



# Plynt Certification Criteria

Version 3.0

Effective Date: April 15, 2010



The awarding of the Plynt Certificate establishes that a web application has adequate measures to guard against remote adversaries and protect against a wide range of threats.

This document defines the Plynt Certification Standard. It details the criteria that an application must meet in order to be awarded the Certificate.

This document is organized in two parts:

1. Part I lists and defines the certification criteria, and
2. Part II explains the logic and rationale of the criteria.



## Part1: Plynt Certification Criteria

The certification standard is composed of 16 criteria. These are organized in two sections:

Section 1, “*Security Protection Criteria*” identifies the defenses an application must demonstrate to meet the certification standard.

Section 2, “*Security Requirements Criteria*”, specifies the features and behavior an application must have to meet the certification standard



## Section 1: Security Protection Criteria

- 1 Safe against popular attacks:** The application must demonstrate that it is not vulnerable to popular attacks.

*“Popular attacks” include but are not limited to exploits documented by organizations such as, The Open Web Application Security Project (owasp.org), The Web Application Security Consortium (webappsec.org), The Open Source Security Testing Methodology Manual (osstmm.org), SANS Common Weakness Enumeration (CWE) TOP 25 (cwe.mitre.org/top25), etc.*
- 2 Defend against Threat Profile:** The application must demonstrate that it defends against the threats specified in a threat profile that has been developed specifically for this application.

*Note I: A threat describes the goal of an adversary. According to the National Information Systems Security Glossary, a threat is any circumstance or event that has the potential to harm an information system through unauthorized access, destruction, disclosure, modification of data, and/or denial of service.*

*Note II: The threat profile is a list of all possible threats to an application. These threats include violating the business rules and authorization rules of the application.*

*Note III: This criterion applies to high and medium risk vulnerabilities that let threats be realized.*
- 3 Protect sensitive data in transmission:** The application must take adequate measures to protect sensitive data from being stolen over the network.
- 4 Safeguard passwords:** The application must demonstrate that a remote adversary cannot steal user passwords from the application.

*Note I: The criterion is inclusive of post-login. That is, it requires that even after a user logs out the user’s password must be protected from theft.*

*Note II: The criterion recognizes that there are social engineering methods that could be used to steal passwords without access to the application. These are not within the scope of this criterion.*
- 5 Protect against password guessing:** If the password used by the application has less than 10,000 possible values, the application must protect against manual password guessing attacks.

*Note I: A 4-digit numeric PIN is an example of a password with less than 10,000 possible values.*

*Note II: The risk will vary depending on the predictability of the usernames used by the application.*
- 6 Secure Forgot Password Implementation:** If the application provides a password recovery or ‘forgot password’ feature with secret question(s), it must protect against an adversary guessing the answer to the secret question(s). In addition the application should ensure that upon issuance of a new password, it is securely communicated to the user.
- 7 Insecure Configuration Settings on servers accessible directly by users:** All services that a user can reach must be securely configured so there is no leakage of sensitive information. This includes securing all directories and configuration files that are publicly accessible .

*Note I: Configuration files, as used in this criterion, include OS, web server and application configuration files.*

*Note II: Directory listings, as used in this criterion, refer to a listing of all files and directories in a folder, regardless of whether there are links to those objects from the application. While an adversary can compile a list of links in the application by studying the html pages, such compilations are not considered directory listings .*

## Section 2: Security Requirements Criteria

**8 Safe Sensitive data not stored on client:** The application must not store sensitive data on the client machine in easily accessible locations.

*Note I: Easily accessible locations include the browser cache, the browser history, other browser menus and persistent cookies on the client machine.*

*Note II: The browser memory is not considered an easily accessible location for this criterion.*

**9 Sensitive data not hidden in pages:** The application must not hide sensitive data in html comments or hidden form fields embedded in the pages.

**10 No sensitive data in error messages:** The application must not reveal sensitive information in error messages.

*Note: Sensitive information includes not only business sensitive information but also details regarding application architecture that could aid an adversary when attempting to launch an attack against the application.*

**11 Code obfuscation for secrets:** If Javascripts, Applets or ActiveX controls contain secrets, they must use strong code obfuscation techniques to protect the secrets .

*Note: The secrets the above criterion refers to include cryptographic keys, passwords, and algorithms considered a trade secret.*

**12 Re-authentication required for sensitive activities:** The application must re-authenticate the user before allowing the user to perform an operation involving sensitive data. A few examples of an operation involving sensitive data are – Change Password, Transfer Funds and Transaction Approval.

*Note: This criterion does not apply to the special case where the user has forgotten the password and resets the password using the application's password recovery or "Forgot Password" feature .*

**13 No sensitive data in requests to external sites:** If the application maintains links to external sites, it must not disclose to the external site any sensitive data other than that required to conduct the operation with the external site

*Note: Sensitive data, as used in this criterion, includes business sensitive information, as well as session token(s) and authentication token(s).*

**14 Webserver protected against known vulnerabilities:** The web server service running on the server which houses the application must not be vulnerable to publicly known exploitable vulnerabilities. Private information on the webserver that is accessible remotely must be revealed only after the user is securely authenticated.

**15 No sample or test applications:** The server must not make available any sample or test application to remote users.

**16 No sensitive data in source code:** The application must not disclose sensitive data in any source code that is accessible to remote users.



## Part2: Guide to the Plynt Certification Criteria

### What's the logic behind the Plynt Certification Criteria?

The certification criteria specify the standards that an application must meet to establish that it has adequate security measures. An application must prove that it resists attacks as well as demonstrate the implementation of security features that enhance its security. Accordingly, the certification criteria reflect these two aspects. The “Security Protection Criteria”, defined above in Section I, defines the threats that the application must show itself to be protected against. Section II, “Security Requirements Criteria” specifies the features and characteristics the application must maintain to enhance its security.

### My application faces threats that you don't specify in the criteria. How will you address those?

It's quite likely that your application faces threats that are specific to its business context. For instance, a banking application needs to protect against the threat of funds being siphoned off, and a gaming application needs to defend against the rules of the game being violated. The certification criteria require the application to protect against these threats in the Criterion 2, above, Defend against Threat Profile. The threat profile is a customized listing of the threats relevant to that specific application. The threat profile is developed by the Plynt team with inputs from the application owner. The security engineers then develop test cases specific to the threat profile to verify that the application is safe against those threats.

### Why doesn't the criteria talk about SQL Injection, Cross Site Scripting etc.?

When writing the criteria, the authors first considered listing each individual attack that would be tested. However since the criteria is revised with the discovery of every new attack, and there was no agreed to standard taxonomy of attacks, it is reasonable to categorize all of the popular attacks under Criterion 1. Criterion 1, Safe against popular attacks, catches the attacks like those listed in this question. Some of the popular attacks the application must defend against are listed here:

Variable Manipulation	Denial of Service	Brute Force
Bypass input validation	Session Hijacking	Session Fixation
Content Spoofing	Cross-site Scripting	Cross Site Tracing
Buffer Overflow	Format String Attack	SQL Injection
OS Command Injection	LDAP Injection	XPath Injection
SSI Injection	Blind Injection attacks	Directory Indexing
Directory Traversal	Authentication Bypass	Filter evasion

### You missed criterion xyz!

Thank you for pointing out. We are constantly looking at ways to improve the criteria – so your suggestions will help.



### **What do you mean by sensitive data in the criteria?**

*Sensitive data is any data that if compromised could lead to:*

- *Loss to the business in financial or reputation terms*
- *Invasion of the privacy of individuals*
- *Adversaries gaining an advantage to launch attacks*

*Sensitive data includes business sensitive information, authentication credentials, session token(s), authentication token(s) and any details regarding application architecture that could aid an adversary in launching a successful attack against the application .*

### **The criteria say that the application must not allow passwords to be stolen even after the user logs out. Please explain this?**

*In general, the application must safeguard passwords. The note mentioned in Criteria 4 in Section 1 highlights that the password should be protected even when the user has logged out. Vulnerabilities in some authentication schemes enable the password to be stolen if a user who has logged out leaves the browser window open. This is described in the Paladion paper "Stealing Passwords via Browser Refresh".*

### **An application must re-authenticate the user after log out – Isn't this what all applications do?**

*Experience shows that not all applications enforce re-authentication after log out. To make matters worse, some platforms like .Net do not invalidate a user session immediately when their signout method is called. To learn more, please read the paper "ASP.Net Forms Authentication" from Foundstone. As the note points out, if an application does not invalidate a session immediately, it should protect against the session being reused. Also, this Microsoft support document (<http://support.microsoft.com/default.aspx?scid=kb;en-us;900111>) has examples of such protection.*

### **You don't mention SSL anywhere in the criteria!**

*The Criteria doesn't mention SSL, or any specific encryption technology. The criterion "Protect sensitive data in transmission" requires that the application take adequate protection to safeguard sensitive data in transit. SSL is a good example of such protection. The criteria, however, does not demand SSL. The application is free to choose the technology, as long as it demonstrated that it protects sensitive data during transmission.*

### **Please explain, "No sensitive data in requests to external sites"?**

*Links to external sites are often a source of data leakage. Along with the URL, the HTTP request also includes a referrer field that contains the URL of the page from which the request originated. If that URL (and the referrer field in turn) contains sensitive data, then that data will get logged in the web server logs of the external site. Administrators of the external site, or anyone who might gain administrator access, have access to those logs and can collect the sensitive information.*

### **"No sensitive data in source code available to users". What does this specifically describe?**

*This criterion refers to both client-side javascripts and the instances that occur when web server mis-configurations lead to server-side source code being available to adversaries. The criterion requires that any source code that's available to users not contain sensitive data .*

### **"Why only these criteria? Does this cover everything?"**

*The Plynt criteria focus on the high impact threats popular today. The certificate assures that an application is safe against those threats. The authors chose to omit criteria that are not essential or do not have a material impact on security. For example, the certificate does not require that the application catch all error messages. While that's a good practice to follow, it would have been unduly stringent for the certificate. The standard only demands that no sensitive data be revealed in error messages as that has direct impact on the application's security .*



### **Why a new version of the criteria?**

New technologies, new threats, and new attacks are continually being created. The Plynt team periodically reviews the Plynt certification criteria to ensure the criteria address all known current threats. Details of the changes made to the Plynt Certification Criteria version 3.0 are listed below. We deleted a few criteria:

1. Version 2.0 criteria 15, Warning required for “Remember Me”
2. Version 2.0 criteria 16, Password not stored in plain text for “Remember Me”
3. Version 2.0 criteria 13, Session timed out after period of inactivity
4. Version 2.0 criteria 14, Re-authentication required after logout
5. Version 2.0 criteria 18, Random token to track authenticated sessions

Version 2.0 Criteria 15 & 16 have been merged into Version 3.0 Criteria 8 - **“Sensitive data not stored on client”**

With the criteria Password not stored in plain text for **“Remember Me”** we were trying to protect from storing the password in plain text on the client’s machine.

The **“Warning required for Remember Me”** is just a message to inform the user that his credentials will be stored on the computer he is accessing the application from. If the user chooses to ignore the warning and click OK, the application will not protect him. We felt it didn't really deserve a criteria on its own.

Version 2.0 Criteria 13, 14 & 18 have been merged into Version 3.0 Criteria 1 - **“Safe against popular attacks”**

When we are talking about strong session management, we are worried about our session being hijacked. Session hijacking is one of the popular attacks. So, session timeout, re-authentication required on logout as well as the necessity of a random token become a part of strong session management and are hence merged under **“Safe against popular attacks”**.

### **Following Criteria have been modified**

Version 2.0 Criteria 17 Old password required before changing password has been modified to Version 3.0 Criteria 12 Re-authentication required on sensitive activities

Application must re-authenticate the user before allowing the user to change the password.

Note: This criterion does not apply for the special case where the user has forgotten the password and resets the password using the application’s password recovery or **“Forgot Password”** feature.

Along with change password option there can be other critical operations where one needs to re-authenticate; reauthentication could mean entering the same password as login or a different password as in the case of banking transactions. To ensure that all such areas are also included, this criteria has been modified accordingly.

Version 2.0 Criteria 7 Protect configuration files and directory lists has been modified to Version 3.0 Criteria 7 Insecure configuration settings on servers accessible directly by users

Insecure configuration includes much more than the config files and the directories. It includes the Operating System, web servers and the application configuration files, hence renamed to Insecure configuration settings on servers accessible directly by users.

Version 2.0 Criteria 6 Protect Secret Questions from guessing attacks has been modified to Version 3.0 Criteria 6 Secure Forgot Password Implementation

A secret question/answer implementation is just a part of a complete Forgot Password implementation. Protecting just the secret question didn't make sense if the entire forgot password implementation itself was insecure. .



**Following Criteria have been Merged**

*Version 2.0 Criteria 20 - Services patched & Criteria 21 - Access to server filtered have been merged to Version 3.0 Criteria 14 Webservice protected against known vulnerabilities*

*The webservice must be protected against all known vulnerabilities. While it's most important that every remotely accessible service is securely implemented, the Plynt Certification concerns itself just with the components directly associated with the application; this includes the web-server. This is hence mandatory to get the Plynt Certification.*

*However since data used by the application can still be stolen through other misconfigured remote services, we strongly recommend you still fix all findings brought to your notice through the report. This however is not mandatory for the Plynt Certification as you might not always have complete control over these services; specially in the case of shared hosting.*