Security Issues with HTML5 Web Local Storage and Co-Hosting

Slides: http://tinyurl.com/lsscpl13

Available via GitHub at:

https://github.com/RichardRoda/CodePaLOUsa/tree/master/2013/LocalStorageSecurity

Speaker Introduction

- Richard Roda’s linked in profile:

  http://www.linkedin.com/in/richardroda
- Over 15 years of IT experience.
- Sr. Software Engineer for Hewlett Packard for the Army at Ft. Knox
- Lead developer for common security framework used by multiple Java applications.
- Headquarters Support Structure application Technical Lead.
- Certifications: Security+, ITILv3 Foundation
- BA Business with minor in Computer Science from Warren Wilson College

What is Local Storage?

- It is a way for web applications to store and use data in the browser.
- Allows data manipulation using browser side mobile code such as JavaScript.
- Differs from cookies:
  - Not sent to server with each request
  - Cookies are primarily a product of http responses. HTML5 local storage is manipulated by browser side code.

Typical Local Storage Uses

- Example: Email Application
  - Offline Reading
  - Offline Composition
  - Offline Organization
- Example: Caching
  - A cached page may use local storage to display dynamic or per-user content.
  - Eliminates the need to re-transmit content for each request.

The Evercookie

- An “Evercookie” is browser side data that is difficult to remove or block from a browser.
- Its JavaScript implementation uses various storage mechanisms to id and track visitors.
- Localstorage is one of many mechanisms used.
- Disabling Localstorage won’t stop evercookie.
- Clearing everything every time might stop it.
- Source ”Evercookie – Never Forget” http://samy.pl/evercookie/, pulled April 19, 2013

How is Local Storage Bound?

- “The localStorage object provides a Storage object for an origin.”… “If the Document's origin is not a scheme/host/port tuple, then throw a SecurityError exception and abort these steps.” (source: http://www.w3.org/TR/webstorage)
- http://www.w3.org/TR/webstorage is a http/www.w3.org/80 tuple (port 80 is implied)
- Everything on http/www.w3.org uses the same local storage object.

What Do We Mean By Security?

- Security is commonly defined by the CIA triad as Confidentiality, Integrity, and Availability.
  - **Confidentiality** – Only authorized users may access data and information.
  - **Integrity** – Only authorized users may correctly change data. Damage is reversible.
  - **Accessibility** – System and all authorized functions are available for authorized users.
- Source, “The CIA Triad”, http://www.techrepublic.com/blog/security/the-cia-triad/488, pulled April 13, 2013.

Security Implications of Local Storage Applications Sharing an Origin

- An application may display data belonging to another application for which the user is not authorized, violating **Confidentiality**.
- An application may alter data belonging to another application, violating **Integrity**.
- An application that has its data altered may crash, violating **Accessibility**.

Disclaimer: These demonstrations show the effect that changes in an origin can have on the security of local storage and Single Sign On. They do not show overall good security practice.

Demo: Single to Co-Hosting

This demo uses a virtual host configured with completely separate host names, and another virtual host configured for hostname “myappserver.mydomain.com.”

How Is This Different from AJAX?

- JavaScript document and window objects are inherently a page (request) bound entity.
- Likewise, JavaScript variable and function declarations within a page are bound to it.
- The data consumed by client side AJAX originates and is managed by the server. Any requests/updates are routed by the server to the correct application.
- Application servers manage cookies and are able to distinguish session identifiers.

Application Co-hosting Good Practice

- Each application* should have its own origin.
- A separate network address for each application is not required, merely a different name.
- DNS aliasing may be used to provide each application with its origin by making the <u>host</u> part of the scheme/<u>host</u>/port tuple unique.
- A DNS wildcard (e.g. *.appserver.domain.com) should be used. It minimizes DNS administration and allows remediation of server session issues.

*Or groups of applications designed to share data using local storage

Demo: Co-Hosting with subdomains

This demonstration uses a virtual host configured to handle requests for hosts within the domain “myappserver.mydomain.com”

Server Session Implications

- By default, the browser binds cookies using the same protocol, host, port tuple as Local Storage.
- Most application servers use cookies to bind sessions, with SSO (Single Sign On) being a form of shared session.
- Changing applications on a server to use separate host name may effectively unbind any shared session state.

Server Session Remediation

- When using the DNS wildcarding strategy for co-hosting, session state may be shared with a domain wide cookie.
- A web browser delivers domain wide cookies to all hosts or subdomains of the cookie.
- Domain would be part of DNS name that is not wild carded.
  - Example: *.appserver.mydomain.com would have a domain of appserver.mydomain.com set.

Demo: Subdomain co-hosting with domain SSO cookie.

This demo uses a virtual host configured with the domain fixedappserver.mydomain.com. This host creates a shared SSO cookie within the domain.

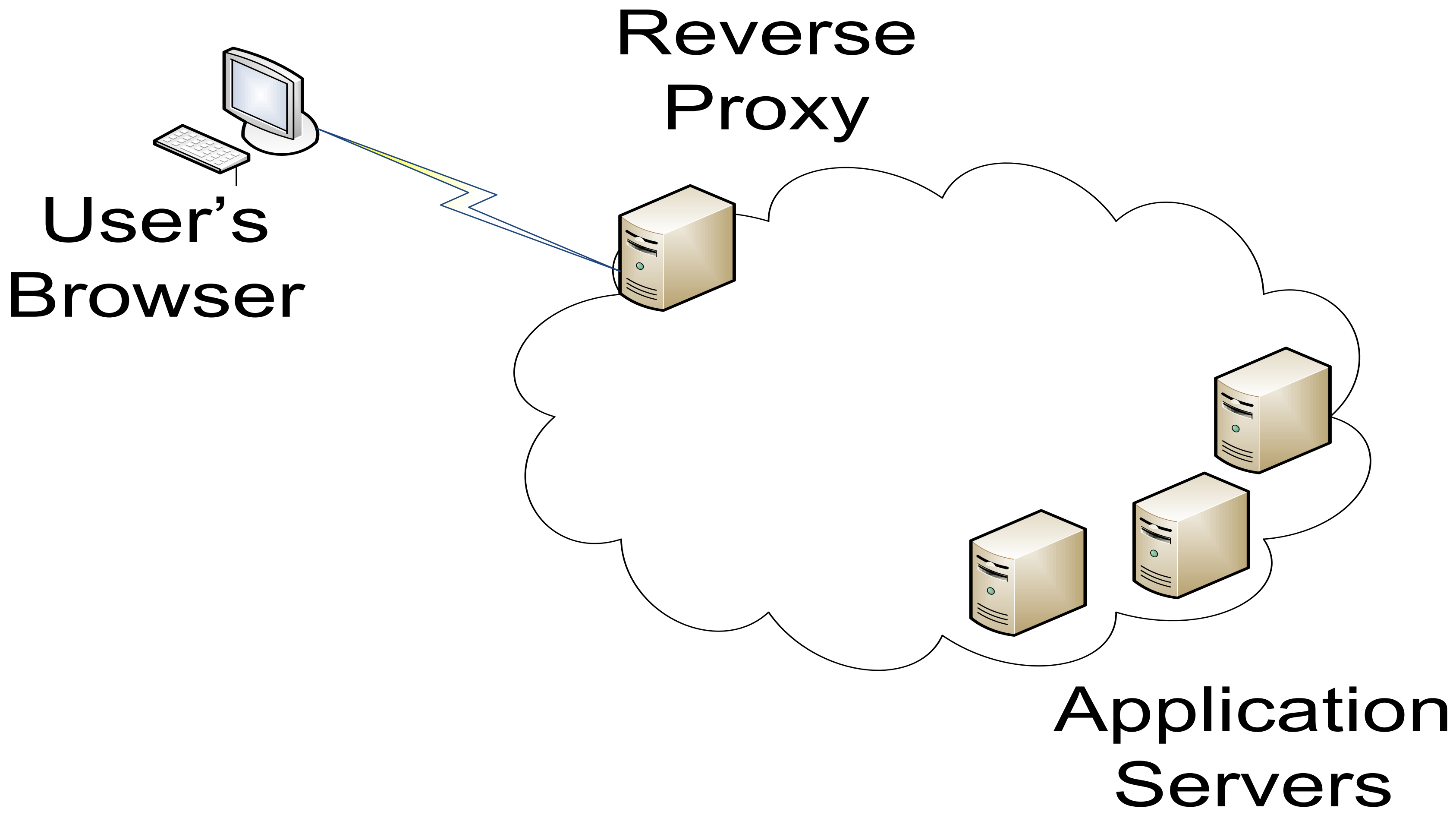
URL Rewriting Proxies

Or, we’re not out of the forest yet

Reverse Proxy Servers

- A technique used to provide protected access to protected resources over an untrusted network using the https scheme.
- Primary advantage: It works with a standard web browser. No need to deploy software.
- Because they rewrite the URL to go through the proxy, they can map everything to a single origin. This effectively “flattens” the origin host namespace.

Reverse Proxy Configuration



Reverse Proxy Solution

- Set up virtual hosts on the reverse proxy for each application, or a virtual host that allows for wildcard host names.
- Set up Single Sign On (SSO) for all of the reverse proxy virtual hosts.
- Set up a portal for the reverse proxy, or set up rules on the reverse proxy to redirect the browser to the correct host based on the application context requested.

Summary

- HTML5 web local storage is a useful technology that has security implications.
- Applications from the same origin share local storage, which can violate Confidentiality, Integrity, and Accessibility (CIA triad).
- Avoid these risks by giving each application a unique host name.
- Avoid server session issues by placing the application names in a common domain with a domain wide cookie.

Questions?

GitHub Location:

https://github.com/RichardRoda/CodePaLOUsa/tree/master/2013/LocalStorageSecurity