



# Security Issues with HTML5 Web Local Storage and Co- Hosting

Slides: <https://bit.ly/1Xovm2x>

Source Available via GitHub at:  
<https://github.com/RichardRoda/AFCEALocalStorageSecurity>



# 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 Enterprise
- Headquarters Support System application Technical Lead.
- VP Scholarships for AFCEA Gold Vault Chapter
- Certifications: Security+, ITILv3 Foundation
- BA from Warren Wilson College



# Security Requirements Change

- In the early 1970s analog phone system was controlled using tones.
- A tone of 2600Hz requested a trunk line. Trunk lines had operator privileges.
- The only known instrument that could reproduce the tone was an organ. Sound reproduction equipment was not portable
- What could go wrong...?



# The Portable 2600hz Instrument







# 2015 Cost of Lax Security

- OPM – 20 million people, including highly sensitive SF-86 applications for classified access.
- Ashley Madison – 11 million people, including payment details
- Anthem – 11 million highly sensitive healthcare records.
- And Many more...



# Breaches Result in Loss Of

- Trust
- Loyalty
- Business
- Brand Value
- Money
- Time



# Developer Reviews Fortify Findings



June 2009





# Damocles Discusses False Positives with IA







# Penitent Seeks Forgiveness for Exploited Vulnerability



June 2009



# What is Local Storage?

- It is a way for web based applications to store and use data in the browser.
- Allows data manipulation using browser side mobile code such as JavaScript.
- Does not rely on cookies:
  - All data is stored in the browser, not the server.
  - Assumes browser is inherently bound to the user. Not for public computers.



# 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.





# 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)



# 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

- 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**.





# 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.”



# 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 host part of the scheme/host/port tuple unique.
- A DNS wildcard (e.g. \*.appserver.domain.com) may be used. It minimizes DNS administration and helps support Single Sign On.

\*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.



# 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



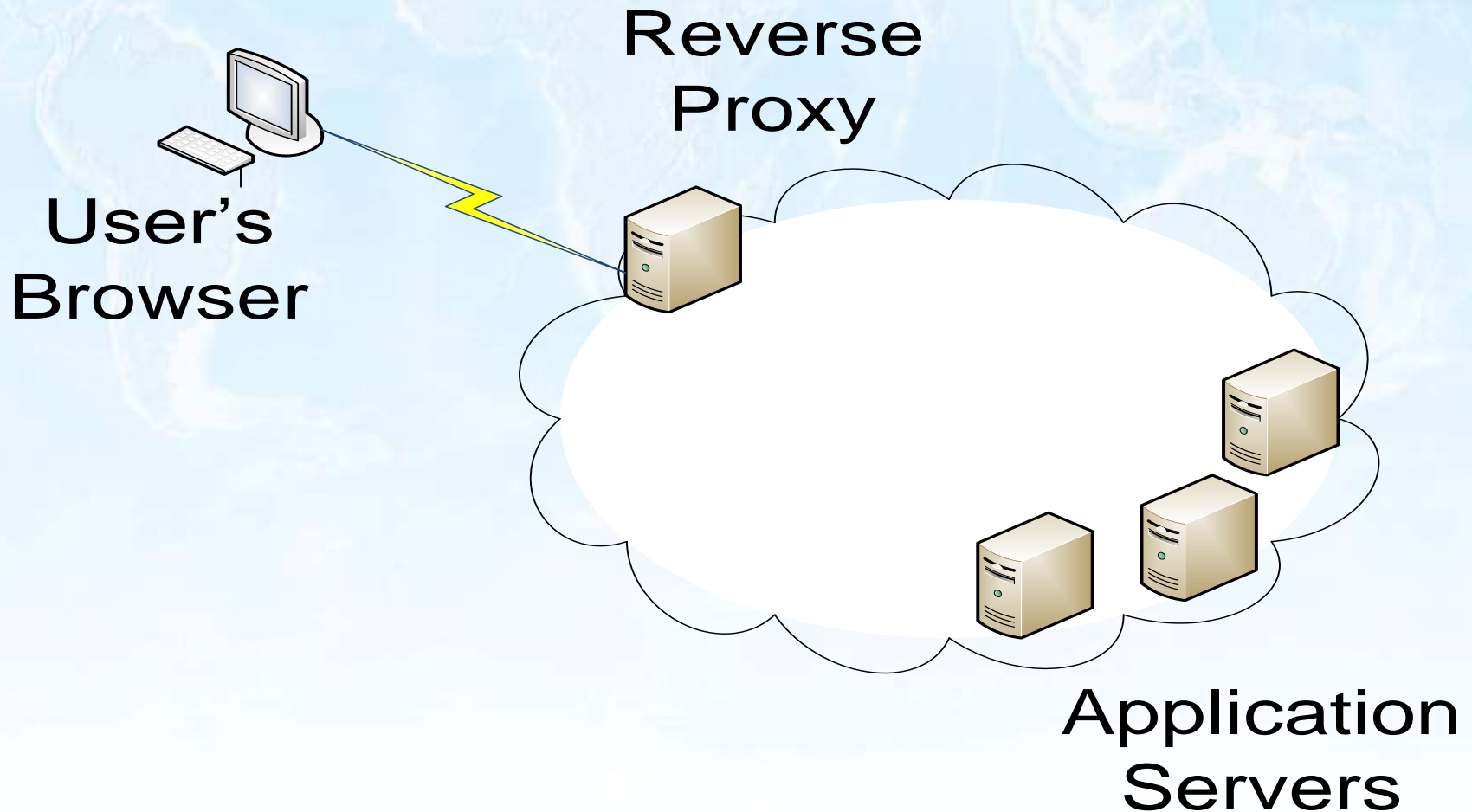


# Reverse Proxy Servers

- 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.



# LocalStorage Data Remains Vulnerable

- DNS cache poisoning can direct to an impostor site that exfiltrates the data (similar to the co-hosting example).
- A browser vulnerability may allow an escalation of privilege attack or code to escape the browser “sandbox”





# Protect Sensitive Data

- Such data should only come from an HTTPS page.
- Encrypt the data and include a MAC code with it.
- Message authentication codes (MAC) should be used instead of the keys.
- The keys deserve protection because they can reveal clues about the data.



# Co-Hosting Demo with Encryption

- This demonstration will repeat the co-hosting demo, but with encryption and MAC codes.
- The use of encryption limits the damage that a compromise may cause.
- Two attacks remain: A deletion or corruption attack, and a replay attack.
- These attacks must be mitigated (taken into consideration) when the app is designed.



# Encryption Benefits

- Confidentiality – Strong. Only brute force can decrypt data.
- Integrity – Weak. Data may not be altered, but it may be replayed or deleted. Such attacks must be remediated by application design.
- Accessibility – None. Is erased using `localStorage.clear()`.



# 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, and encrypt any sensitive data.





# “This Will Never Be Exploited”



7/17/2016



# Questions?

Slides: <https://bit.ly/1Xovm2x>

GitHub:

<https://github.com/RichardRoda/AFCEALocalStorageSecurity>