Web Security: Browsers

CS 161: Computer Security

Prof. David Wagner

February 19, 2013

Announcements

- Midterm 1: in class, next Monday, here
- Midterm review session:
  Saturday 2/22, 2-4pm, 100 GPB
- Project 1 is now out; due Monday 3/3
- HW1 solutions are posted
- No discussion sections next week

Goals For Today

- Web security challenges that are specific to web browsers
  - Quick reminder: web “driveby” attacks
  - Social engineering users: Clickjacking
- Server-side solutions cannot fix these problems

Dynamic Web Pages

- Rather than static HTML, web pages can be expressed as a program, say written in *Javascript*:

```
<title>Javascript demo page</title>

<font size=30>
Hello, <b>
<script>
var a = 1;
var b = 2;
document.write("world: ", a+b, "</b>");
</script>
```

Threats?

Or what else?
Java, Flash,
Active-X, PDF …

Drive-By Downloads

55846 : Mozilla Firefox Just-in-time (JIT) JavaScript Compiler js/src/jstracer.cpp font HTML Tag Handling Memory Corruption

Printer | http://osvdb.org/55846 | Email This | **Edit Vulnerability**

Views This Week	Views All Time	Added to OSVDB	Last Modified	Modified (since 2008)	Percent Complete
6	571	about 1 year ago	about 1 month ago	24 times	90%

generously sponsored by TENABLE Network Security

Timeline

Disclosure Date	Exploit Publish Date	Vendor Solution Date
2009-07-13	2009-07-13	2009-07-16

Days of Exposure
3 days

Keywords

6868125, 6861719

Description

A memory corruption flaw exists in Firefox. The Just-in-Time (JIT) compiler can enter a corrupt state following native function calls resulting in memory corruption. With a specially crafted request, an attacker can cause arbitrary code execution resulting in a loss of integrity.

Classification

Location: Remote / Network Access, Context Dependent
Attack Type: Input Manipulation
Impact: Loss of Integrity
Solution: Workaround, Upgrade
Exploit: Exploit Public, Exploit Commercial
Disclosure: Vendor Verified, Uncoordinated Disclosure, Discovered in the Wild
OSVDB: Web Related

Solution

Upgrade to version 3.5.1 or higher, as it has been reported to fix this vulnerability. It is also possible to correct the flaw by implementing the following workaround: disable JavaScript.

Drive-By download = attack that infects your system just by you visiting a (malicious) web page. Your are now 0wnd!

PUBLIC ADVISORY: 02.22.07

VeriSign ConfigChk ActiveX Control Buffer Overflow Vulnerability

I. BACKGROUND

The ConfigChk ActiveX Control is part of VeriSign Inc.'s MPKI, Secure Messaging for Microsoft Exchange and Go Secure! products. It looks for the Microsoft Enhanced Cryptographic Provider in order to support 1024-bit cryptography.

II. DESCRIPTION

Remote exploitation of a buffer overflow vulnerability in VeriSign Inc.'s ConfigChk ActiveX Control could allow an attacker to execute arbitrary code within the security context of the victim.

The ActiveX control in question, identified by CLSID 08F04139-8DFC-11D2-80E9-006008B066EE, is marked as being safe for scripting.

The vulnerability specifically exists when processing lengthy parameters passed to the VerCompare() method. If either of the two parameters passed to this method are longer than 28 bytes, stack memory corruption will occur. This amounts to a trivially exploitable stack-based buffer overflow.

III. ANALYSIS

Successful exploitation of this vulnerability would allow a remote attacker to execute arbitrary code within the context of the victim.

In order to exploit this vulnerability, an attacker would need to persuade the victim into viewing a malicious web site. This is usually accomplished by getting the victim into clicking a link in a form of electronic communication such as e-mail or instant messaging.

About the security content of Java for Mac OS X 10.6 Update 2

Last Modified: May 18, 2010

Java for Mac OS X 10.6 Update 2

- **Java**

CVE-ID: CVE-2009-1105, CVE-2009-3555, CVE-2009-3910, CVE-2010-0082, CVE-2010-0084, CVE-2010-0085, CVE-2010-0087, CVE-2010-0088, CVE-2010-0089, CVE-2010-0090, CVE-2010-0091, CVE-2010-0092, CVE-2010-0093, CVE-2010-0094, CVE-2010-0095, CVE-2010-0837, CVE-2010-0838, CVE-2010-0840, CVE-2010-0841, CVE-2010-0842, CVE-2010-0843, CVE-2010-0844, CVE-2010-0846, CVE-2010-0847, CVE-2010-0848, CVE-2010-0849, CVE-2010-0886, CVE-2010-0887

Available for: Mac OS X v10.6.3, Mac OS X Server v10.6.3

Impact: Multiple vulnerabilities in Java 1.6.0_17

Description: Multiple vulnerabilities exist in Java 1.6.0_17, the most serious of which may allow an untrusted Java applet to execute arbitrary code outside the Java sandbox. Visiting a web page containing a maliciously crafted untrusted Java applet may lead to arbitrary code execution with the privileges of the current user. These issues are addressed by updating to Java version 1.6.0_20. Further information is available via the Sun Java website at http://java.sun.com/javase/6/webnotes/ReleaseNotes.html

Category: Application (Web Browser) > Opera

Vendors: Opera Software

Opera JPEG DHT Marker Buffer Overflow and createSVGTransformFromMatrix Request Validation Flaw Lets Remote Users Execute Arbitrary Code

SecurityTracker Alert ID: 1017473
SecurityTracker URL: http://securitytracker.com/id/1017473
CVE Reference: CVE-2007-0126, CVE-2007-0127 *(Links to External Site)*
Updated: May 20 2008
Original Entry Date: Jan 5 2007
Impact: Execution of arbitrary code via network, User access via network
Fix Available: Yes **Vendor Confirmed:** Yes
Version(s): prior to 9.10
Description: Two vulnerabilities were reported in Opera. A remote user can cause arbitrary code to be executed on the target user's system.

A remote user can create a specially crafted JPEG image that, when loaded by the target user, will trigger a heap overflow and execute arbitrary code on the target system. The code will run with the privileges of the target user.

A specially crafted JPEG DHT marker can trigger the flaw.

Christoph Diehl reported this vulnerability to iDefense.

A remote user can create Javascript with a specially crafted createSVGTransformFromMatrix request parameter that, when processed by the target user, will execute arbitrary code on the target system. The code will run with the privileges of the target user.

MS-ISAC Multi-State Information Sharing and Analysis Center

Alert Level: GUARDED

Home / Cyber Advisories

MS-ISAC ADVISORY NUMBER:

2009-008

DATE(S) ISSUED:

2/20/2009

SUBJECT:

Vulnerability in Adobe Reader and Adobe Acrobat Could Allow Remote Code Execution

OVERVIEW:

A new vulnerability has been discovered in the Adobe Acrobat and Adobe Reader applications that allows attackers to execute arbitrary code on the affected systems. Adobe Reader allows users to view Portable Document Format (PDF) files. Adobe Acrobat offers users additional features such as the ability to create PDF files.

Depending on the privileges associated with the user, an attacker could then install programs; view, change, or delete data; or create new accounts with full user rights. Unsuccessful exploitation attempts may cause these programs to crash.

It should be noted that this vulnerability is being actively exploited on the Internet.

Vulnerability Notes Database

Search Vulnerability Notes

Vulnerability Notes Help Information

Report a Vulnerability

View Notes By
Name

ID Number

CVE Name

Date Public

Date Published

Date Updated

Severity Metric

Vulnerability Note VU#593409

Adobe Reader and Acrobat util.printf() JavaScript function stack buffer overflow

Overview

Adobe Reader and Acrobat contain a stack buffer overflow in the `util.printf()` JavaScript function, which may allow a remote, unauthenticated attacker to execute arbitrary code on a vulnerable system.

I. Description

Adobe Reader is software designed to view Portable Document Format (PDF) files. Adobe Acrobat is software that can create PDF files. Adobe Reader and Acrobat support JavaScript in PDF documents. According to the Acrobat Forms JavaScript Object Specification, the `util.printf()` function "... *will format one or more values as a string according to a format string. This is similar to the C function of the same name.*"

Adobe Reader and Acrobat fail to sufficiently validate input to the `util.printf()` JavaScript function, which can result in a stack buffer overflow. Exploit code for this vulnerability is publicly available.

II. Impact

By convincing a user to open a specially-crafted PDF file, a remote, unauthenticated attacker may be

Adobe - Flash Player

http://www.adobe.com/software/flash/about/

Home / Products / Flash Player /

Adobe Flash Player

Adobe Flash Player is the standard for delivering high-impact, rich Web content. Designs, animation, and application user interfaces are deployed immediately across all browsers and platforms, attracting and engaging users with a rich Web experience.

Version Information

You have version 11,5,502,149 installed

The table below contains the latest Flash Player version information. Adobe recommends that all Flash Player users upgrade to the most recent version of the player through the Player Download Center to take advantage of security updates.

Platform	Browser	Player version
Windows	Internet Explorer (and other browsers that support Internet Explorer ActiveX controls and plug-ins)	11.6.602.171
	Internet Explorer (Windows 8)	11.6.602.171
	Firefox, Mozilla, Netscape, Opera (and other plugin-based browsers)	11.6.602.171
	Chrome (Pepper-based Flash Player)	11.6.602.171
Macintosh OS X	Firefox, Opera, Safari	11.6.602.171
	Chrome (Pepper-based Flash Player)	11.6.602.171
Linux	Mozilla, Firefox, SeaMonkey (Flash Player 11.2 is the last supported Flash Player version for Linux. Adobe will continue to provide security updates.)	11.2.202.273

COMPUTERWORLD

News

Adobe springs emergency Flash update, says hackers hitting Firefox

Second 'out-of-band' patch this month, fourth fix overall in 2013

By Gregg Keizer

February 26, 2013 04:11 PM ET 3 Comments

Computerworld - Adobe today patched new vulnerabilities in Flash Player that hackers are now exploiting in attacks aimed at Firefox users, the company said.

Today's surprise update to Flash Player was the second emergency fix this month, the third overall for February, and the fourth since the start of 2013.

Defenses Against Driveby Attacks

- Sandboxing: rich content (PDF, Flash, …) runs in a *constrained environment*
  - Implements Least Privilege
- Disable unneeded functionality
  - Excessive featurism kills!
  - But not always practical
- Patching / autoupdate
  - Still a race, and can be disruptive
- Control exposure to untrusted sites
  - E.g., *Google Safe Browsing*: dynamically updated list of malware & phishing sites
  - Browser warns on any access …

Misleading Users

- Browser assumes clicks & keystrokes = *clear indication of what the user wants to do*
  - Constitutes part of the user's *trusted path*
- Attacker can meddle with integrity of this relationship in all sorts of ways …


Mozilla Firefox
www.attacker.com
Most Visited
Getting Started
Latest Headlines
(Untitled)
BEST GAME EVER!
PLAY!
twitter
Is this goodbye?
This action is permanent.
Before you deactivate your account, know this:
Okay, fine, deactivate my account
© 2010 Twitter

Stealing Keystrokes (demo)

Misleading Users

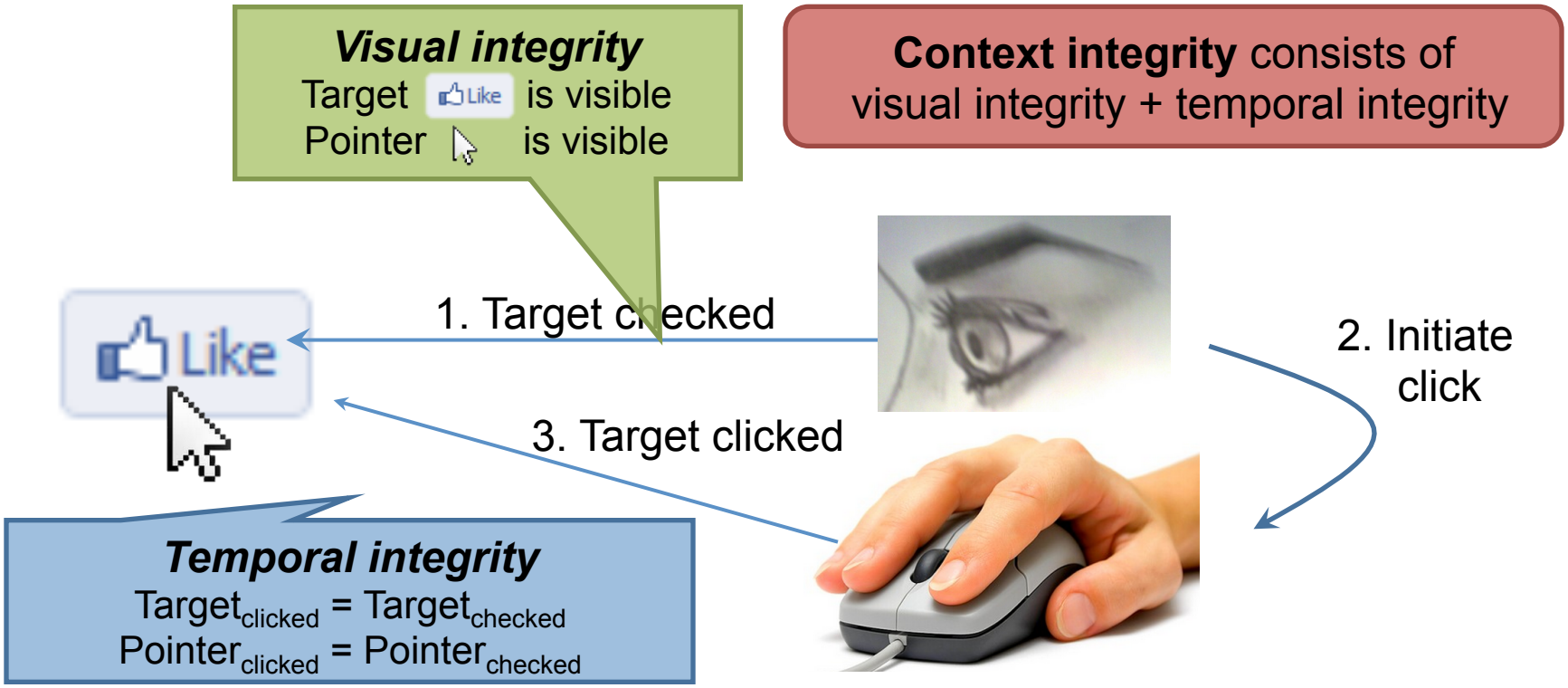
- Browser assumes clicks & keystrokes = *clear indication of what the user wants to do*
  - Constitutes part of the user's *trusted path*
- Attacker can meddle with integrity of this relationship in all sorts of ways …
- Especially, recall the power of Javascript!
  - Alter page contents *(dynamically)*
  - Track events (mouse clicks, motion, keystrokes)
  - Read/set cookies
  - Issue web requests, read replies

Using JS to Steal Facebook *Likes*

- *Bait-and-switch*
- Note: many of these attacks are similar to *TOCTTOU* (Time of Check to Time of Use) vulnerabilities

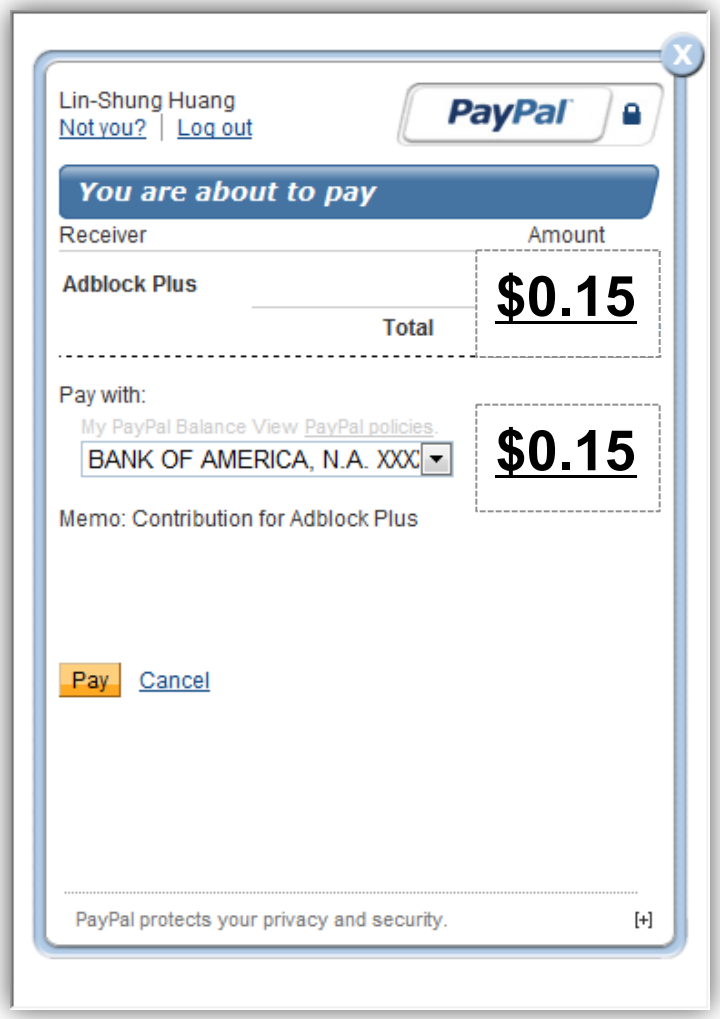
UI Subversion: *Clickjacking*

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



Compromise visual integrity – target

- Hiding the target
- Partial overlays



Compromise visual integrity – pointer

- Manipulating cursor feedback

Clickjacking to Access the User's Webcam

Some Clickjacking Defenses

- Require confirmation for actions (annoys users)
- *Frame-busting:* Web site ensures that its "vulnerable" pages can't be included as a frame inside another browser frame
  - So user can't be looking at it with something invisible overlaid on top …
  - … nor have the site invisible above something else

Attacker implements this attack by placing Twitter's page in a "Frame" inside their own page. Otherwise the two pages wouldn't overlap.

Some Clickjacking Defenses

- Require confirmation for actions (annoys users)
- *Frame-busting:* Web site ensures that its "vulnerable" pages can't be included as a frame inside another browser frame
  - So user can't be looking at it with something invisible overlaid on top …
  - … nor have the site invisible above something else
- Conceptually implemented with Javascript like:

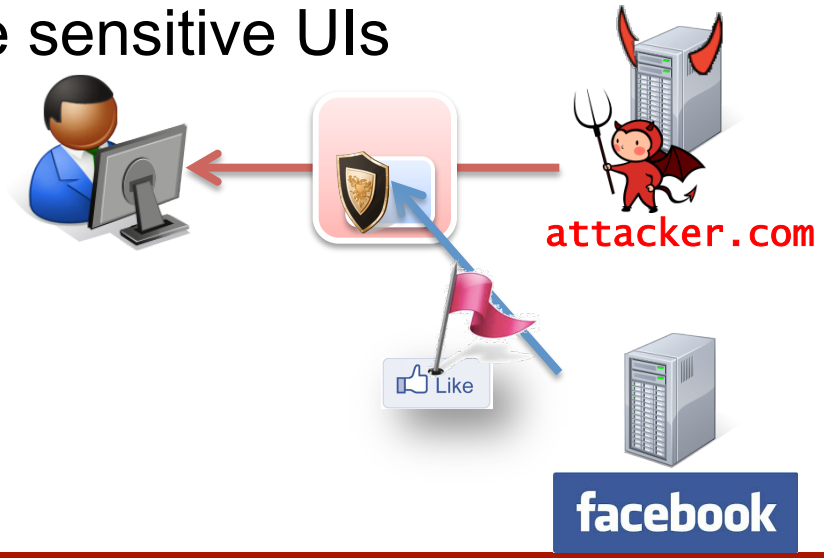
```
if (top.location != self.location)
      top.location = self.location;
```

(Note: actually quite tricky to get this right!)

- Current research considers more general approach …

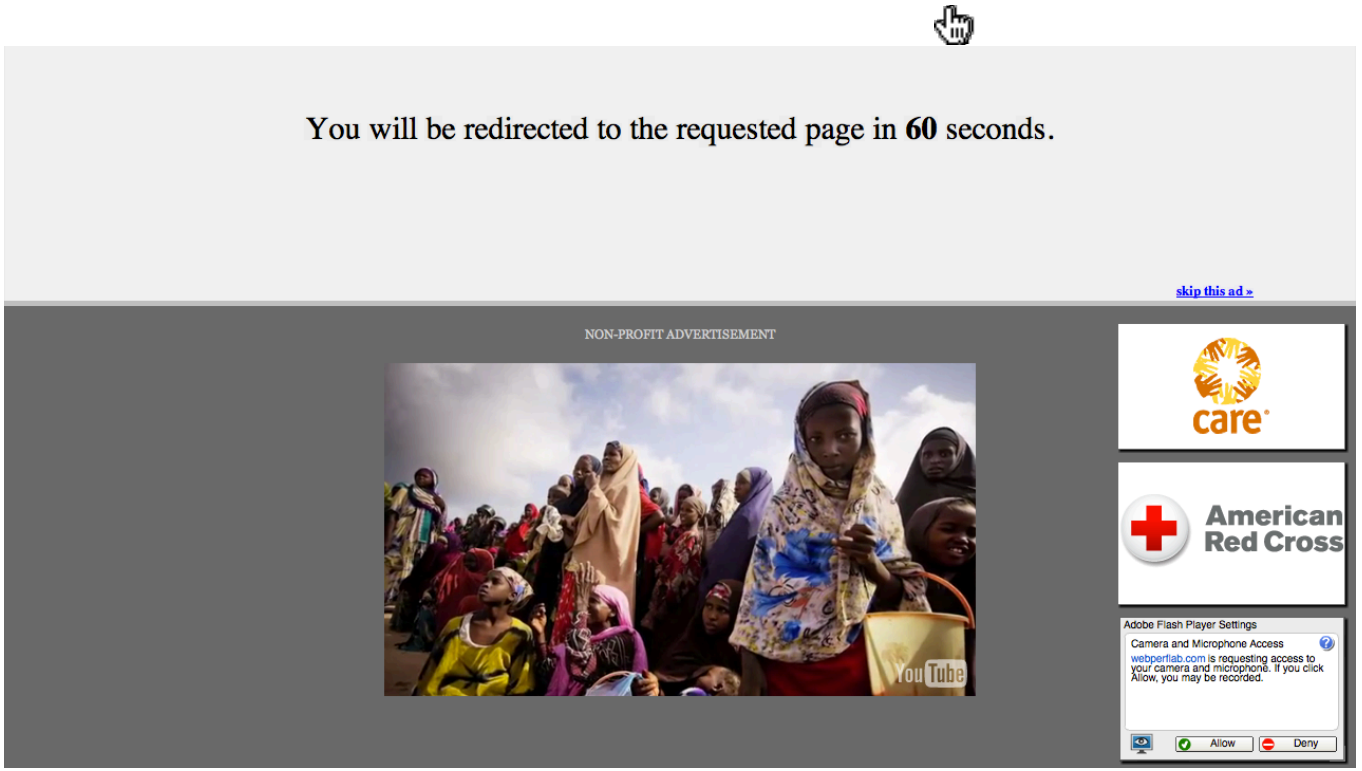
InContext Defense (Research)

- A set of techniques to ensure *context integrity* for user actions
- Server opt-in approach
  - Let websites *indicate* their sensitive UIs
  - Let browsers *enforce* context integrity when users act on the sensitive UIs



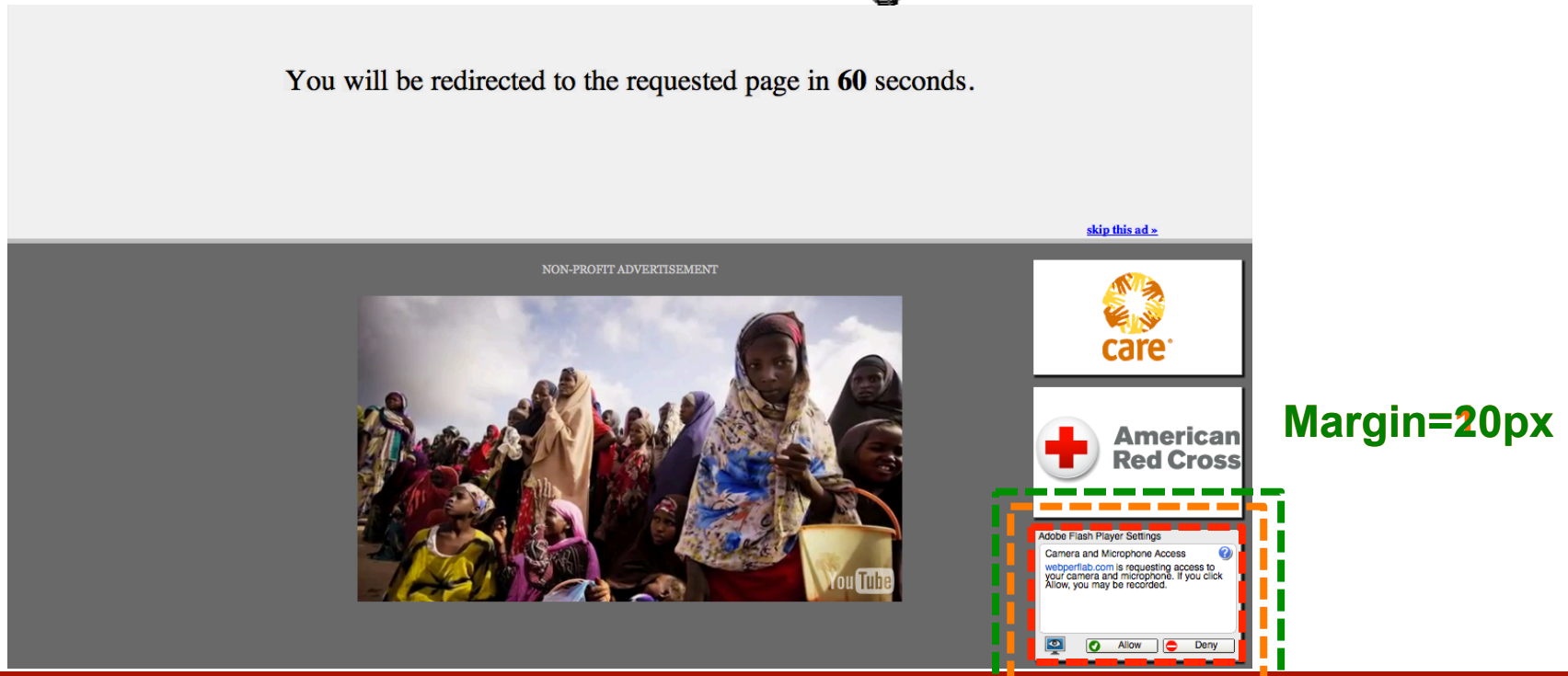
Ensuring visual integrity of pointer

- Remove cursor customization
  - Attack success: 43% -> 16%



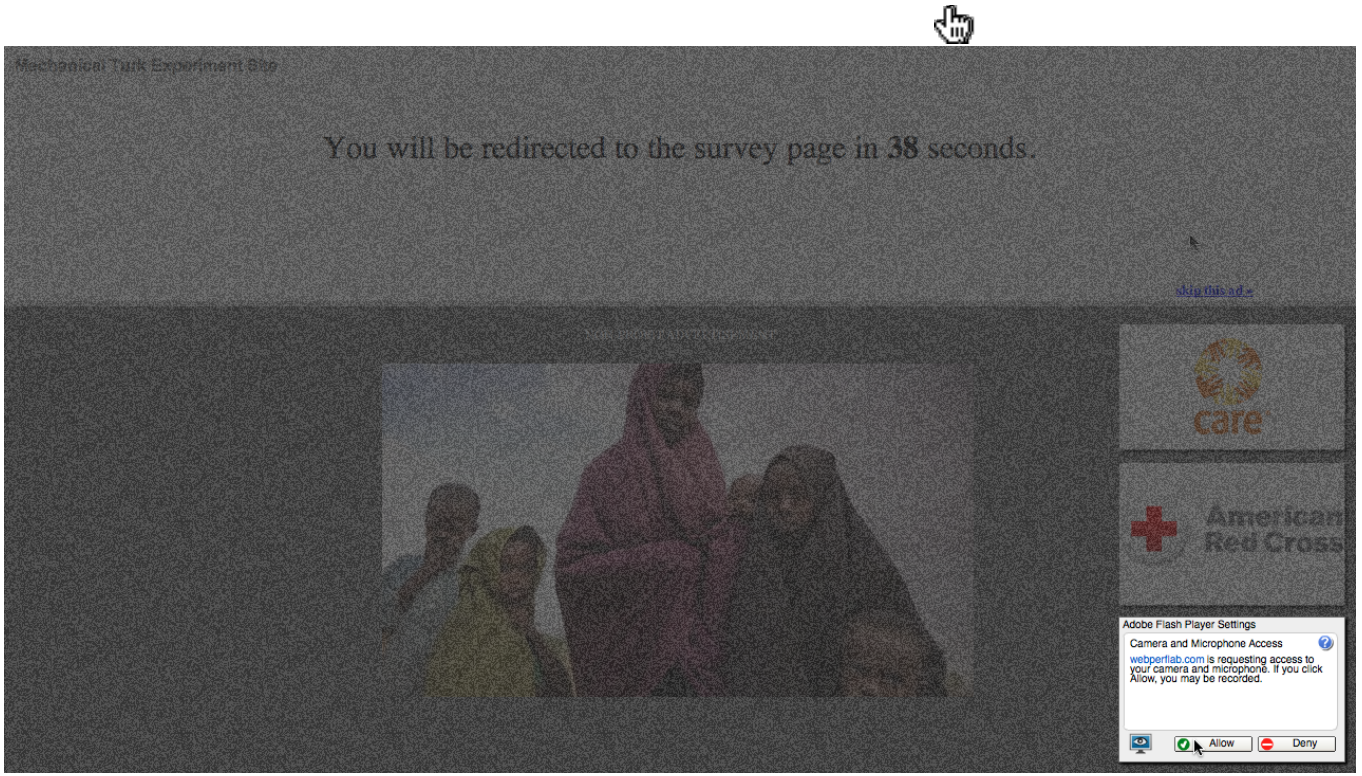
Ensuring visual integrity of pointer

- Freeze screen around target on pointer entry
  - Attack success: 43% -> 15%
  - Attack success (margin=10px): 12%
  - Attack success (margin=20px): 4% (baseline:5%)



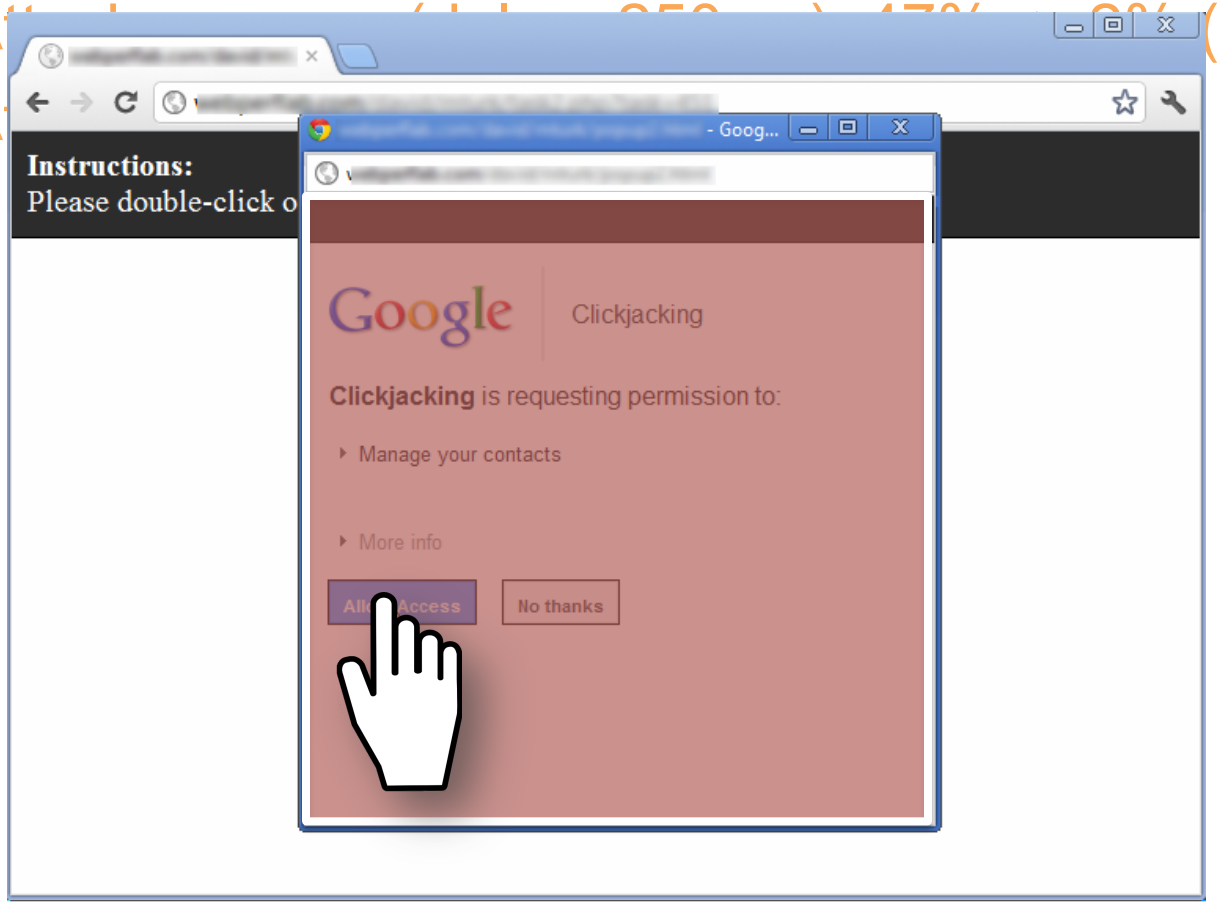
Ensuring visual integrity of pointer

- Lightbox effect around target on pointer entry
  - Attack success (Freezing + lightbox): 2%



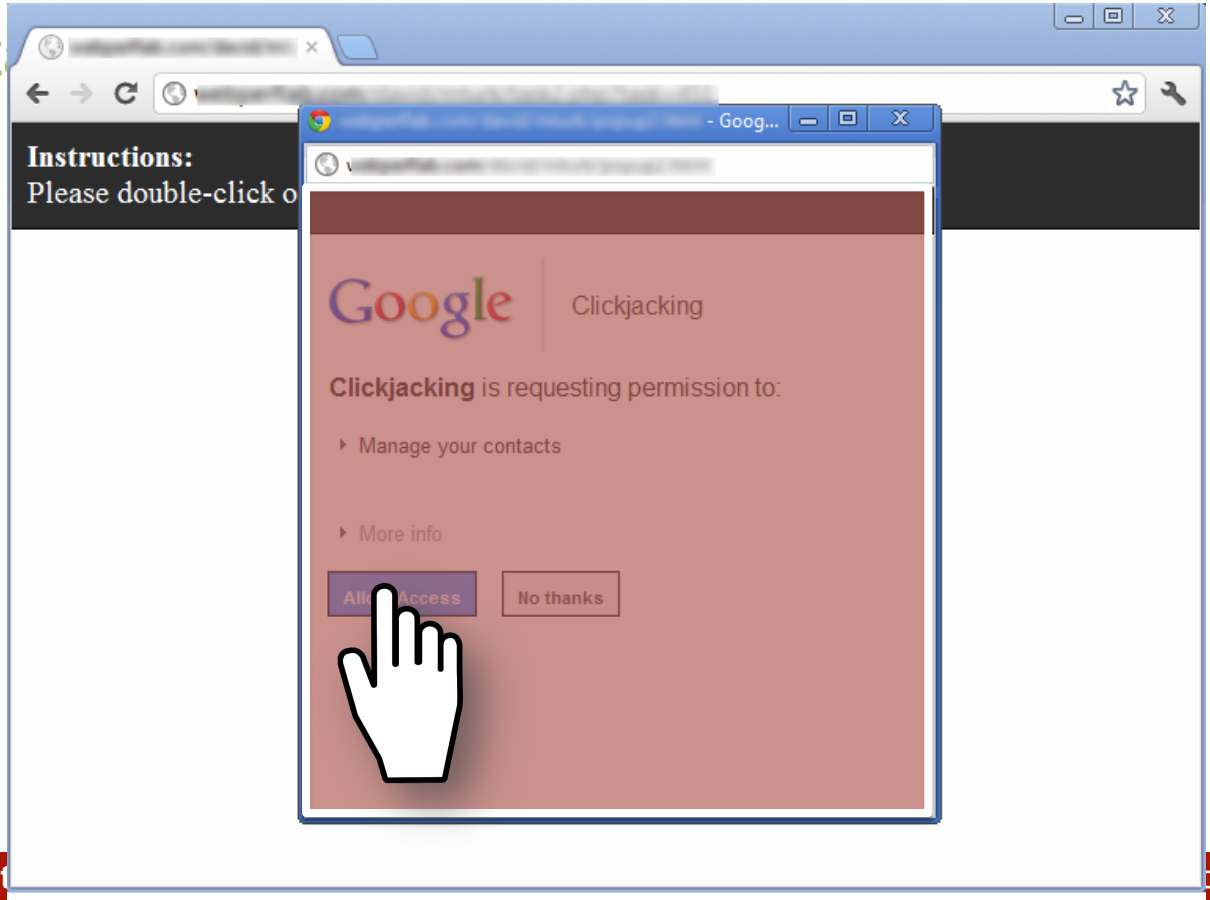
Enforcing temporal integrity

- UI delay: after visual changes on target or pointer, invalidate clicks for X ms
  - A (2/91)
  - A



Enforcing temporal integrity

- Pointer re-entry: after visual changes on target, invalidate clicks until pointer re-enters target
  - Att

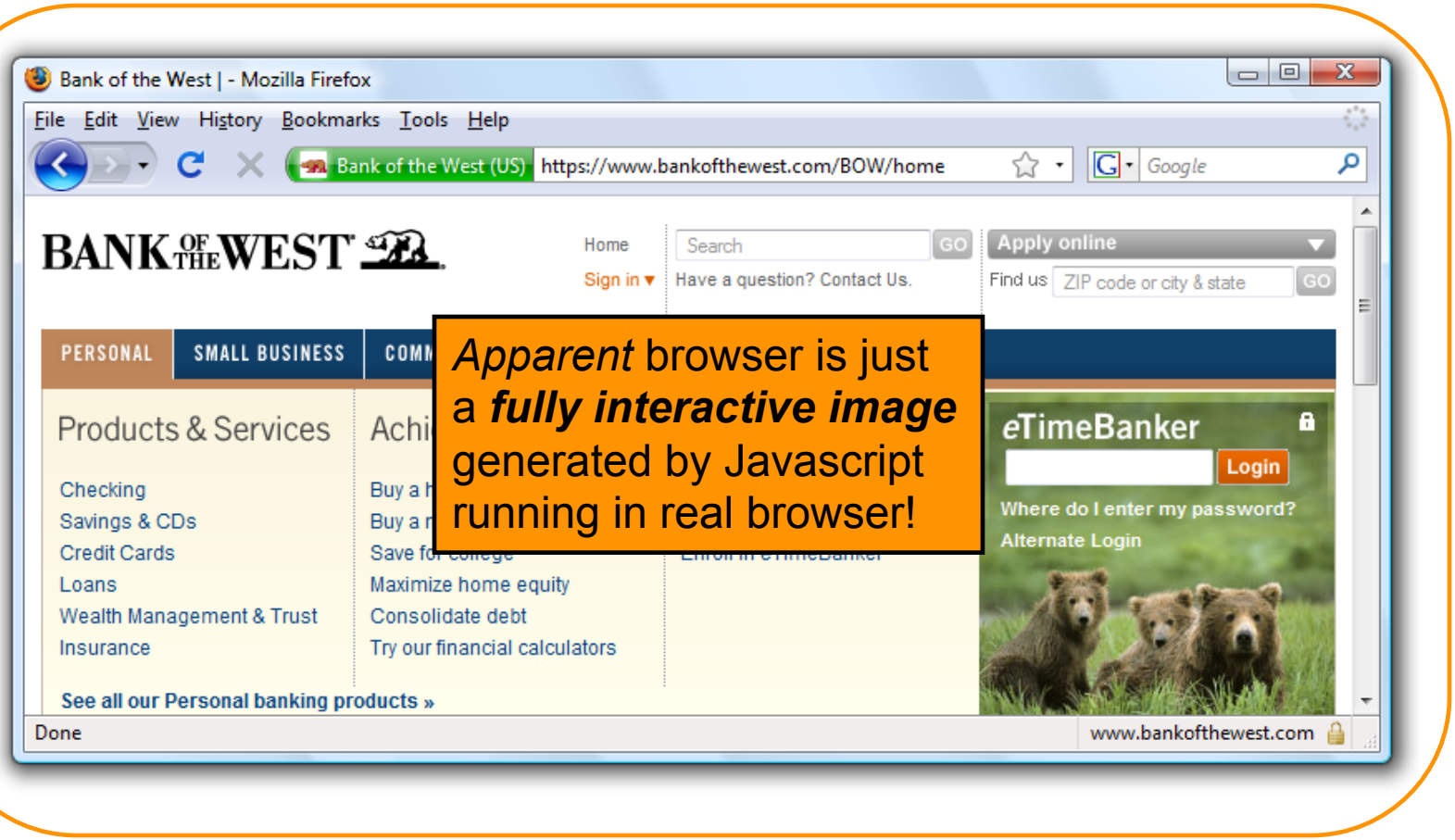


Other Forms of UI Sneakiness

- Along with stealing events, attackers can use power of Javascript customization / dynamic changes to mess with the user's mind …

- For example, the user may not be paying sufficient attention ...
  - *Tabnabbing*
- Or they might find themselves living in *The Matrix …*

“Browser in Browser”

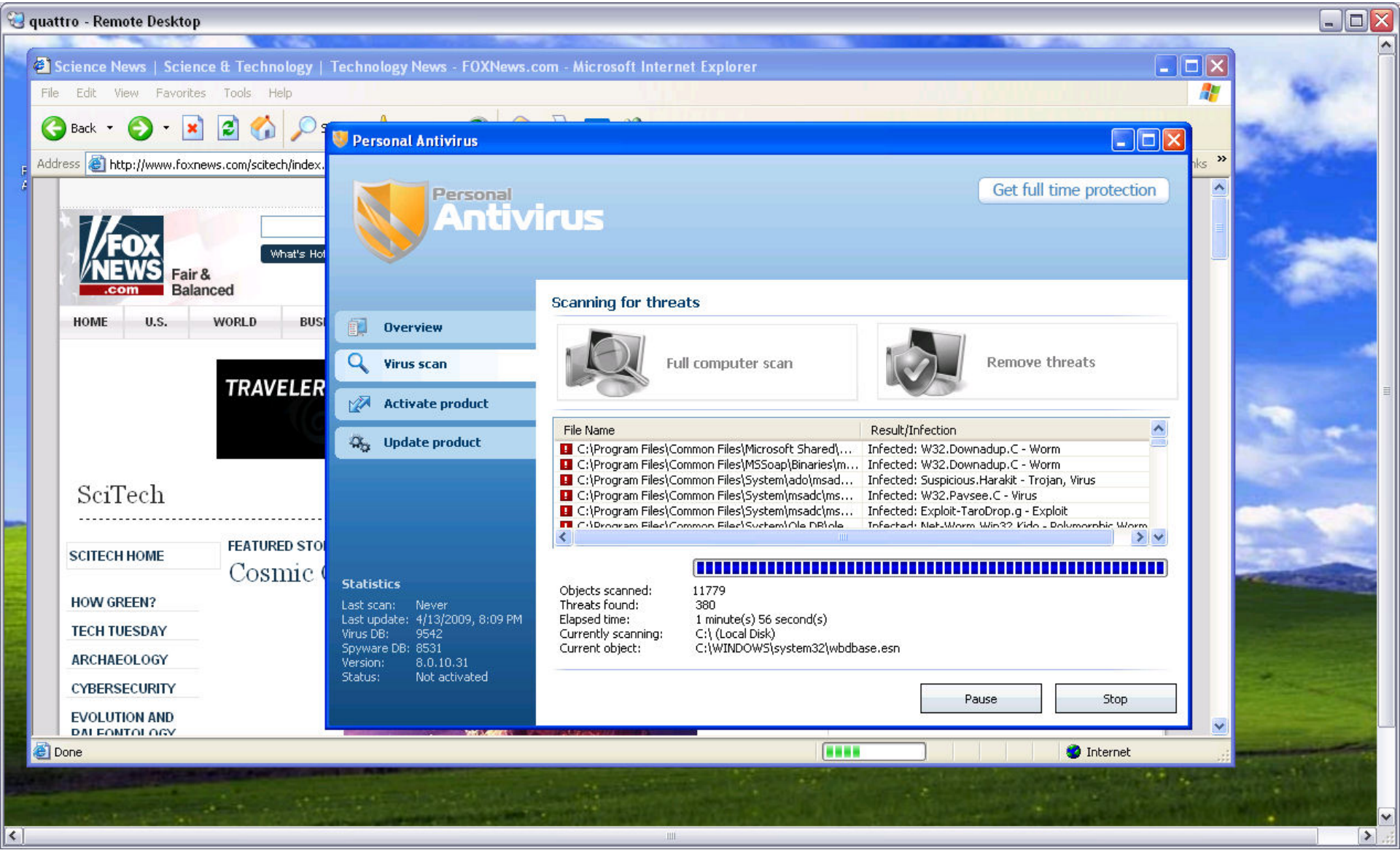


Lessons

- Clickjacking is an injection attack on the human brain
- Trusted path is critical to security
- The web security model was not designed with trusted path in mind
- Changing the web security model is challenging, because of legacy constraints

Discussion

- So, how do these lessons apply to desktop applications?
- Compare the security model for desktop apps:
  - Are desktop apps safer against these attacks?
  - Are desktop apps riskier against these attacks?


quattro - Remote Desktop
Science News | Science & Technology | Technology News - FOXNews.com - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://www.foxnews.com/scitech/index.
FOX NEWS .com
Fair & Balanced
What's Hot
HOME U.S. WORLD
TRAVELER
SciTech
SCITECH HOME
FEATURED STO
Cosmic
HOW GREEN?
TECH TUESDAY
ARCHAEOLOGY
CYBERSECURITY
EVOLUTION AND
Done
Internet
Personal Antivirus
Personal Antivirus
Get full time protection
Overview
Virus scan
Activate product
Update product
Scanning for threats
Full computer scan
Remove threats
File Name
Result/Infection
C:\Program Files\Common Files\Microsoft Shared\...
Infected: W32.Downadup.C - Worm
C:\Program Files\Common Files\MSSoap\Binaries\m...
Infected: W32.Downadup.C - Worm
C:\Program Files\Common Files\System\ado\msad...
Infected: Suspicious.Harakit - Trojan, Virus
C:\Program Files\Common Files\System\msadc\ms...
Infected: W32.Pavsee.C - Virus
C:\Program Files\Common Files\System\msadc\ms...
Infected: Exploit-TaroDrop.g - Exploit
Statistics
Last scan: Never
Last update: 4/13/2009, 8:09 PM
Virus DB: 9542
Spyware DB: 8531
Version: 8.0.10.31
Status: Not activated
Objects scanned: 11779
Threats found: 380
Elapsed time: 1 minute(s) 56 second(s)
Currently scanning: C:\ (Local Disk)
Current object: C:\WINDOWS\system32\wbdbase.esn
Pause
Stop

Discussion

- So, how do these lessons apply to mobile (smartphone/tablet) apps?
- Compare the security model for mobile apps:
  - Are mobile apps safer against these attacks?
  - Are mobile apps riskier against these attacks?