of an OOM condition. The only attack vector here is denial of service. However, if you allow attacker-controlled, unbounded allocations you have a DoS vector regardless of GMP's OOM behavior."
7.5	**CVE-2017-8923:** The zend_string_extend function in Zend/zend_string.h in PHP through 7.1.5 does not prevent changes to string objects that result in a negative length, which allows remote attackers to cause a denial of service (application crash) or possibly have unspecified other impact by leveraging a script's use of .= with a long string.
5.0	**CVE-2018-20783:** In PHP before 5.6.39, 7.x before 7.0.33, 7.1.x before 7.1.25, and 7.2.x before 7.2.13, a buffer over-read in PHAR reading functions may allow an attacker to read allocated or unallocated memory past the actual data when trying to parse a .phar file. This is related to phar_parse_pharfile in ext/phar/phar.c.
4.3	**CVE-2018-5711:** gd_gif_in.c in the GD Graphics Library (aka libgd), as used in PHP before 5.6.33, 7.0.x before 7.0.27, 7.1.x before 7.1.13, and 7.2.x before 7.2.1, has an integer signedness error that leads to an infinite loop via a crafted GIF file, as demonstrated by a call to the imagecreatefromgif or imagecreatefromstring PHP function. This is related to GetCode_ and gdImageCreateFromGifCtx.
4.3	**CVE-2018-5712:** An issue was discovered in PHP before 5.6.33, 7.0.x before 7.0.27, 7.1.x before 7.1.13, and 7.2.x before 7.2.1. There is Reflected XSS on the PHAR 404 error page via the URI of a request for a .phar file.
7.5	**CVE-2018-7584:** In PHP through 5.6.33, 7.0.x before 7.0.28, 7.1.x through 7.1.14, and 7.2.x through 7.2.2, there is a stack-based buffer under-read while parsing an HTTP response in the php_stream_url_wrap_http_ex function in ext/standard/http_fopen_wrapper.c. This subsequently results in copying a large string.
7.5	**CVE-2014-9427:** sapi/cgi/cgi_main.c in the CGI component in PHP through 5.4.36, 5.5.x through 5.5.20, and 5.6.x through 5.6.4, when mmap is used to read a .php file, does not properly consider the mapping's length during processing of an invalid file that begins with a # character and lacks a newline character, which causes an out-of-bounds read and might (1) allow remote attackers to obtain sensitive information from php-cgi process memory by leveraging the ability to upload a .php file or (2) trigger unexpected code execution if a valid PHP script is present in memory locations adjacent to the mapping.

STATE	PHP 5.5.9
6.8	**CVE-2015-8994:** An issue was discovered in PHP 5.x and 7.x, when the configuration uses apache2handler/mod_php or php-fpm with OpCache enabled. With 5.x after 5.6.28 or 7.x after 7.0.13, the issue is resolved in a non-default configuration with the opcache.validate_permission=1 setting. The vulnerability details are as follows. In PHP SAPIs where PHP interpreters share a common parent process, Zend OpCache creates a shared memory object owned by the common parent during initialization. Child PHP processes inherit the SHM descriptor, using it to cache and retrieve compiled script bytecode ("opcode" in PHP jargon). Cache keys vary depending on configuration, but filename is a central key component, and compiled opcode can generally be run if a script's filename is known or can be guessed. Many common shared-hosting configurations change EUID in child processes to enforce privilege separation among hosted users (for example using mod_ruid2 for the Apache HTTP Server, or php-fpm user settings). In these scenarios, the default Zend OpCache behavior defeats script file permissions by sharing a single SHM cache among all child PHP processes. PHP scripts often contain sensitive information: Think of CMS configurations where reading or running another user's script usually means gaining privileges to the CMS database.
5.0	**CVE-2016-7478:** Zend/zend_exceptions.c in PHP, possibly 5.x before 5.6.28 and 7.x before 7.0.13, allows remote attackers to cause a denial of service (infinite loop) via a crafted Exception object in serialized data, a related issue to CVE-2015-8876. CWE-835: Loop with Unreachable Exit Condition ('Infinite Loop')
5.0	**CVE-2018-19395:** ext/standard/var.c in PHP 5.x through 7.1.24 on Windows allows attackers to cause a denial of service (NULL pointer dereference and application crash) because com and com_safearray_proxy return NULL in com_properties_get in ext/com_dotnet/com_handlers.c, as demonstrated by a serialize call on COM("WScript.Shell").
5.0	**CVE-2018-19396:** ext/standard/var_unserializer.c in PHP 5.x through 7.1.24 allows attackers to cause a denial of service (application crash) via an unserialize call for the com, dotnet, or variant class.
6.5	**CVE-2018-19520:** An issue was discovered in SDCMS 1.6 with PHP 5.x. app/admin/controller/themecontroller.php uses a check_bad function in an attempt to block certain PHP functions such as eval, but does not prevent use of preg_replace 'e' calls, allowing users to execute arbitrary code by leveraging access to admin template management.
5.0	**CVE-2018-19935:** ext/imap/php_imap.c in PHP 5.x and 7.x before 7.3.0 allows remote attackers to cause a denial of service (NULL pointer dereference and application crash) via an empty string in the message argument to the imap_mail function.
6.4	**CVE-2014-5120:** gd_ctx.c in the GD component in PHP 5.4.x before 5.4.32 and 5.5.x before 5.5.16 does not ensure that pathnames lack %00 sequences, which might allow remote attackers to overwrite arbitrary files via crafted input to an application that calls the (1) imagegd, (2) imagegd2, (3) imagegif, (4) imagejpeg, (5) imagepng, (6) imagewbmp, or (7) imagewebp function.
5.0	**CVE-2015-8867:** The openssl_random_pseudo_bytes function in ext/openssl/openssl.c in PHP before 5.4.44, 5.5.x before 5.5.28, and 5.6.x before 5.6.12 incorrectly relies on the deprecated RAND_pseudo_bytes function, which makes it easier for remote attackers to defeat cryptographic protection mechanisms via unspecified vectors.

STATE	PHP 5.5.9
7.5	**CVE-2015-8876:** Zend/zend_exceptions.c in PHP before 5.4.44, 5.5.x before 5.5.28, and 5.6.x before 5.6.12 does not validate certain Exception objects, which allows remote attackers to cause a denial of service (NULL pointer dereference and application crash) or trigger unintended method execution via crafted serialized data. CWE-476: NULL Pointer Dereference
7.5	**CVE-2015-6831:** Multiple use-after-free vulnerabilities in SPL in PHP before 5.4.44, 5.5.x before 5.5.28, and 5.6.x before 5.6.12 allow remote attackers to execute arbitrary code via vectors involving (1) ArrayObject, (2) SplObjectStorage, and (3) SplDoublyLinkedList, which are mishandled during unserialization. CWE-416: Use After Free
7.5	**CVE-2015-2301:** Use-after-free vulnerability in the phar_rename_archive function in phar_object.c in PHP before 5.5.22 and 5.6.x before 5.6.6 allows remote attackers to cause a denial of service or possibly have unspecified other impact via vectors that trigger an attempted renaming of a Phar archive to the name of an existing file. CWE-416: Use After Free
6.8	**CVE-2015-8866:** ext/libxml/libxml.c in PHP before 5.5.22 and 5.6.x before 5.6.6, when PHP-FPM is used, does not isolate each thread from libxml_disable_entity_loader changes in other threads, which allows remote attackers to conduct XML External Entity (XXE) and XML Entity Expansion (XEE) attacks via a crafted XML document, a related issue to CVE-2015-5161. CWE-611: Improper Restriction of XML External Entity Reference ('XXE')
7.1	**CVE-2015-8878:** main/php_open_temporary_file.c in PHP before 5.5.28 and 5.6.x before 5.6.12 does not ensure thread safety, which allows remote attackers to cause a denial of service (race condition and heap memory corruption) by leveraging an application that performs many temporary-file accesses.
7.5	**CVE-2016-4071:** Format string vulnerability in the php_snmp_error function in ext/snmp/snmp.c in PHP before 5.5.34, 5.6.x before 5.6.20, and 7.x before 7.0.5 allows remote attackers to execute arbitrary code via format string specifiers in an SNMP::get call.
7.5	**CVE-2016-4072:** The Phar extension in PHP before 5.5.34, 5.6.x before 5.6.20, and 7.x before 7.0.5 allows remote attackers to execute arbitrary code via a crafted filename, as demonstrated by mishandling of characters by the phar_analyze_path function in ext/phar/phar.c.
7.5	**CVE-2016-4073:** Multiple integer overflows in the mbfl_strcut function in ext/mbstring/libmbfl/mbfl/mbfilter.c in PHP before 5.5.34, 5.6.x before 5.6.20, and 7.x before 7.0.5 allow remote attackers to cause a denial of service (application crash) or possibly execute arbitrary code via a crafted mb_strcut call.
7.2	**CVE-2014-0185:** sapi/fpm/fpm/fpm_unix.c in the FastCGI Process Manager (FPM) in PHP before 5.4.28 and 5.5.x before 5.5.12 uses 0666 permissions for the UNIX socket, which allows local users to gain privileges via a crafted FastCGI client.

CRASHTEST **SECURITY**



Crashtest Security is a German IT security company specialized in automated web application security testing. The fully automated penetration test lets developers discover vulnerabilities in real-time and supports the remediation through an integrated knowledge base.

CONTACT US:

Crashtest Security GmbH
Leopoldstr. 21
80802 München
+49 (0) 89 215 41 665