PLYNT SECURE APPLICATION CERTIFICATE

The Plynt Certification Program has tested and certified the following application.

Application Name : _____

Version Number : _____

The above application version has been tested and found to meet or exceed the Plynt Certification Criteria. During testing the above application has been subjected to logical and technical application security attacks and found to be resilient. The certification criteria is listed overleaf and the most current version is published online at http://www.plynt.com/criteria

Application Owner : _____

Test Conducted : _____

Certificate Issued : _____

_____
Roshen Chandran
Plynt Program Director

Certificate Validity: Limited to application version tested and limited by checks present in Certification Criteria on date issued.

(This is a sample of the Plynt Certificate. The actual certificate may not appear the same.)

The Plynt Certification Standard

Version 1.0, Effective Date: February 1, 2006

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 Plynt Certification Standard document defines the criteria used to evaluate an application for the Certificate. An application must demonstrate through testing that these security criteria are met before it is awarded the Plynt Certificate. The application is tested remotely to verify that it meets the Plynt criteria.

Plynt Certification Criteria

The Certification Standard is composed of 25 criteria. These are organized in two sections: Section 1, "Security Protection Criteria" identifies the defenses an application must demonstrate to get the Plynt Certificate. Section 2, "Security Requirements Criteria", specifies the features and behavior an application must have to get the Plynt Certificate.

Section 1: Security Protection Criteria

1. **Safe against popular attacks**: The application must demonstrate through testing that it is not vulnerable to popular attacks. Note i: "Popular attacks" include but are not limited to exploits documented in web sites like www.owasp.org, www.webappsec.org, www.osstm.org, etc.

2. **Defend against Threat Profile**: The application must demonstrate through testing that it defends against the threats specified in its threat profile. 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 with 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.

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 through testing that a remote adversary cannot steal user passwords from the application. Note i: The criterion requires that even after a user logs out, it must not be possible to steal the user's password. Note ii: The criterion recognizes that there are social engineering methods that could be used to steal passwords outside the application. They are not within the scope of this criterion.

5. **Protect against automated password guessing**: If the application uses passwords for authentication, it must protect against brute force password guessing attacks.

6. **Protect against manual 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.

7. **Protect secret questions from guessing attacks**: 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).

8. **Protect configuration files and directory lists**: The application must not reveal configuration files and directory listings to remote users. 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 or not. 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

9. **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 and persistent cookies on the client machine. Note ii: The browser memory is not considered an easily accessible location for this criterion.

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

11. **No sensitive data in error messages**: The application must not reveal sensitive information in error messages. Note i: Sensitive information includes not only business sensitive information, but also details regarding application architecture that could aid an adversary launch a successful attack against the application.

12. **Known, strong cryptographic algorithms**: Any cryptographic algorithm used on the client to protect sensitive data must be a publicly known algorithm that is secure for the intended use.

13. **Code obfuscation for secrets**: If Javascripts, Applets or ActiveX controls contain secrets, they must use strong code obfuscation techniques to protect the secrets. Note i: The secrets the above criterion refers to include cryptographic keys, passwords, and algorithms considered a trade secret.

14. **Session timed out after period of inactivity**: The user session must be timed out after an appropriately defined period of inactivity. Note i: The criterion requires that the application define and enforce a value for the minimum period of inactivity. It does not specify the minimum value. Note ii: The criterion requires that the inactivity period chosen to timeout a user be adequate to address the threat of an inactive session being hijacked by an adversary. The criterion recognizes that the minimum period could be different for different applications, based on their pattern of usage.

15. **Re-authentication required after log out**: Once the user has logged out, the application must require authentication before granting access to private pages. Note i: The criterion requires the application to invalidate the authenticated session at the server when the user logs out. If the application does not invalidate a user session at the server immediately on log out, it must take adequate protection against the session being reused.

16. **Warning required for "Remember Me"**: If the application provides a "Remember Me" feature, it must warn the user against enabling while using shared computers to access the application.

17. **Password not stored in plain text for "Remember Me"**: If the application provides a "Remember Me" feature, it must not store user passwords on the client machine in plain text or in a form that can be decrypted by an adversary.

18. **Old password required before changing password**: The application must re-authenticate the user before allowing the user to change the password. Note i: 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.

19. **Random session token**: Session token(s) must be random and difficult to predict. Note i: "Session token(s)", as used in this criterion, is the set of tokens that the application uses to track the state of the user session, regardless of whether the token is implemented as a HTTP Cookie, a URL query-string variable, a Hidden form field value, or a combination of any of these.

20. **New authentication token on log in**: The application must assign new, random authentication token(s) to a user when the user logs in. Note i: "Authentication token(s)", for the purpose of this criterion, is the set of tokens the application uses to track the authentication status of a session and the identity of the user of that session. Note ii: If the application uses session token(s) to play the role of the authentication token, this criterion requires that the session token take a new, random value when a user logs in.

21. **No sensitive data in requests to external sites**: If the application links to external sites, it must not disclose to the external site any more sensitive data than is required by the external site. Note i: Sensitive data, as used in this criterion, includes business sensitive information, as well as session token(s) and authentication token(s).

22. **Services patched**: The services exposed by the server must not be vulnerable to publicly known, remotely exploitable bugs. "Exposed", as used in this criterion, refers to services accessible remotely.

23. **Access to server filtered**: The server must be protected by a filter that allows access only to ports required for the use and administration of the server. Note i: The criterion does not specify the ports that must be blocked or allowed; the specific ports may vary with applications. The criterion however requires that only those required by remote users and administrators be allowed by the filter.

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

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

Disclaimer

Plynt Certificate does not certify that the concerned Client Product is completely secure or free from all security vulnerabilities/holes and that there will not be any security breaches with respect to any such certified Client Product. The Plynt Certificate merely evidences that such Client Product has passed various universally recognized security checks, which are applied by Plynt during the program.

(This is a sample of the Plynt Certificate. The actual certificate may not appear the same.)