acunetix

WEB APPLICATION SECURITY

Acunetix Website Audit

29 July, 2015

Web Application Security Consortium: Threat Classification

~ compliance report ~

Web Application Security Consortium: Threat Classification

compliance report

Description

The Web Security Threat Classification is a cooperative effort to clarify and organize the threats to the security of a web site. The members of the Web Application Security Consortium have created this project to develop and promote industry standard terminology for describing these issues. Application developers, security professionals, software vendors, and compliance auditors will have the ability to access a consistent language for web security related issues.

The Web Security Threat Classification will compile and distill the known unique classes of attack, which have presented a threat to web sites in the past.

Disclaimer

This document or any of its content cannot account for, or be included in any form of legal advice. The outcome of a vulnerability scan (or security evaluation) should be utilized to ensure that diligent measures are taken to lower the risk of potential exploits carried out to compromise data.

Legal advice must be supplied according to its legal context. All laws and the environments in which they are applied, are constantly changed and revised. Therefore no information provided in this document may ever be used as an alternative to a qualified legal body or representative.

A portion of the information in this report is taken from Web Application Security Consortium "Threat Classification" document, that can be found at http://www.webappsec.org/.

Scan

URL	**http://agid.internetsoluzioni.it**
Scan date	7/28/2015 7:43:51 PM
Duration	8 hours, 30 minutes
Profile	Default

Compliance at a Glance

This section of the report is a summary and lists the number of alerts found according to individual compliance categories.

- Authentication: Brute Force (1.1)
  No alerts in this category
- Insufficient Authentication (1.2)
  No alerts in this category
- Weak Password Recovery Validation (1.3)
  No alerts in this category
- Credential/Session Prediction (2.1)
  No alerts in this category
- Insufficient Authorization (2.2)
  No alerts in this category
- Insufficient Session Expiration (2.3)
  No alerts in this category
- Session Fixation (2.4)
  No alerts in this category
- Content Spoofing (3.1)
  No alerts in this category

- Cross-site Scripting (3.2)
  - **No alerts in this category**
- Buffer Overflow (4.1)
  - **No alerts in this category**
- Format String Attack (4.2)
  - **No alerts in this category**
- LDAP Injection (4.3)
  - **No alerts in this category**
- OS Commanding (4.4)
  - **No alerts in this category**
- SQL Injection (4.5)
  - **No alerts in this category**
- SSI Injection (4.6)
  - **No alerts in this category**
- XPath Injection (4.7)
  - **No alerts in this category**
- Directory Indexing (5.1)
  - **No alerts in this category**
- Information Leakage (5.2)
  - **Total number of alerts in this category: 32**
- Path Traversal (5.3)
  - **No alerts in this category**
- Predictable Resource Location (5.4)
  - **Total number of alerts in this category: 2**
- Abuse of Functionality (6.1)
  - **Total number of alerts in this category: 2**
- Denial of Service (6.2)
  - **Total number of alerts in this category: 1**
- Insufficient Anti-automation (6.3)
  - **No alerts in this category**
- Insufficient Process Validation (6.4)
  - **No alerts in this category**

Compliance According to Categories: A Detailed Report

This section is a detailed report that explains each vulnerability found according to individual compliance categories.

(1.1) Authentication: Brute Force

A Brute Force attack is an automated process of trial and error used to guess a person's username, password, credit-card number or cryptographic key.

Acunetix authentication tester can be used to bruteforce authentication schemes based either on HTTP protocol NTLM or Basic authentication or HTML form based authentication.

No alerts in this category.

(1.2) Insufficient Authentication

Insufficient Authentication occurs when a web site permits an attacker to access sensitive content or functionality without having to properly authenticate. Web-based administration tools are a good example of web sites providing access to sensitive functionality. Depending on the specific online resource, these web applications should not be directly accessible without the user required to properly verify their identity.

To get around setting up authentication, some resources are protected by "hiding" the specific location and not linking the location into the main web site or other public places. However, this approach is nothing more than "Security Through Obscurity". Its important to understand that simply because a resource is unknown to an attacker, it still remains accessible directly through a specific URL. The specific URL could be discovered through a Brute Force probing for common file and directory locations (/admin for example), error messages, referrer logs, or perhaps documented in help files. These resources, whether they are content or functionality driven, should be adequately protected.

No alerts in this category.

(1.3) Weak Password Recovery Validation

Weak Password Recovery Validation is when a web site permits an attacker to illegally obtain, change or recover another user's password. Conventional web site authentication methods require users to select and remember a password or passphrase. The user should be the only person that knows the password and it must be remembered precisely. As time passes, a user's ability to remember a password fades. The matter is further complicated when the average user visits 20 sites requiring them to supply a password. (RSA Survey: http://news.bbc.co.uk/1/hi/technology/3639679.stm) Thus, Password Recovery is an important part in servicing online users.

No alerts in this category.

(2.1) Credential/Session Prediction

Credential/Session Prediction is a method of hijacking or impersonating a web site user. Deducing or guessing the unique value that identifies a particular session or user accomplishes the attack. Also known as Session Hijacking, the consequences could allow attackers the ability to issue web site requests with the compromised user's privileges.

No alerts in this category.

(2.2) Insufficient Authorization

Insufficient Authorization is when a web site permits access to sensitive content or functionality that should require increased access control restrictions. When a user is authenticated to a web site, it does not necessarily mean that he should have full access to all content and that functionality should be granted arbitrarily.

Authorization procedures are performed after authentication, enforcing what a user, service or application is permitted to do. Thoughtful restrictions should govern particular web site activity according to policy. Sensitive portions of a web site may need to be restricted to everyone expect to perhaps an administrator.

No alerts in this category.

(2.3) Insufficient Session Expiration

Insufficient Session Expiration is when a web site permits an attacker to reuse old session credentials or session IDs for authorization. Insufficient Session Expiration increases a web site's exposure to attacks that steal or impersonate other users.

No alerts in this category.

(2.4) Session Fixation

Session Fixation is an attack technique that forces a user's session ID to an explicit value. Depending on the functionality of the target web site, a number of techniques can be utilized to "fix" the session ID value. These techniques range from Cross-site Scripting exploits to peppering the web site with previously made HTTP requests. After a user's session ID has been fixed, the attacker will wait for them to login. Once the user does so, the attacker uses the predefined session ID value to assume their online identity.

No alerts in this category.

(3.1) Content Spoofing

Content Spoofing is an attack technique used to trick a user into believing that certain content appearing on a web site is legitimate and not from an external source.

Some web pages are served using dynamically built HTML content sources. For example, the source location of a frame
<frame src="http://foo.example/file.html">
could be specified by a URL parameter value:

http://foo.example/page?frame_src=http://foo.example/file.html

An attacker may be able to replace the "frame_src" parameter value with "
frame_src=http://attacker.example/spoof.html
". When the resulting web page is served, the browser location bar visibly remains under the user expected domain (foo.example), but the foreign data (attacker.example) is shrouded by legitimate content.

No alerts in this category.

(3.2) Cross-site Scripting

Cross-site Scripting (XSS) is an attack technique that forces a web site to echo attacker-supplied executable code, which loads in a user's browser. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.

When an attacker gets a user's browser to execute his code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Crosssite Scripting attacks essentially compromise the trust relationship between a user and the web site.

No alerts in this category.

(4.1) Buffer Overflow

Buffer Overflow exploits are attacks that alter the flow of an application by overwriting parts of memory. Buffer Overflow is a common software flaw that results in an error condition. This error condition occurs when data written to memory exceed the allocated size of the buffer. As the buffer is overflowed, adjacent memory addresses are overwritten causing the software to fault or crash. When unrestricted, properly-crafted input can be used to overflow the buffer resulting in a number of security issues.

A Buffer Overflow can be used as a Denial of Service attack when memory is corrupted, resulting in software failure. Even more critical is the ability of a Buffer Overflow attack to alter application flow and force unintended actions. This scenario can occur in several ways. Buffer Overflow vulnerabilities have been used to overwrite stack pointers and redirect the program to execute malicious instructions. Buffer Overflows have also been used to change program variables.

No alerts in this category.

(4.2) Format String Attack

Format String Attacks alter the flow of an application by using string formatting library features to access other memory space. Vulnerabilities occur when user-supplied data are used directly as formatting string input for certain C/C++ functions (e.g. fprintf, printf, sprintf, setproctitle, syslog, ...).

If an attacker passes a format string consisting of printf conversion characters (e.g. "%f", "%p", "%n", etc.) as parameter value to the web application, they may:

- Execute arbitrary code on the server
- Read values off the stack
- Cause segmentation faults / software crashes

No alerts in this category.

(4.3) LDAP Injection

LDAP Injection is an attack technique used to exploit web sites that construct LDAP statements from user-supplied input.

Lightweight Directory Access Protocol (LDAP) is an open-standard protocol for both querying and manipulating X.500 directory services. The LDAP protocol runs over Internet transport protocols, such as TCP. Web applications may use user-supplied input to create custom LDAP statements for dynamic web page requests.

No alerts in this category.

(4.4) OS Commanding

OS Commanding is an attack technique used to exploit web sites by executing Operating System commands through manipulation of application input.

When a web application does not properly sanitize user-supplied input before using it within application code, it may be possible to trick the application into executing Operating System commands. The executed commands will run with the same permissions of the component that executed the command (e.g. Database server, Web application server, Web server, etc.).

No alerts in this category.

(4.5) SQL Injection

SQL Injection is an attack technique used to exploit web sites that construct SQL statements from user-supplied input.

When a web application fails to properly sanitize user-supplied input, it is possible for an attacker to alter the construction of backend SQL statements. When an attacker is able to modify a SQL statement, the process will run with the same permissions as the component that executed the command. (e.g. Database server, Web application server, Web server, etc.). The impact of this attack can allow attackers to gain total control of the database or even execute commands on the system.

No alerts in this category.

(4.6) SSI Injection

SSI Injection (Server-side Include) is a server-side exploit technique that allows an attacker to send code into a web application, which will later be executed locally by the web server. SSI Injection exploits a web application's failure to sanitize user-supplied data before they are inserted into a server-side interpreted HTML file.

If an attacker submits a Server-side Include statement, he may have the ability to execute arbitrary operating system commands, or include a restricted file's contents the next time the page is served.

No alerts in this category.

(4.7) XPath Injection

XPath Injection is an attack technique used to exploit web sites that construct XPath queries from user-supplied input.

XPath 1.0 is a language used to refer to parts of an XML document. It can be used directly by an application to query an XML document, or as part of a larger operation such as applying an XSLT transformation to an XML document, or applying an XQuery to an XML document.

No alerts in this category.

(5.1) Directory Indexing

Automatic directory listing/indexing is a web server function that lists all of the files within a requested directory if the normal base file (index.html/home.html/default.htm) is not present. When a user requests the main page of a web site, they normally type in a URL such as: http://www.example.com - using the domain name and excluding a specific file. The web server processes this request and searches the document root directory for the default file name and sends this page to the client. If this page is not present, the web server will issue a directory listing and send the output to the client. Essentially, this is equivalent to issuing an "ls" (Unix) or "dir" (Windows) command within this directory and showing the results in HTML form. From an attack and countermeasure perspective, it is important to realize that unintended directory listings may be possible due to software vulnerabilities.

No alerts in this category.

(5.2) Information Leakage

Information Leakage is when a web site reveals sensitive data, such as developer comments or error messages, which may aid an attacker in exploiting the system. Sensitive information may be present within HTML comments, error messages, source code, or simply left in plain sight. There are many ways a web site can be coaxed into revealing this type of information. While leakage does not necessarily represent a breach in security, it does give an attacker useful guidance for future exploitation. Leakage of sensitive information may carry various levels of risk and should be limited whenever possible.

Total number of alerts in this category: 32

Alerts in this category

Possible sensitive directories

A possible sensitive directory has been found. This directory is not directly linked from the website.This check looks for common sensitive resources like backup directories, database dumps, administration pages, temporary directories. Each one of these directories could help an attacker to learn more about his target.

CVSS	Base Score: 5.0 - Access Vector: Network - Access Complexity: Low - Authentication: None - Confidentiality Impact: Partial - Integrity Impact: None - Availability Impact: None
CWE	CWE-200

Affected item	**/editor/ckeditor**
Affected parameter	
Variants	1

Affected item	**/editor/ckeditor/_source**
Affected parameter	
Variants	1

Affected item	**/editor/ckeditor/_source/plugins/save**
Affected parameter	
Variants	1

Affected item	**/editor/ckeditor/_source/tests**
Affected parameter	
Variants	1

Affected item	**/editor/ckeditor/_tests**
Affected parameter	
Variants	1

Affected item	**/editor/ckeditor/plugins/save**
Affected parameter	
Variants	1

Affected item	**/editor/fckeditor**

Affected parameter	
Variants	1

Affected item	**/editor/fckeditor/editor/_source**
Affected parameter	
Variants	1

Affected item	**/editor/fckeditor/editor/_source/globals**
Affected parameter	
Variants	1

Affected item	**/editor/fckeditor/editor/filemanager**
Affected parameter	
Variants	1

Affected item	**/editor/filemanager**
Affected parameter	
Variants	1

Affected item	**/editor/filemanager/include**
Affected parameter	
Variants	1

Affected item	**/editor/filemanager/uploader**
Affected parameter	
Variants	1

Affected item	**/grafica/admin**
Affected parameter	
Variants	1

Affected item	**/inc**
Affected parameter	
Variants	1

Affected item	**/temp**
Affected parameter	
Variants	1

Affected item	**/tmp**
Affected parameter	
Variants	1

Possible sensitive files

A possible sensitive file has been found. This file is not directly linked from the website. This check looks for common sensitive resources like password files, configuration files, log files, include files, statistics data, database dumps. Each one of these files could help an attacker to learn more about his target.

CVSS	Base Score: 5.0 - Access Vector: Network - Access Complexity: Low - Authentication: None - Confidentiality Impact: Partial - Integrity Impact: None - Availability Impact: None
CWE	CWE-200

Affected item	**/admin.php**
Affected parameter	
Variants	1

Affected item	**/personalizzazioni/popover/.gitignore**

Affected parameter	
Variants	1

Email address found

One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.

CVSS	Base Score: 5.0 - Access Vector: Network - Access Complexity: Low - Authentication: None - Confidentiality Impact: Partial - Integrity Impact: None - Availability Impact: None
CWE	CWE-200

Affected item	**/archivio13_strutture-organizzative_0_9861_0.html**
Affected parameter	
Variants	1

Affected item	**/archivio13_strutture-organizzative_0_9864_0.html**
Affected parameter	
Variants	1

Affected item	**/archivio13_strutture-organizzative_0_9882_0.html**
Affected parameter	
Variants	1

Affected item	**/archivio13_strutture-organizzative_0_9892_0.html**
Affected parameter	
Variants	1

Affected item	**/archivio13_strutture-organizzative_0_9902_0.html**
Affected parameter	
Variants	1

Affected item	**/archivio13_strutture-organizzative_0_9908_0.html**
Affected parameter	
Variants	1

Affected item	**/archivio3_personale-ente_0_40509_0.html**
Affected parameter	
Variants	1

Affected item	**/archivio3_personale-ente_0_40510_0.html**
Affected parameter	
Variants	1

Affected item	**/archivio3_personale-ente_0_40511_0.html**
Affected parameter	
Variants	1

Affected item	**/archivio3_personale-ente_0_40512_0.html**
Affected parameter	
Variants	1

Affected item	**/archivio3_personale-ente_0_40514_0.html**
Affected parameter	
Variants	1

Affected item	**/archivio3_personale-ente_0_40517_0.html**
Affected parameter	
Variants	1

Error page web server version disclosure

By requesting a page that doesn't exist, an error page was returned. This error page contains the web server version number and a list of modules enabled on this server. This information can be used to conduct further attacks.

CVSS	Base Score: 5.0 - Access Vector: Network - Access Complexity: Low - Authentication: None - Confidentiality Impact: Partial - Integrity Impact: None - Availability Impact: None
CWE	CWE-200

Affected item	**Web Server**
Affected parameter	
Variants	1

(5.3) Path Traversal

The Path Traversal attack technique forces access to files, directories, and commands that potentially reside outside the web document root directory. An attacker may manipulate a URL in such a way that the web site will execute or reveal the contents of arbitrary files anywhere on the web server. Any device that exposes an HTTPbased interface is potentially vulnerable to Path Traversal.

Most web sites restrict user access to a specific portion of the filesystem, typically called the "web document root" or "CGI root" directory. These directories contain the files intended for user access and the executables necessary to drive web application functionality. To access files or execute commands anywhere on the file-system, Path Traversal attacks will utilize the ability of special-characters sequences.

No alerts in this category.

(5.4) Predictable Resource Location

Predictable Resource Location is an attack technique used to uncover hidden web site content and functionality. By making educated guesses, the attack is a brute force search looking for content that is not intended for public viewing. Temporary files, backup files, configuration files, and sample files are all examples of potentially leftover files. These brute force searches are easy because hidden files will often have common naming convention and reside in standard locations. These files may disclose sensitive information about web application internals, database information, passwords, machine names, file paths to other sensitive areas, or possibly contain vulnerabilities. Disclosure of this information is valuable to an attacker.

Total number of alerts in this category: 2

Alerts in this category

Possible sensitive files

A possible sensitive file has been found. This file is not directly linked from the website. This check looks for common sensitive resources like password files, configuration files, log files, include files, statistics data, database dumps. Each one of these files could help an attacker to learn more about his target.

CVSS	Base Score: 5.0 - Access Vector: Network - Access Complexity: Low - Authentication: None - Confidentiality Impact: Partial - Integrity Impact: None - Availability Impact: None
CWE	CWE-200

Affected item	**/admin.php**

Affected parameter	
Variants	1

Affected item	**/personalizzazioni/popover/.gitignore**
Affected parameter	
Variants	1

(6.1) Abuse of Functionality

Abuse of Functionality is an attack technique that uses a web site's own features and functionality to consume, defraud, or circumvents access controls mechanisms. Some functionality of a web site, possibly even security features, may be abused to cause unexpected behavior. When a piece of functionality is open to abuse, an attacker could potentially annoy other users or perhaps defraud the system entirely. The potential and level of abuse will vary from web site to web site and application to application.

Total number of alerts in this category: 2

Alerts in this category

Clickjacking: X-Frame-Options header missing

Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages.

The server didn't return an X-Frame-Options header which means that this website could be at risk of a clickjacking attack. The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.

CVSS	Base Score: 6.8 - Access Vector: Network - Access Complexity: Medium - Authentication: None - Confidentiality Impact: Partial - Integrity Impact: Partial - Availability Impact: Partial
CWE	CWE-693

Affected item	**Web Server**
Affected parameter	
Variants	1

Possible relative path overwrite

Manual confirmation is required for this alert.
Gareth Heyes introduced a technique to take advantage of CSS imports with relative URLs by overwriting their target file. This technique can be used by an attacker to trick browsers into importing HTML pages as CSS stylesheets. If the attacker can control a part of the imported HTML pages he can abuse this issue to inject arbitrary CSS rules.

CVSS	Base Score: 0.0 - Access Vector: Network - Access Complexity: Low - Authentication: None - Confidentiality Impact: None - Integrity Impact: None - Availability Impact: None
CWE	CWE-20

Affected item	**/index.php**
Affected parameter	
Variants	1

(6.2) Denial of Service

Denial of Service (DoS) is an attack technique with the intent of preventing a web site from serving normal user activity. DoS attacks, which are easily normally applied to the network layer, are also possible at the application layer. These malicious attacks can succeed by starving a system of critical resources, vulnerability exploit, or abuse of functionality.

Total number of alerts in this category: 1

Alerts in this category

Apache httpd remote denial of service

A denial of service vulnerability has been found in the way the multiple overlapping ranges are handled by the Apache HTTPD server:

http://seclists.org/fulldisclosure/2011/Aug/175

An attack tool is circulating in the wild. Active use of this tools has been observed. The attack can be done remotely and with a modest number of requests can cause very significant memory and CPU usage on the server.

This alert was generated using only banner information. It may be a false positive.
Affected Apache versions (1.3.x, 2.0.x through 2.0.64, and 2.2.x through 2.2.19).

CVSS	Base Score: 7.9 - Access Vector: Network - Access Complexity: Low - Authentication: None - Confidentiality Impact: None - Integrity Impact: None - Availability Impact: Complete
CWE	CWE-399
CVE	CVE-2011-3192

Affected item	**Web Server**
Affected parameter	
Variants	1

(6.3) Insufficient Anti-automation

Insufficient Anti-automation is when a web site permits an attacker to automate a process that should only be performed manually. Certain web site functionalities should be protected against automated attacks.

Left unchecked, automated robots (programs) or attackers could repeatedly exercise web site functionality attempting to exploit or defraud the system. An automated robot could potentially execute thousands of requests a minute, causing potential loss of performance or service.

No alerts in this category.

(6.4) Insufficient Process Validation

Insufficient Process Validation is when a web site permits an attacker to bypass or circumvent the intended flow control of an application. If the user state through a process is not verified and enforced, the web site could be vulnerable to exploitation or fraud.

No alerts in this category.