CISC 7320X - COMPUTER SECURITY

The Web—User Side

Rise of the Hackers

- Rise of the Hackers

Objectives

- Attacks against browsers
- Fake and malicious websites
- Attacks targeting sensitive data
- Injection attacks
- Spam
- Phishing attacks

Browsers

- The software most users use as the gateway to the Internet
- Internet access enables certain security threats and vulnerabilities
- Focus on user side: What harm may come to an individual user interacting with Internet locations?

Security Issues for Browsers

- A browser often connects to more than the one address shown in the browser's address bar
- Fetching data can entail accesses to numerous locations
  - to obtain pictures, audio content, and other linked content.
- Browser software can be malicious or can be corrupted to acquire malicious functionality.

Security Issues for Browsers

- Popular browsers support add-ins, extra code to add new features to the browser
  - these add-ins themselves can include corrupting code.
- Data display involves a rich command set
  - Controls rendering, positioning, motion, layering, invisibility.

Security Issues for Browsers

- Browser typically has the user privileges
  - Can access any data on a user's computer allows by access control restrictions
- Data transfers to and from the user are invisible
  - Occur without the user's knowledge or explicit permission

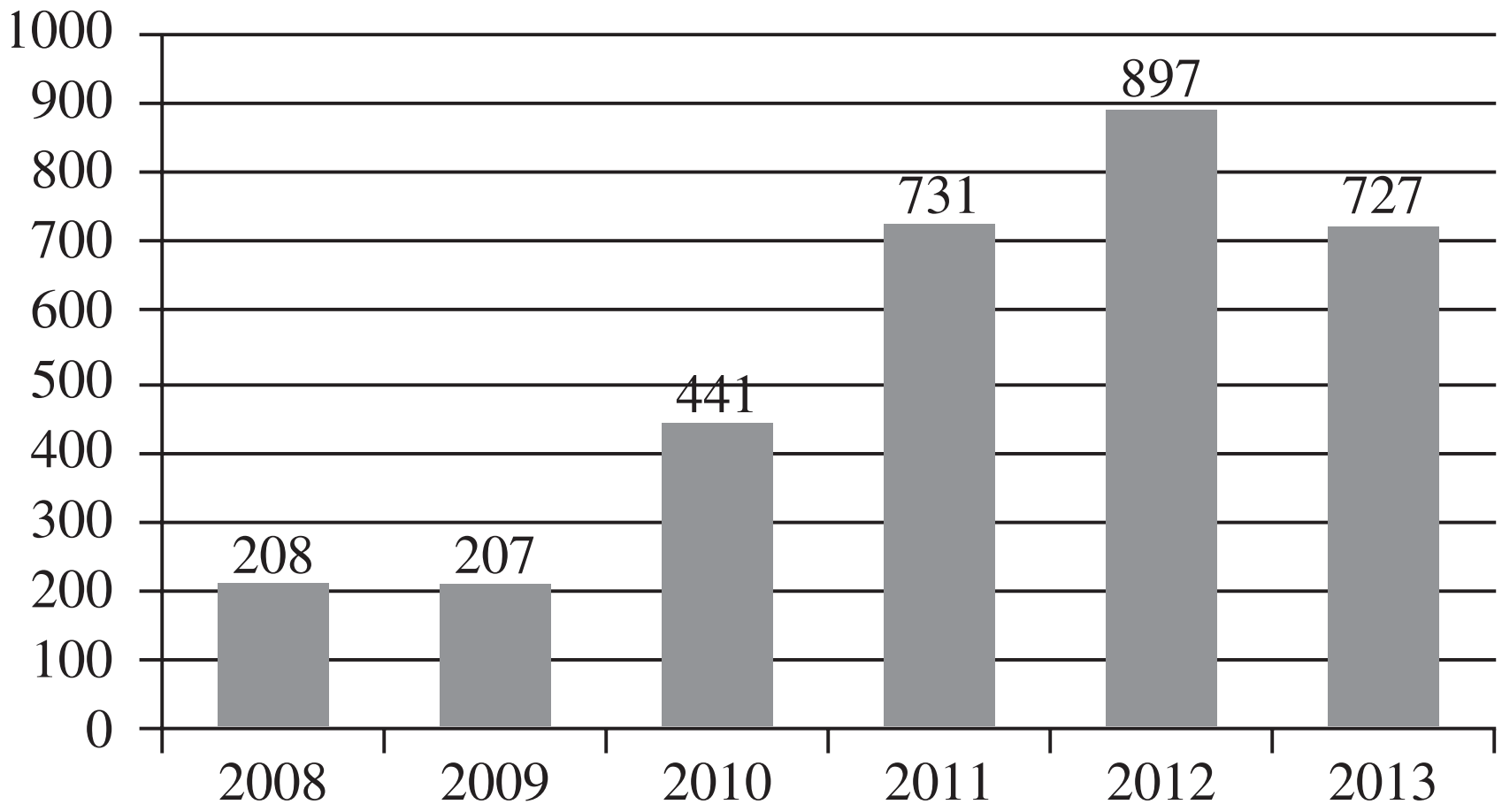
Browser Attacks

- Attacks aim to:
  - Obtain sensitive information
    - E.g., Account numbers or authentication passwords
  - Entice the user
    - for example, using pop-up ads, or to install malware.

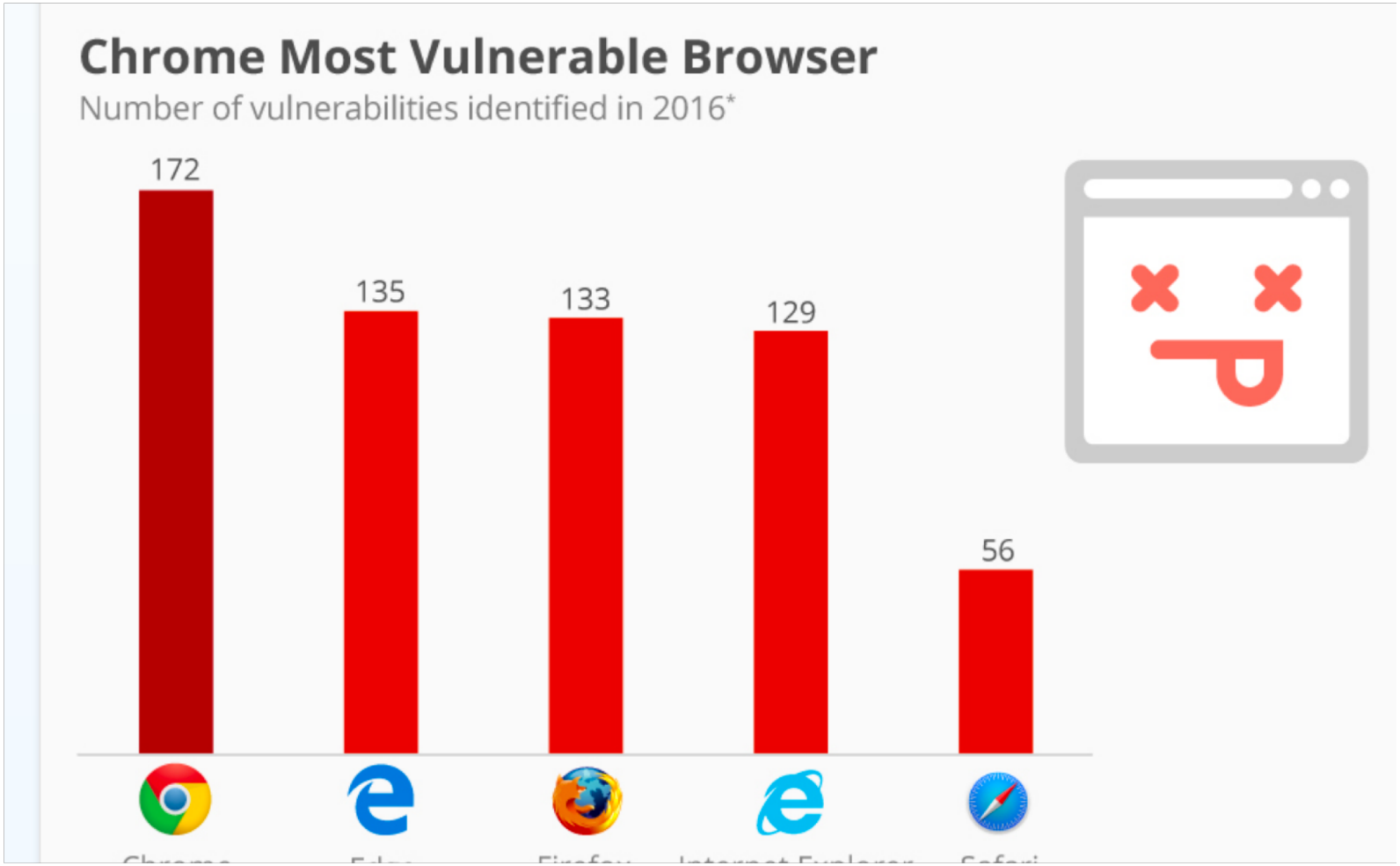
Browser Attacks

- There are three attack vectors against a browser:
  - Go after the operating system
    - => impede the browser's correct and secure functioning.
  - Tackle the browser or one of its components, add-ons, or plug-ins
    - => its activity is altered.
  - Intercept or modify communication to or from the browser

Browser Vulnerabilities



Browser Vulnerabilities



https://www.statista.com/chart/7451/chrome-most-vulnerable-browser/

Browser Attack Types

- Man-in-the-browser
- Keystroke logger
- Page-in-the-middle
- Program download substitution
- User-in-the-middle

Man-in-the-browser

- Attacker modifies web pages
  - in a completely covert fashion
  - invisible to both the user and host web application
- A type of Trojan horse
- Some trojans will be detected and removed by antivirus SW

Man-in-the-browser

- Malicious code that has infected a browser
- Code inserted into the browser can read, copy, and redistribute user browser input
- The threat here is that the attacker will intercept and reuse credentials
  - To access financial accounts and other sensitive data

Man-in-the-browser - Example

- Attack on an internet banking funds transfer:
  - The customer will always be shown, via confirmation screens, the exact payment information as keyed into the browser.
  - The bank, however, will receive a transaction with materially altered instructions
    - i.e. a different destination account number and possibly amount.

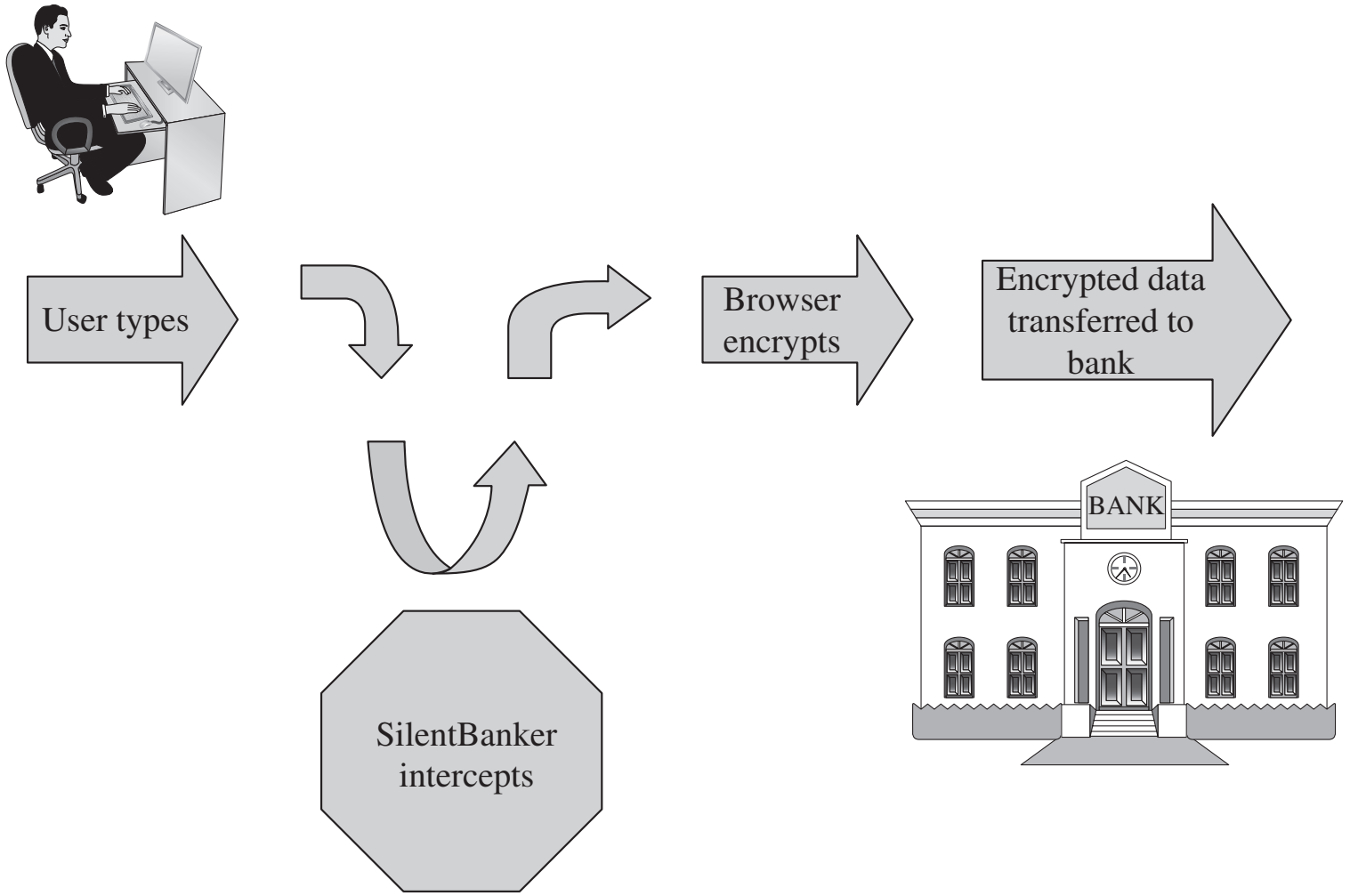
Man-in-the-browser - Example

- Attack on an internet banking funds transfer (cont.):
  - The use of strong authentication tools may create an increased level of misplaced confidence on the part of both customer and bank
    - that the transaction is secure
    - authentication is concerned with the validation of identity credentials.
      - This should not be confused with transaction verification.

SilentBanker

- SilentBanker was a Trojan that generally installed as a browser plug-in
- When it detected the user going to a banking URL, it would:
  - intercept keystrokes and even modify them so that money transfers would go to attackers' accounts.

Man-in-the-Browser



SilentBanker

- SilentBanker started with a list of over 400 URLs of popular banks throughout the world.
- Whenever it saw a user going to one of those sites, it redirected the user's keystrokes
  - recorded customer details that it forwarded to remote computers (presumably malicious bots)
- Detected in 2008 by Liam Omurchu of Symantec

Keystroke Logger

- Hardware or software that records all keystrokes
- May be a small dongle plugged into a USB port or can masquerade as a keyboard
- May also be installed as malware
- Not limited to browsers

Page-in-the-Middle

- User is directed to a different page than believed or intended
- Similar effect to a man-in-the-browser, where attacker can intercept and modify user input

Page-in-the-middle vs. Man-in-the-browser Attacks

- Man-in-the-browser action:
  - An example of an infected browser that may never alter the sites visited by the user
    - but works behind the scenes to capture information
- Page-in-the-middle action:
  - The attacker redirects the user, presenting different web pages for the user to see.

Program Download Substitution

- Attacker creates a page with seemingly innocuous and desirable programs for download
- Instead of, or in addition to, the intended functionality, the user installs malware
- This is a very common technique for spyware

User-in-the-Middle

- Using click-bait to trick users into solving CAPTCHAs on spammers' behalf

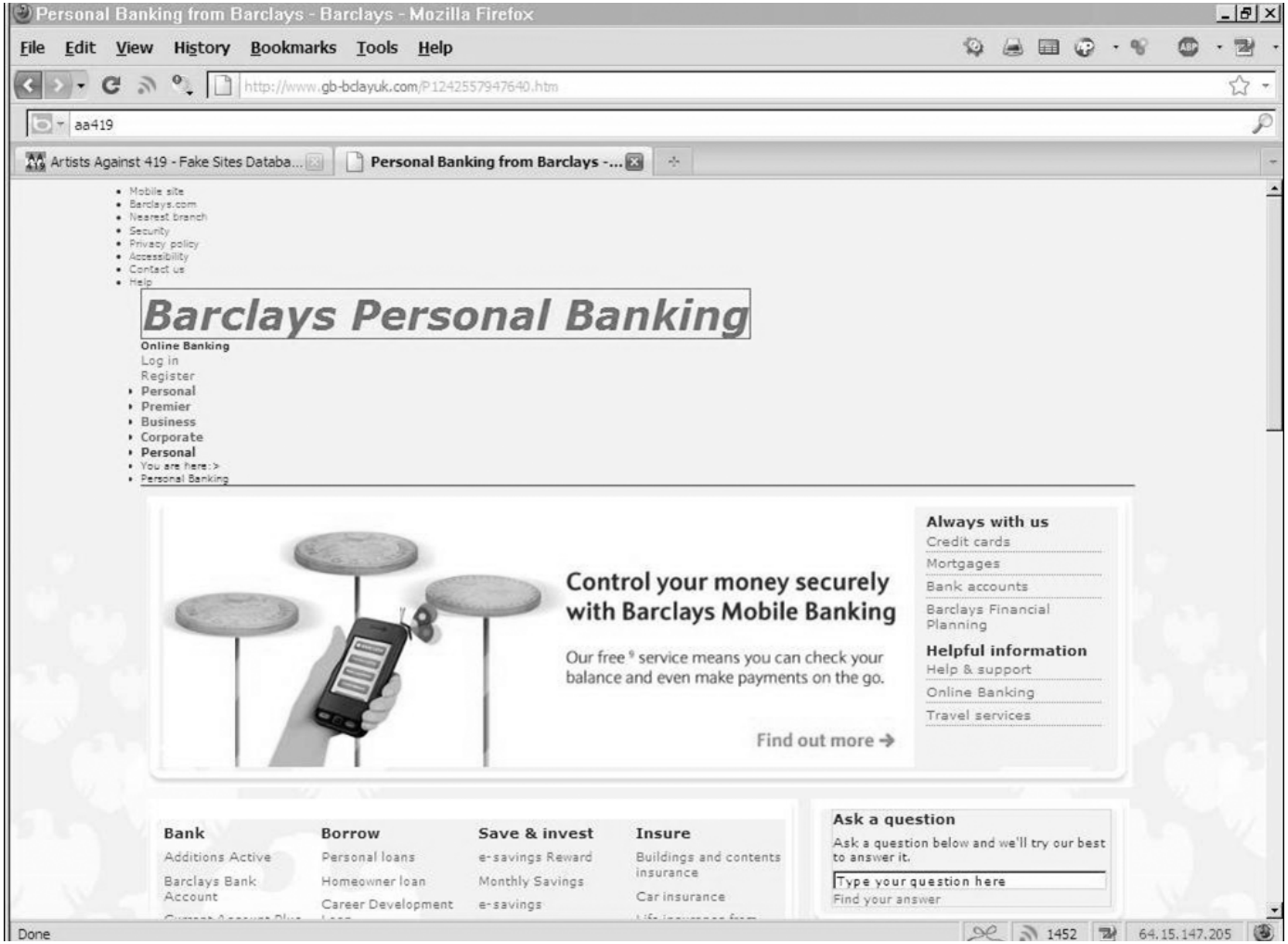
User-in-the-Middle

- CAPTCHAs are used by websites to defeat automation
  - such as by preventing spammers from scripting the creation of massive numbers of email accounts
- By using dummy websites to entice users into solving CAPTCHAs, attackers can effectively defeat the CAPTCHAs at scale

Successful Authentication

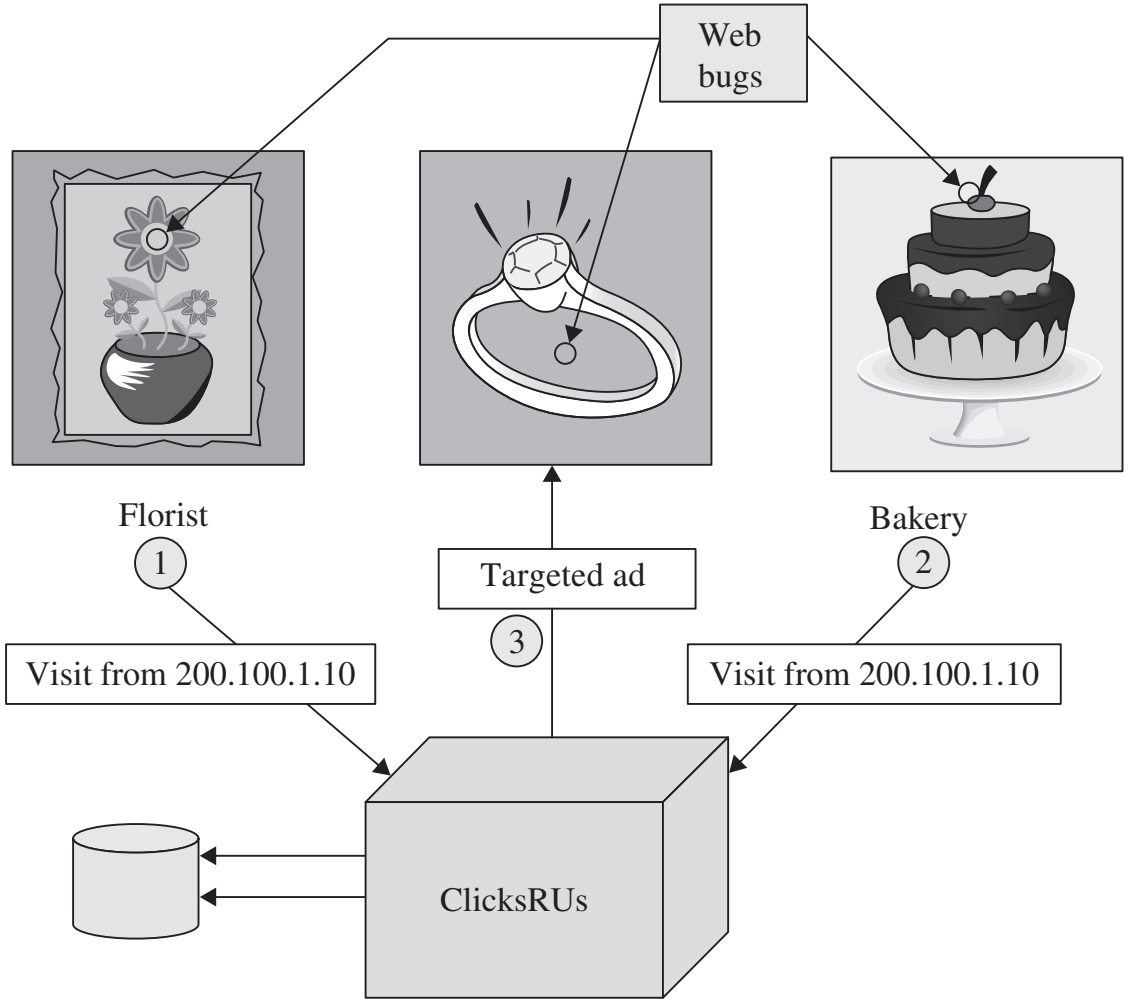
- The attacks listed above are largely failures of authentication
- Can be mitigated with
  - Shared secret
  - One-time password
  - Out-of-band communication

Fake Website



Fake Code

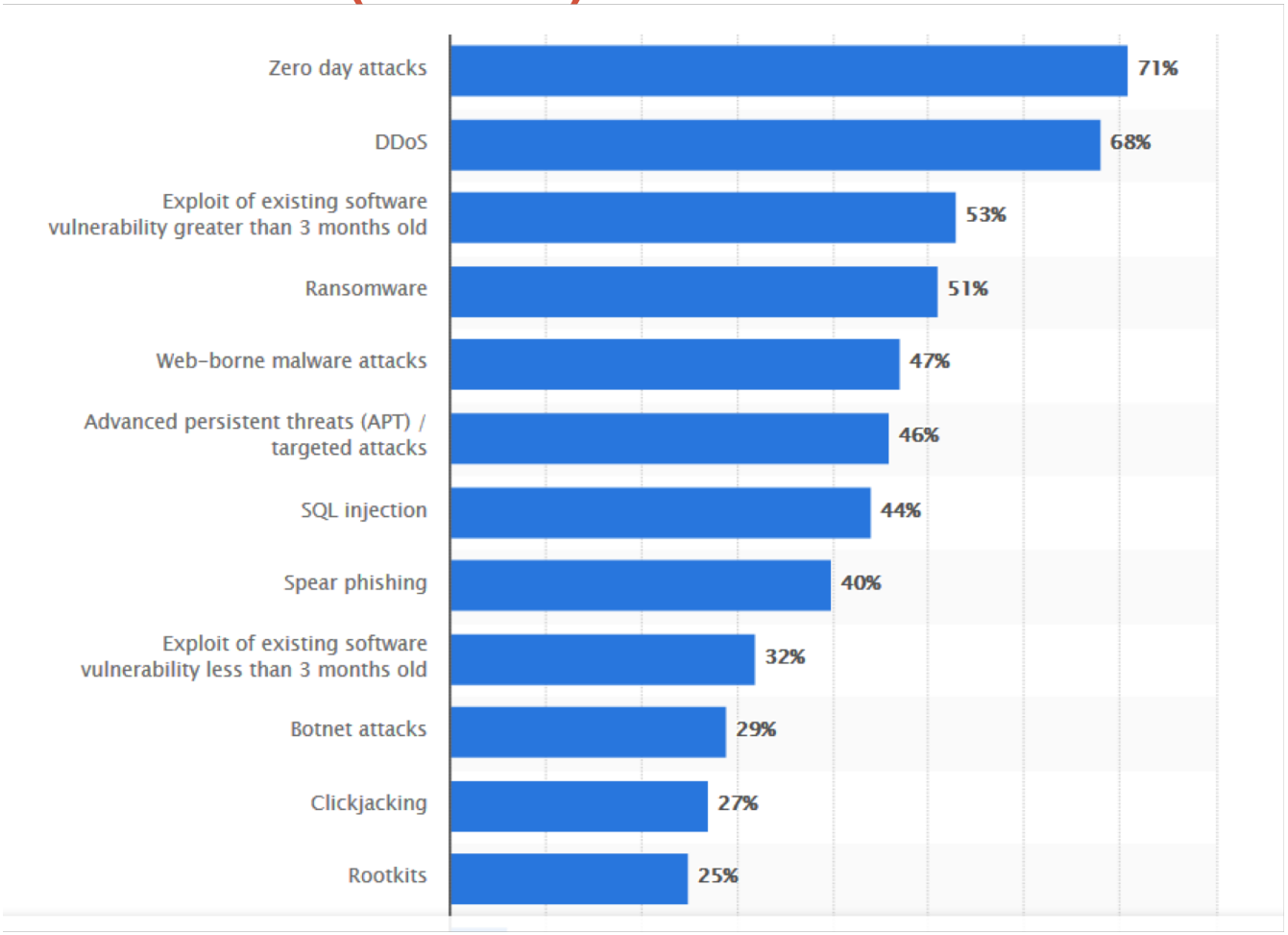
Tracking Bug



Tracking Bugs

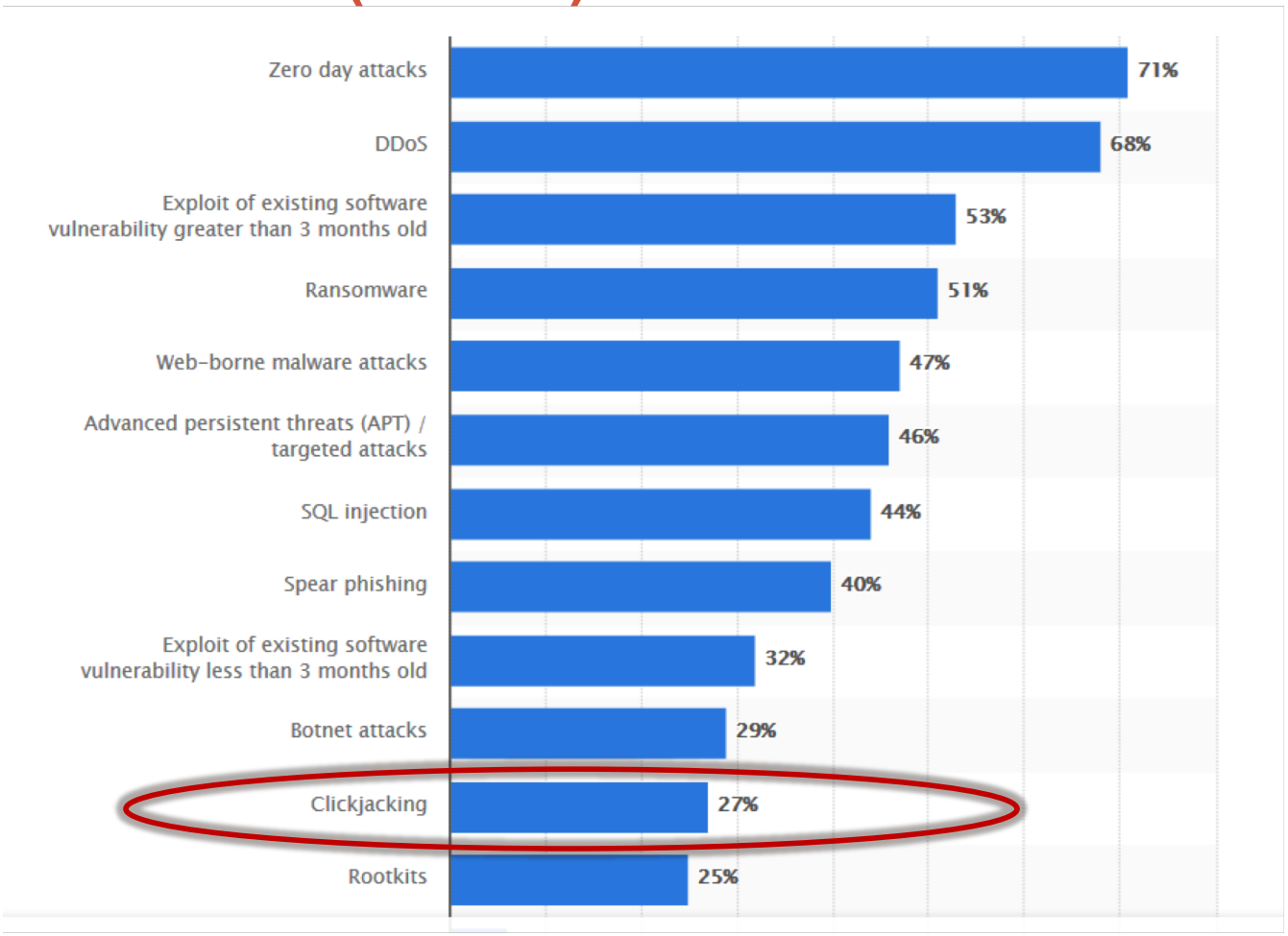
- A tiny image served up from one provider ("ClicksRUs")
  - allows user behavior to be tracked across many sites for advertising purposes
- You may notice this when you see web ads that offer up items very similar to ones you recently been shopping for on other sites
- Web bugs can also be used to track users' reading of advertising emails.

Most serious endpoint security incidents in the US (2016)



https://www.statista.com/statistics/203186/top-endpoint-security-incidents-usa/

Most serious endpoint security incidents in the US (2016)

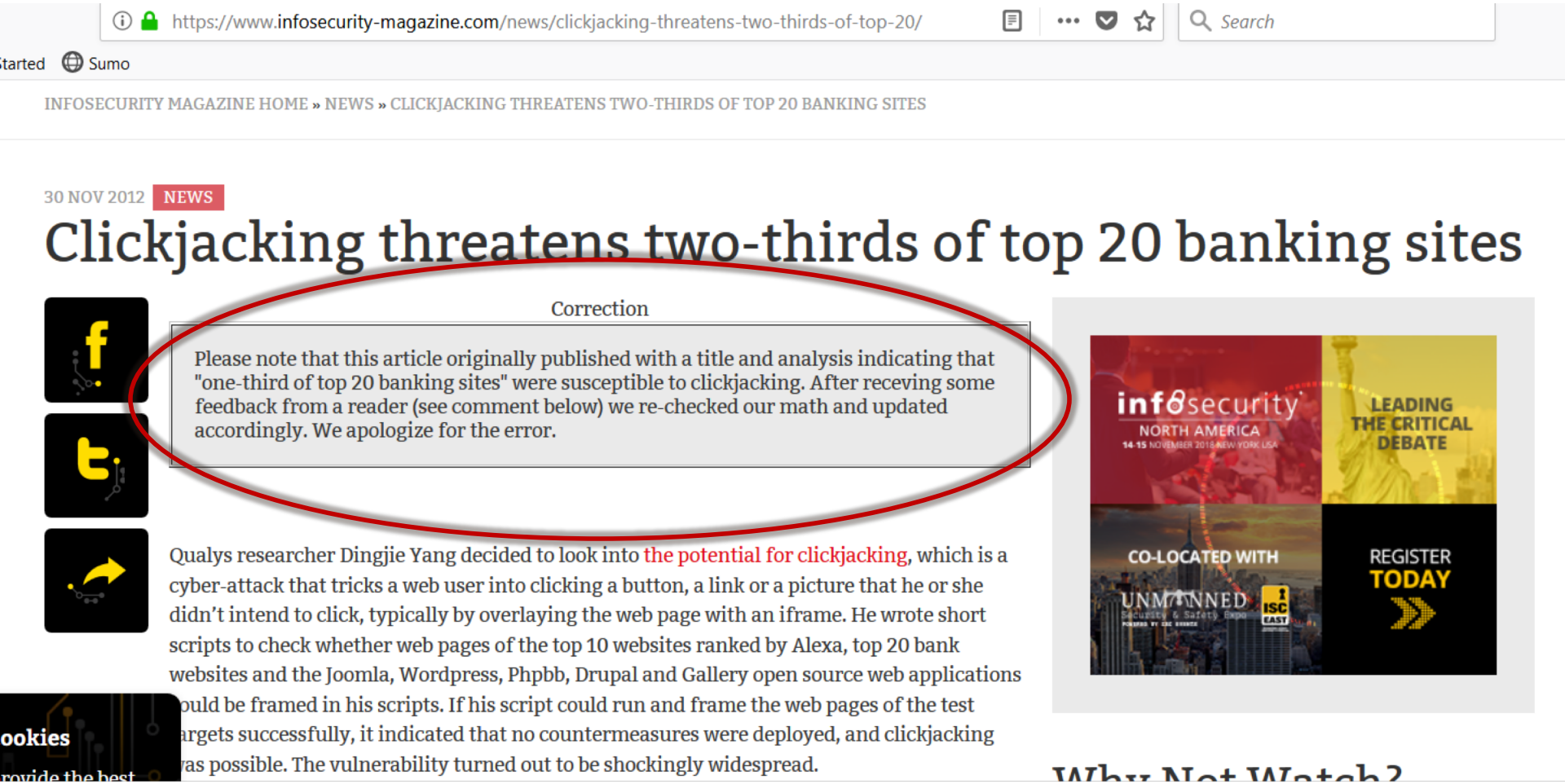


https://www.statista.com/statistics/203186/top-endpoint-security-incidents-usa/

Clickjacking

https://www.infosecurity-magazine.com/news/clickjacking-threatens-two-thirds-of-top-20/

Clickjacking



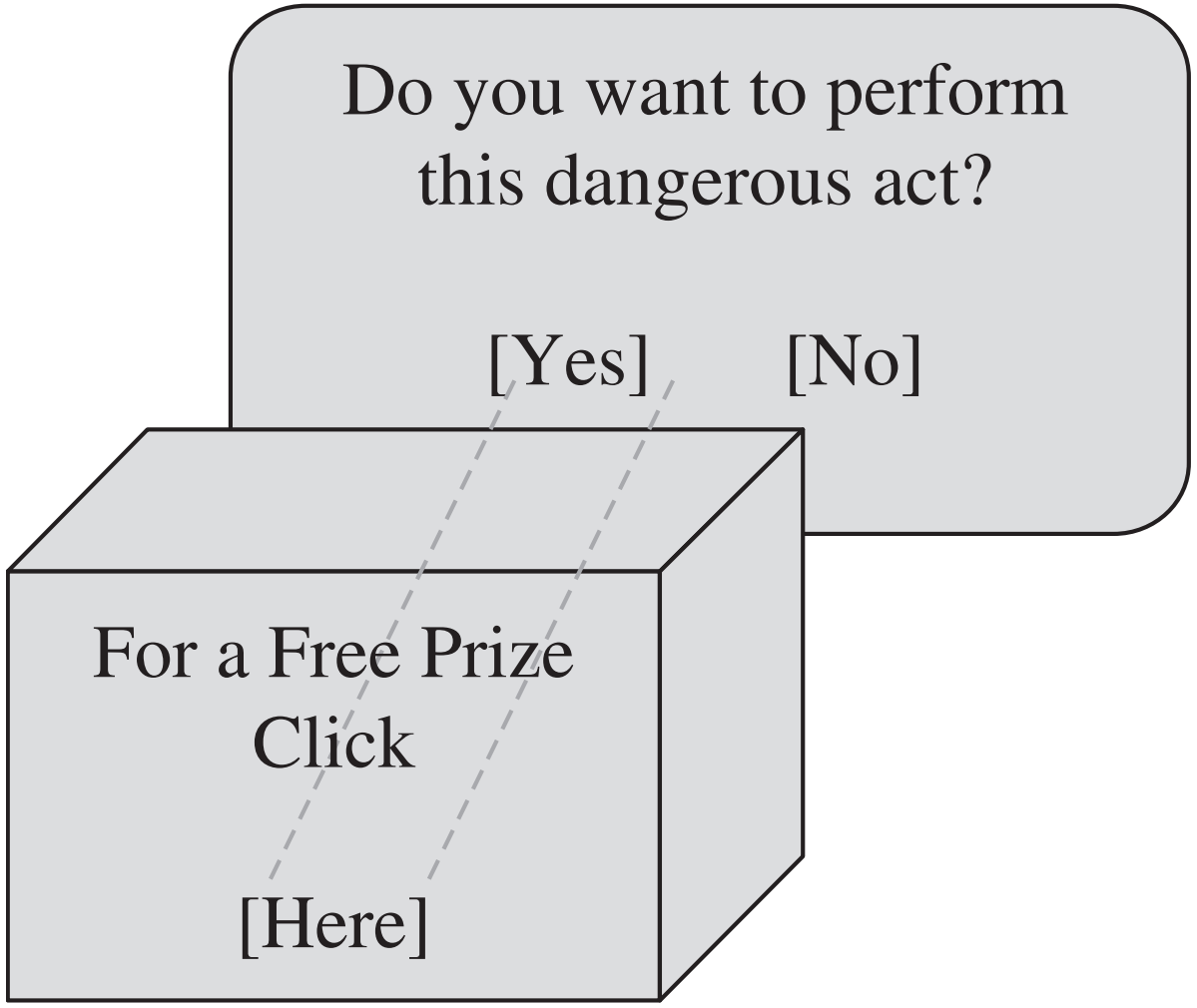
https://www.infosecurity-magazine.com/news/clickjacking-threatens-two-thirds-of-top-20/

Clickjacking

Correction

Please note that this article originally published with a title and analysis indicating that "one-third of top 20 banking sites" were susceptible to clickjacking. After receving some feedback from a reader (see comment below) we re-checked our math and updated accordingly. We apologize for the error.

Clickjacking



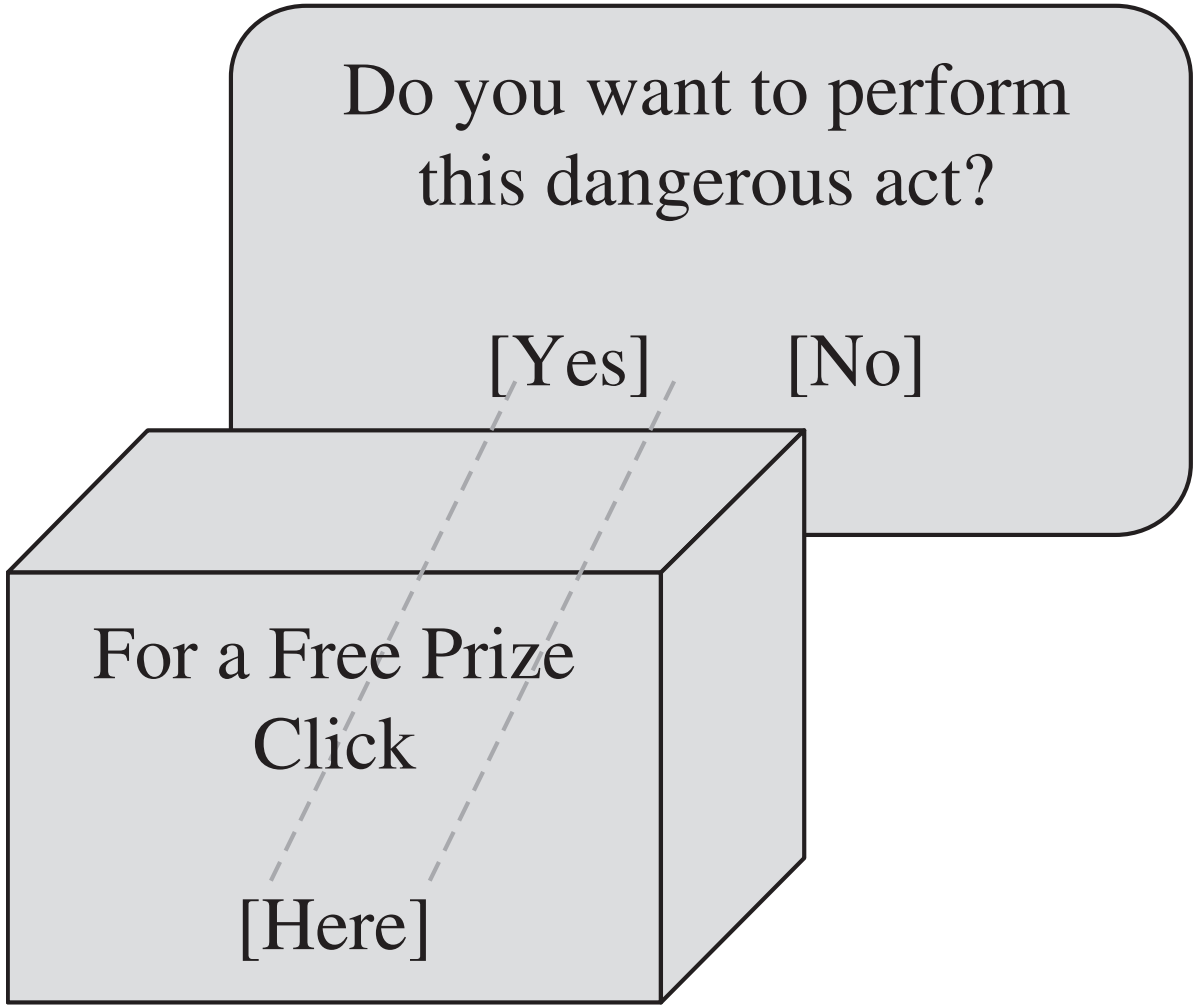
Clickjacking Attacks

- Clickjacking is a way of tricking users into providing desired input
- The attacker makes the input dialog transparent and places an image with an enticement below the transparent dialog

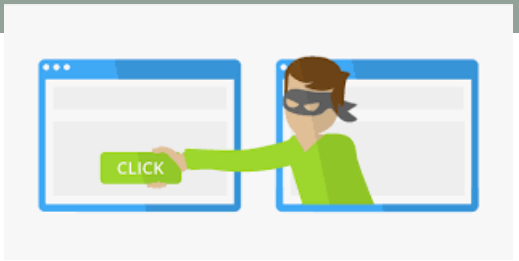
Clickjacking Attacks

- The user ends up answering a question he didn't even know he was being asked
  - unknowingly authorizing his computer to execute the attacker's will
- "Framing"—moving and layering HTML iframes—is an important component of this attack.

Clickjacking



Clickjacking Attacks

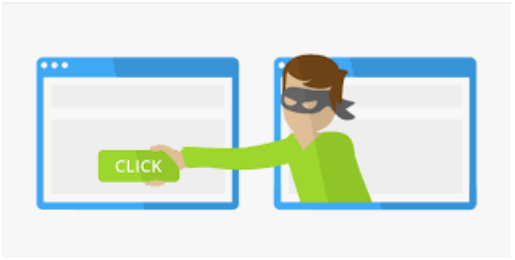


- A.K.A. User Interface (UI) redress attack
- Tricking a user into clicking on something different from what he thinks he is clicking on
- Risks:
  - potentially revealing confidential information
  - Taking control of their computer while clicking on seemingly innocuous web pages
- Exists in a variety of browsers and platforms

https://neelbhatt.com/2018/02/16/secure-net-core-applications-from-click-jacking-net-core-security-part-iii/

Clickjacking Attacks

- How does it happen?



Clickjacking Attacks

- How does it happen?
- Example:

```
<a
    onMountUp=window.open(http://www.evil.com)
    href=http://www.google.com/>
    Go to Google</a>
```

- What happens with this code?
  - A window opens to the attacker website

Clickjacking Attacks

- Example:

```
<a
    onMountUp=window.open(http://www.evil.com)
    href=http://www.google.com/>
    Go to Google</a>
```

- Why include *href* to Google?
  - Browser status bar will show URL when hovering
    - To protect the user
    - User tricked by seeing the wrong reference

Clickjacking Attacks

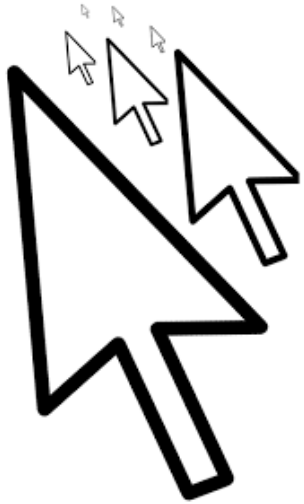
- Example 2:
  - A user might receive an email with a link to a video about a news item
  - Another webpage may be "hidden" on top or underneath the "PLAY" button of the news video
    - E.g., a product page on Amazon
  - The user tries to "play" the video
    - actually "buys" the product from Amazon
    - Attack will work if visitor is both logged into Amazon.com and has 1-click ordering enabled
      - Hacker can only send a single click

Other Scenarios

- Tricking users into enabling their webcam and microphone through Flash
- Downloading and running a malware (malicious software) allowing to a remote attacker to take control of others computers
- Clicking Google AdSense ads to generate pay-per-click revenue
- Etc.

Cursorjacking

- Cursor may be changed
  - Create a more visible fake shifted cursor
    - In addition to the real cursor
    - Will cause the victim to go to evil website, etc.



http://resources.infosecinstitute.com/bypassing-same-origin-policy-part-3-clickjacking-cursorjacking-filejacking/
https://explosivelab.blogspot.com/2012/04/cursor-jacking.html

Clickjacking Known Attacks

- Twitter clickjacking worm:
  - Attack convinced users to click on a button that re-tweeted location of a malicious page
    - Propagated massively
- Facebook attacks:
  - Attackers trickers users into “liking” items
    - Fan pages, links, groups, etc.

Clickjacking Defenses

- Requiring user confirmation
  - Reduces usability, requires extra actions
- Adding random UI elements, randomize location of buttons on page
  - Make it harder for attacker to overlay known elements
  - Difficult to implement, may still be vulnerable
    - Attacker may click multiple locations

Clickjacking Defenses

- Implementing defensive code in the UI
  - Ensure current frame is most top level window
- Preventing websites from framing your site
  - Incorporating frame-breaking methods when programming site

Clickjacking Defenses - Sitekeys

- A mutual authentication web-based technique
  - between end-users and websites
  - To deter phishing
- Owned by RSA
- Was used by Vanguard, Bank of America
  - Discontinued in 2015
    - Still used by some sights
- Found to be ineffective
  - People don’t notice or care about it

Clickjacking Defenses - Sitekeys

https://www.cs.tufts.edu/comp/117/slides/Mez Tufts2018.pdf

Drive-By Download

- Code is downloaded, installed, and executed on a computer without the user's knowledge
- May be the result of clickjacking, fake code, program download subsitution, etc.

Dot-Dot-Slash

- Also known as “directory traversal,” this is when attackers use the term “../” to access files that are on the target web server but not meant to be accessed from outside
- Most commonly entered into the URL bar but may also be combined with other attacks, such as XSS

```
http://yoursite.com/webhits.htw?CiWebHits&File=../../../../../winnt/system32/autoexec.nt
```

Server-Side Include (SSI)

- SSI is an interpreted server-side scripting language that can be used for basic web server directives
  - such as including files and executing commands
- As is the case with XSS, some websites are vulnerable to allowing users to execute SSI directives through text input

```
<!--#exec cmd="/usr/bin/telnet &"-->
```

Countermeasures to Injections

- Filter and sanitize all user input
  - Need to account for every potentially valid encoding
- Make no assumptions about the range of possible user inputs
  - trust nothing, check everything
- Use access control mechanisms on backend servers, such as “stored procedures”

Email Spam

- Experts estimate that 60% to 90% of all email is spam
- Types of spam:
  - Advertising
    - Pharmaceuticals
    - Stocks
  - Malicious code
  - Links for malicious websites

Email Spam

- Spam countermeasures
  - Laws against spam exist but are generally ineffective
  - Email filters have become very effective for most spam
  - Internet service providers use volume limitations to make spammers' jobs more difficult

Countermeasures

- User education
  - Limited effectiveness and very subject to co-evolution with attacks
  - However, Became more scientific over the years
    - with products like PhishMe automating the user training process and focusing on the worst offenders
- PGP and S/MIME
  - Cryptographic solutions that have seen very limited adoption after years on the market
    - solutions for encrypting and signing email

- Questions?