Web Vulnerability Scanning Report

DVWA Demo (04 Sep 19 14:40 CEST)
https://dvwa.test.crashtest.cloud/

1 Overview

1.1 Vulnerability Overview

Based on our testing, we identified **47** vulnerabilities:

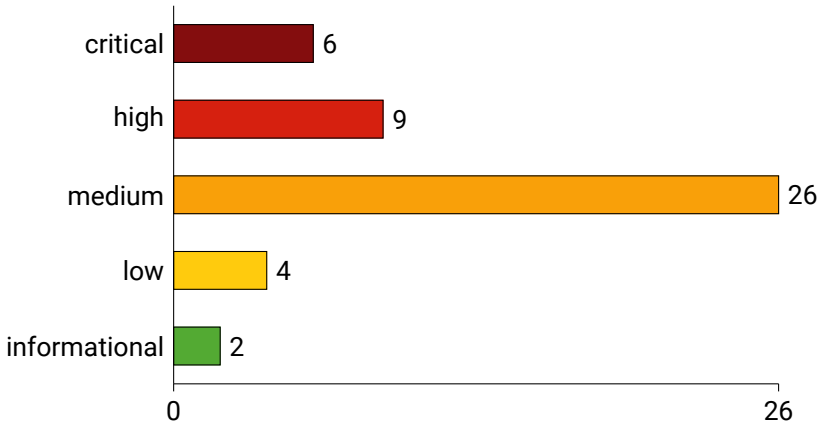

Figure 1.1: Total number of vulnerabilities for ”DVWA Demo”

Risk	Description	Base Score
informational	Informational findings do not pose any threat but have solely informational purpose.	0
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
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
high	Findings in this category pose an immediate threat and should be fixed immediately.	7 - 8.9
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

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
- ✓ 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
- ✓ CVE Comparison
- ✓ Directory Fuzzer
- ✓ File Fuzzer
- ✓ Command Injection
- ✓ XML External Entity Processing (XXE)

1.2.1 Status for executed Scanners

Scanner	Percentage	Status
Portscan	100%	1 completed
Transport Layer Security (TLS/SSL)	100%	1 completed
Command Injection	100%	19 completed
Cross-Site Request Forgery (CSRF)	100%	19 completed
Deserialization	100%	19 completed
Fuzzer	100%	1 completed
SQL Injection	100%	19 completed
Cross-Site Scripting (XSS)	100%	19 completed
XML External Entity (XXE)	100%	19 completed
CVE	100%	2 completed
File Inclusion	100%	19 completed
Fingerprinting	100%	1 completed
	100%	139 completed

1.3 Findings Checklist

1.3.1 XXE

Severity	Finding	Noticed	Fixed
critical	**XXE:** Found XXE in parameter ”xml” with method ”get” for URL ”https://dvwa.test.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>”	☐	☐

1.3.2 XSS

Severity	Finding	Noticed	Fixed
medium	**Cross-Site Scripting (XSS):** Found possible XSS vulnerability on site dvwa.test.crashtest.cloud/vulnerabilities/xss_r/. The parameter 'name' seems vulnerable for payload '<svg ”ons>'	☐	☐
medium	**Cross-Site Scripting (XSS):** Found possible XSS vulnerability on site dvwa.test.crashtest.cloud/vulnerabilities/xss_s/. The parameter 'mtxMessage' seems vulnerable for payload '<svg ”ons>'	☐	☐
medium	**Cross-Site Scripting (XSS):** Found possible XSS vulnerability on site dvwa.test.crashtest.cloud/vulnerabilities/xss_s/. The parameter 'txtName' seems vulnerable for payload '<svg ”ons>'	☐	☐
medium	**Cross-Site Scripting (XSS):** Found possible XSS vulnerability on site dvwa.test.crashtest.cloud/vulnerabilities/deserialize/. The parameter 'data' seems vulnerable for payload '<svg ”ons>'	☐	☐
medium	**Cross-Site Scripting (XSS):** Found possible XSS vulnerability on site dvwa.test.crashtest.cloud/vulnerabilities/nosql/. The parameter 'text' seems vulnerable for payload '<svg ”ons>'	☐	☐

Severity	Finding	Noticed	Fixed
medium	**Cross-Site Scripting (XSS):** Found possible XSS vulnerability on site dvwa.test.crashtest.cloud/vulnerabilities/nosql/. The parameter 'title' seems vulnerable for payload '<svg "ons>'	☐	☐

1.3.3 COMMANDINJECTION

Severity	Finding	Noticed	Fixed
critical	**Command Injection:** Found command injection in parameter "ip" with method "post" for URL "https://dvwa.test.crashtest.cloud/vulnerabilities/exec/", with payload "; echo crashtest-security$((12*12))"	☐	☐

1.3.4 DESERIALIZATION

Severity	Finding	Noticed	Fixed
high	**Insecure Deserialization:** Found insecure deserialization for method "get" with parameter "data" on "https://dvwa.test.crashtest.cloud/vulnerabilities/deserialize/" with payload "phpinfo();"	☐	☐

1.3.5 FILEINCLUSION

Severity	Finding	Noticed	Fixed
critical	**Local File Inclusion:** Found file inclusion with method "get" for parameter "page" on "https://dvwa.test.crashtest.cloud/vulnerabilities/fi/." with payload "/etc/passwd"	☐	☐

1.3.6 FINGERPRINTING

Severity	Finding	Noticed	Fixed
high	**Fingerprint Web Server:** The webserver is running Apache 2.4.7 (**23** 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)	☐	☐

Severity	Finding	Noticed	Fixed
critical	**Fingerprint Web Application Framework:** Found PHP running in version 5.5.9-1ubuntu4.14. (**100** 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)	☐	☐

1.3.7 PORTSCAN

Severity	Finding	Noticed	Fixed
informational	**Portscanner:** Found open port "80/tcp" with service name "Apache httpd"	☐	☐
informational	**Portscanner:** Found open port "443/tcp" with service name "Apache httpd"	☐	☐

1.3.8 FUZZER

Severity	Finding	Noticed	Fixed
medium	**Sensitive Data Exposure:** Retrieved https://dvwa.test.crashtest.cloud/about.php by using a GET request on the URL without prior knowledge.	☐	☐
medium	**Sensitive Data Exposure:** Retrieved https://dvwa.test.crashtest.cloud/config/ by using a GET request on the URL without prior knowledge.	☐	☐
medium	**Sensitive Data Exposure:** Retrieved https://dvwa.test.crashtest.cloud/docs/ by using a GET request on the URL without prior knowledge.	☐	☐
medium	**Sensitive Data Exposure:** Retrieved https://dvwa.test.crashtest.cloud/.git/ by using a GET request on the URL without prior knowledge.	☐	☐
medium	**Sensitive Data Exposure:** Retrieved https://dvwa.test.crashtest.cloud/instructions.php by using a GET request on the URL without prior knowledge.	☐	☐
medium	**Sensitive Data Exposure:** Retrieved https://dvwa.test.crashtest.cloud/phpinfo.php by using a GET request on the URL without prior knowledge.	☐	☐
medium	**Sensitive Data Exposure:** Retrieved https://dvwa.test.crashtest.cloud/php.ini by using a GET request on the URL without prior knowledge.	☐	☐
medium	**Sensitive Data Exposure:** Retrieved https://dvwa.test.crashtest.cloud/README.md by using a GET request on the URL without prior knowledge.	☐	☐

Severity	Finding	Noticed	Fixed
medium	**Sensitive Data Exposure:** Retrieved https://dvwa.test.crashtest.cloud/setup.php by using a GET request on the URL without prior knowledge.	☐	☐

1.3.9 SQLINJECTION

Severity	Finding	Noticed	Fixed
critical	**SQL Injection:** Found boolean-based blind sqlinjection for parameter id (GET) on https://dvwa.test.crashtest.cloud/vulnerabilities/sqli/ with payload Submit=Submit&id=xyz' AND 7605=(SELECT (CASE WHEN (7605=7605) THEN 7605 ELSE (SELECT 5454 UNION SELECT 5045) END))–xZyY	☐	☐
critical	**SQL Injection:** Found boolean-based blind sqlinjection for parameter username (GET) on https://dvwa.test.crashtest.cloud/vulnerabilities/brute/ with payload Login=Login&password=Crashtest123!&username=xyz' AND 3920=(SELECT (CASE WHEN (3920=3920) THEN 3920 ELSE (SELECT 1453 UNION SELECT 9149) END))–wLMA	☐	☐

1.3.10 SSL/TLS

Severity	Finding	Noticed	Fixed
medium	**SSL Cipher Order:** NOT a cipher order configured	☐	☐
medium	**SSL Insecure Algorithm:** Signature Algorithm: MD5	☐	☐
low	**SSL LOGJAM Common Primes:** LOGJAM vulnerability detected CVE-2015-4000	☐	☐
medium	**SSL SWEET32:** Uses 64 bit block ciphers	☐	☐
high	**SSL Cipherlist LOW:** Cipherlist_LOW is offered by the server.	☐	☐
medium	**TLS Key Size:** The certificate key size is RSA 1024 bits.	☐	☐
low	**SSL Cipher Block Chaining SSL3:** BEAST SSL3: The BEAST attack leverages weakness in the cipher block chaining (CBC) which allows man in the middle attacks.	☐	☐
medium	**TLS Configuration:** DHE-RSA-AES256-SHA256, 1024 bit DH (cbc) (matching cipher in list missing)	☐	☐
medium	**TLS Configuration:** LOGJAM vulnerability detected CVE-2015-4000	☐	☐
medium	**Missing Security Headers:** No security headers detected	☐	☐
medium	**Missing HSTS:** HSTS is not offered by the server.	☐	☐

Severity	Finding	Noticed	Fixed
low	**SSL Cipherlist AVERAGE:** Cipherlist_AVERAGE is offered by the server.	☐	☐
medium	**SSL BEAST:** VULNERABLE – but also supports higher protocols TLSv1.1 TLSv1.2 (likely mitigated)	☐	☐
high	**SSL Trust:** Certificate is selfsigned.	☐	☐
high	**SSL Trust:** Certificate does not match supplied URI (same w/o SNI)	☐	☐
high	**SSL Trust:** There is no subject alt name defined. Browsers are complaining.	☐	☐
medium	**SSL Cipher Block Chaining TLS1:** BEAST TLS1 The BEAST attack leverages weakness in the cipher block chaining (CBC) which allows man in the middle attacks.	☐	☐
low	**SSL POODLE:** VULNERABLE, uses SSLv3+CBC	☐	☐
medium	**SSL RC4:** VULNERABLE, Detected ciphers: ECDHE-RSA-RC4-SHA RC4-SHA RC4-MD5	☐	☐
high	**Certificate Revocation:** Neither CRL nor OCSP URI provided	☐	☐
high	**SSL Cipherlist 3DES IDEA:** Cipherlist_3DES_IDEA is offered by the server.	☐	☐
high	**SSL Protocol Version:** SSLv3 is offered by the server.	☐	☐

Contents

2 Findings

2.1 COMMANDINJECTION

2.1.1 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.

2.1.2 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 Schema 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.test.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

2.2 DESERIALIZATION

2.2.1 What is this?

Insecure Deserialization is an attack where a manipulated object is injected into the context of the web application. If the application is vulnerable, the object is deserialized and executed, which can result in SQL Injection, Path Traversal, Application Denial of Service and Remote Code Execution.

2.2.2 Insecure Deserialization

Severity

Base Score:	high (8.1/10)
Impact:	medium (5.9/10)
Exploitability:	low (2.2/10)

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

Description

Insecure Deserialization allows an attacker to inject a manipulated object into the web application.

Finding

- Found insecure deserialization for method ”get” with parameter ”data” on ”https://dvwa.test.crashtest.cloud/vulnerabilities/deserialize/” with payload ”phpinfo();”

How to fix

Do not pass untrusted serialized objects to the unserialize function

Recommendations

https://wiki.crashtest-security.com/insecure-deserialization

2.3 FILEINCLUSION

2.3.1 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.

2.3.2 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 Schema 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.test.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

2.4 FINGERPRINTING

2.4.1 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.

2.4.2 Fingerprint Web Server

Severity

Base Score:	high (7.5/10)
Impact:	-
Exploitability:	-

All values are based on the Common Vulnerability Scoring Schema 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 (**23** 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

2.4.3 Fingerprint Web Application Framework

Severity

Base Score:	critical (10/10)
Impact:	-
Exploitability:	-

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

Description

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

Finding

- Found PHP running in version 5.5.9-1ubuntu4.14. (**100** 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

2.5 FUZZER

2.5.1 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.

2.5.2 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 Schema 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.test.crashtest.cloud/about.php by using a GET request on the URL without prior knowledge.
- Retrieved https://dvwa.test.crashtest.cloud/config/ by using a GET request on the URL without prior knowledge.
- Retrieved https://dvwa.test.crashtest.cloud/docs/ by using a GET request on the URL without prior knowledge.
- Retrieved https://dvwa.test.crashtest.cloud/.git/ by using a GET request on the URL without prior knowledge.
- Retrieved https://dvwa.test.crashtest.cloud/instructions.php by using a GET request on the URL without prior knowledge.
- Retrieved https://dvwa.test.crashtest.cloud/phpinfo.php by using a GET request on the URL without prior knowledge.
- Retrieved https://dvwa.test.crashtest.cloud/php.ini by using a GET request on the URL without prior knowledge.
- Retrieved https://dvwa.test.crashtest.cloud/README.md by using a GET request on the URL without prior knowledge.
- Retrieved https://dvwa.test.crashtest.cloud/setup.php 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

2.6 PORTSCAN

2.6.1 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.

2.6.2 Portscanner

Severity

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

All values are based on the Common Vulnerability Scoring Schema 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 ”80/tcp” with service name ”Apache httpd”
- Found open port ”443/tcp” with service name ”Apache httpd”

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

2.7 XXE

2.7.1 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.

2.7.2 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 Schema 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.test.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

2.8 SQLINJECTION

2.8.1 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.

2.8.2 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 Schema 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 id (GET) on https://dvwa.test.crashtest.cloud/vulnerabilities/sqli/ with payload Submit=Submit&id=xyz' AND 7605=(SELECT (CASE WHEN (7605=7605) THEN 7605 ELSE (SELECT 5454 UNION SELECT 5045) END))– xZyY
- Found boolean-based blind sqlinjection for parameter username (GET) on https://dvwa.test.crashtest.cloud/vulnerabilities/brute/ with payload Login=Login&password=Crashtest123!&username=xyz' AND 3920=(SELECT (CASE WHEN (3920=3920) THEN 3920 ELSE (SELECT 1453 UNION SELECT 9149) END))– wLMA

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

2.9 SSL/TLS

2.9.1 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.

2.9.2 SSL Cipher Order

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 Schema v3.

Description

There is no cipher order for HTTPS ciphers set or the cipher order includes an insecure cipher. This means, that an attacker can make use of an insecure SSL/TLS connection.

Finding

- NOT a cipher order configured

How to fix

There is no cipher order for HTTPS ciphers set or the cipher order includes an insecure cipher. 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/configure-ssl-cipher-order

2.9.3 SSL Insecure Algorithm

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 Schema v3.

Description

The used encryption algorithm has severe security issues.

Finding

- Signature Algorithm: MD5

How to fix

One of the used encryption algorithms has severe security issues and needs to be replaced with a newer algorithm. More details on which cipher suites have strong encryption algorithms can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/disable-ssl-insecure-algorithm

2.9.4 SSL LOGJAM Common Primes

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 Schema v3.

Description

The server is vulnerable for LOGJAM, a security vulnerability against a Diffie-Hellman key exchange using 512 to 1024 bit keys. The algorithm uses in most cases the same pregenerated prime numbers which makes it way easier (and cheaper) to crack such an encryption.

Finding

- LOGJAM vulnerability detected CVE-2015-4000

How to fix

LOGJAM attacks can be prevented by using strong ciphers and avoiding weak primes. More details on what ciphers are considered strong can be found in the knowledge database (see Recommendations)

Recommendations

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

2.9.5 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 Schema 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

- Uses 64 bit block ciphers

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

2.9.6 SSL Cipherlist LOW

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 Schema v3.

Description

The server is configured to support low Ciphers like ”LOW:DES:RC2:RC4”. This means, that an attacker can make use of an insecure SSL/TLS connection.

Finding

- Cipherlist_LOW is offered by the server.

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

2.9.7 TLS Key Size

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 Schema v3.

Description

The security of a TLS connection heavily depends on the used keysize. The server offers a keysize which will result in a weak encryption.

Finding

- The certificate key size is RSA 1024 bits.

How to fix

The used TLS connection key is too small and therefore can be easily broken. This can be solved by choosing a certificate with a larger key size. More details on which certificates to choose can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/increase-tls-key-size

2.9.8 SSL Cipher Block Chaining SSL3

Severity

Base Score:	low (3.1/10)
Impact:	low (1.4/10)
Exploitability:	low (1.6/10)

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

Description

The webserver is configured to allow connections encrypted with SSL V3 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 SSL3: The BEAST attack leverages weakness in the cipher block chaining (CBC) which allows man in the middle attacks.

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)More details on how to fix this problem can be found in the knowledge database (see Recommendations)

Recommendations

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

2.9.9 TLS Configuration

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 Schema v3.

Description

There is a misconfiguration with your SSL/TLS configuration. SSL/TLS is responsible for encrypting traffic between your web application and a user's browser to prevent eavesdropping.

Finding

- DHE-RSA-AES256-SHA256, 1024 bit DH (cbc) (matching cipher in list missing)
- LOGJAM vulnerability detected CVE-2015-4000

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

2.9.10 Missing Security Headers

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 Schema v3.

Description

Security headers can effectly prevent certain hacking attempts. You should consider headers like Strict-Transport-Security, Content-Security-Policy, X-Frame-Options or X-XSS-Protection

Finding

- No security headers detected

How to fix

Security headers can effectively prevent a variety of hacking attempts. The following headers are the most used ones: Strict-Transport-Security, Content-Security-Policy, X-Frame-Options or X-XSS-Protection. More details on how to set these headers can be found in the knowledge database (see Recommendations)

Recommendations

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

2.9.11 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 Schema 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

2.9.12 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 Schema v3.

Description

The server is configured to support average Ciphers like ”HIGH:MEDIUM:AES:CAMELLIA:ARIA”. This means, that an attacker can make use of an insecure SSL/TLS connection.

Finding

- Cipherlist_AVERAGE is offered by the server.

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

2.9.13 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 Schema 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

2.9.14 SSL Trust

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 Schema v3.

Description

The X.509 certificate issued for this domain cannot be trusted. Clients such as browsers will show warnings or not be able to connect if they cannot trust the certificate. Trust issues arise if the common name in the certificate does not match the webserver domain or if the certificate is self-signed.

Finding

- Certificate is selfsigned.
- Certificate does not match supplied URI (same w/o SNI)
- There is no subject alt name defined. Browsers are complaining.

How to fix

The issued certificate is not consistent with the domain that delivered the certificate. To issue a trusted certificate, the certificate needs to contain the correct information for the web application such as the domain name as common name of the certificate. The certificate must be signed by a certificate authority (CA) that the users' browser trust. The webserver has then to be configured to present the certificate on incoming https requests. Guides on how to generate and use a trusted certificate can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/configure-trusted-certificates

2.9.15 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 Schema 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

2.9.16 SSL POODLE

Severity

Base Score:	low (3.1/10)
Impact:	low (1.4/10)
Exploitability:	low (1.6/10)

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

Description

The server is vulnerable for POODLE (Padding Oracle On Downgraded Legacy Encryption) attacks. With the Man-In-The-Middle attack using the SSL 3.0 Fallback, an attacker can expose data of encrypted connections.

Finding

- VULNERABLE, uses SSLv3+CBC

How to fix

POODLE attacks can be prevented by ensuring that TLS_FALLBACK_SCSV is enabled and a secure TLS configuration is used. More details on how to enable TLS_FALLBACK_SCSV and which configurations are secure, can be found in the knowledge database (see Recommendations)

Recommendations

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

2.9.17 SSL RC4

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 Schema v3.

Description

The server supports RC4 (Rivest Cipher 4), which is a cipher stream that is considered insecure due to multiple known vulnerabilities.

Finding

- VULNERABLE, Detected ciphers: ECDHE-RSA-RC4-SHA RC4-SHA RC4-MD5

How to fix

”Rivest Cipher 4” is considered insecure as there are multiple known vulnerabilities for it. It is recommended to replace the cipher with a strong encryption algorithm. More details on which ciphers to choose, can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/disable-ssl-rc4

2.9.18 Certificate Revocation

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 Schema v3.

Description

The webserver is badly configured regarding revoked certificates. Certificate Revocation Lists (CRLs) and the Online Certificate Status Protocol (OCSP) make sure, that users can verify the integraty of a server certificate.

Finding

- Neither CRL nor OCSP URI provided

How to fix

The webserver is badly configured regarding revoked certificates. Certificate Revocation Lists (CRLs) and the Online Certificate Status Protocol (OCSP) make sure, that users can verify the integrity of a server certificate. If the certificate is compromised, these techniques allow the user, respectively the used certificate authority (CA) to revoke the compromised certificate. Therefore one can issue a new (valid) certificate and the compromised certificate (used by an attacker) will produce warnings when a user accesses their website. OCSP is the newer method to revoke certificates, as it allows certificate authorities to revoke certificates much faster without the need to update complete revocation lists potentially containing thousands of certificates. More details on how to enable OCSP can be found in the knowledge database (see Recommendations)

Recommendations

https://wiki.crashtest-security.com/renew-tls-certificates

2.9.19 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 Schema 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

- Cipherlist_3DES_IDEA is offered by the server.

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

2.9.20 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 Schema 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.

Finding

- SSLv3 is offered by the server.

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)More details on how to fix this problem can be found in the knowledge database (see Recommendations)

Recommendations

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

2.10 XSS

2.10.1 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.

2.10.2 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 Schema 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.test.crashtest.cloud/vulnerabilities/xss_r/. The parameter 'name' seems vulnerable for payload '<svg "ons>'
- Found possible XSS vulnerability on site dvwa.test.crashtest.cloud/vulnerabilities/xss_s/. The parameter 'mtxMessage' seems vulnerable for payload '<svg "ons>'
- Found possible XSS vulnerability on site dvwa.test.crashtest.cloud/vulnerabilities/xss_s/. The parameter 'txtName' seems vulnerable for payload '<svg "ons>'
- Found possible XSS vulnerability on site dvwa.test.crashtest.cloud/vulnerabilities/deserialize/. The parameter 'data' seems vulnerable for payload '<svg "ons>'
- Found possible XSS vulnerability on site dvwa.test.crashtest.cloud/vulnerabilities/nosql/. The parameter 'text' seems vulnerable for payload '<svg "ons>'
- Found possible XSS vulnerability on site dvwa.test.crashtest.cloud/vulnerabilities/nosql/. The parameter 'title' seems vulnerable for payload '<svg "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

2.11 Appendix

2.11.1 Apache 2.4.7 CVE Findings

Apache 2.4.7	
medium	**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.
medium	**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.
medium	**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.
low	**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.
medium	**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.

Apache 2.4.7

medium	**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.
medium	**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.
high	**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.
medium	**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).
medium	**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.
medium	**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.
medium	**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.
low	**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.

Apache 2.4.7

medium	**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.
medium	**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.
medium	**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.
medium	**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.
medium	**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.
medium	**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.
medium	**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.
medium	**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.

Apache 2.4.7

medium	**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.
medium	**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.

2.11.2 PHP 5.5.9 CVE Findings

PHP 5.5.9

medium	**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.
medium	**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.
medium	**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.
high	**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.
high	**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.

PHP 5.5.9

high	**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.
high	**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.
medium	**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.
medium	**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.
medium	**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.
high	**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.
medium	**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.
medium	**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.

PHP 5.5.9

critical	**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.
critical	**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.
high	**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.
high	**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.
high	**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.
medium	**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.
high	**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.
medium	**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.

PHP 5.5.9

medium	**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.
medium	**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.
medium	**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."
high	**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.
high	**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.
medium	**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.
medium	**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.
medium	**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.

PHP 5.5.9

medium	**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.
low	**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.
medium	**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.
medium	**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.
medium	**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.
medium	**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.
medium	**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.
medium	**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.
medium	**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.

PHP 5.5.9

medium	**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.
medium	**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").
medium	**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.
high	**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.
medium	**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.
high	**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.
medium	**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 underread and application crash) via a long name.
medium	**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.

PHP 5.5.9

high	**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.
medium	**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.
high	**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.
high	**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.
high	**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.
medium	**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.
high	**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.
critical	**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.

PHP 5.5.9

critical	**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.
medium	**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.
high	**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.
high	**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.
medium	**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.
medium	**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.
high	**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.
medium	**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.

PHP 5.5.9

medium	**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.
medium	**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.
high	**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.
high	**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.
high	**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.
medium	**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.
high	**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.
high	**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.

PHP 5.5.9

high	**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.
critical	**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.
critical	**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.
medium	**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.
medium	**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.

PHP 5.5.9

high	**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.
low	**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.
medium	**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.
medium	**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.
low	**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.
medium	**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.
high	**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) SPLObjectStorage.
medium	**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.

PHP 5.5.9

medium	**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.
high	**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.
high	**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.
high	**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.
high	**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.
high	**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.
medium	**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.

PHP 5.5.9

medium	**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.
medium	**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.
medium	**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.
medium	**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.
medium	**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.
medium	**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.
low	**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.
medium	**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.

PHP 5.5.9

medium	**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.
medium	**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.

CRASHTEST SECURITY

Crashtest Security is a german IT security company specialized on 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
c/o WERK1
Atelierstr. 29
81671 München

+49 (0) 89 215 41 665