CMSC414 Computer and Network Security

UI Attacks, CAPTCHAs, Security Principles

Yizheng Chen | University of Maryland
surrealyz.github.io

Feb 19, 2026

Announcement

- Project 1, **due today!**
- Project 2 deadline Thursday, March 5
- Don’t wait!

Agenda

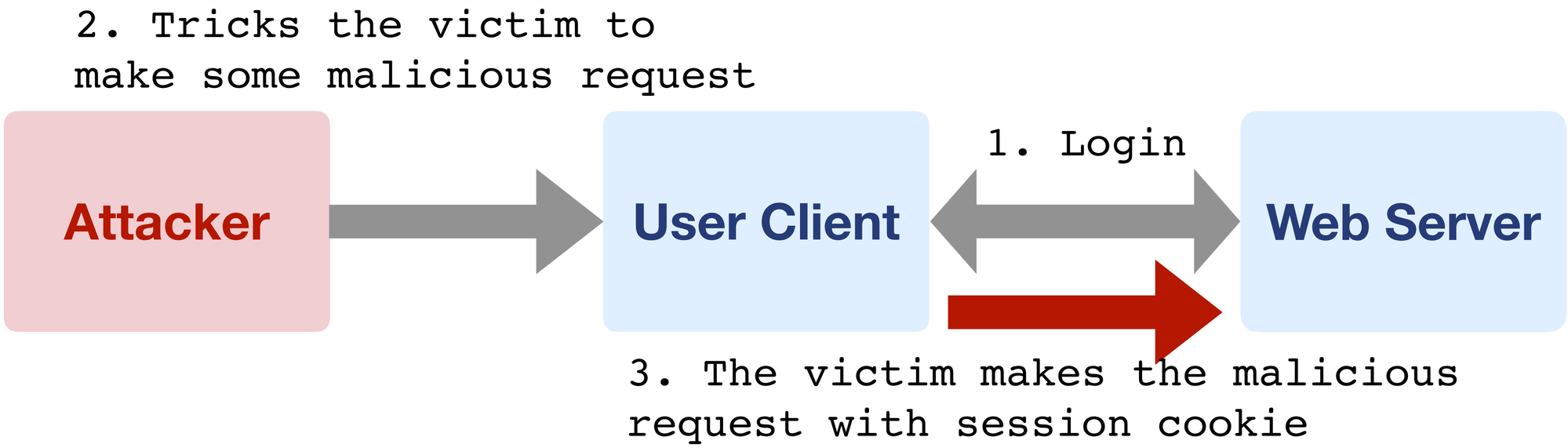
- Recap
- UI Attacks
- CAPTCHAs
- Security Principles

Cross-Site Request Forgery (CSRF)

- **Cross-site request forgery (CSRF or XSRF)**: An attack that exploits cookie-based authentication to perform an action as the victim

Steps of a CSRF Attack

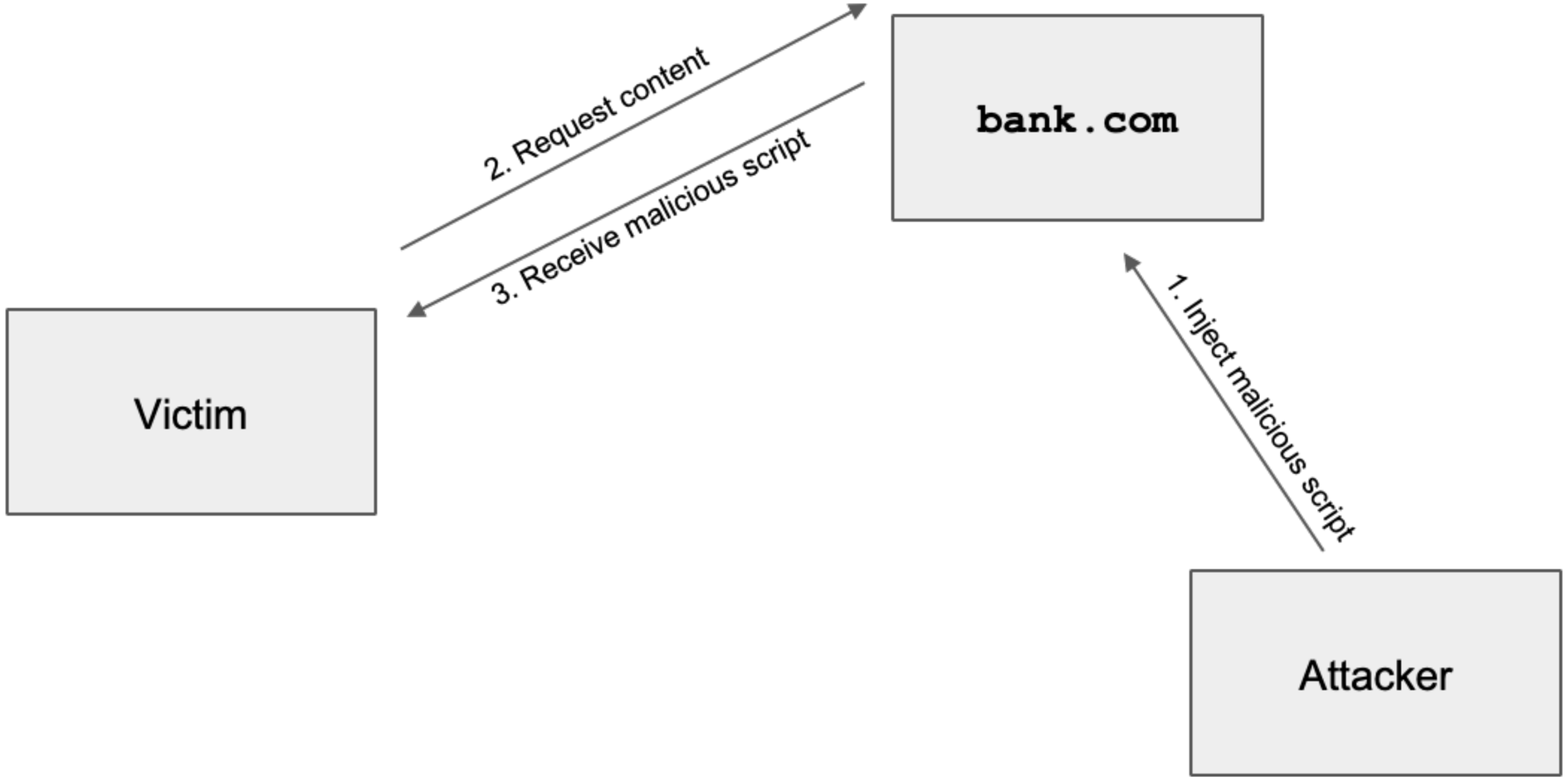
1. User authenticates to the server, receives a **cookie** with a valid **session token**
2. Attacker **tricks** the victim into making a malicious request to the server
3. The victim **makes the malicious request**, attaching the cookie, server accepts it



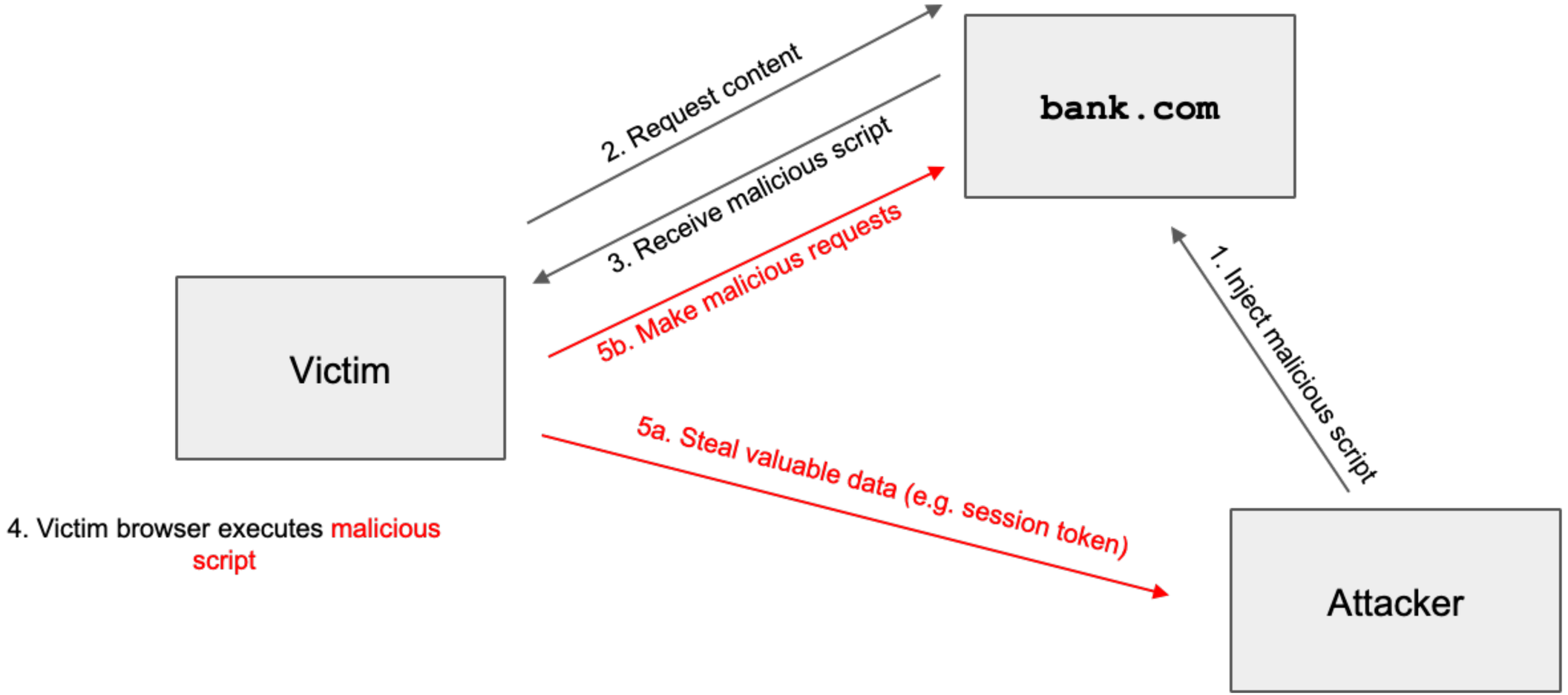
Cross-Site Scripting (XSS)

- **Cross-site scripting (XSS)**: Injecting JavaScript into websites that are viewed by other users
  - Cross-site scripting subverts the same-origin policy
- Two main types of XSS
  - Stored XSS
  - Reflected XSS

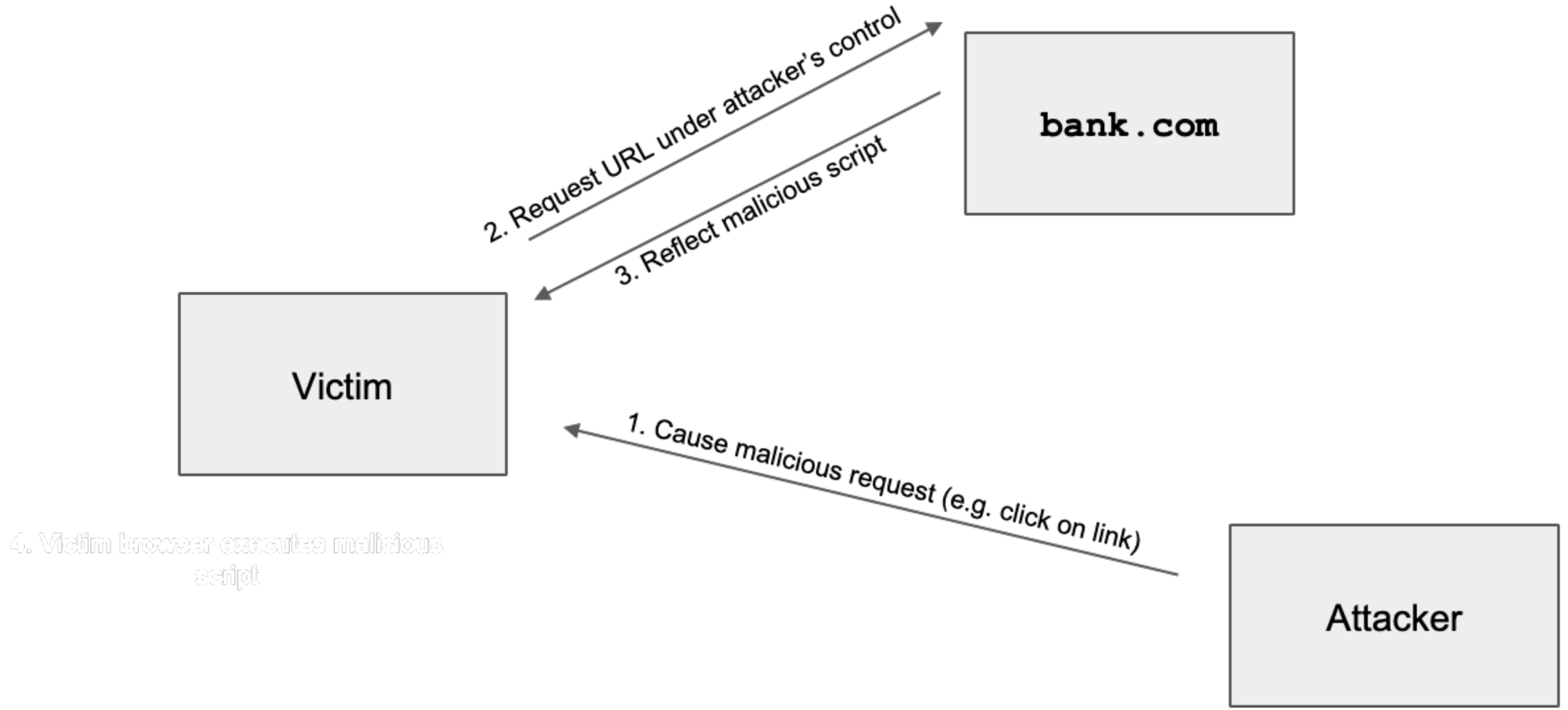
Stored XSS


bank.com
2. Request content
3. Receive malicious script
Victim
1. Inject malicious script
Attacker

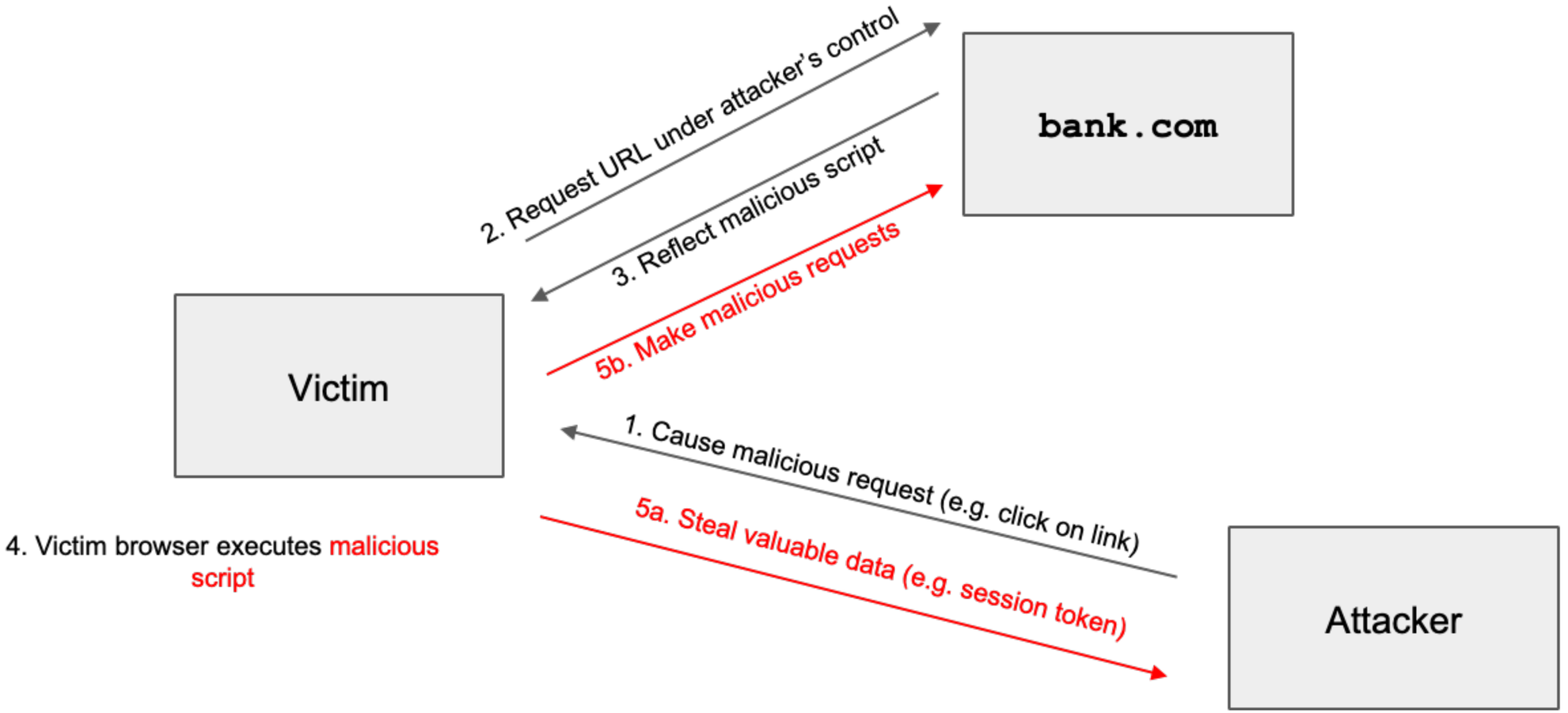
Stored XSS


bank.com
2. Request content
3. Receive malicious script
5b. Make malicious requests
1. Inject malicious script
Victim
5a. Steal valuable data (e.g. session token)
Attacker
4. Victim browser executes malicious script

Reflected XSS



Reflected XSS



Reflected XSS vs CSRF

- Reflected XSS and CSRF both require the victim to make a request to a link
- Reflected XSS: An HTTP response contains maliciously inserted JavaScript, executed on the client side
- CSRF: A malicious HTTP request is made (containing the user's cookies), executing an effect on the server side

XSS Defense: HTML Sanitization

- Checking for malicious input that might cause JavaScript to run, such as `<script>` tags. Remove these tags.
- What about `<scr<script>ipt>`

XSS Defense: HTML Sanitization

- Treat untrusted user input as data, not HTML.
  - Escape the input
- Example: <script>alert(1)</script>
  - Start with & and end with a ;
  - Instead of <, use <
  - Instead of ", use "
  - Escape all dangerous characters

```
<html>
<body>
Hello &lt;script&gt;alert(1)&lt;/script&gt;!
</body>
</html>
```

- Note: You should always rely on trusted libraries to do this for you!

XSS Defense: Content Security Policy (CSP)

- Defined by a web server and enforced by a browser
- Instruct the browser to only use resources loaded from specific places
  - Disallow inline scripts, e.g., `<script>alert(1)</script>`
  - Only allow scripts from some domains <script src="https://example.com/jsfile.js">
  - Also works with iframes, images, etc.
- Uses additional headers to specify the policy
  - Content-Security-Policy

XSS Defense: Content Security Policy (CSP)

- Defined by a web server and enforced by a browser
- Instruct the browser to only use resources loaded from specific places
  - Disallow inline scripts, e.g., `<script>alert(1)</script>`
  - Only allow scripts from some domains <script src="https://example.com/jsfile.js">
  - Also works with iframes, images, etc.
- Uses additional headers to specify the policy
  - Content-Security-Policy

Use allowlist, not blocklist

How to trick the user into making a HTTP request?

Agenda

- Recap
- UI Attacks
- CAPTCHAs
- Security Principles

UI Attacks

- General theme: The attacker tricks the victim into thinking they are taking an **intended** action, when they are actually taking a **malicious** action
- Two main types of UI attacks
  - **Clickjacking**: Trick the victim into clicking on something from the attacker
  - **Phishing**: Trick the victim into sending the attacker personal information

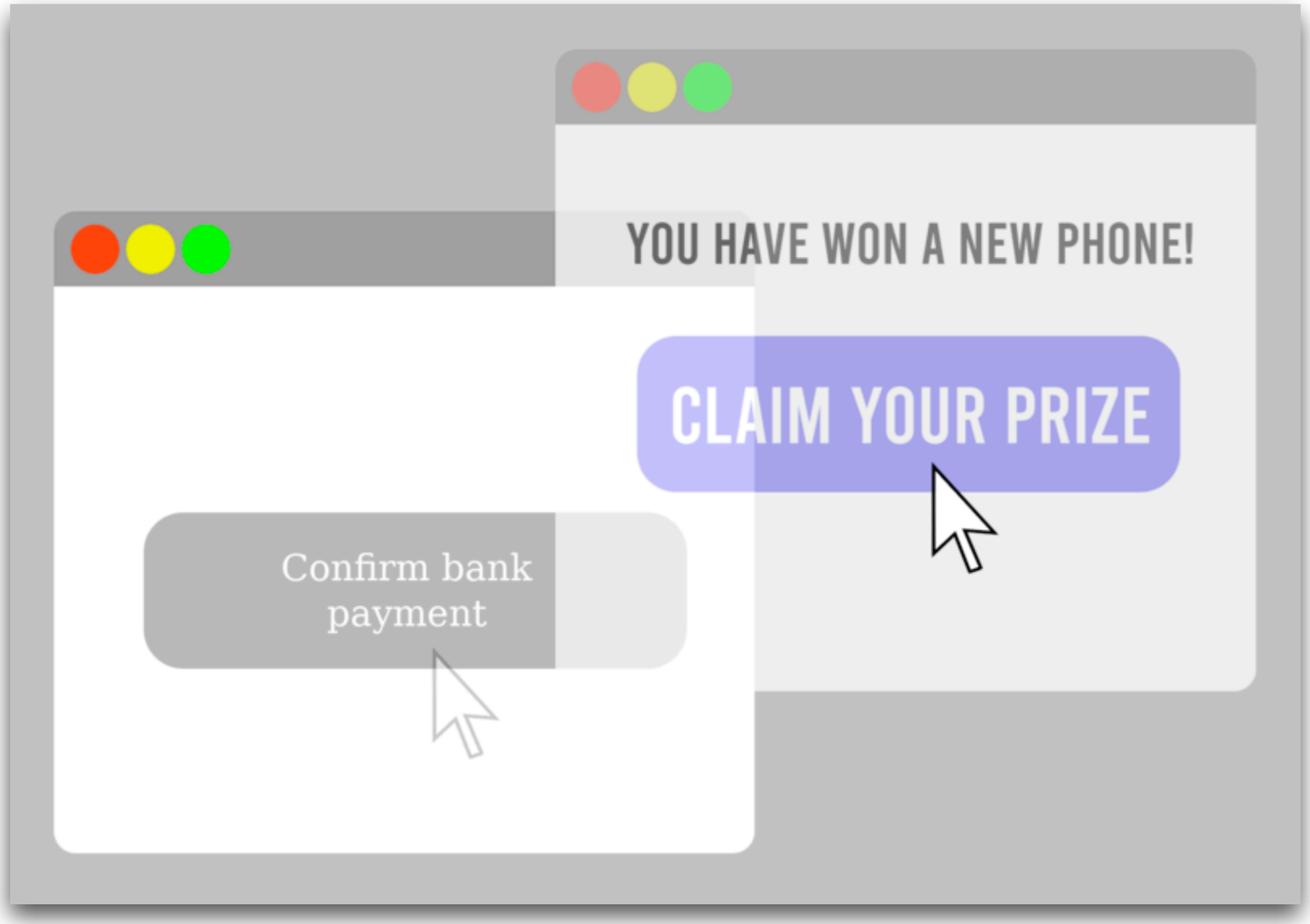
Clickjacking

- **Clickjacking**: Trick the victim into clicking on something from the attacker
- The browser trusts the user's clicks
- Why steal clicks?
  - Download a malicious program
  - Like a Facebook page/YouTube video
  - Delete an online account

Clickjacking: Download Buttons

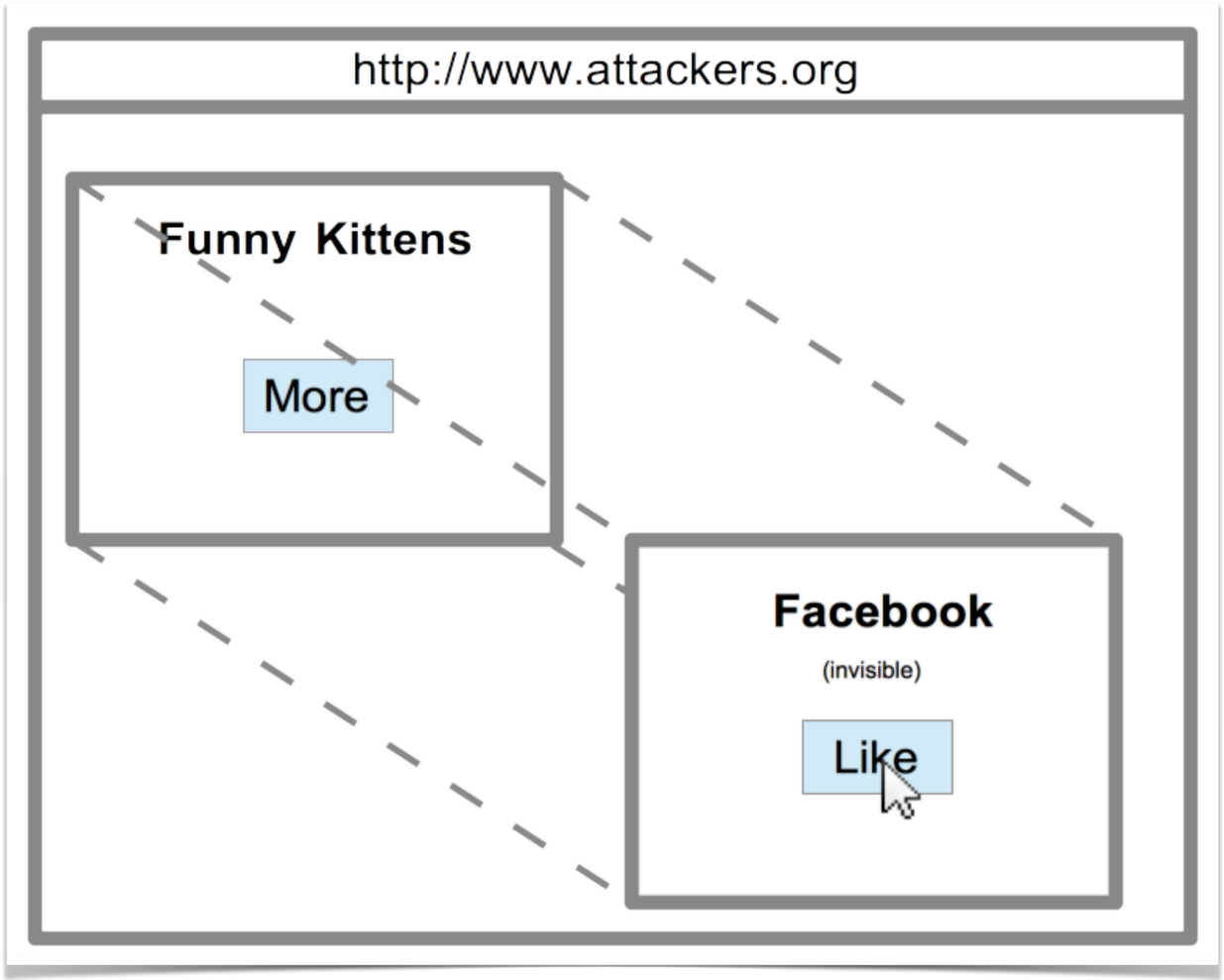
- Which is the real download button?
- What if the user clicks the wrong one?

Invisible iframe Variant #1



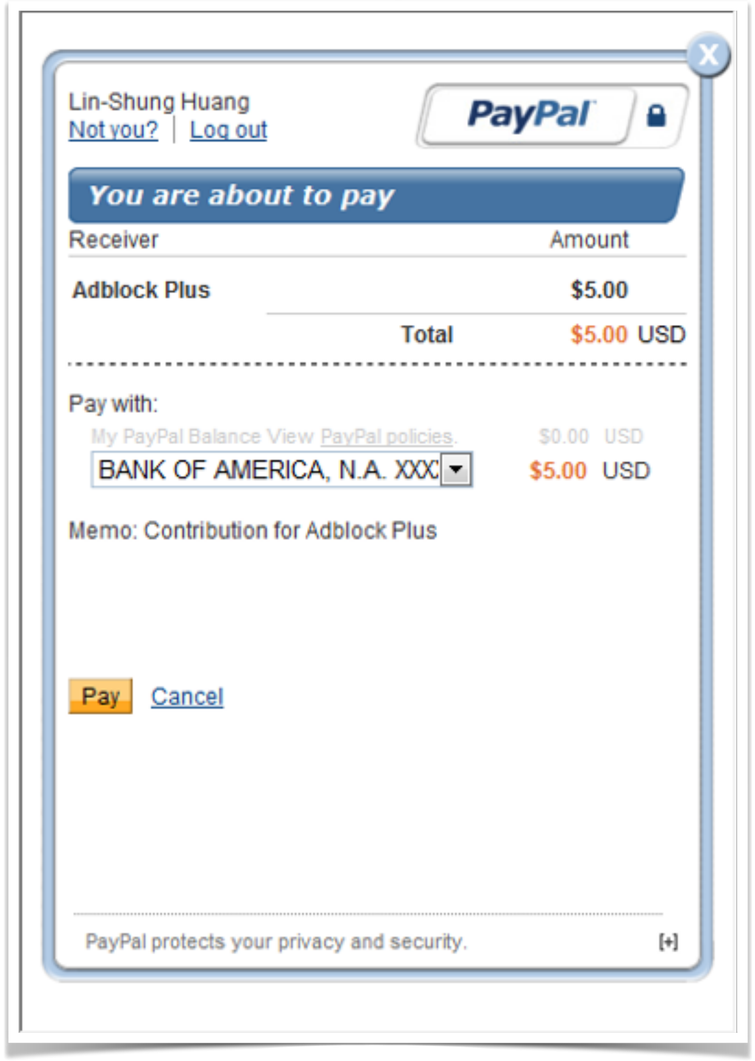
- Frame the legitimate site invisibly, over visible, enticing content
- Victims think they are clicking on the enticing site, but they click on the legitimate site, e.g., pay the attacker's account

Invisible iframe Variant #2

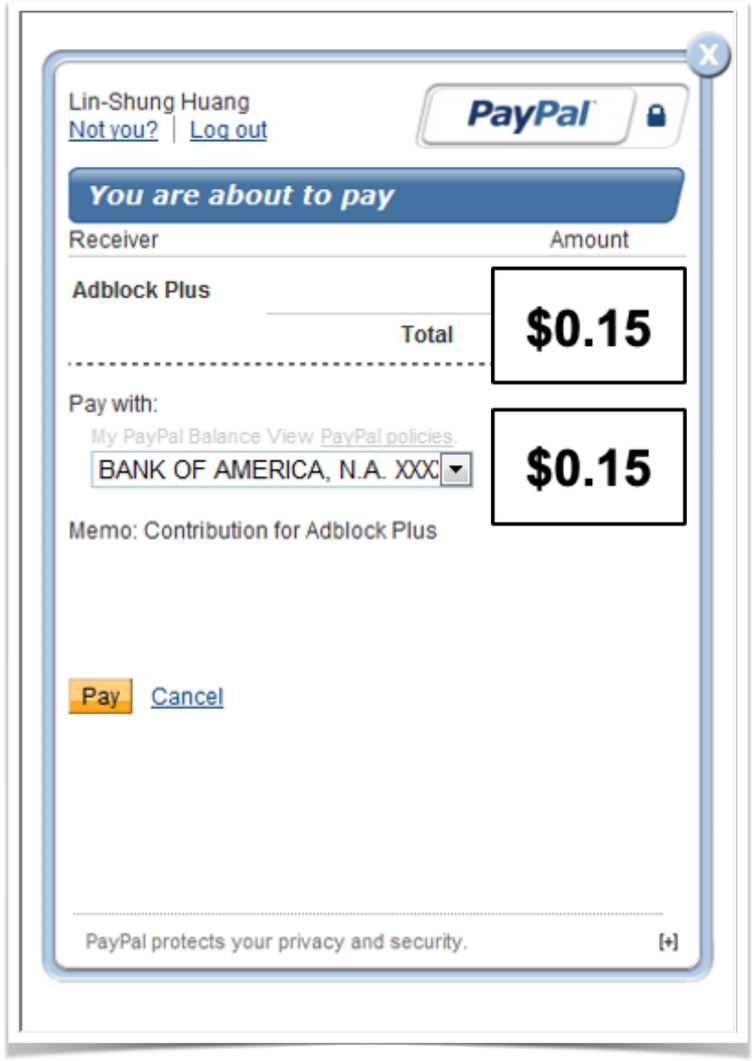


- Frame the legitimate site visibly, under invisible malicious content
- Victims think they are clicking on the visible legitimate site, but their click happens on the malicious site, e.g., fake likes, download malicious software

Invisible iframe Variant #3



Invisible iframe Variant #3



- Frame the legitimate site visibly, under malicious content partially overlaying the site
- The attacker can change the appearance of the site without breaking the Single-Origin Policy

Clickjacking: Temporal Attack

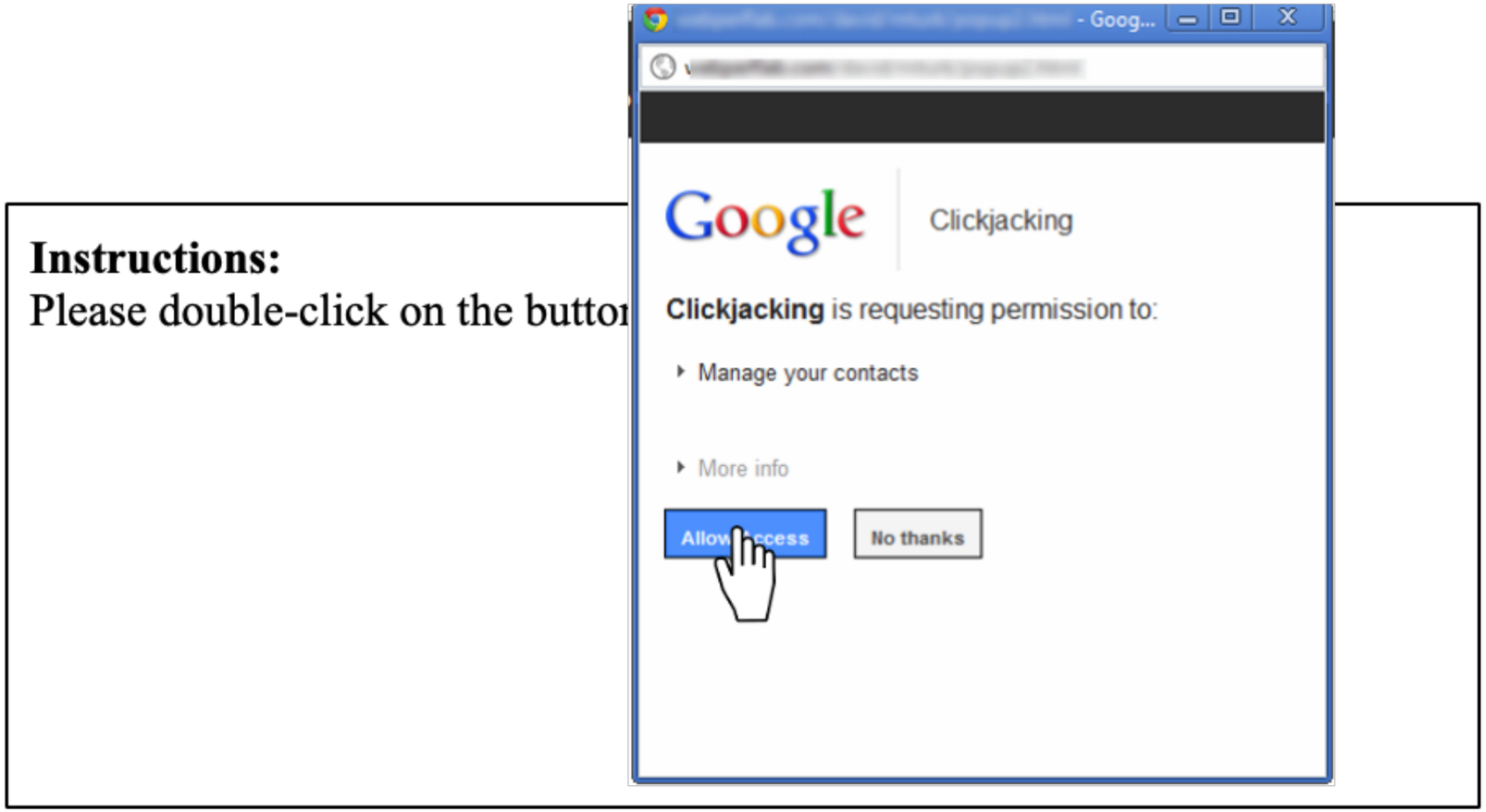
- Attacker uses JavaScript to detect the position of the cursor and change the website right before the user clicks on something
- The user clicks on the malicious input (embedded iframe, download button, etc.) before they notice that something changed

Clickjacking: Temporal Attack

Instructions:
Please double-click on the button below to continue to your content

Click here

Clickjacking: Temporal Attack



Clickjacking: Cursorjacking

Fake cursor, created with CSS and/or JavaScript

Real cursor, hidden or less visible with CSS

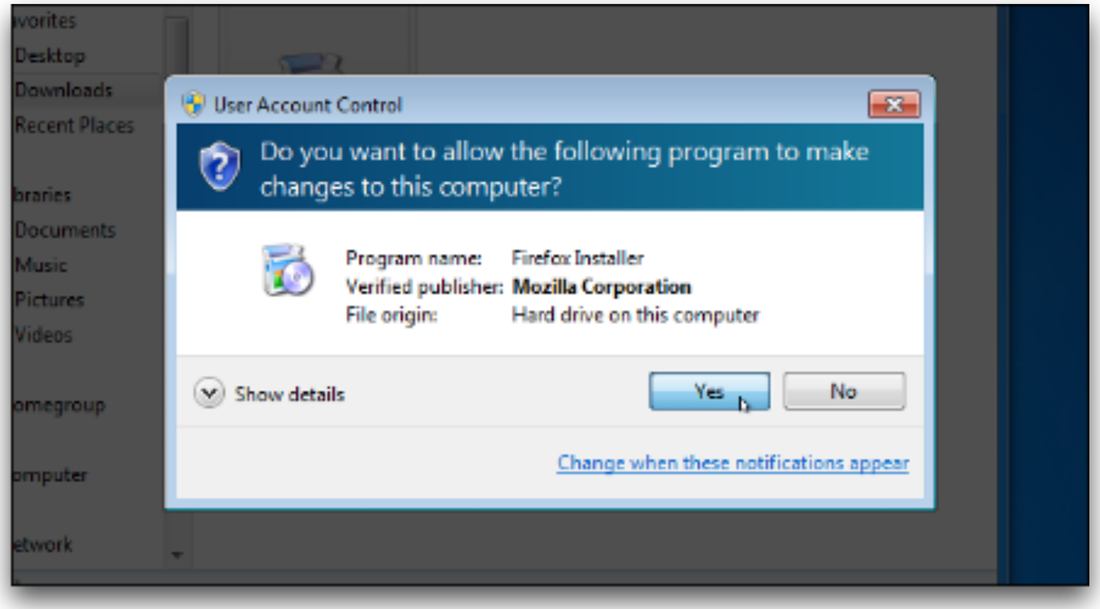
- Arrange a fixed distance between them

Clickjacking: Cursorjacking

- leads victims to misinterpret a click’s target

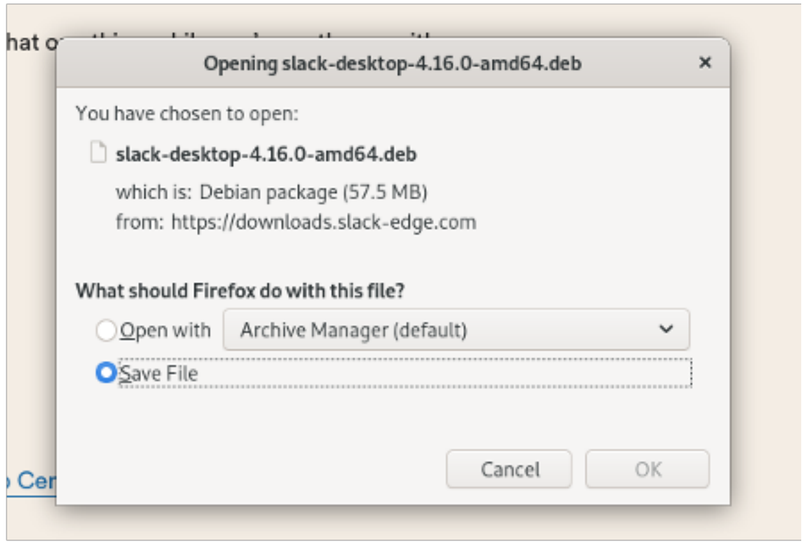
Clickjacking Defense

- **Direct the user's attention to their click:** Ensure clear visual separation between important dialogs and content, e.g., darken the background

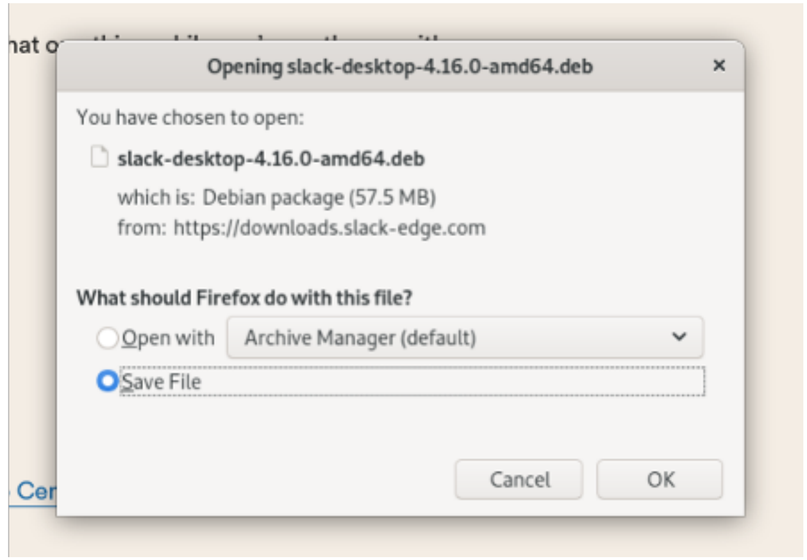


Clickjacking Defense

- **Delay the click:** Force the user to hover over the desired button for some amount of time before allowing the user to click the button.

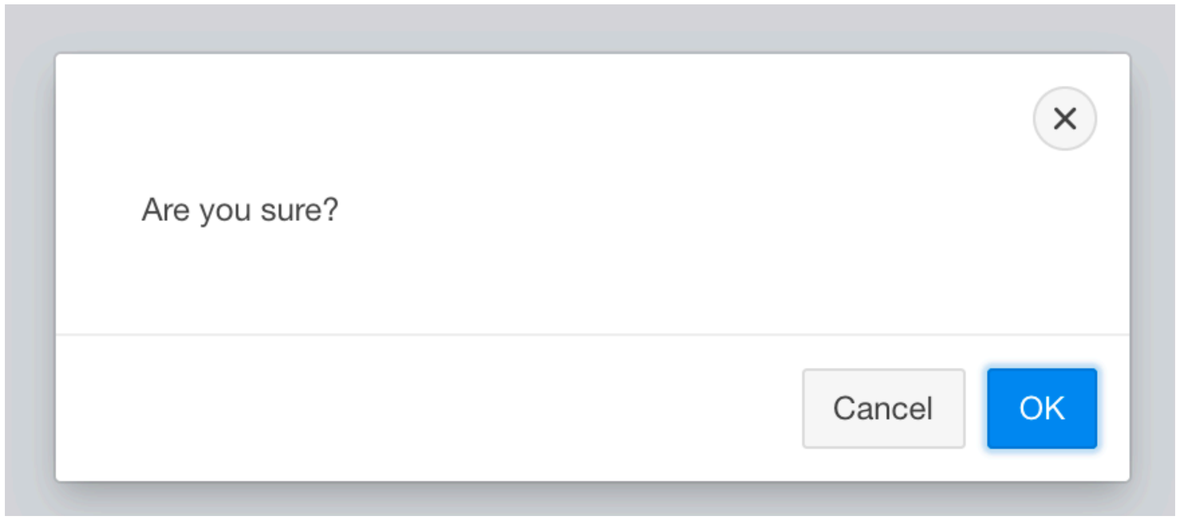


Wait 1 second before allowing click on the OK button



Clickjacking Defense

- **Confirmation pop-ups:**
  - The browser needs to confirm that the user's click was intentional
  - Drawbacks: Asking for confirmation annoys users



Clickjacking Defense

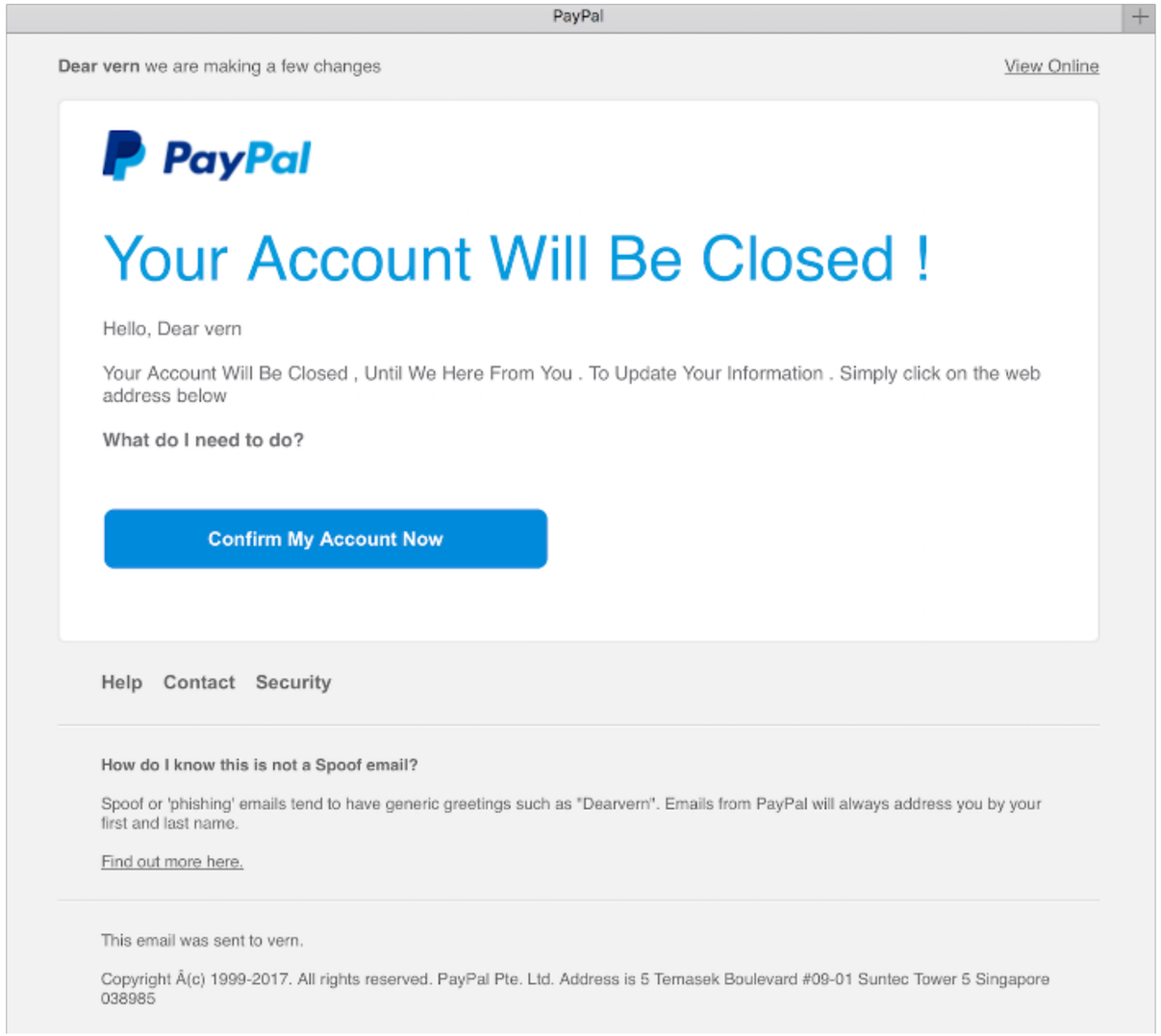
- **Frame-busting:** The legitimate website forbids other websites from embedding it in an iframe
  - Defeats the invisible iframe attacks
  - Can be enforced by Content Security Policy (CSP)
  - Can be enforced by X-Frame-Options (an HTTP header)
  - Drawbacks: relies on the end-user's browser enforcing their own security. This makes the method unreliable.

Phishing

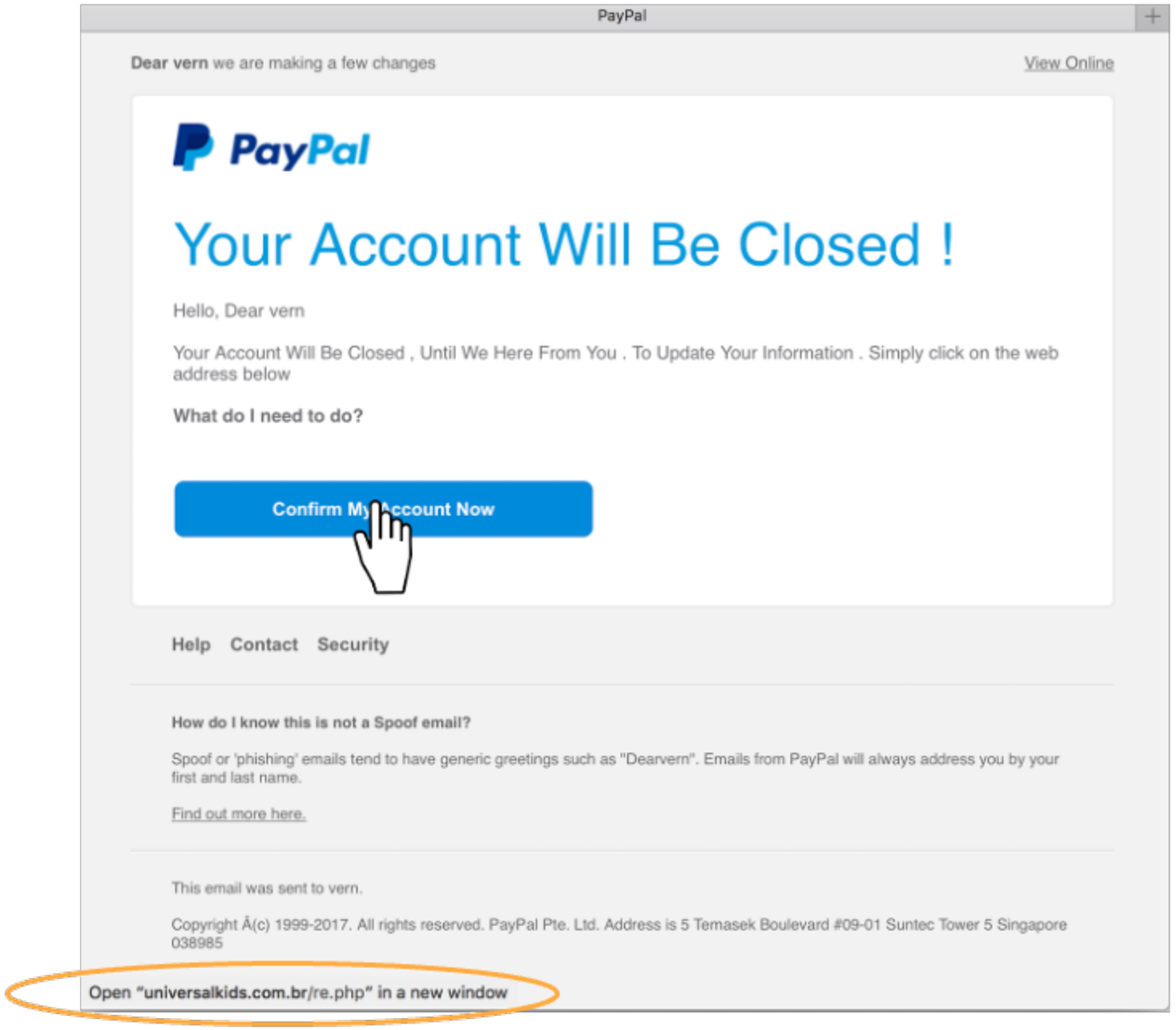
- **Phishing** is a form of social engineering and scam where attackers deceive people into revealing sensitive information or installing malware such as ransomware.

- The user can’t distinguish between a legitimate website and a website impersonating the legitimate website

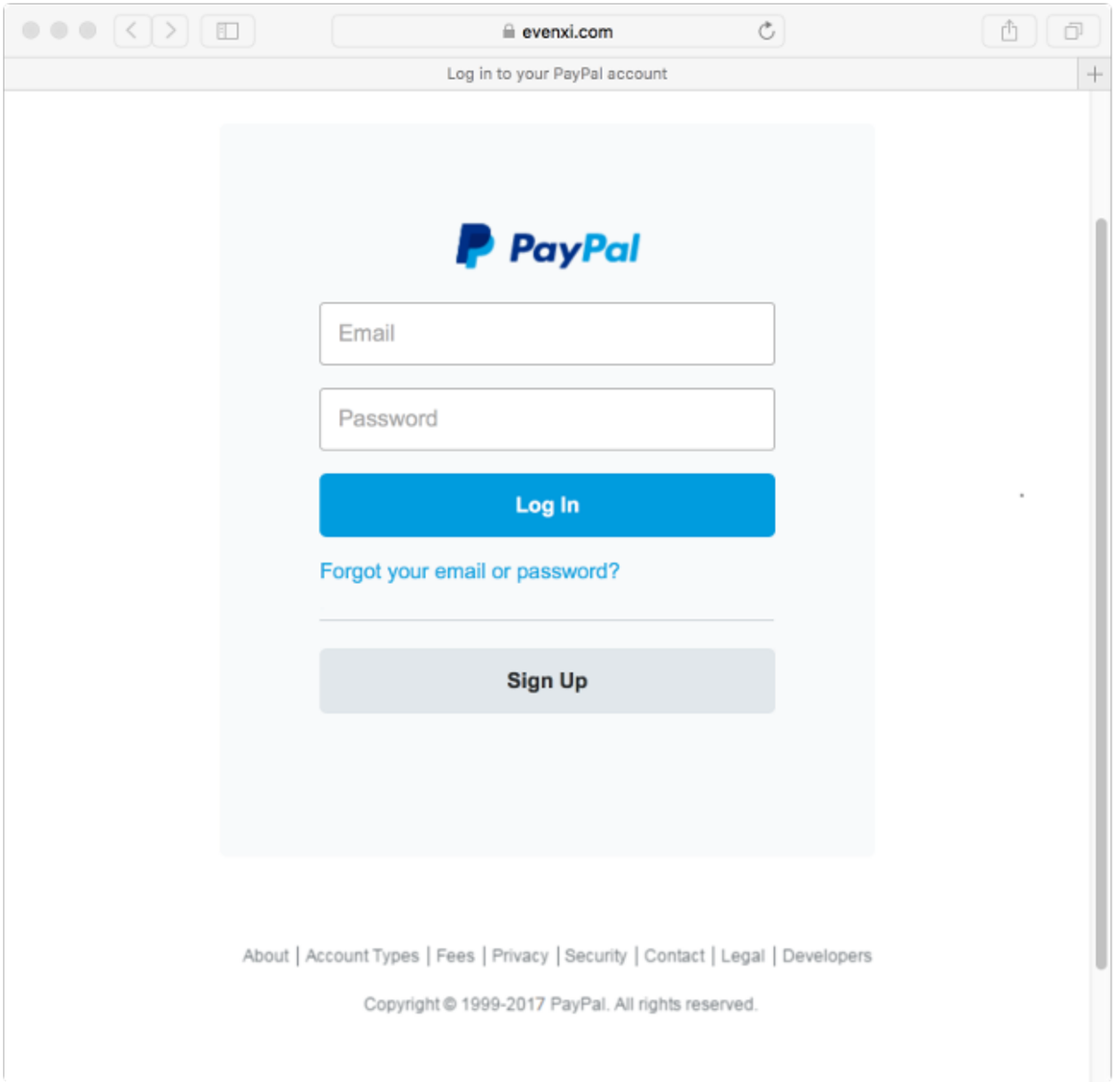
Phishing


PayPal
Dear vern we are making a few changes
View Online
PayPal
Your Account Will Be Closed !
Hello, Dear vern
Your Account Will Be Closed , Until We Here From You . To Update Your Information . Simply click on the web address below
What do I need to do?
Confirm My Account Now
Help Contact Security
How do I know this is not a Spoof email?
Spoof or 'phishing' emails tend to have generic greetings such as "Dearvern". Emails from PayPal will always address you by your first and last name.
Find out more here.
This email was sent to vern.
Copyright Â(c) 1999-2017. All rights reserved. PayPal Pte. Ltd. Address is 5 Temasek Boulevard #09-01 Suntec Tower 5 Singapore 038985

Phishing

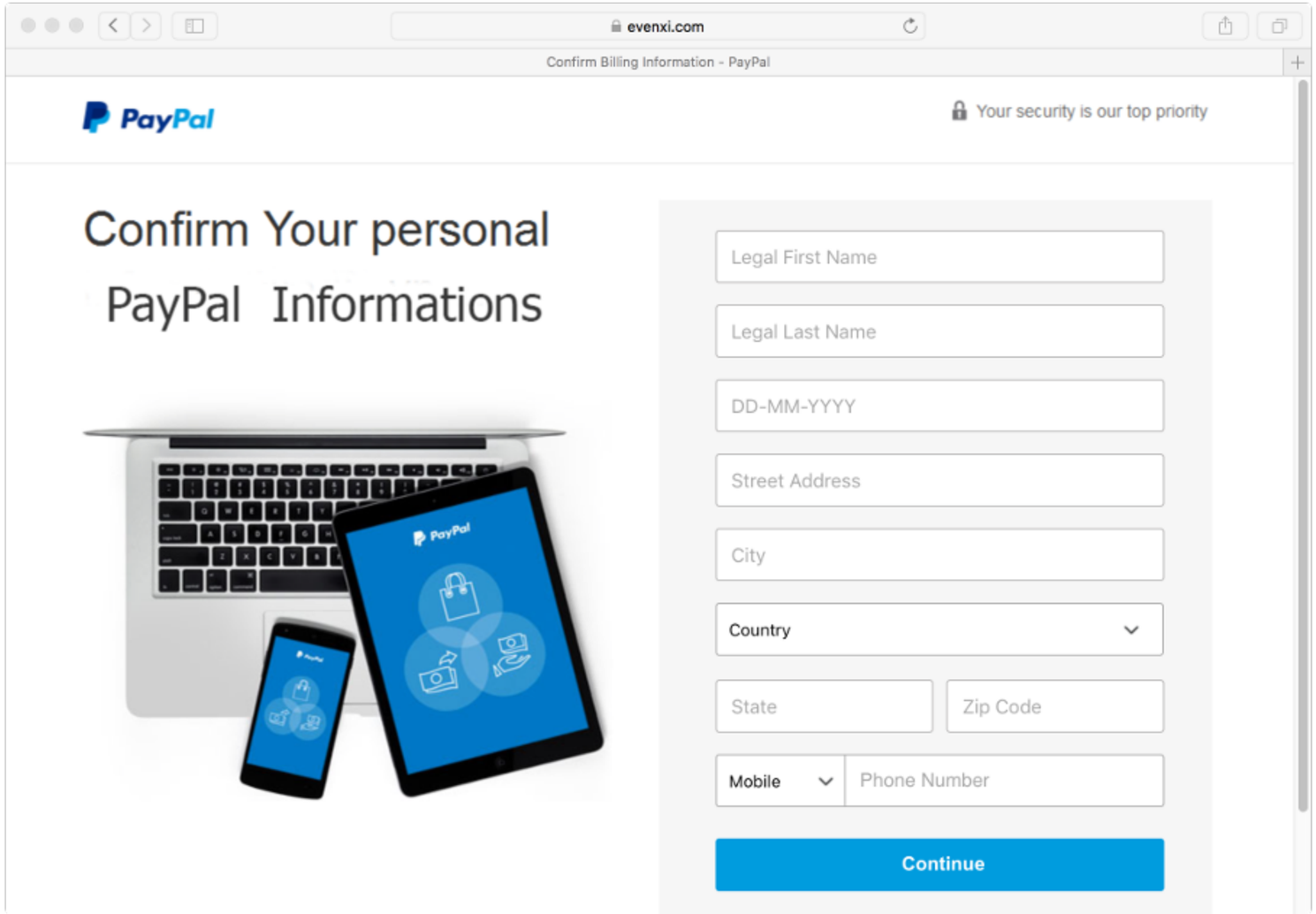

PayPal
Dear vern we are making a few changes
View Online
PayPal
Your Account Will Be Closed !
Hello, Dear vern
Your Account Will Be Closed , Until We Here From You . To Update Your Information . Simply click on the web address below
What do I need to do?
Confirm My Account Now
Help Contact Security
How do I know this is not a Spoof email?
Spoof or 'phishing' emails tend to have generic greetings such as "Dearvern". Emails from PayPal will always address you by your first and last name.
Find out more here.
This email was sent to vern.
Copyright Ã(c) 1999-2017. All rights reserved. PayPal Pte. Ltd. Address is 5 Temasek Boulevard #09-01 Suntec Tower 5 Singapore 038985
Open "universalkids.com.br/re.php" in a new window

Phishing



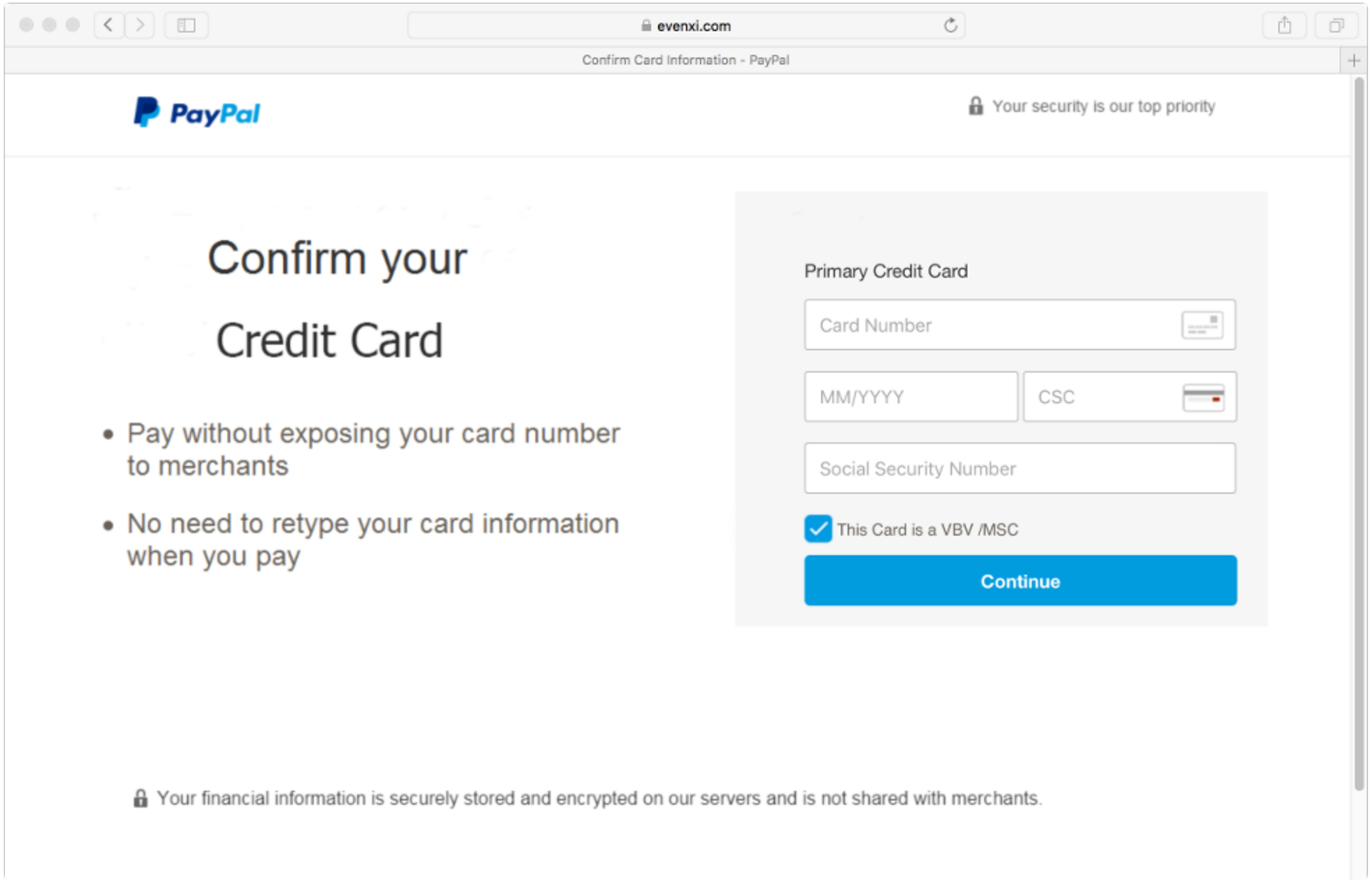
Is this PayPal?

Phishing



After filling out the previous boxes

Phishing

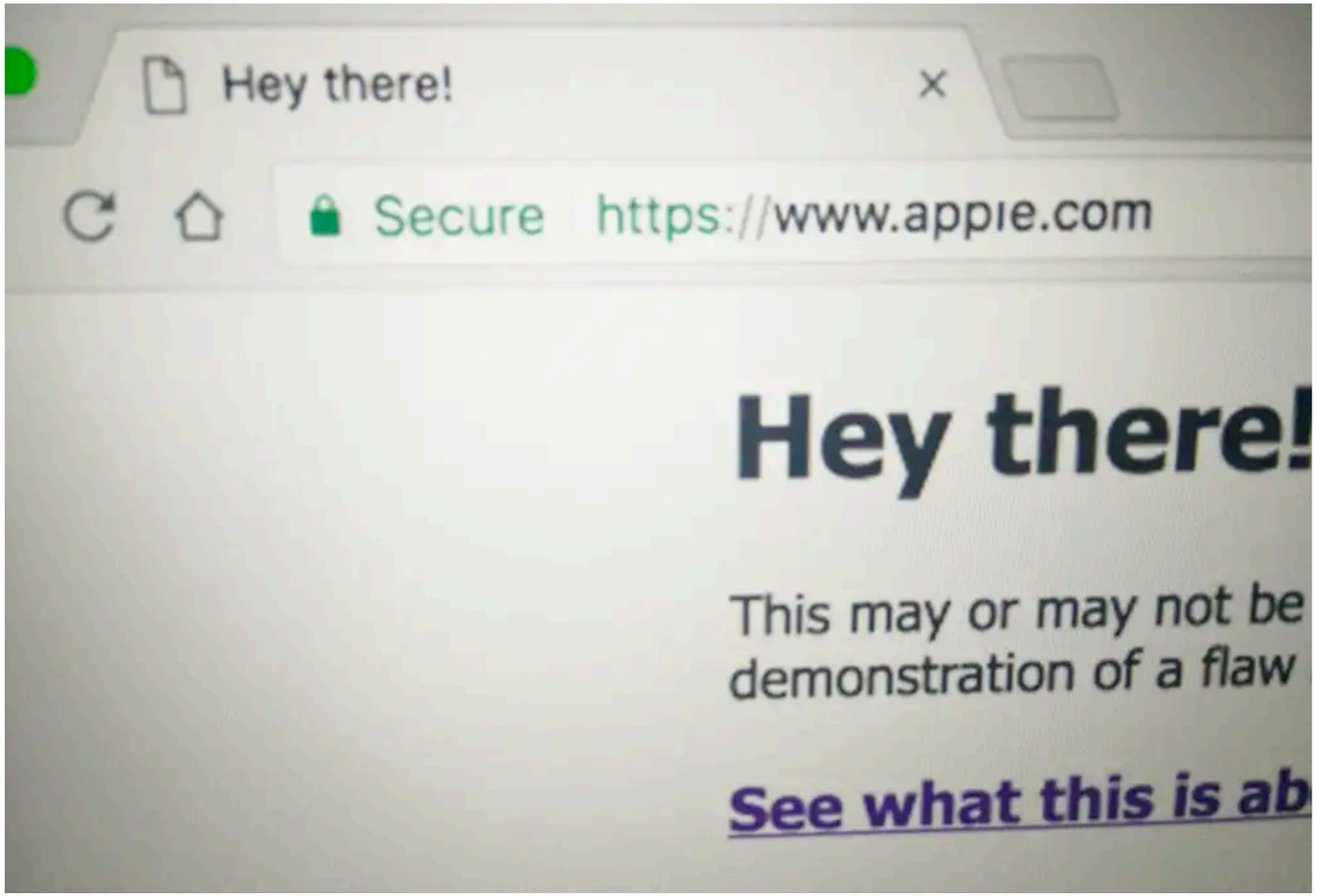


After filling out the previous boxes

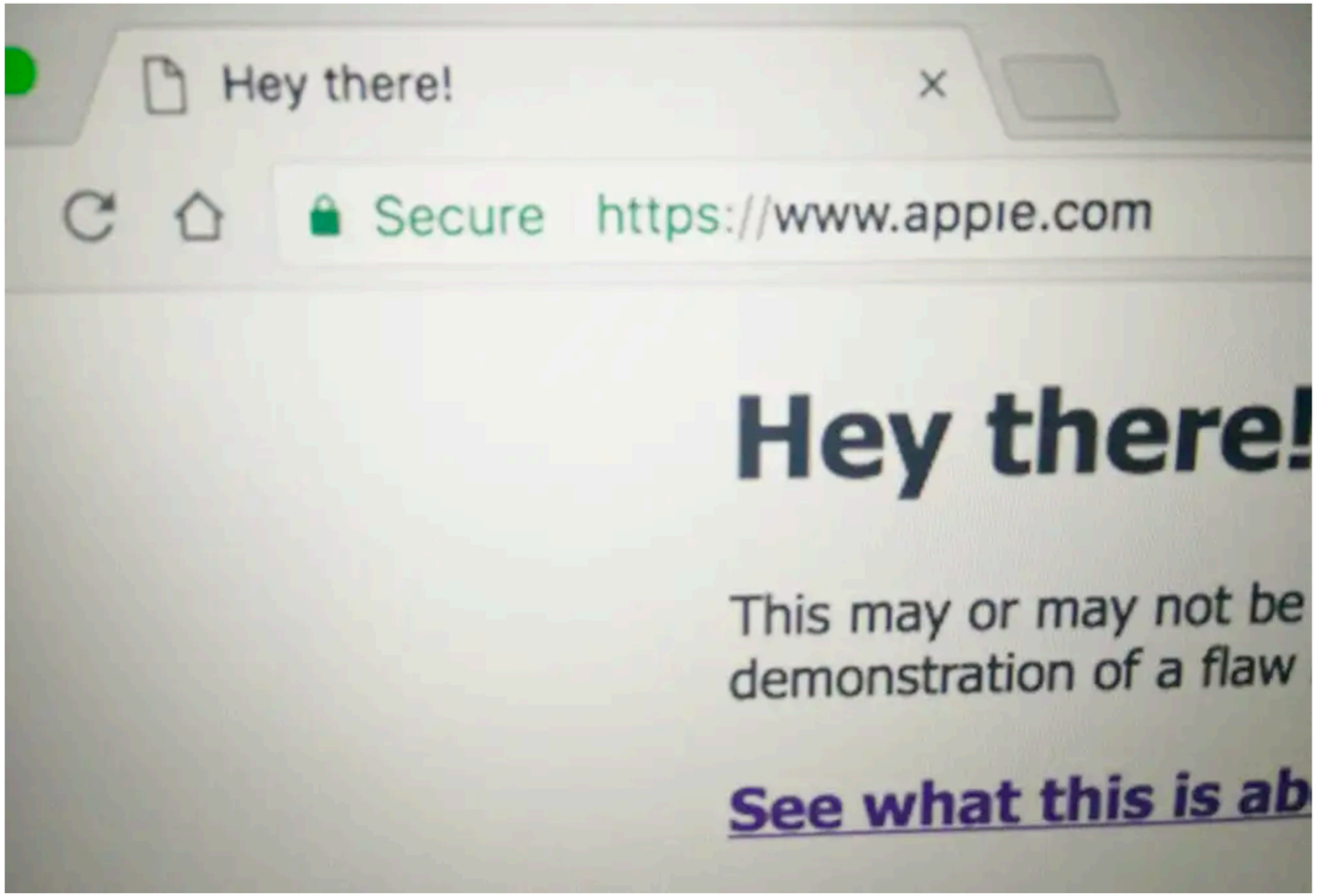
Phishing: Homograph Attacks

- Homograph: Two words that look the same, but have different meanings
- Homograph attack: Creating malicious URLs that look similar (or the same) to legitimate URLs

Phishing: Homograph Attacks



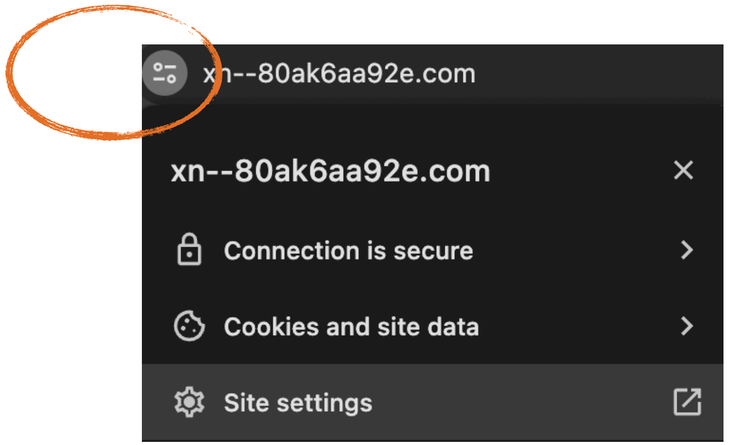
Phishing: Homograph Attacks



- Cyrllic alphabet
- Written in unicode
- Certificate under xn--80ak6aa92e.com
- Looks more real in some browsers

https://www.xudongz.com/blog/2017/idn-phishing/

Phishing: Homograph Attacks



Certificate Viewer: www.xn--80ak6aa92e.com

General | Details

Issued To

Common Name (CN)	www.xn--80ak6aa92e.com
Organization (O)	<Not Part Of Certificate>
Organizational Unit (OU)	<Not Part Of Certificate>

Issued By

Common Name (CN)	R3
Organization (O)	Let's Encrypt
Organizational Unit (OU)	<Not Part Of Certificate>

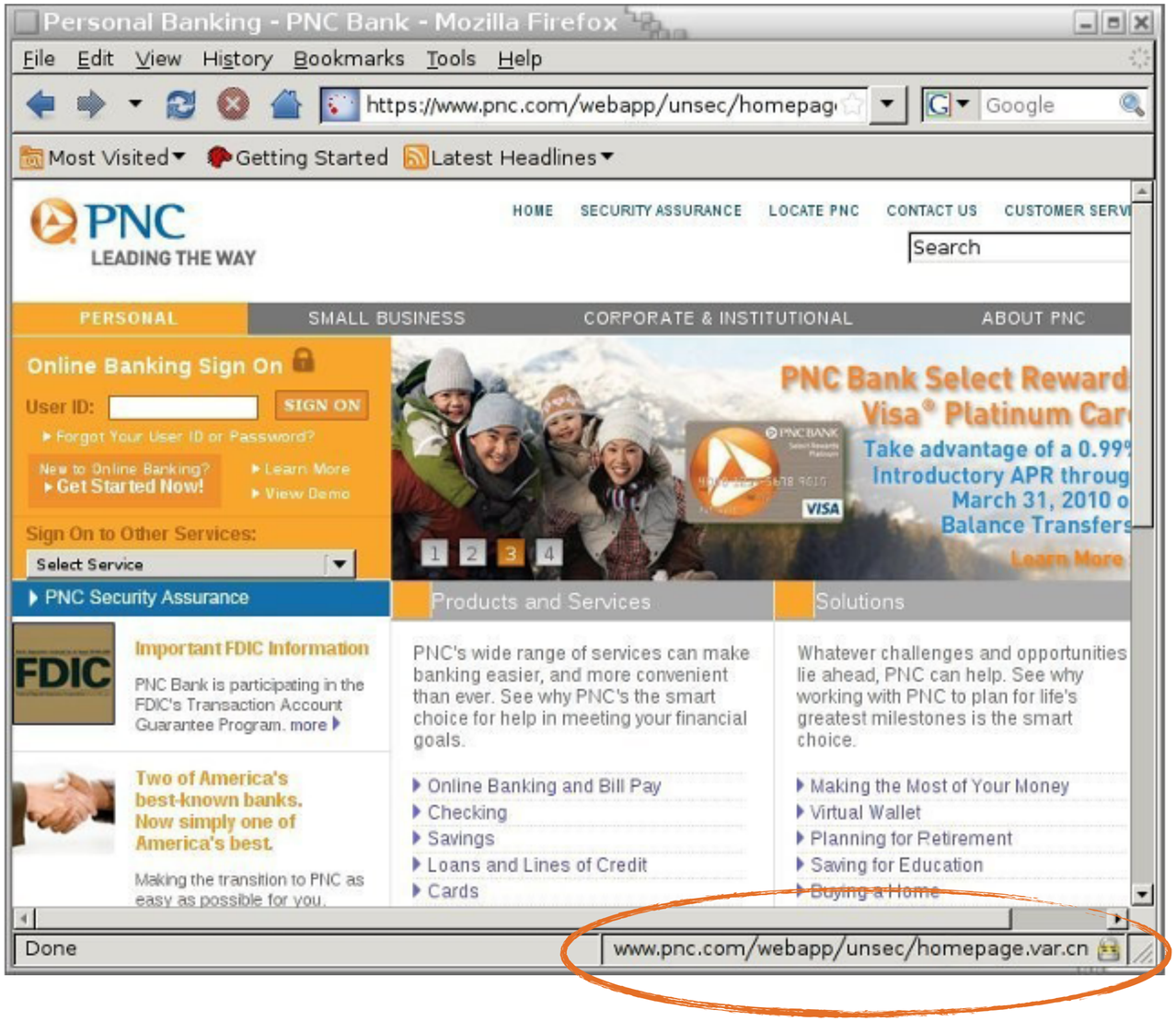
Validity Period

Issued On	Tuesday, February 13, 2024 at 6:40:10 AM
Expires On	Monday, May 13, 2024 at 7:40:09 AM

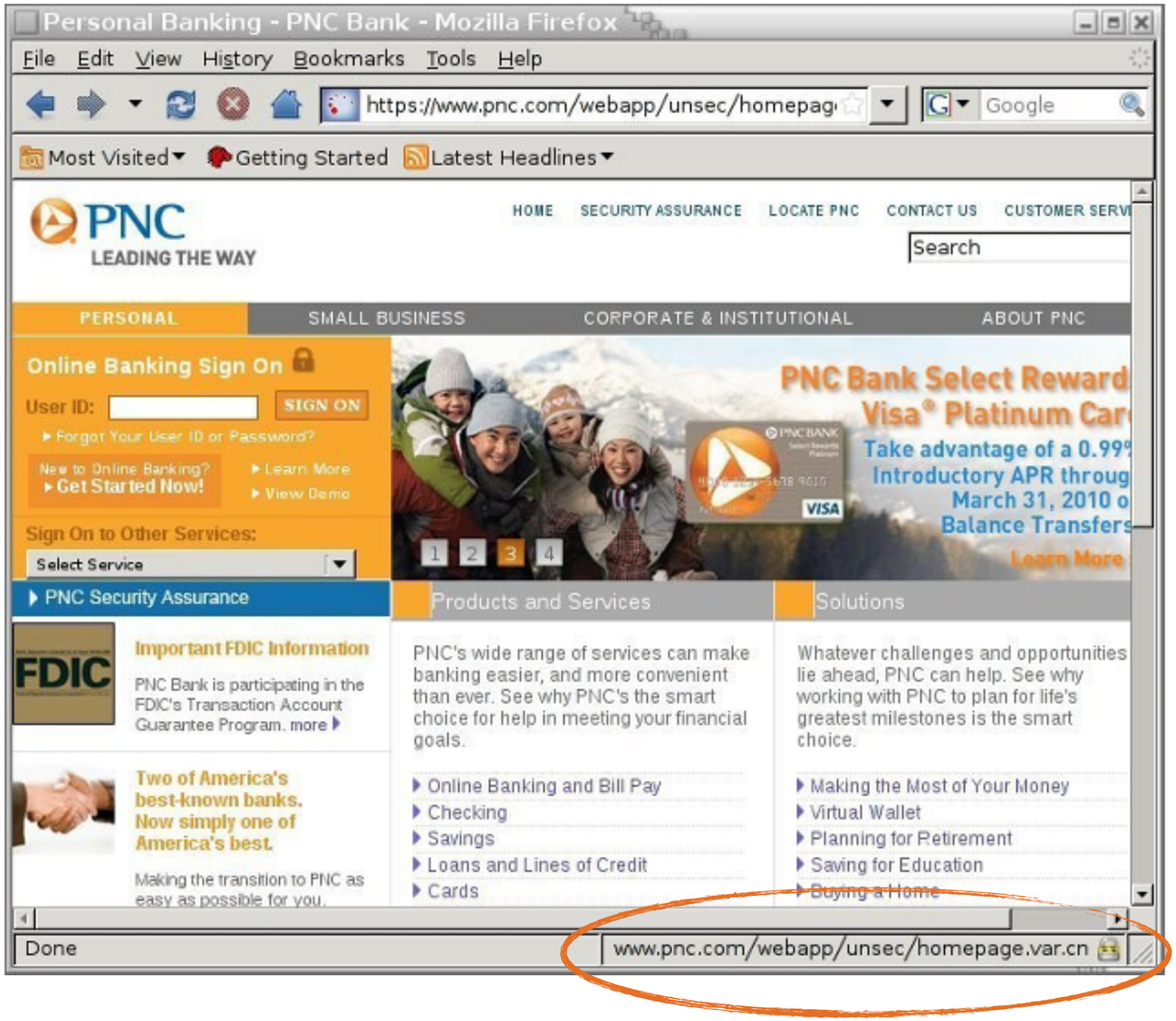
SHA-256 Fingerprints

Certificate	35ba295b1ef463f81382aae94f78046f4597bee69b5ed608a6114277702fef86
Public Key	eec04ffa1e48615fa9cf2b0b728c7543415a6a7a2691dabb08f8a7991b71e50c

Phishing: Homograph Attacks

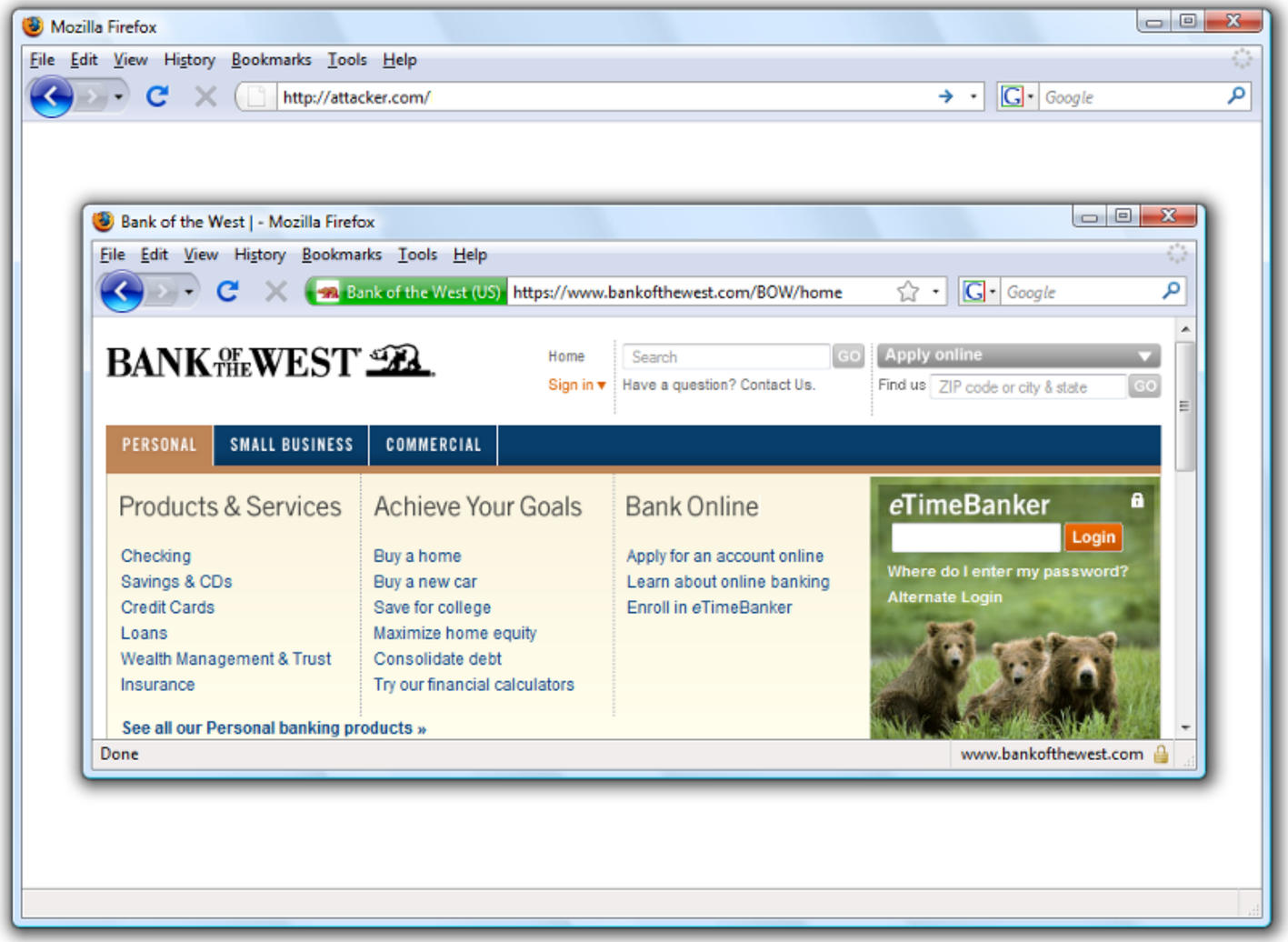


Phishing: Homograph Attacks



- Unicode characters 2044 (⁄) and 2215 (∕) are allowed in hostnames.
- Confusing chars

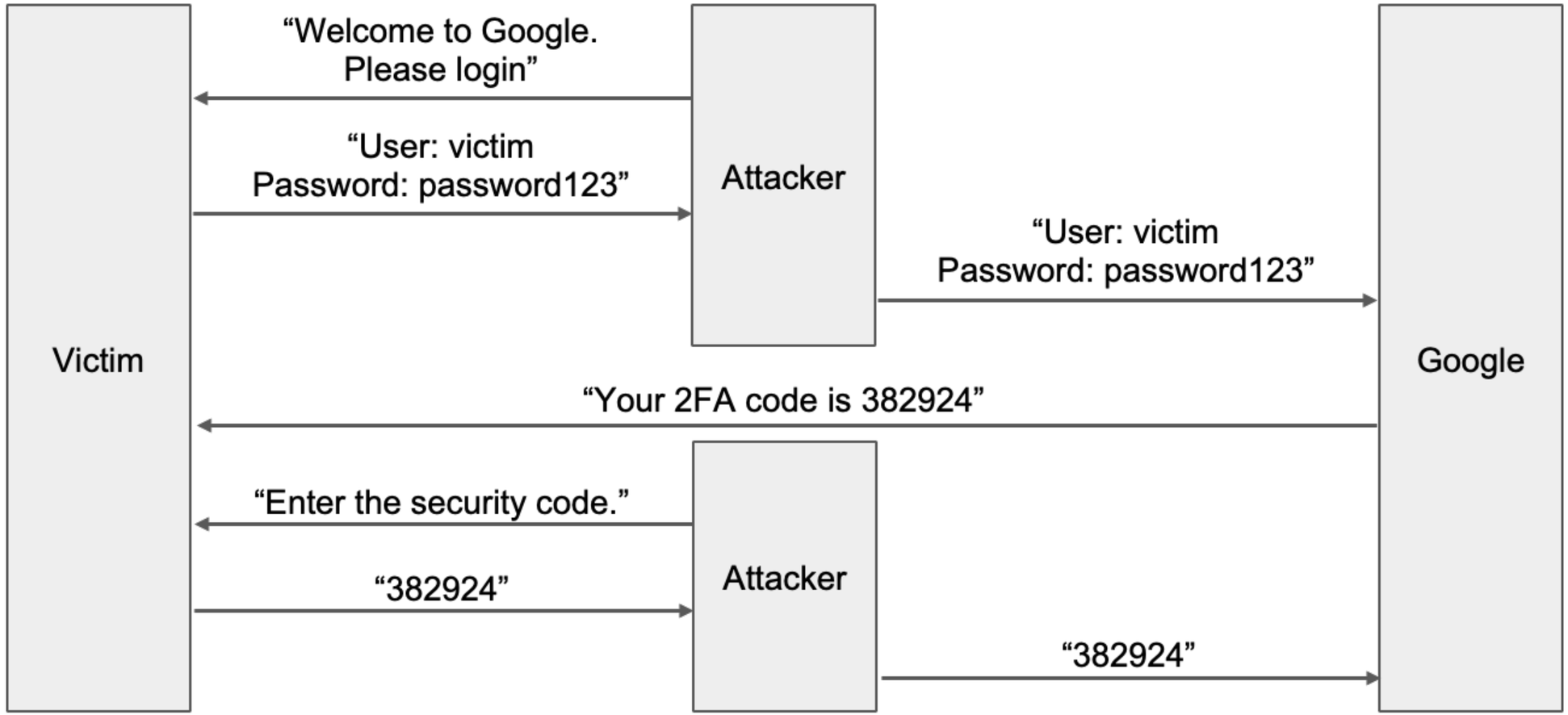
Phishing: Browser in Browser Attack



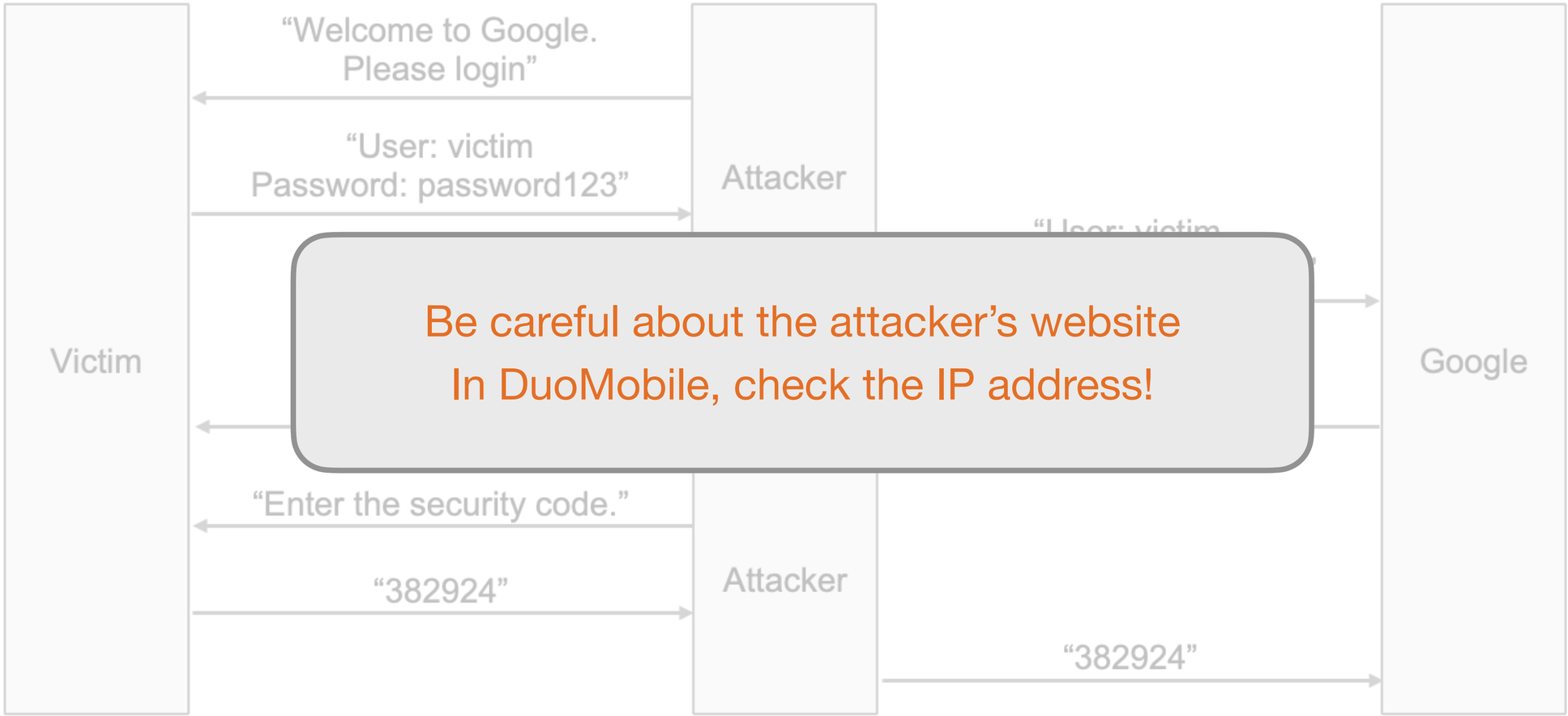
Two-Factor Authentication

- Problem: Phishing attacks allow attackers to learn passwords
- Idea: Require more than passwords to log in
- **Two-factor authentication (2FA):** The user must prove their identity in two different ways before successfully authenticating
- Three main ways for a user to prove their identity
  - **Something the user knows:** Password, security question (e.g. name of your first pet)
  - **Something the user has:** Their phone, their security key
  - **Something the user is:** Fingerprint, face ID
- Even if the attacker steals the user's password with phishing, they don't have the second factor!

Subverting 2FA: Relay Attacks / MiTM (Man-in-the-middle)



Subverting 2FA: Relay Attacks / MiTM



2FA Example: Authentication Tokens

- Authentication token: A device that generates secure second-factor codes (Something the user owns)
- Examples: RSA SecurID, Google Authenticator, DuoMobile
  - The token and the server share a common secret key k
  - When the user wants to log in, the token generates a code HMAC(k, time)
  - The user submits the code to the website
  - The website uses its secret key to verify the HMAC
- Drawback: Vulnerable to online brute-force attacks; Possible fix: Add a timeout
- Drawback: Vulnerable to relay attacks; Fix: User needs to be more careful, read the IP address

Subverting 2FA: Social Engineering

- Some 2FA schemes text a one-time code to a phone number
  - Attackers can call your phone provider (e.g. Verizon) and tell them to activate the attacker's SIM card, so they receive your texts!
  - 2FA via SMS is not great but better than nothing
- Some 2FA schemes can be bypassed with customer support
  - Attackers can call customer support and ask them to deactivate 2FA!
  - Companies should validate identity if you ask to do this (but not all do)

Agenda

- Recap
- UI Attacks
- CAPTCHAs
- Security Principles

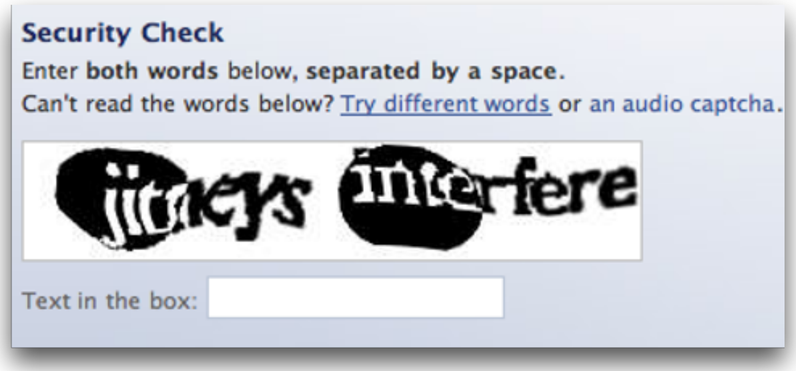
Websites are for Humans

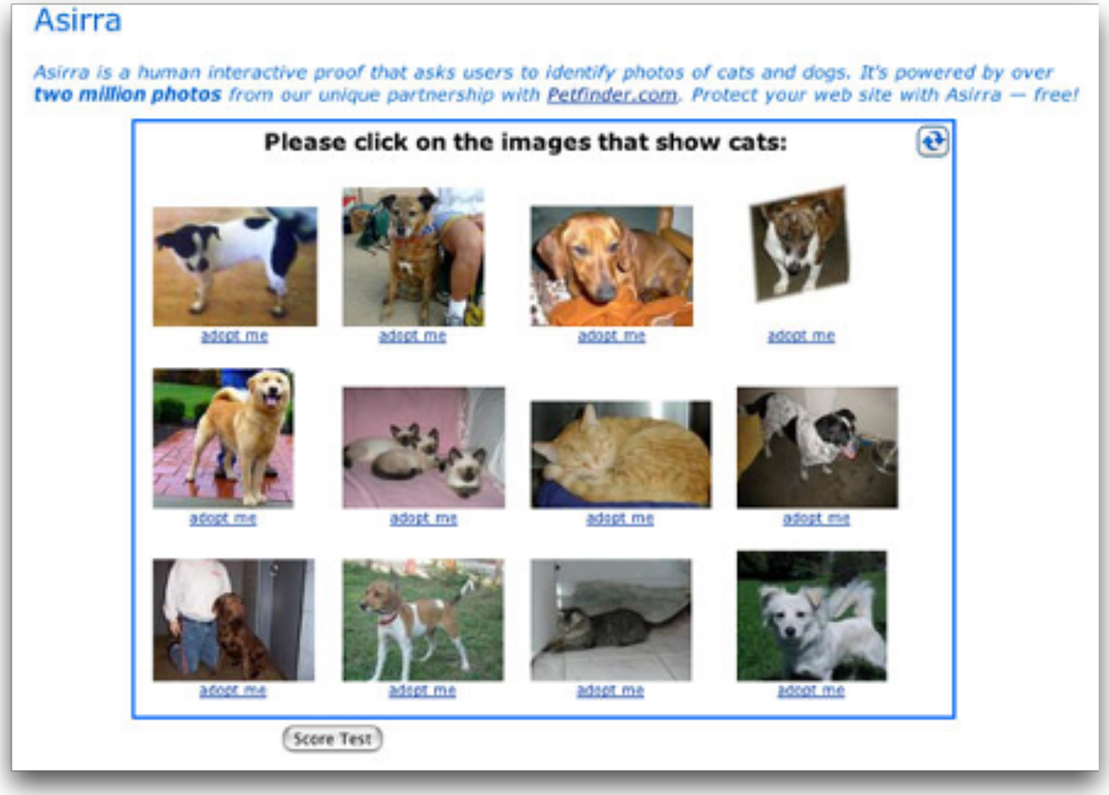
- Most websites are designed for human usage, not robot usage
  - Example: A login page is for users to submit their password, not for an attacker to automate a brute-force attack
- Robot access of websites can lead to attacks
  - Example: Denial of service: Overwhelming a web server by flooding it with requests

CAPTCHAs: Definition

- **CAPTCHA:** A challenge that is easy for a human to solve, but hard for a computer to solve
  - "**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part"
  - Sometimes called a "reverse Turing test"
  - Used to distinguish web requests made by humans and web requests made by robots
- Usage: Administer a CAPTCHA, and if it passes, assume that the user is human and allow access

CAPTCHAs: Examples

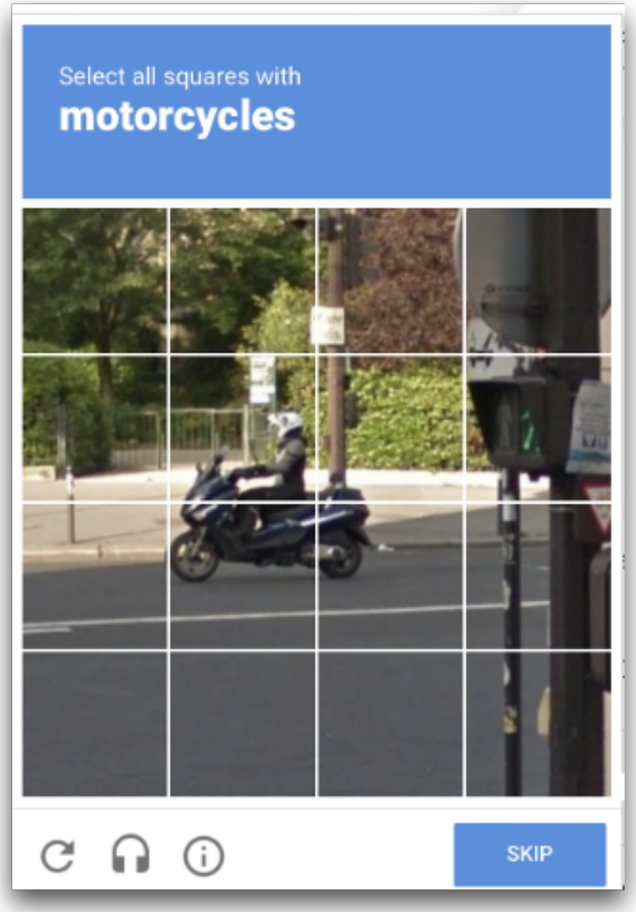




- Reading distorted text
- Identifying images
- Listening to an audio clip and typing out the words spoken

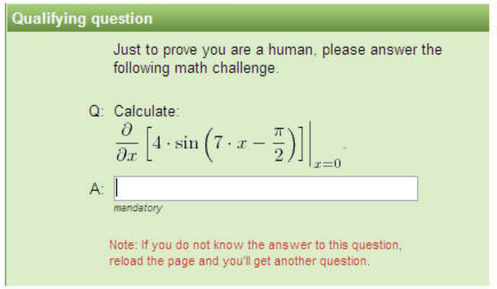
CAPTCHAs and Machine Learning

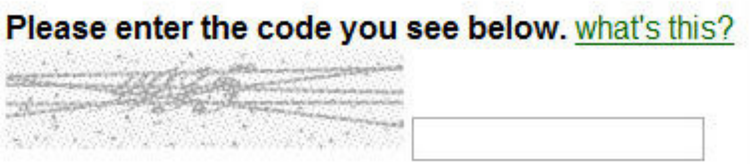
- Modern CAPTCHAs have another purpose: Training machine learning algorithms
  - Machine learning often requires manually-labeled datasets
  - CAPTCHAs crowdsource human power to help manually label these big datasets
  - Example: Machine vision problems require manually-labeled examples: “This is a stop sign”



CAPTCHAs: Issues

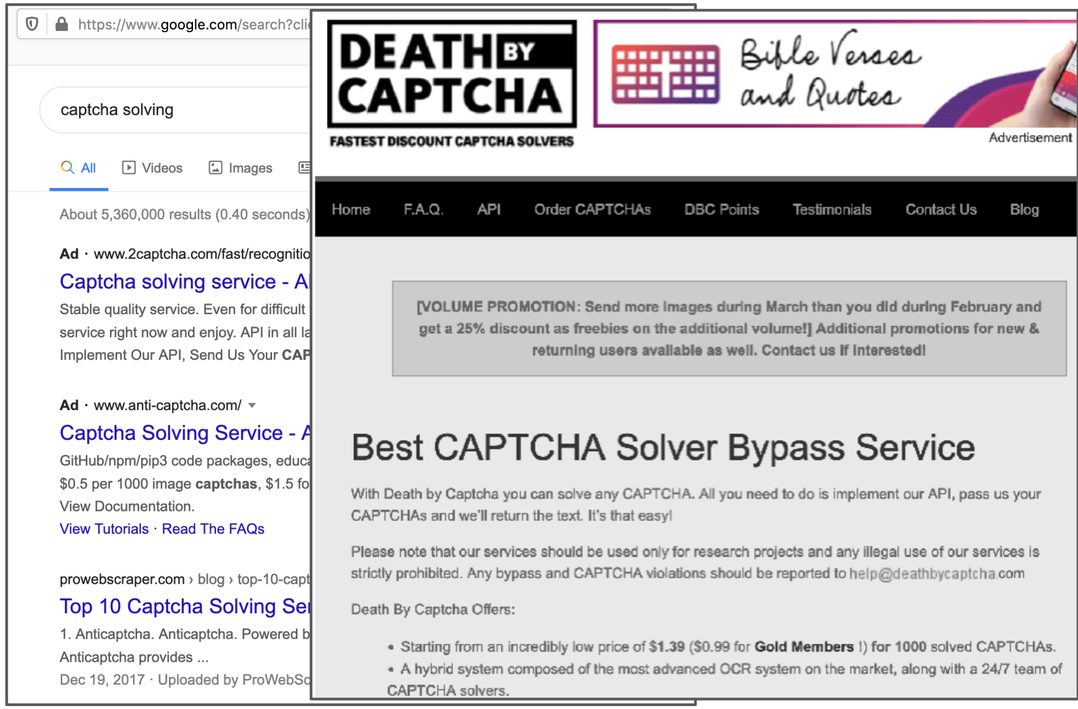
- Arms race: As computer algorithms get smarter, CAPTCHAs need to get harder
- Accessibility: As CAPTCHAs get harder, not all humans are able to solve them easily
- Ambiguity: CAPTCHAs might be so hard that the validator doesn't know the solution either!
- Not all bots are bad: CAPTCHAs can distinguish bots from humans, but not good bots from bad bots
  - Example: Crawler bots help archive webpages





CAPTCHAs: Attacks

- Outsourcing attack: Pay humans to solve CAPTCHAs for you
  - CAPTCHAs only verify that there is a human in the loop; everything else can be automated
  - Usually costs a few cents per CAPTCHA
  - CAPTCHAs end up just distinguishing which attackers are willing to spend money
    - Security is economics!



Agenda

- Recap
- UI Attacks
- CAPTCHAs
- Security Principles

What is Security?

- Enforcing a desired property in the presence of an attacker
  - Data **Confidentiality**
  - Data and Computation **Integrity**
  - **Authenticity**
  - **Availability**
  - User **Privacy**
  - …

Why is Security Important?

- It is important for our
  - physical safety
  - confidentiality/privacy
  - functionality
  - protecting our assets
  - successful business
  - a country’s economy and safety
  - and so on…

Three Main Goals of Cryptography

- In cryptography, there are three common properties that we want on our data
- **Confidentiality**: An adversary cannot read our messages.
- **Integrity**: An adversary cannot change our messages without being detected.
- **Authenticity**: I can prove that this message came from the person who claims to have written it.

Authenticity vs Authentication

- **Authenticity**: I can prove that this message came from the person who claims to have written it.
- **Authentication**: verification of identity (are you who you say you are). Examples include username/password and biometrics.

Adversarial Thinking

“Security requires a particular mindset. Security professionals -- at least the good ones -- see the world differently. They can't walk into a store without noticing how they might shoplift. They can't use a computer without wondering about the security vulnerabilities. They can't vote without trying to figure out how to vote twice. They just can't help it.”

–Bruce Schneier

Key Questions

- **Security Goals**
- **Threat model**: who is the adversary? What actions can the adversary perform? What kind of access and resources does the adversary have?
- **Mechanisms**: What security mechanisms can be used to achieve the security goals given the adversarial model

Exercise: CAPTCHA

- Security Goals:
- Threat model:
- Mechanisms:

Is it secure now?

What does it mean to be secure?

- Too difficult for attackers
- Too expensive
- Lower ROI than the next target
- …
- We raise the bar for attackers to succeed

Security is Economics!