Information Security

Networking 2: With A Single Click

Winter 2023/2024

Jakob Heher, www.iaik.tugraz.at
he/his

Recall from last time

Lecture ground rules

- We color technologies, algorithms, etc. for your convenience
  - State-of-the-art tech, no known vulnerabilities ✔
    - This is generally safe to use!
  - Outdated tech, known issues, covered for demonstration purposes ✗
    - You should not use this!

- Coloring provides a very quick-and-dirty categorization for you
  - Want to know *why*? That's what the lecture is for ☺

Recall from last time

Meet the players

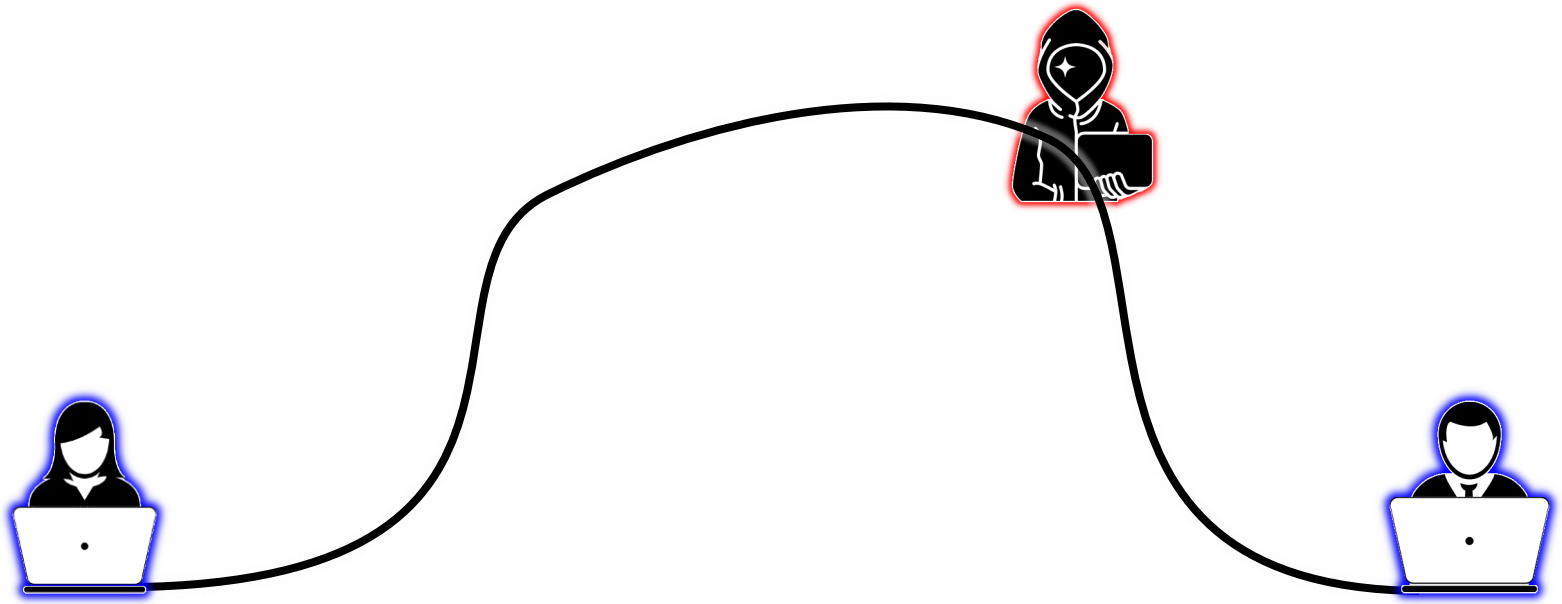
Alice
she/hers

Bob
he/his

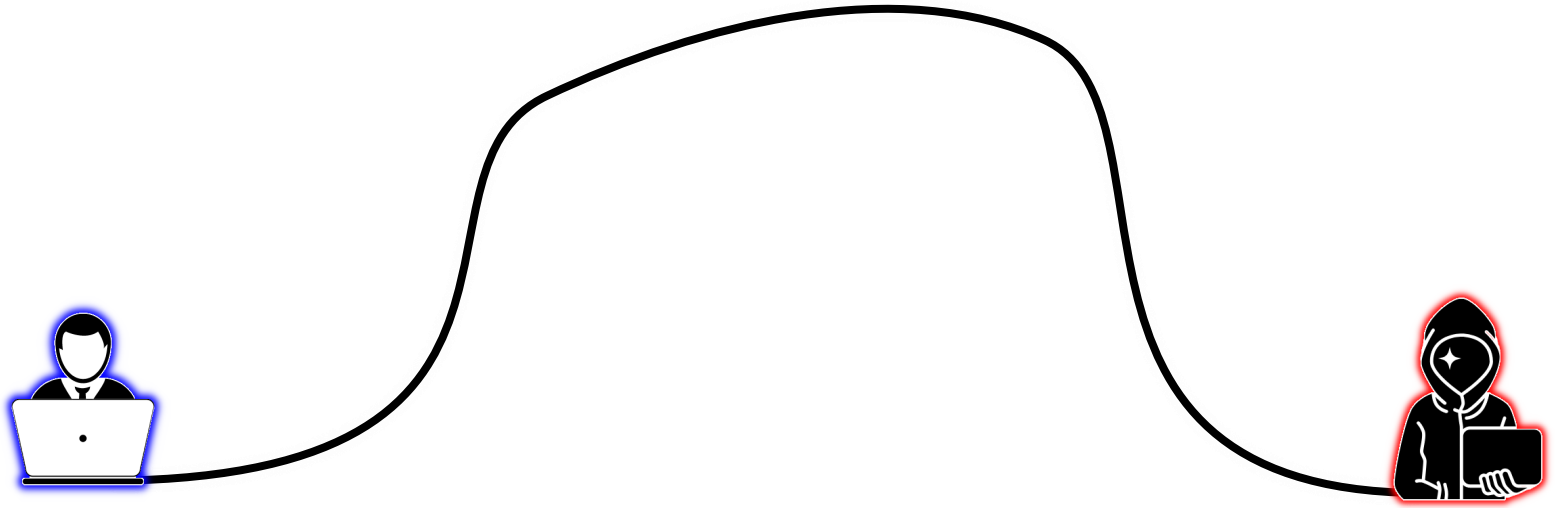
Eve
????

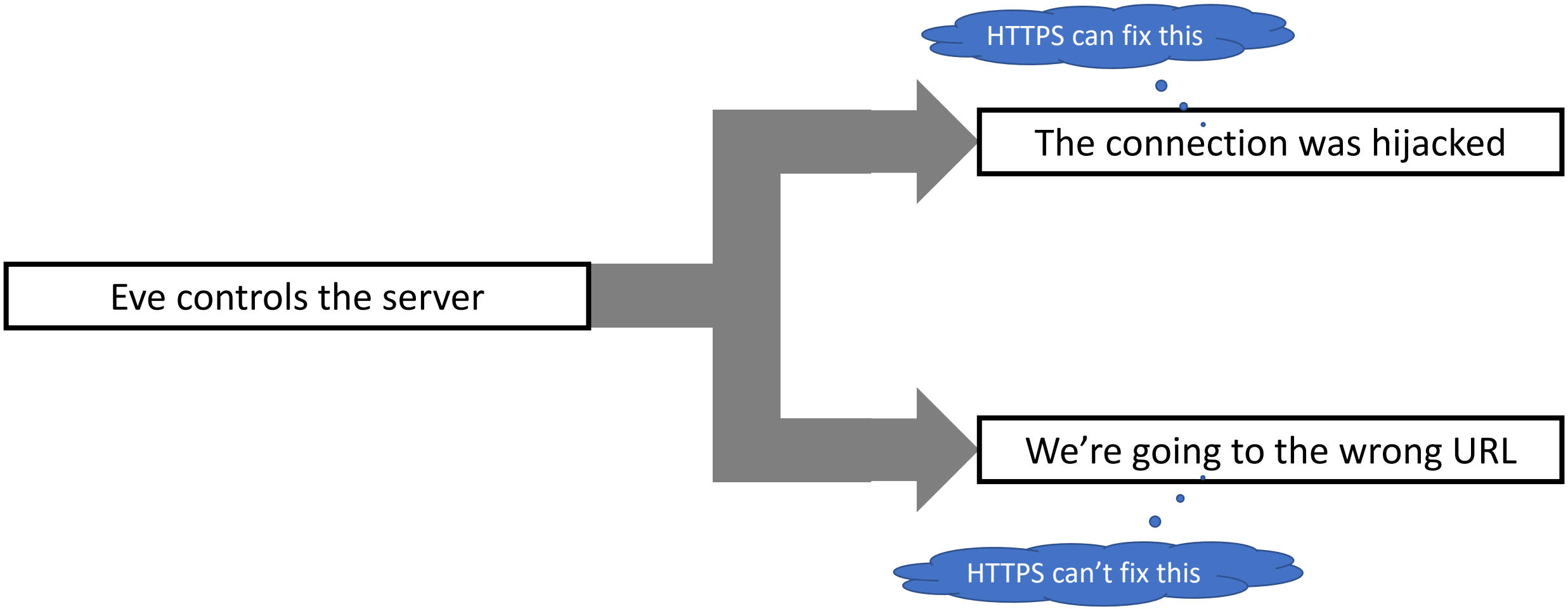
Smith
she/hers

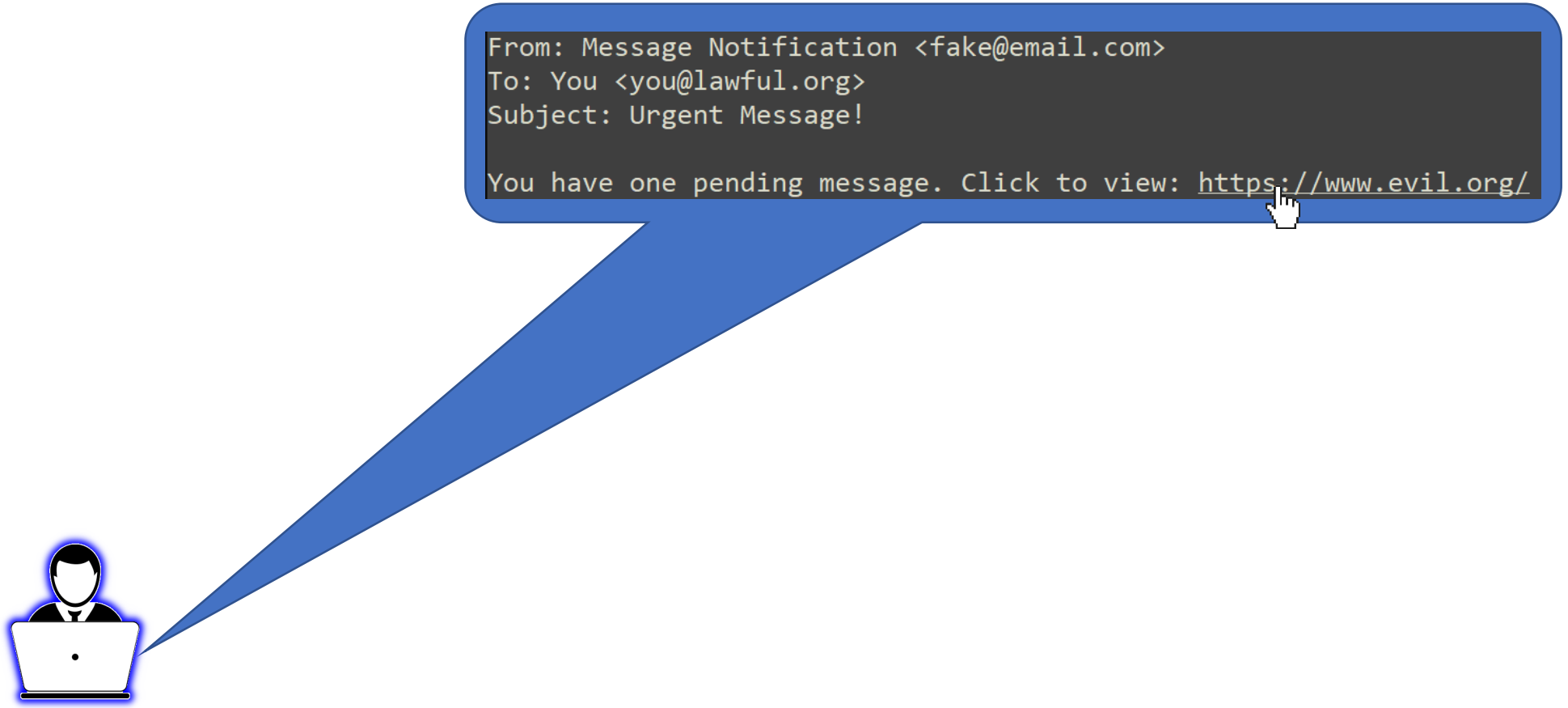
Last time:

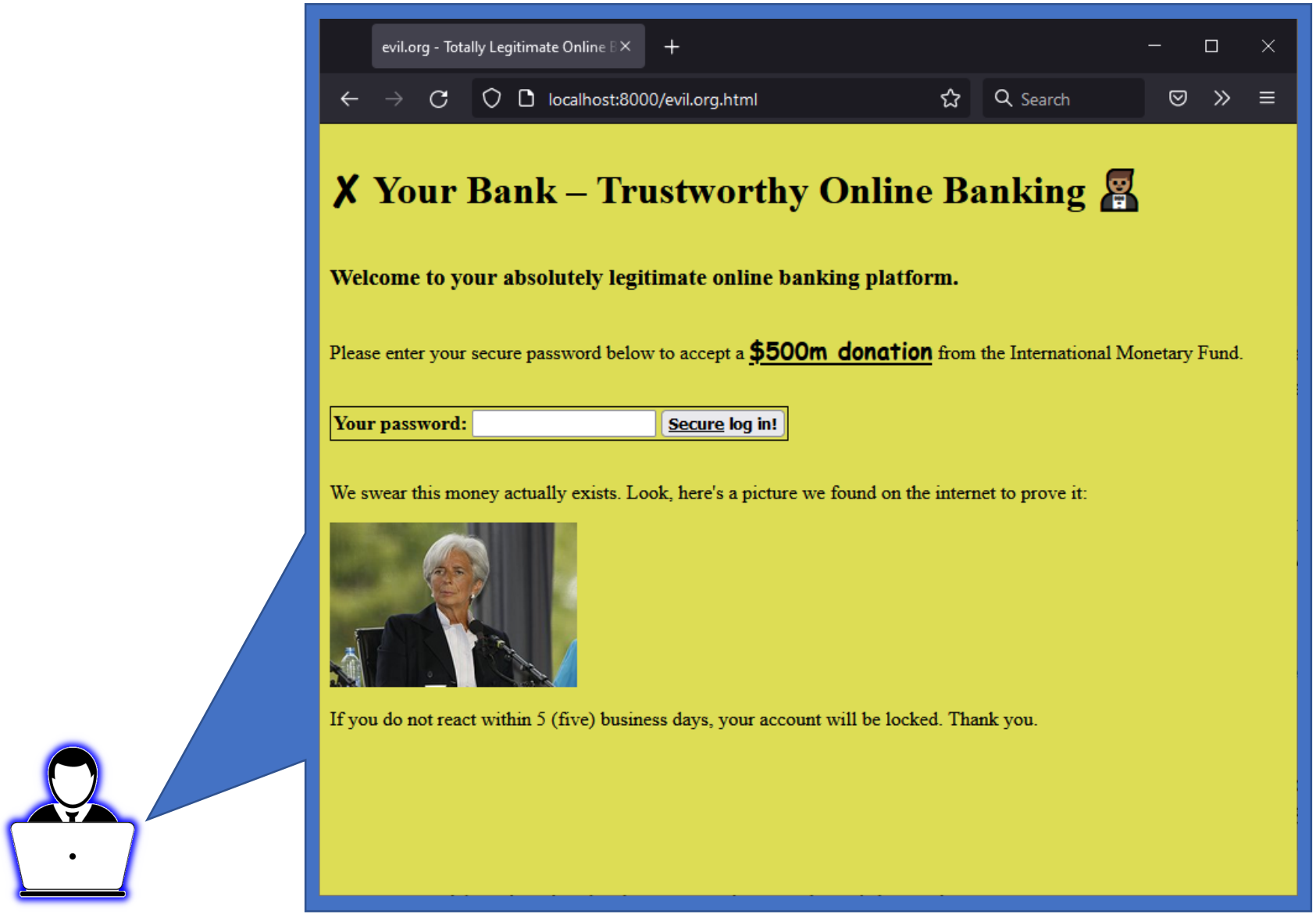


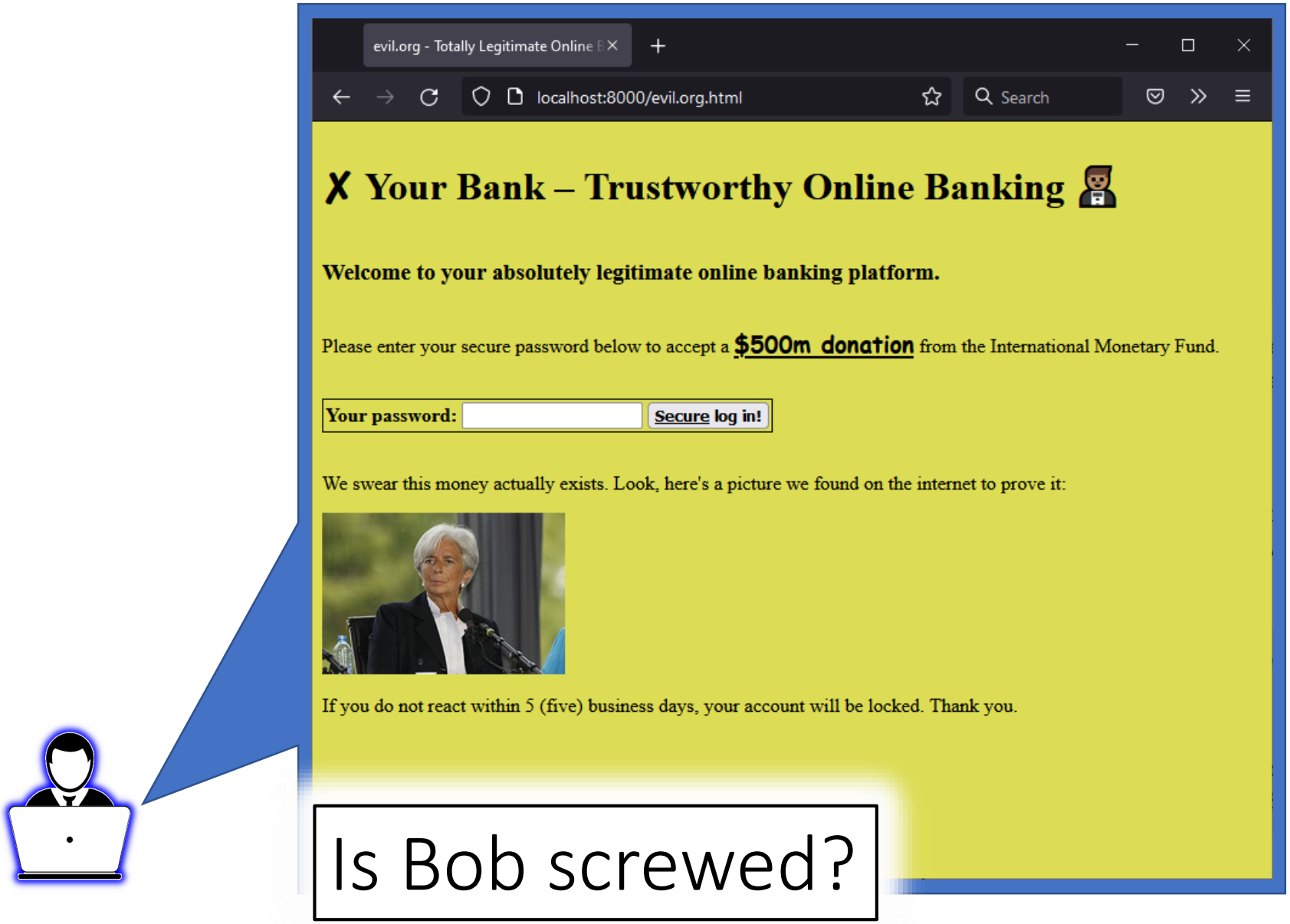
This time:



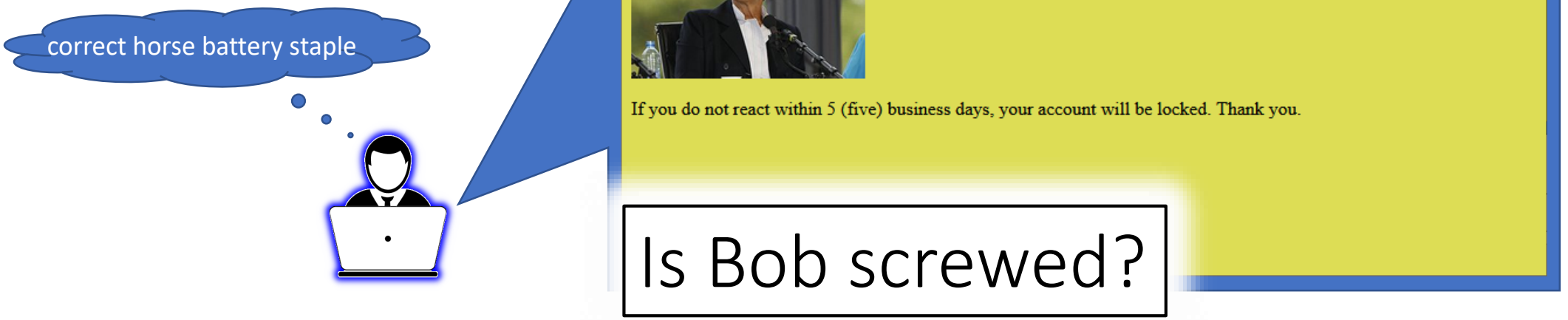

HTTPS can fix this
The connection was hijacked
Eve controls the server
We're going to the wrong URL
HTTPS can't fix this

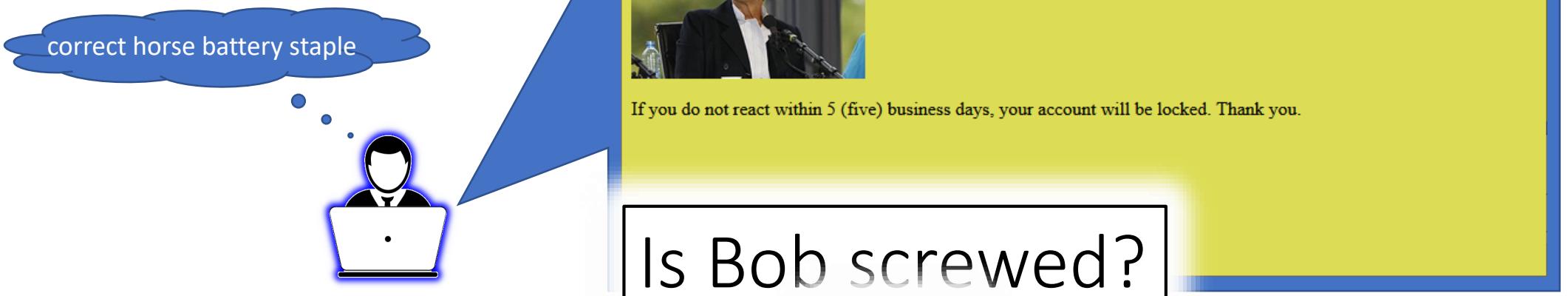

From: Message Notification <fake@email.com>
To: You <you@lawful.org>
Subject: Urgent Message!
You have one pending message. Click to view: https://www.evil.org/





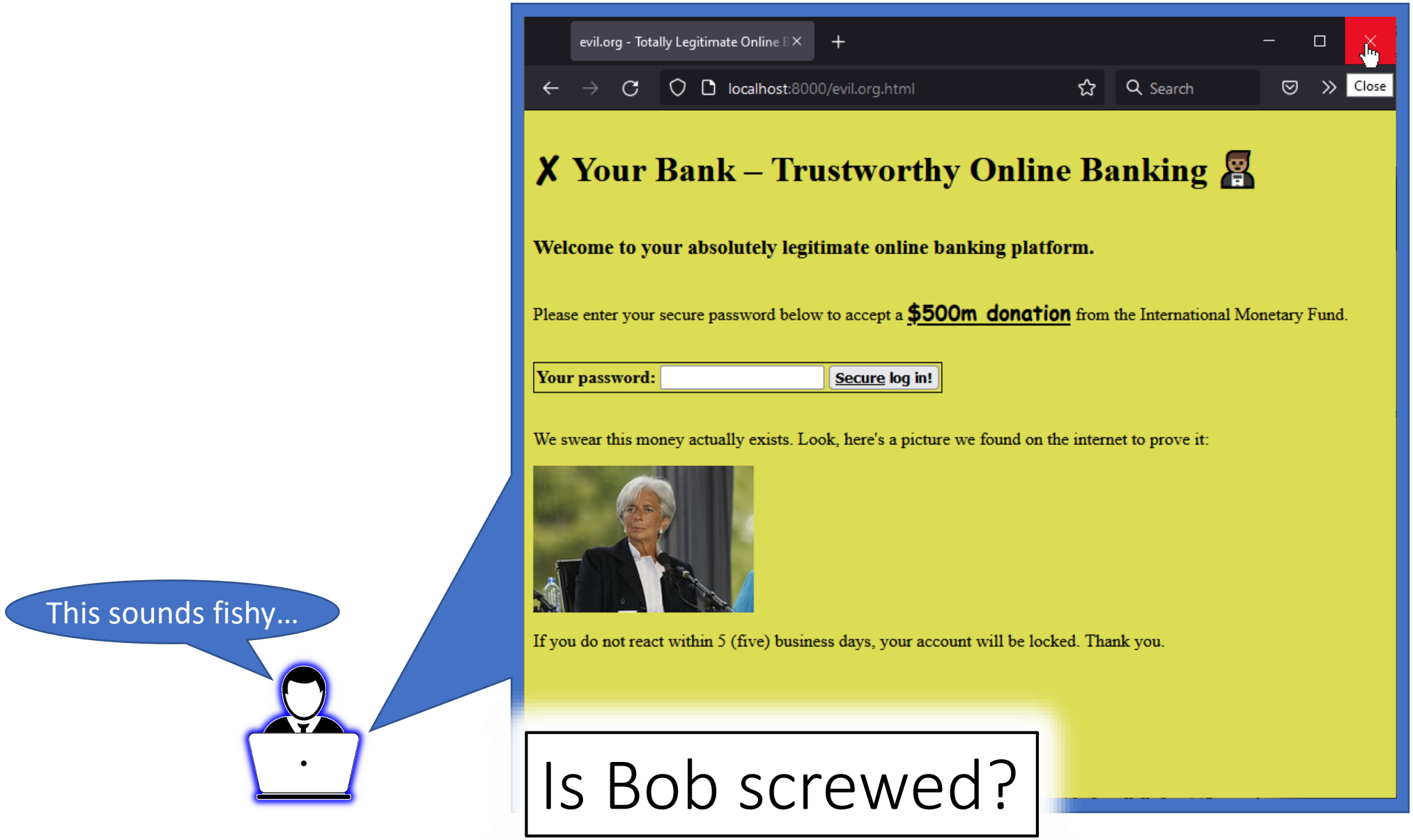
Is Bob screwed?


evil.org - Totally Legitimate Online B
localhost:8000/evil.org.html
Search
✗ Your Bank – Trustworthy Online Banking
Welcome to your absolutely legitimate online banking platform.
Please enter your secure password below to accept a $500m donation from the International Monetary Fund.
Your password:
Secure log in!
We swear this money actually exists. Look, here's a picture we found on the internet to prove it:
If you do not react within 5 (five) business days, your account will be locked. Thank you.
correct horse battery staple
Is Bob screwed?

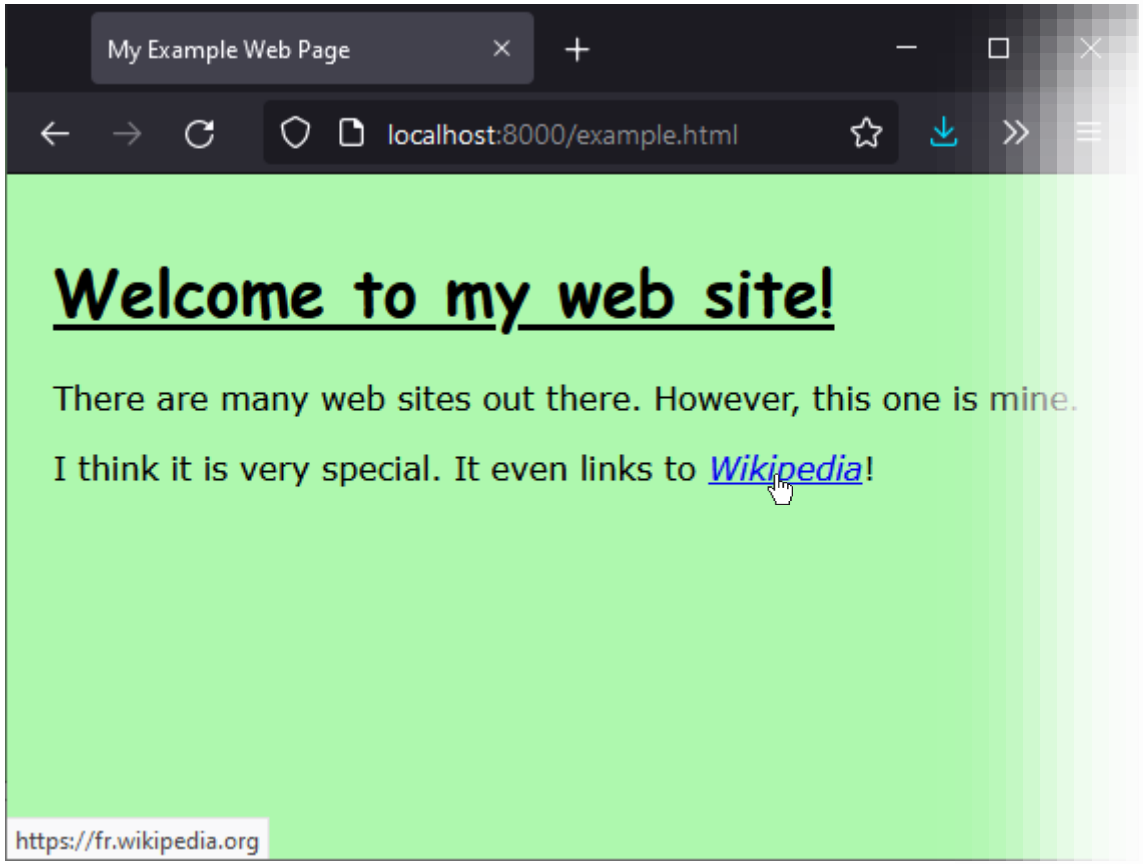


Is Bob screwed?

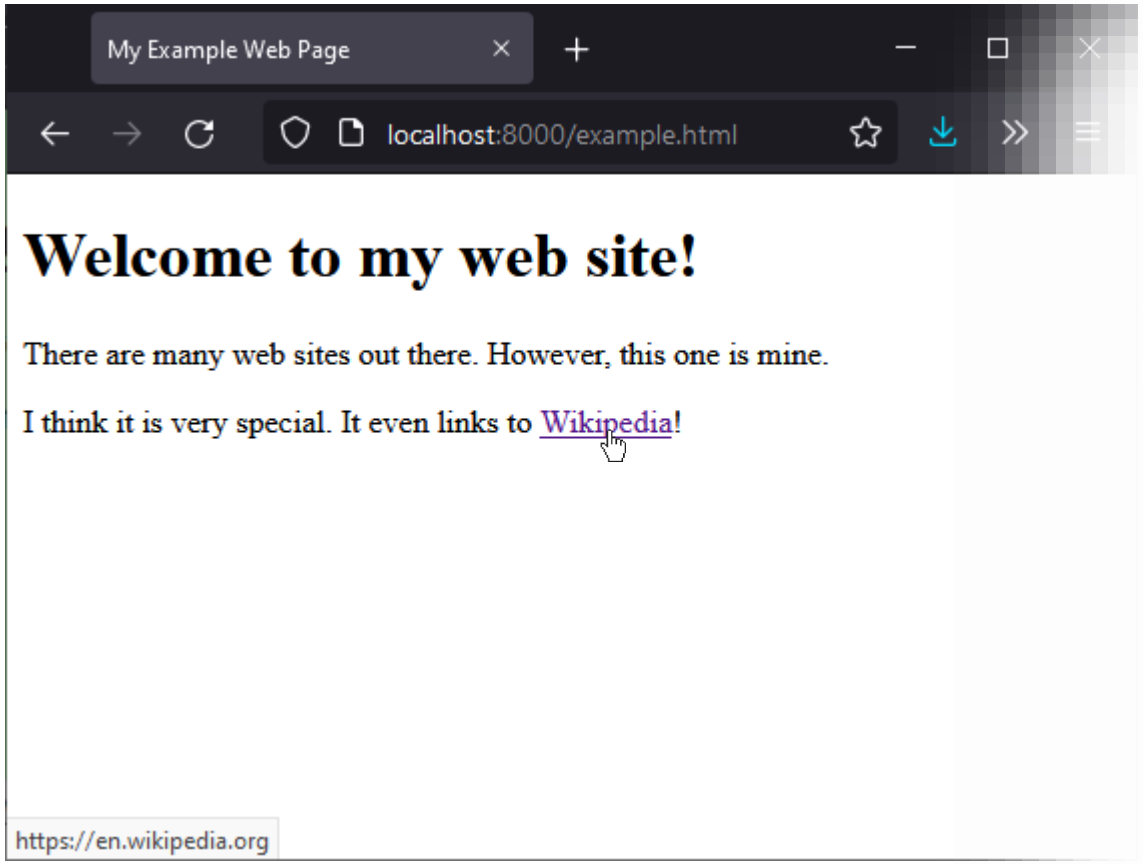
Yes.


evil.org - Totally Legitimate Online B
localhost:8000/evil.org.html
Search
Close
✗ Your Bank – Trustworthy Online Banking
Welcome to your absolutely legitimate online banking platform.
Please enter your secure password below to accept a $500m donation from the International Monetary Fund.
Your password:
Secure log in!
We swear this money actually exists. Look, here's a picture we found on the internet to prove it:
If you do not react within 5 (five) business days, your account will be locked. Thank you.
This sounds fishy…
Is Bob screwed?


evil.org - Totally Legitimate Online B
localhost:8000/evil.org.html
Search
Close
✗ Your Bank – Trustworthy Online Banking
Welcome to your absolutely legitimate online banking platform.
Please enter your secure password below to accept a $500m donation from the International Monetary Fund.
Your password:
Secure log in!
We swear this money actually exists. Look, here's a picture we found on the internet to prove it:
If you do not react within 5 (five) business days, your account will be locked. Thank you.
This sounds fishy…
Is Bob screwed?
Maybe…



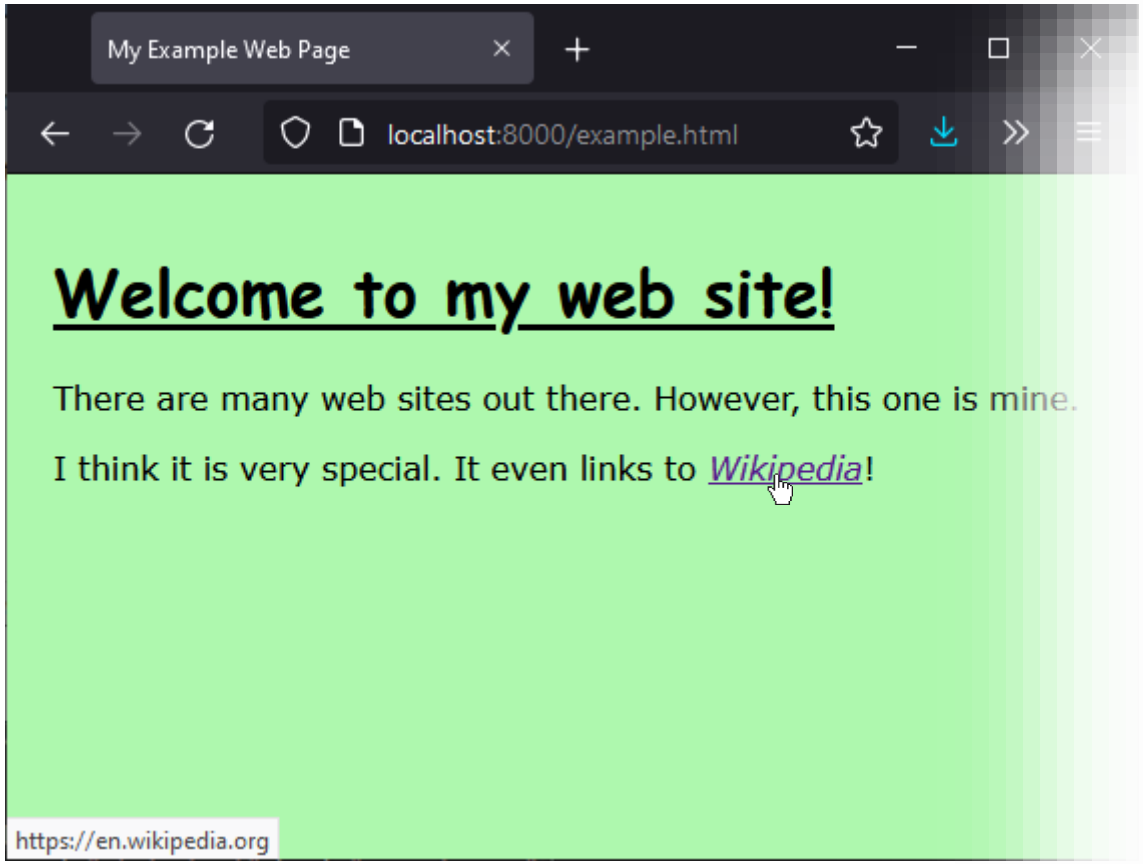
What kinds of files make up a web page?



What kinds of files make up a web page?

- **H**yper**T**ext **M**arkup **L**anguage
  - Page content and structure

```
<!doctype HTML>
<html>
    <head>
        <title>My Example Web Page</title>
    </head>
    <body>
        <h1>Welcome to my web site!</h1>
        <p>There are many web sites out there. However, this one is mine.</p>
        <p>I think it is very special. It even links to <a href="https://en.wikipedia.org/">Wikipedia</a>!</p>
    </body>
</html>
```

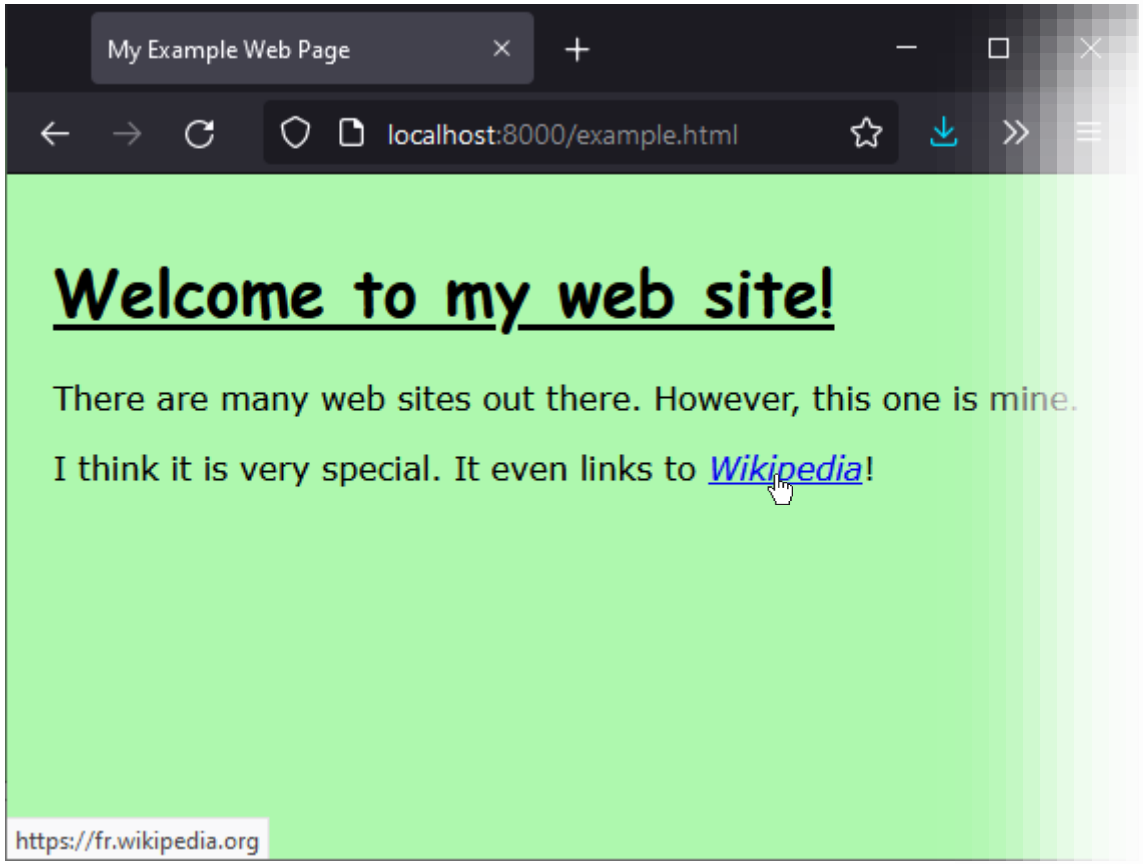


What kinds of files make up a web page?

- **H**yper**T**ext **M**arkup **L**anguage
  - Page content and structure
- **C**ascading **S**tyle **S**heets
  - Page layout and formatting

```html
<!doctype HTML>
<html>
    <head>
        <title>My Example Web Page</title>
        <link rel="stylesheet" href="example.css">
    </head>
    <body>
        <h1>Welcome to my web site!</h1>
        <p>There are many web sites out there. However, this one is mine.</p>
        <p>I think it is very special. It even links to <a href="https://en.wikipedia.org/">Wikipedia</a>!</p>
    </body>
</html>
```

```css
html
{
    background: rgb(174, 248, 174);
    font-family: Verdana;
    padding: 15px;
}

h1
{
    font-family: "Comic Sans MS";
    text-decoration: underline;
}

a
{
    font-style: italic;
}
```



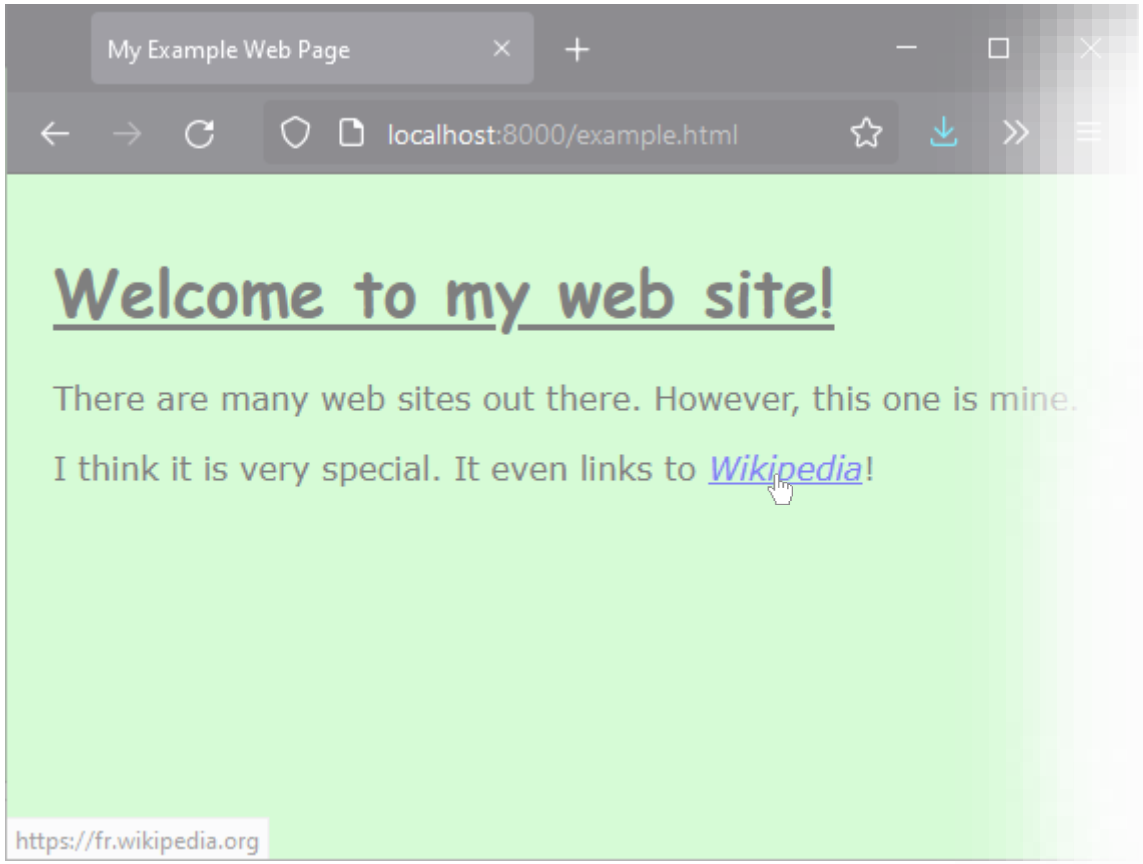
What kinds of files make up a web page?

- **H**yper**T**ext **M**arkup **L**anguage
  - Page content and structure
- **C**ascading **S**tyle **S**heets
  - Page layout and formatting
- **J**ava**S**cript
  - Dynamically modify the web page
  - Request additional data on demand
  - and many more…

```html
<!doctype HTML>
<html>
    <head>
        <title>My Example Web Page</title>
        <link rel="stylesheet" href="example.css">
        <script src="example.js" defer></script>
    </head>
    <body>
        <h1>Welcome to my web site!</h1>
        <p>There are many web sites out there. However, this one is mine.</p>
        <p>I think it is very special. It even links to <a href="https://en.wik
    </body>
</html>
```

```css
html
{
    background: rgb(174, 248, 174);
    font-family: Verdana;
    padding: 15px;
}

h1
{
    font-family: "Comic Sans MS";
    text-decoration: underline;
}
```

```javascript
document.getElementsByTagName('a')[0].addEventListener('mouseover', function()
{
    let languages = ['en','de','it','fr','es'];
    this.href = ('https://'+languages[(Math.random() * languages.length) | 0]+'.wikipedia.org/');
});
```



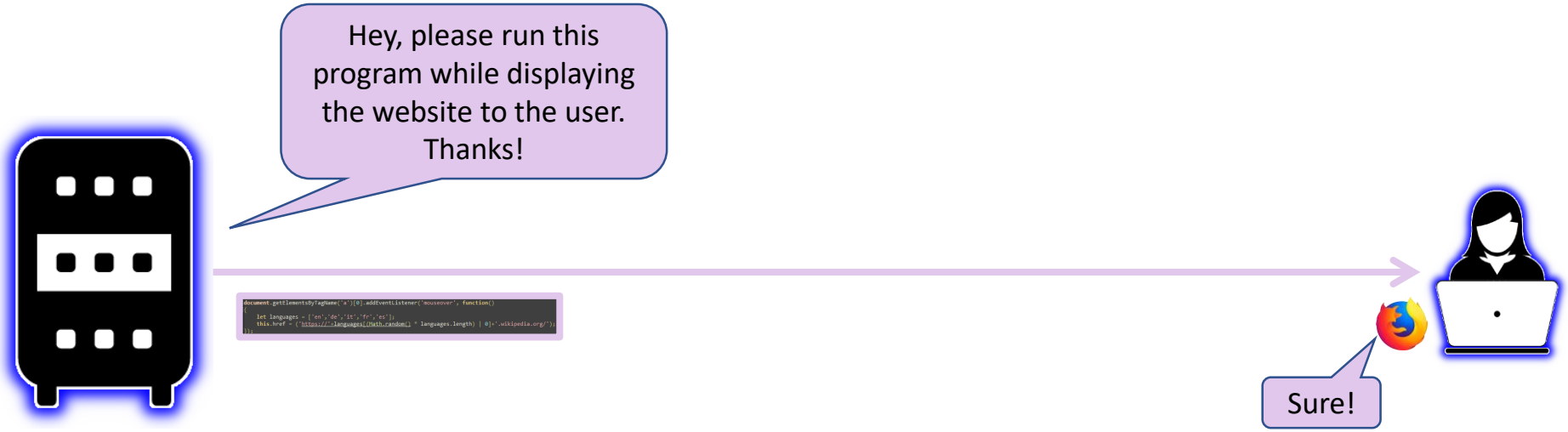
What kinds of files make up a web page?

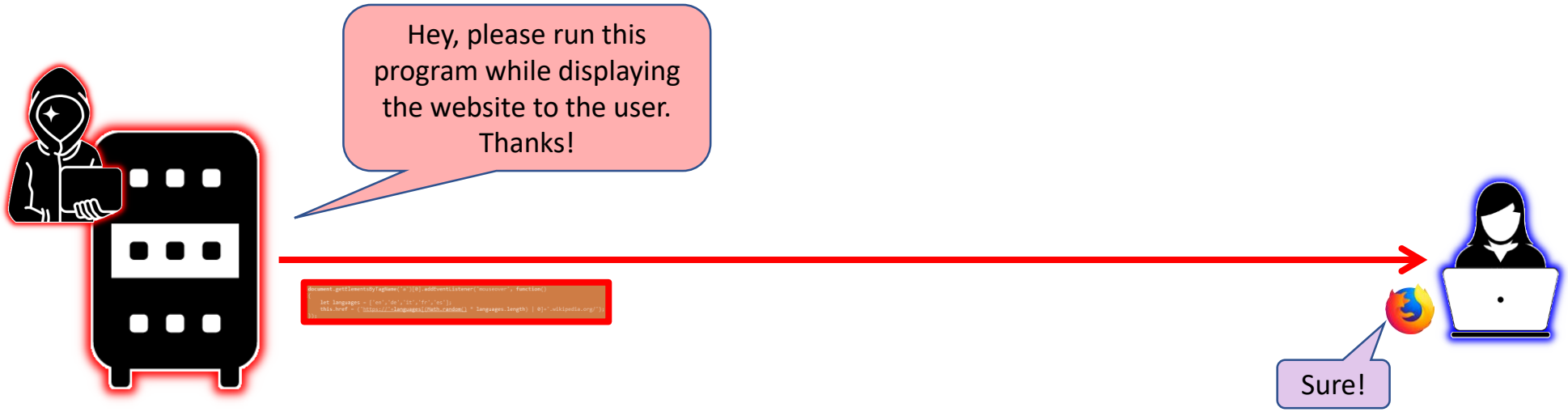
- HyperText Markup Language
  - Page content and structure
- Cascading Style Sheets
  - Page layout and formatting
- JavaScript
  - Dynamically modify the web page
  - Request additional data on demand
  - and many more…

```
<!doctype HTML>
<html>
    <head>
        <title>My Example Web Page</title>
        <link rel="stylesheet" href="example.css">
        <script src="example.js" defer></script>
    </head>
    <body>
        <h1>Welcome to my web site!</h1>
        <p>There are many web sites out there. However, this one is mine.</p>
        <p>I think it is very special. It even links to <a href="https://en.wik
    </body>
</html>
```

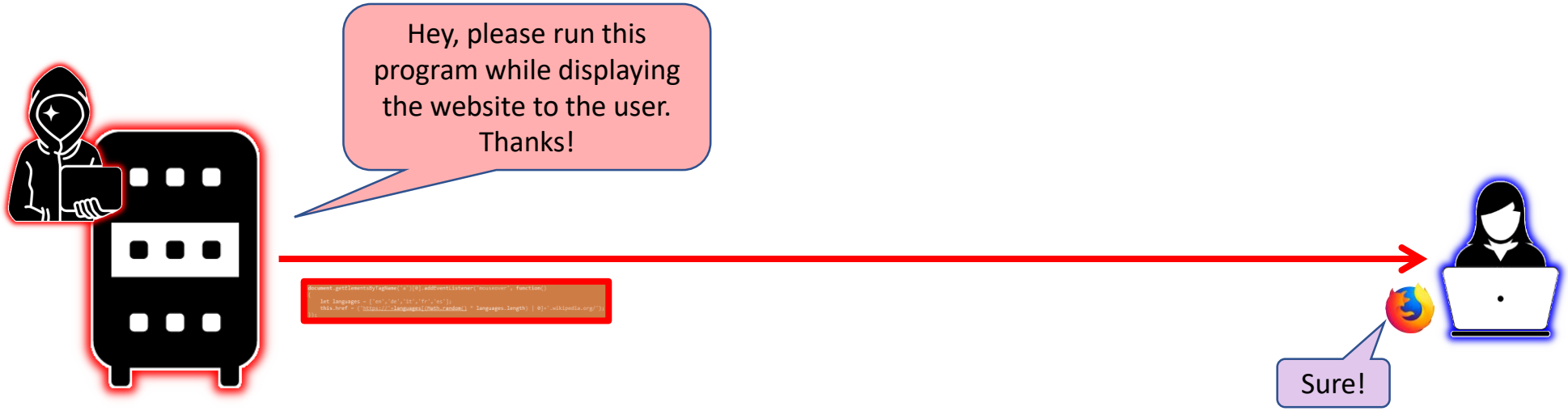
```
html
{
    background: rgb(174, 248, 174);
    font-family: Verdana;
    padding: 15px;
}

h1
{
    font-family: "Comic Sans MS";
    text-decoration: underline;
```

```
document.getElementsByTagName('a')[0].addEventListener('mouseover', function()
{
    let languages = ['en','de','it','fr','es'];
    this.href = ('https://'+languages[(Math.random() * languages.length) | 0]+'.wikipedia.org/');
});
```



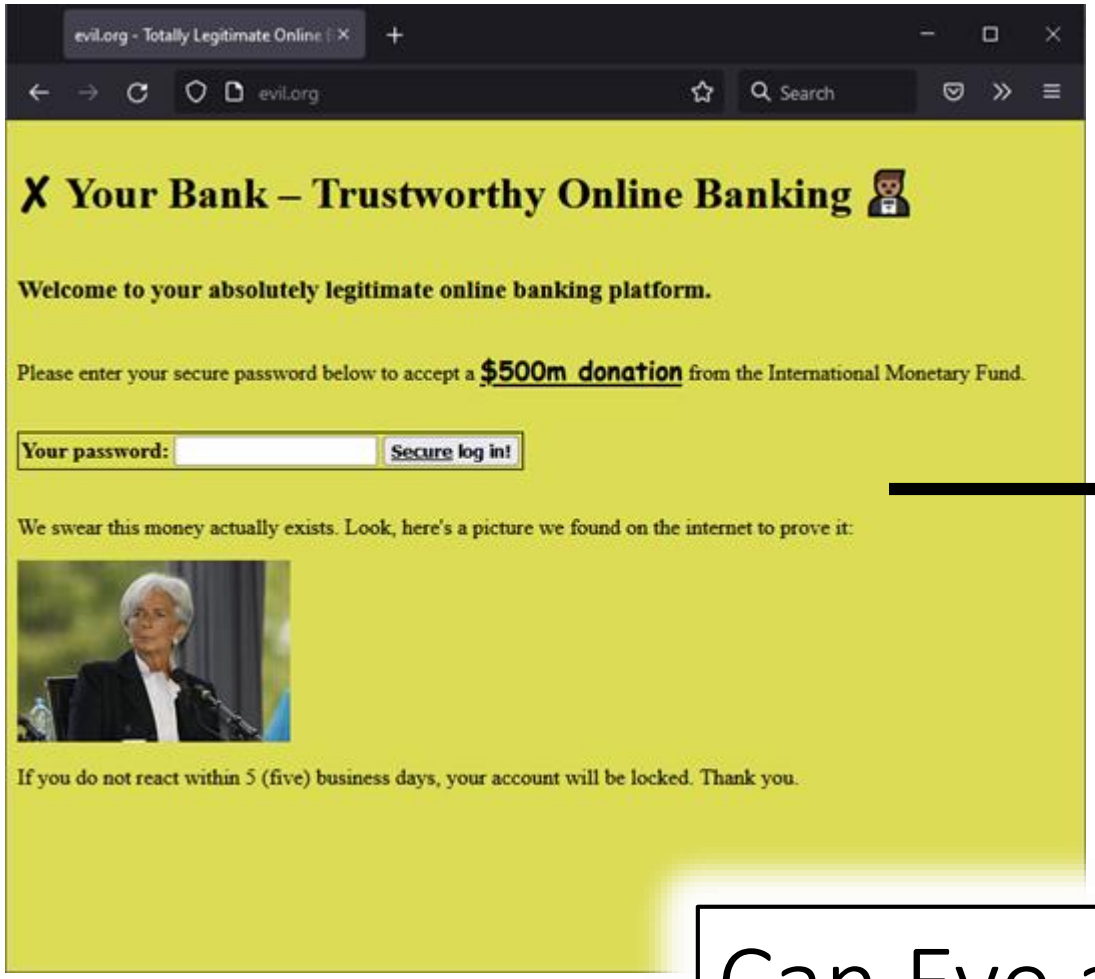


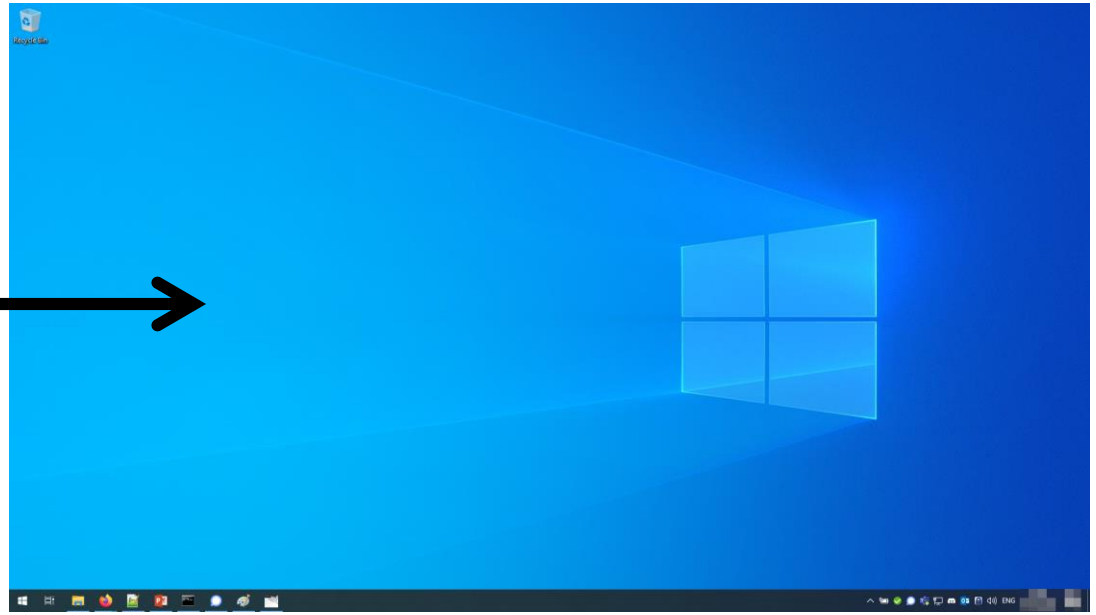
What could possibly go wrong?



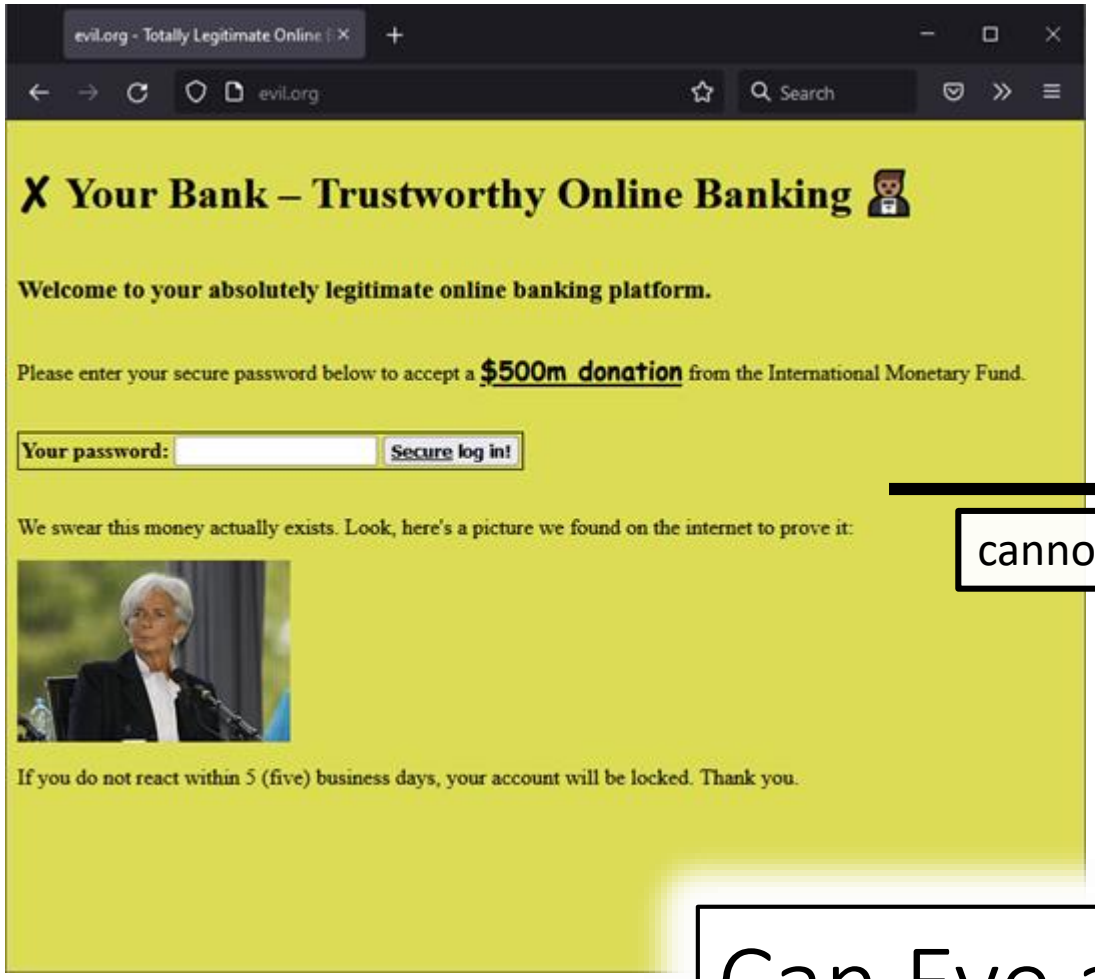
What could possibly go wrong?

Yeah, no, seriously. What could?

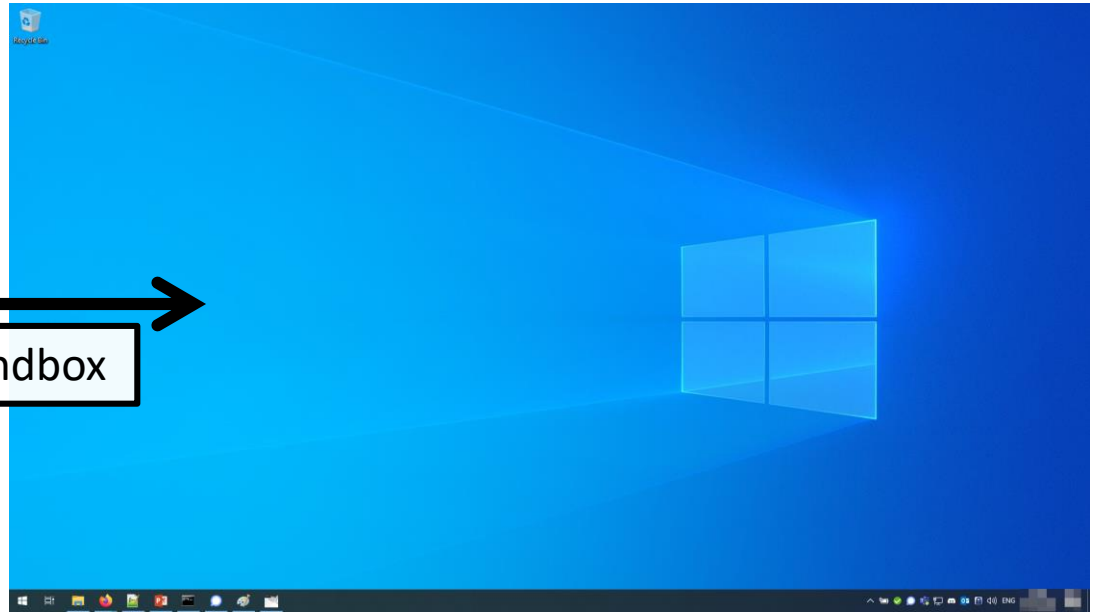




Can Eve attack the OS?



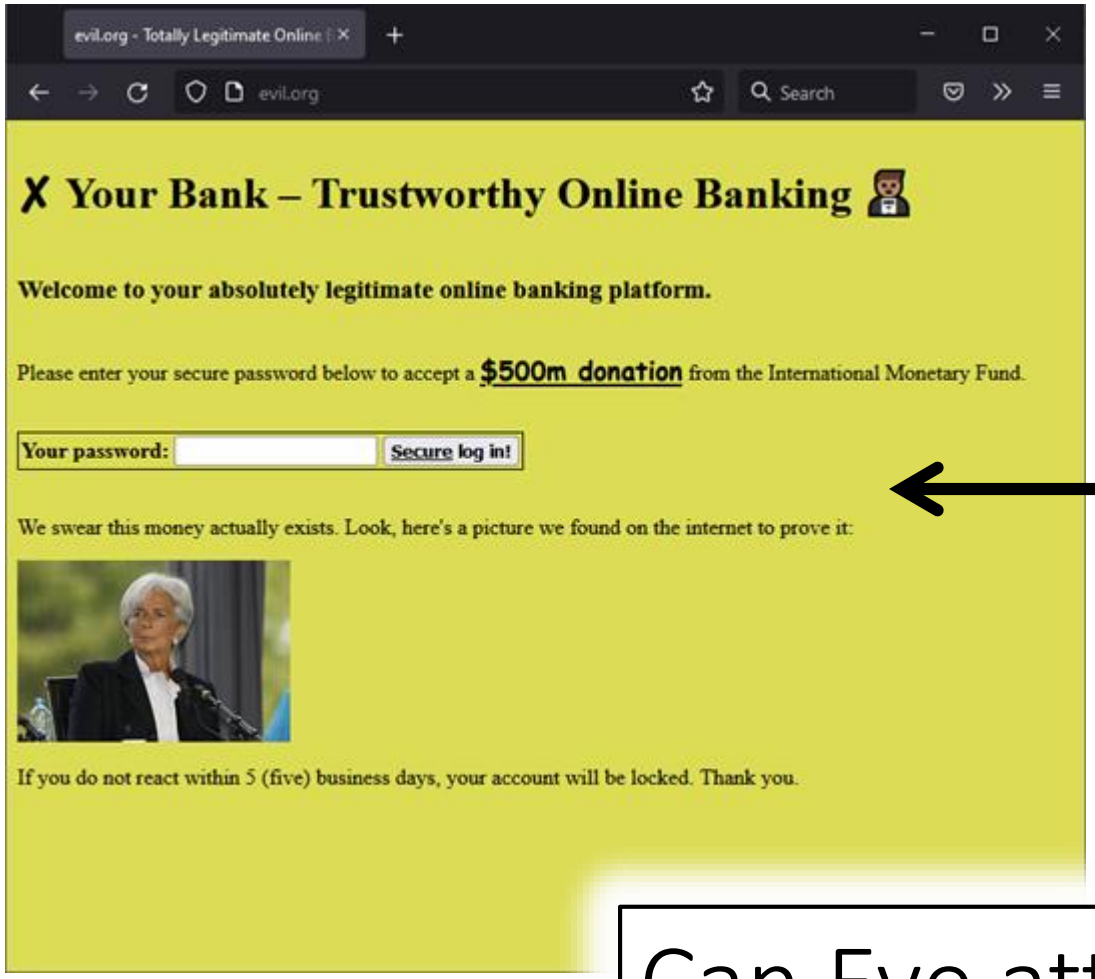
cannot escape sandbox

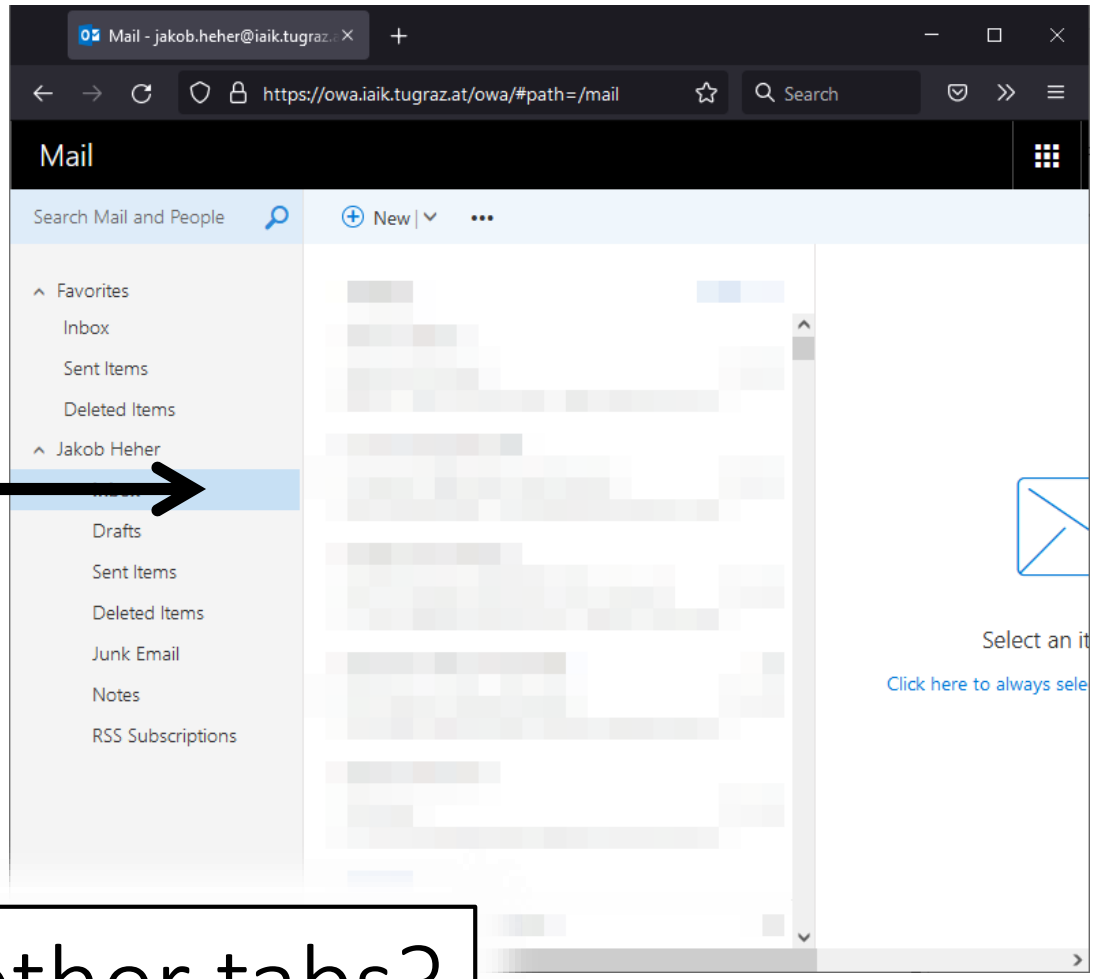


Can Eve attack the OS?

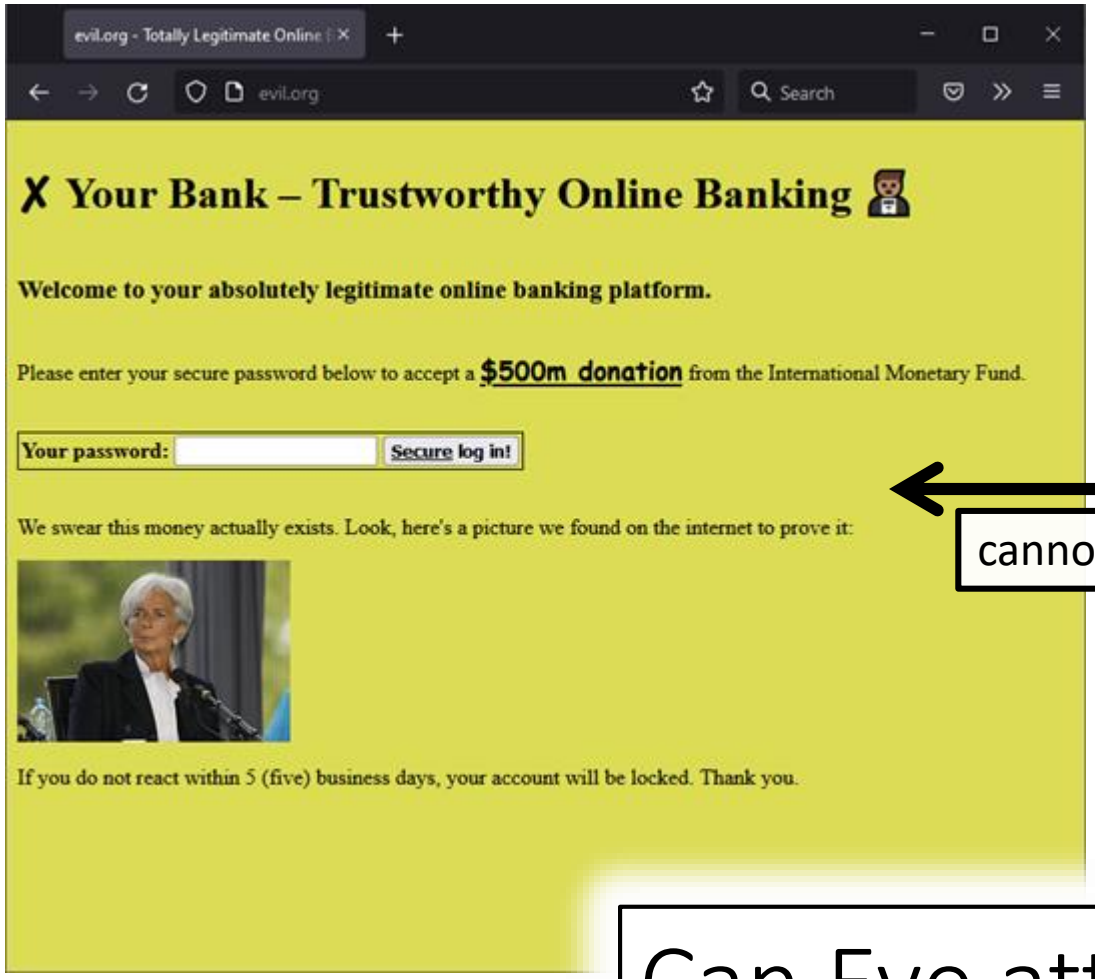
No.

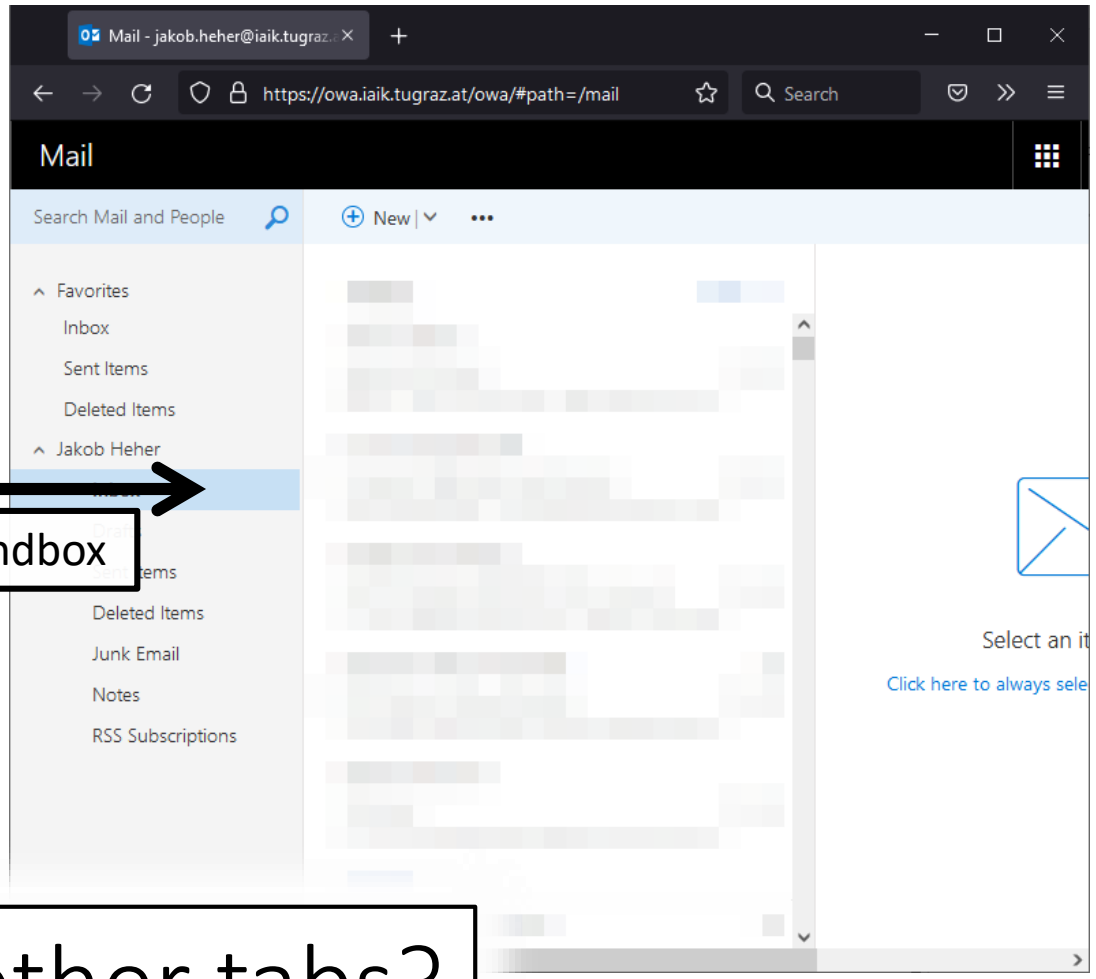
(unless something has gone horribly wrong)





Can Eve attack other tabs?



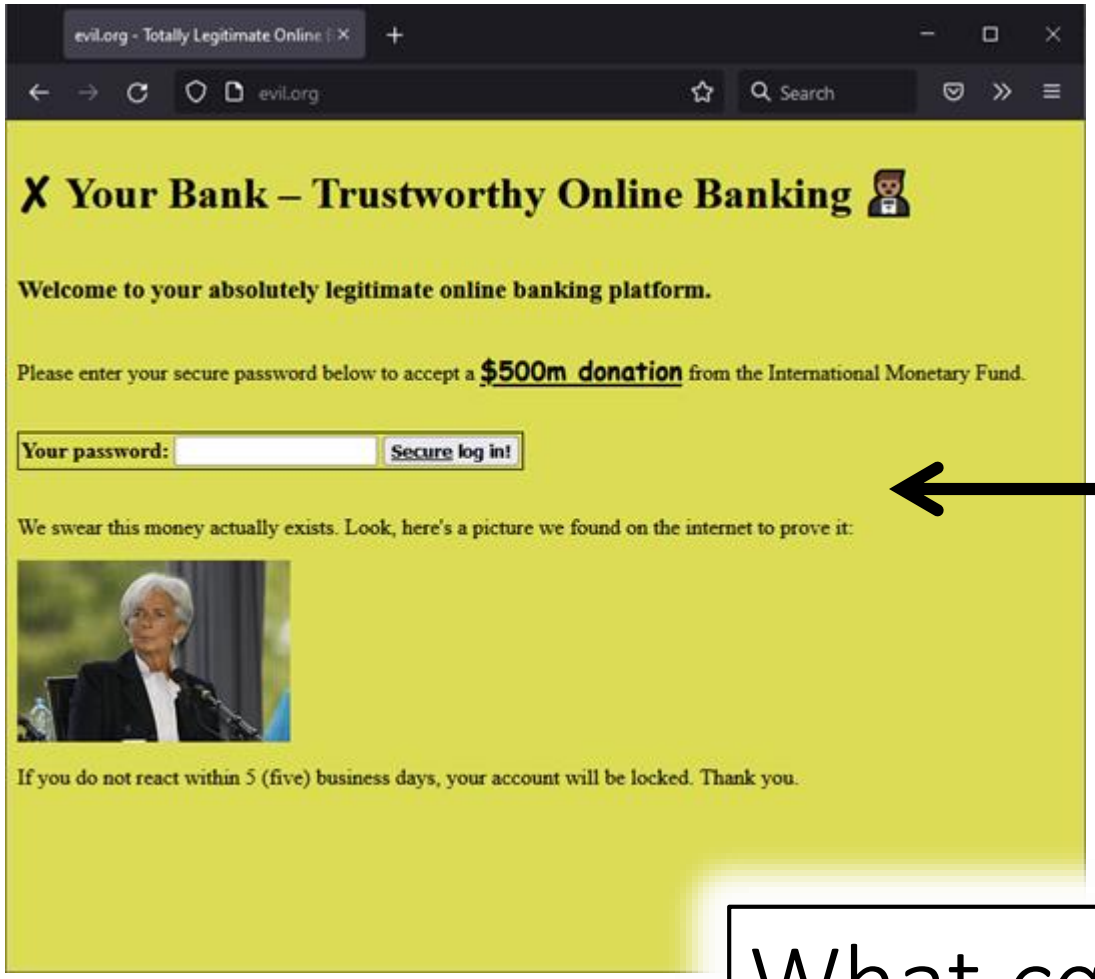


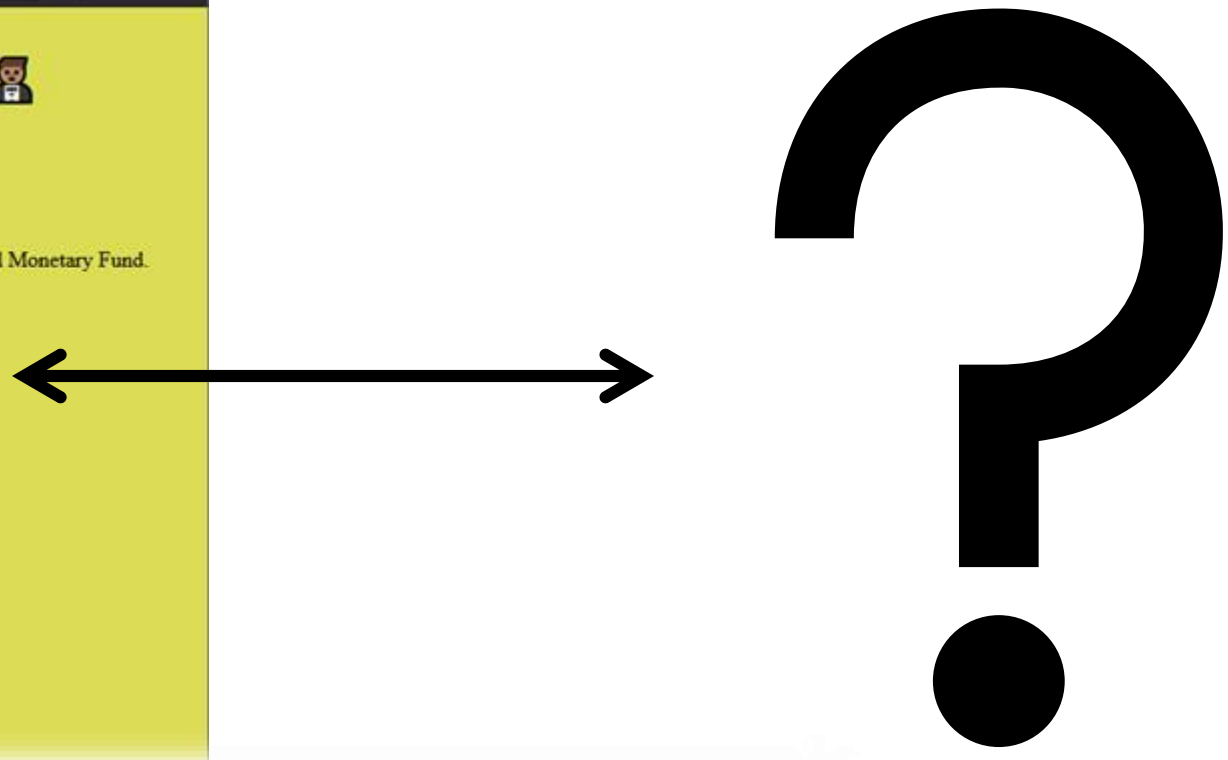
cannot escape sandbox

Can Eve attack other tabs?

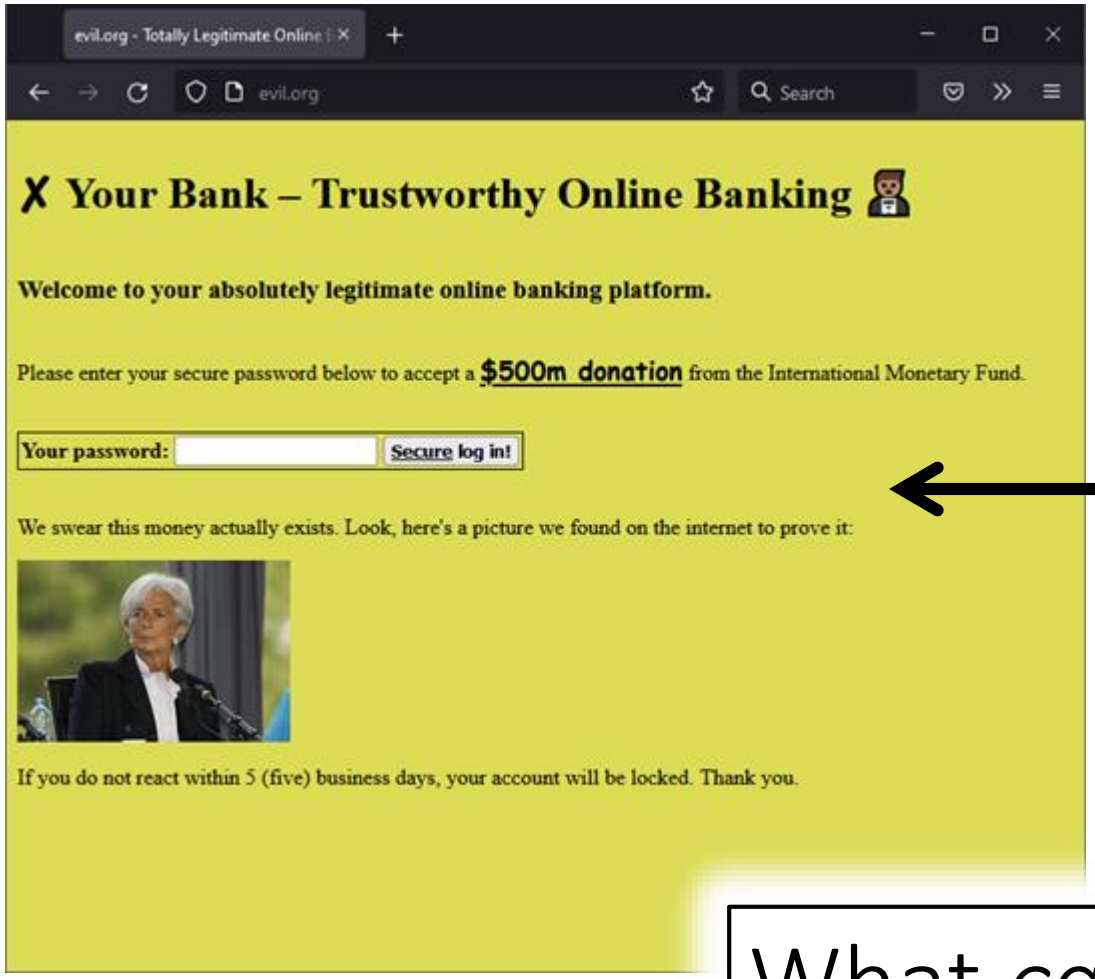
No.

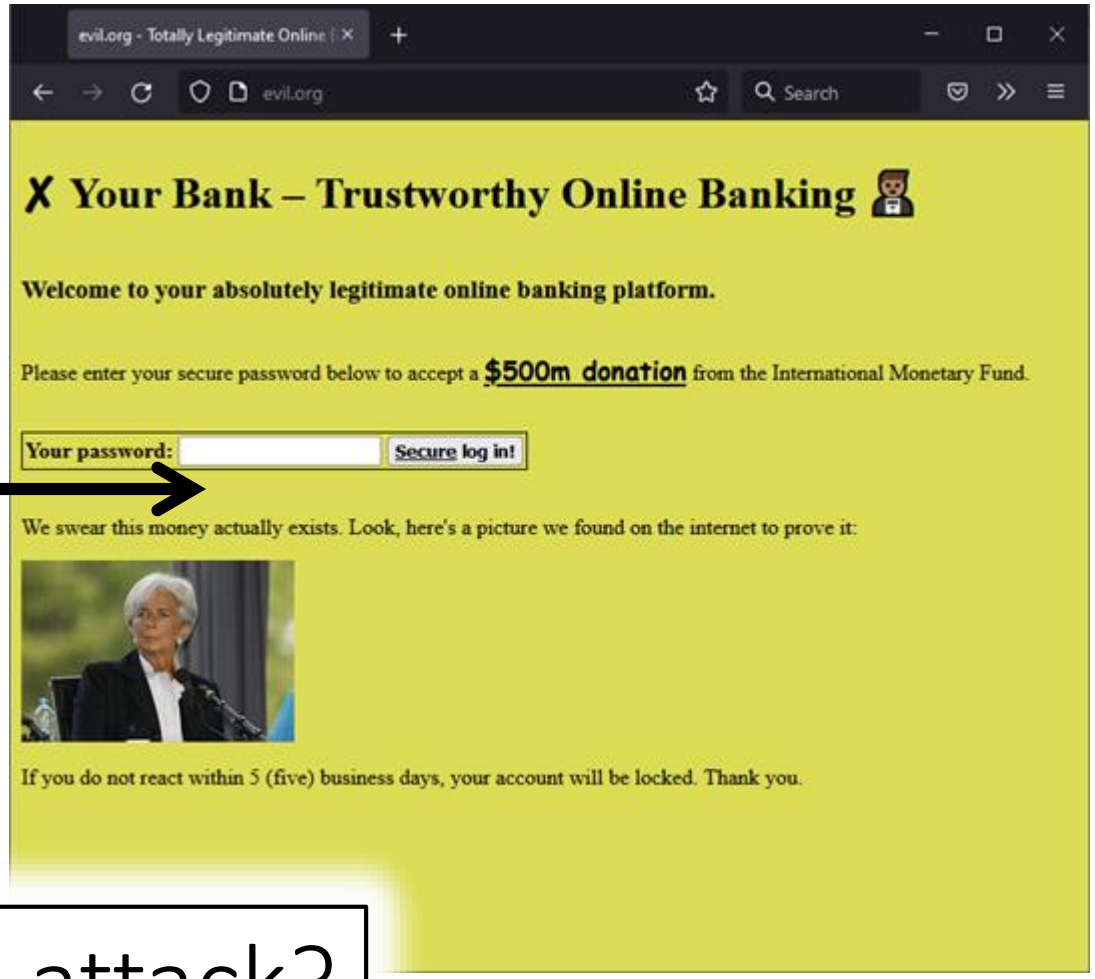
(unless something has gone horribly wrong)





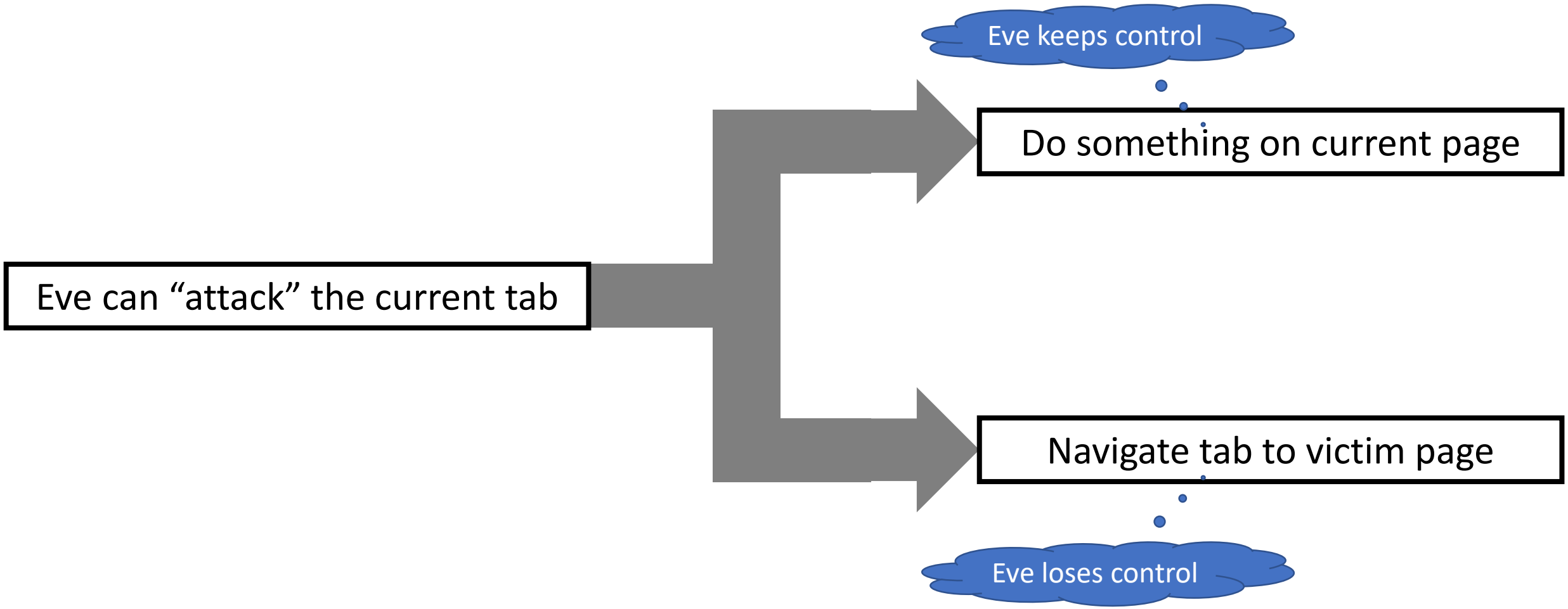
What *can* Eve attack?





What *can* Eve attack?

The *current* tab.


Eve keeps control
Do something on current page
Eve can “attack” the current tab
Navigate tab to victim page
Eve loses control

evil.org - Totally Legitimate Online

evil.org

Search

✗ Your Bank – Trustworthy Online Banking

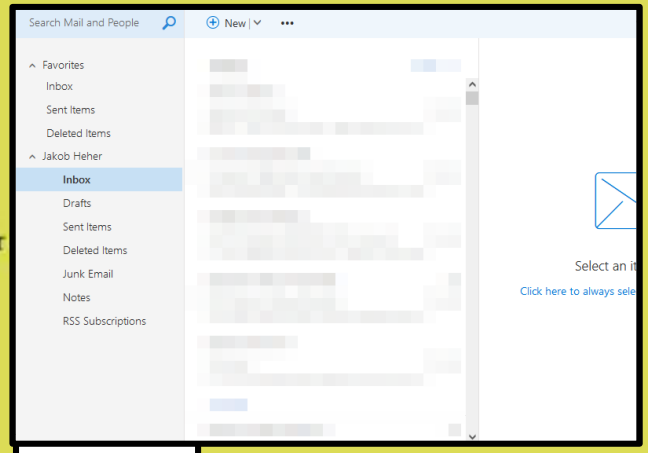
Welcome to your absolutely legitimate online banking platform.

Please enter your secure password below to accept a **$500m donation** from the International Monetary Fund.

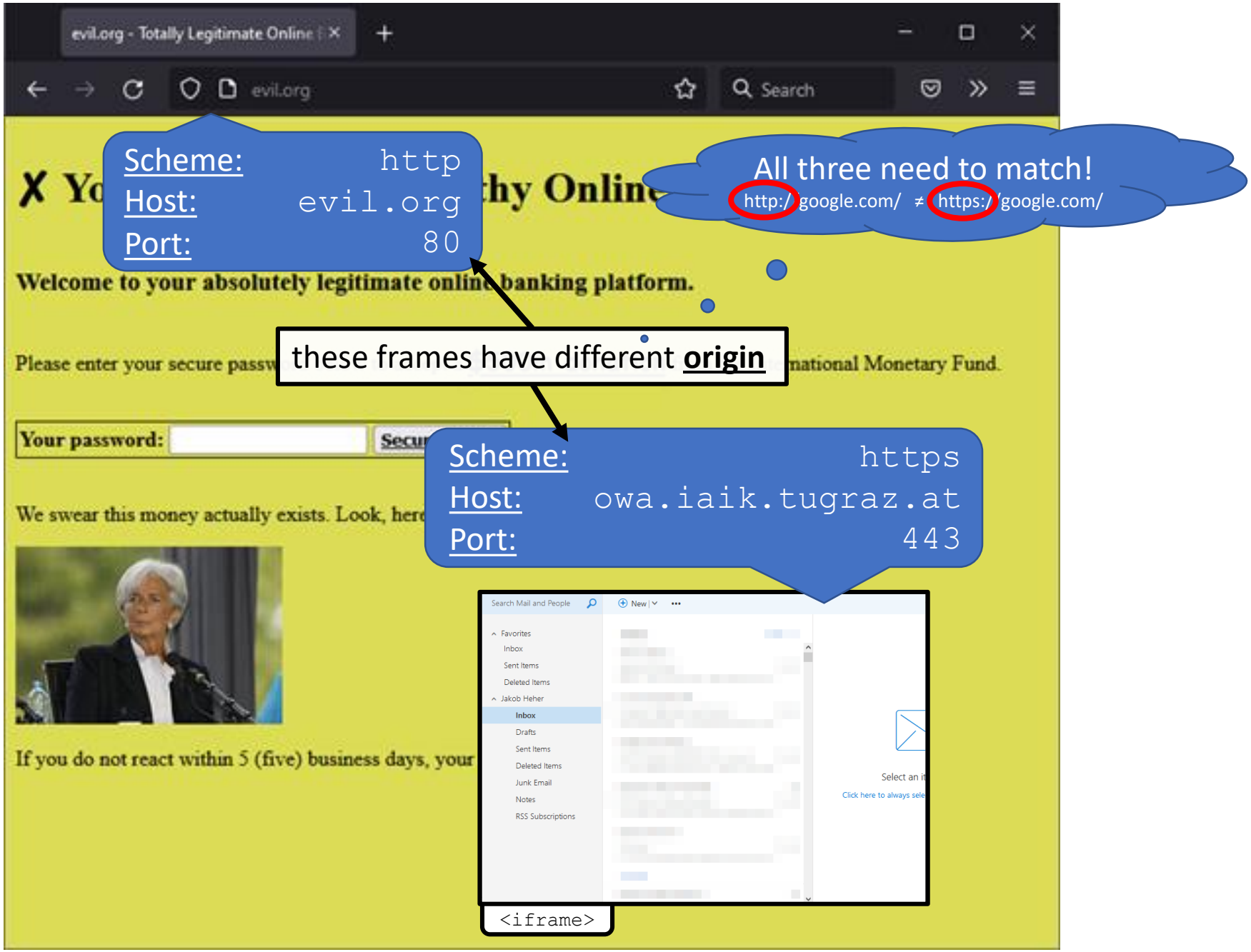
Your password: **Secure log in!**

We swear this money actually exists. Look, here's a picture we found on the internet to prove it:

If you do not react within 5 (five) business days, your



<iframe>


evil.org - Totally Legitimate Online
evil.org
Search
Scheme: http
Host: evil.org
Port: 80
All three need to match!
http://google.com/ ≠ https://google.com/
these frames have different origin
Scheme: https
Host: owa.iaik.tugraz.at
Port: 443
Welcome to your absolutely legitimate online banking platform.
Please enter your secure passw
national Monetary Fund.
Your password:
We swear this money actually exists. Look, here
If you do not react within 5 (five) business days, your
Search Mail and People
New
Favorites
Inbox
Sent Items
Deleted Items
Jakob Heher
Inbox
Drafts
Sent Items
Deleted Items
Junk Email
Notes
RSS Subscriptions
<iframe>

evil.org - Totally Legitimate Online

evil.org

Search

✗ Your Bank – Trustworthy Online Banking

Welcome to your absolutely legitimate online banking platform.

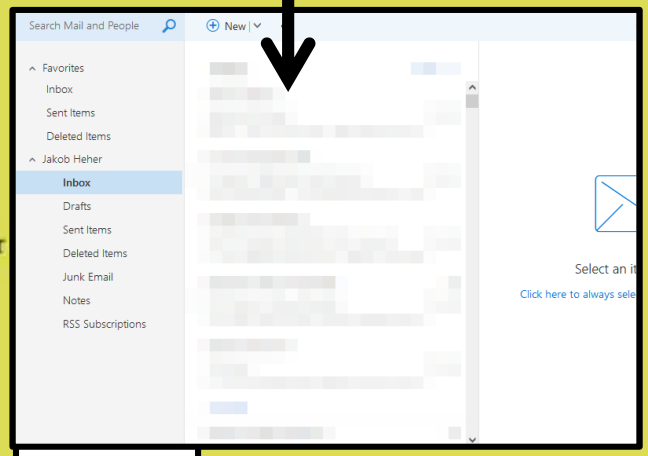
Please enter your secure password below to accept a **$500m donation** from the International Monetary Fund.

Your password: **Secure log in!**

We swear this money actually exists. Look,

If you do not react within 5 (five) business days, your

no interaction with cross-origin iframe



<iframe>

evil.org - Totally Legitimate Online

evil.org

Search

✗ Your Bank – Trustworthy Online Banking

Welcome to your absolutely legitimate online banking platform.

Please enter your secure password below to accept a **$500m donation** from the International Monetary Fund.

Your password: Secure log in!

We swear this money actually exists. Look, here's a picture we found on the internet to prove it:

If you do not react within 5 (five) business days, your account will be locked. Thank you.

```
await fetch('https://secure.lawful.org/all_the_dataz.json');
```

<script>

evil.org - Totally Legitimate Online

evil.org

Search

✗ Your Bank – Trustworthy Online Banking

Welcome to your absolutely legitimate online banking platform.

Please enter your secure password below to accept a **$500m donation** from the International Monetary Fund.

Your password: **Secure log in!**

found on the internet to prove it:

If you do not react within 5 (five) business days, your account will be locked. Thank you.

- Can only send *safe* requests
- Cannot access response data

limited interaction with cross-origin URL

```
await fetch('https://secure.lawful.org/all_the_dataz.json');
```

<script>

Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at *https://secure.lawful.org/all_the_dataz.json*. (Reason: CORS header 'Access-Control-Allow-Origin' missing). [Learn More]

What *can* Eve do?

- Send arbitrary GET requests
  - But: response data is inaccessible!

```
await fetch('https://secure.lawful.org/all_the_dataz.json');
```

- Send arbitrary POST requests

```
<form action="https://secure.lawful.org/add_admin_account" method="POST">
```

- Embed arbitrary JavaScript

```
<script src="https://secure.lawful.org/userdata.js?callback=insecure"></script>
```

- Embed arbitrary web pages as iframes

```
<iframe src="http://twitter.com/home?status=Don't%20Click:%20http://tinyurl.com/amgzs6" scrolling="no"></iframe>
```

What *can* Eve do?

- OK, but why even involve Bob?
  - Eve can make these requests already…

```
await fetch('https://secure.lawful.org/all_the_dataz.json');
```

```
<form action="https://secure.lawful.org/add_admin_account" method="POST">
```

```
<script src="https://secure.lawful.org/userdata.js?callback=insecure"></script>
```

```
<iframe src="http://twitter.com/home?status=Don't%20Click:%20http://tinyurl.com/amgzs6" scrolling="no"></iframe>
```

What *can* Eve do?

- OK, but why even involve Bob?
  - Eve can make these requests already…

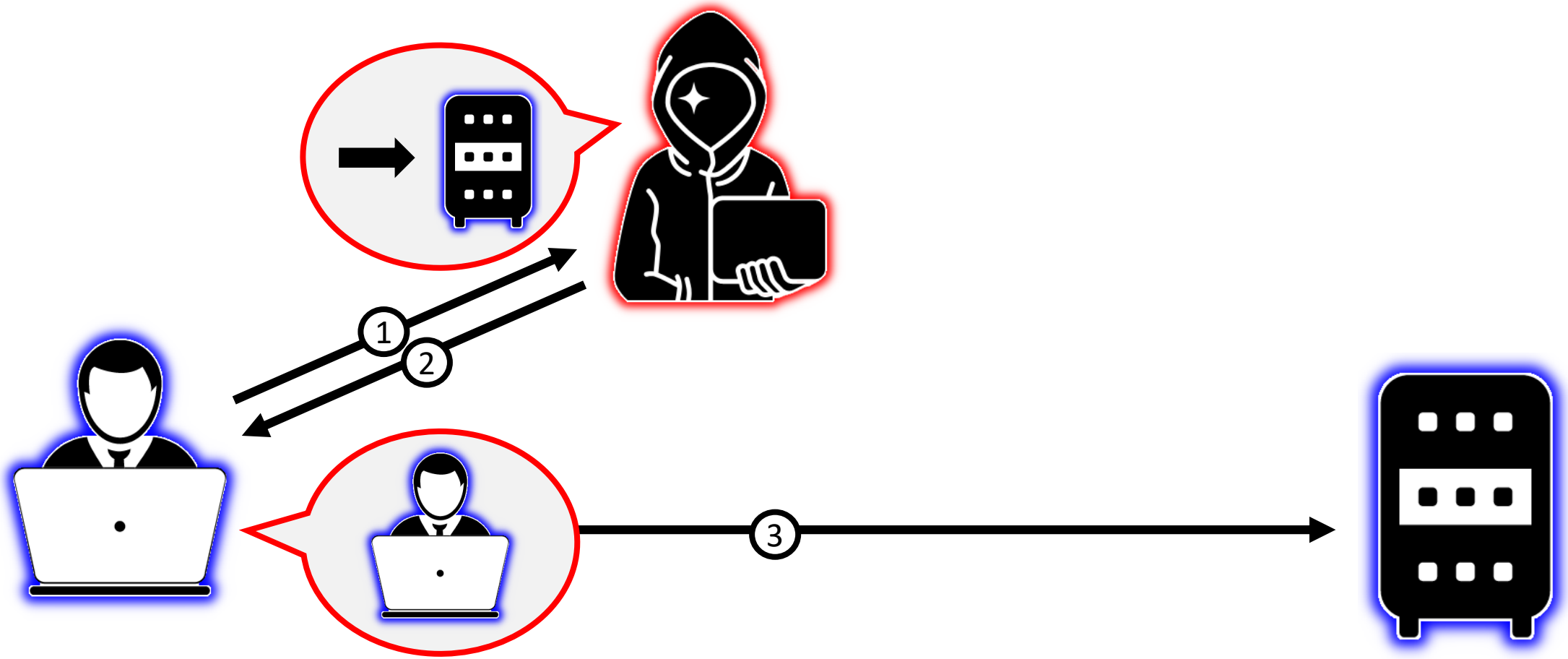
```
await fetch('https://secure.lawful.org/all_the_dataz.json');
```

```
<form action="https://secure.lawful.org/add_admin_account" method="POST">
```

```
<script src="https://secure.lawful.org/userdata.js?callback=insecure"></script>
```

```
<iframe src="http://twitter.com/home?status=Don't%20Click:%20http://tinyurl.com/amgzs6" scrolling="no"></iframe>
```

- Eve cares that it's *Bob* making these requests!
  - Eve cares about *permissions* that Bob has
  - Eve cares about the action being *attributed* to Bob
- Bob's browser is *authenticated* as Bob



Web Authentication Techniques

How does a web server know it’s you?

IP Address

✗

- Is the request is coming from some “secure” address range?
  - Only allow the request if this is true

IP Address

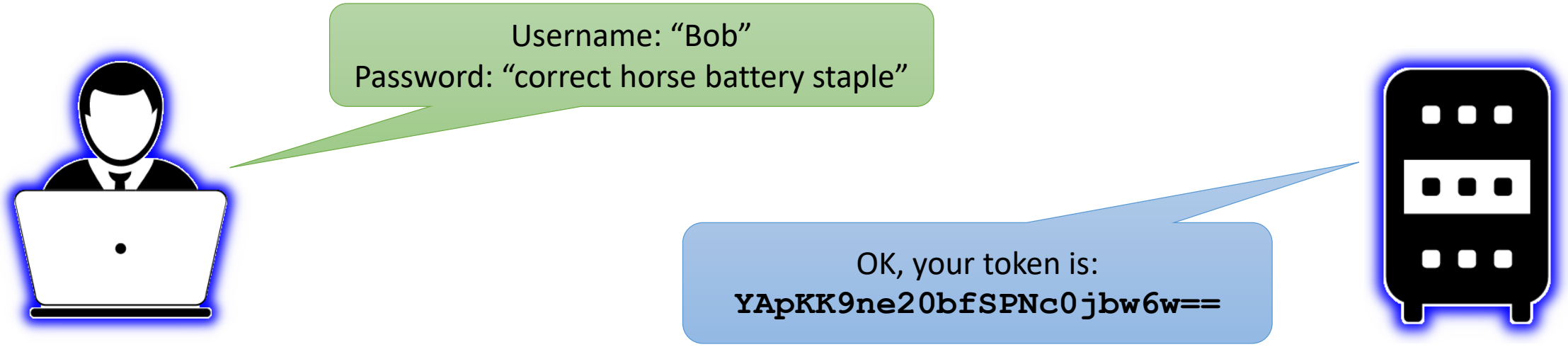
✗

- Is the request is coming from some “secure” address range?
  - Only allow the request if this is true

- Eve can make arbitrary requests in our attack scenario
  - They will all be coming from Bob’s computer!
- But: Eve can’t access the response data
  - Unless we make further mistakes…

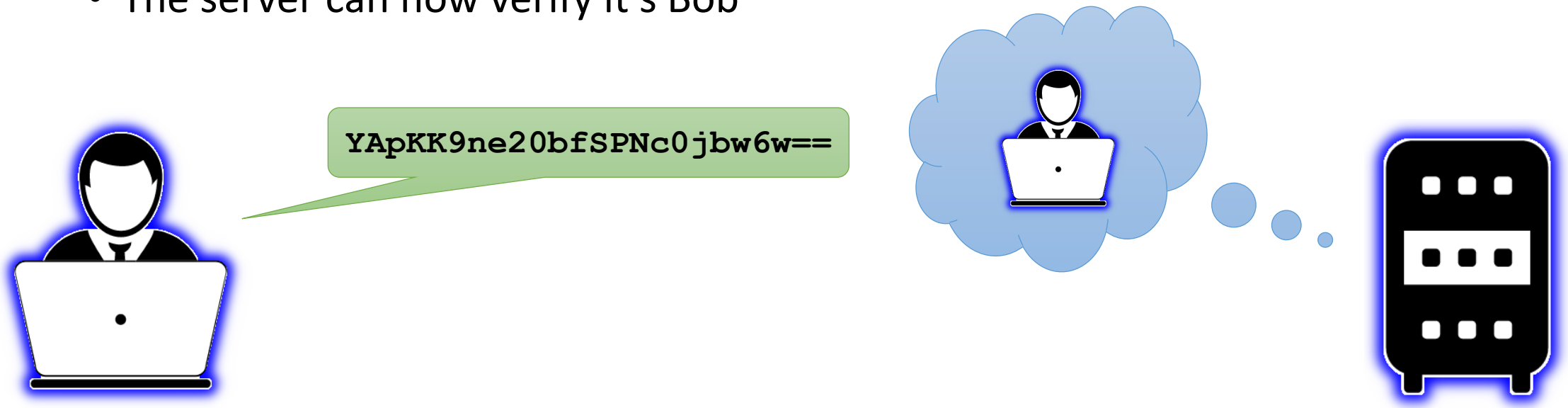
Token-based authentication

- When Bob logs in, Bob gets a *token*
  - Some kind of “special” string



Token-based authentication

- When Bob logs in, Bob gets a *token*
  - Some kind of “special” string
- To prove it’s Bob, the browser sends the token back
  - The server can now verify it’s Bob



Token-based authentication

Token generation

"What's in the token?"

Token storage

"Where does Bob keep the token?"

In the browser, somehow…

- If Eve compromises the token, Eve wins by default
- There are different ways to ask Bob's browser to store information…

Token storage:

URL rewriting

✗

- Store the token as a parameter in the URL
  - e.g. **`http://genuine.org/view.php?S=YApKK9ne20bfSPNc0jbw`**
- Dynamically adjust all links on the site to include the token

Token storage:

URL rewriting

✗

- Store the token as a parameter in the URL
  - e.g. **`http://genuine.org/view.php?S=YApKK9ne20bfSPNc0jbw`**
- Dynamically adjust all links on the site to include the token

- Users commonly copy links to your website
  - If Bob copies this URL and sends it to Alice, Alice will be logged in as Bob!

- Thankfully, URL rewriting has (mostly) died out…

Token storage:

HTTP Cookies

- Server stores a string in the Bob's browser
- Bob's browser sends this string back with any*) request
- Server can verify that it's Bob from this *cookie*

Token storage:

HTTP Cookies

- Server stores a string in the Bob's browser
- Bob's browser sends this string back with any*) request
- Server can verify that it's Bob from this *cookie*

- But won't Bob's browser just send the cookie when Eve asks it to?
  - **Since 2021: No!**

Token storage:

HTTP Cookies

- HTTP cookies come with a variety of *attributes*:
  - **`SameSite`**: do not send this for requests started by a different origin
  - **`Secure`**: only send this over HTTPS
  - **`HttpOnly`**: not accessible from JavaScript
  - *and others…*
- This attribute lets us protect against Eve's shenanigans!

Token storage:

HTTP Cookies

- HTTP cookies come with a variety of *attributes*:
  - **`SameSite`**: do not send this for requests started by a different origin
    - ✔ **`SameSite=Strict`**: Never send this with a cross-origin request
    - ✔ **`SameSite=Lax`**: Don’t send this with cross-origin requests, except top-level navigation
    - ✗ **`SameSite=None`**: Send this with any request, even cross-origin

Token storage:

HTTP Cookies

- HTTP cookies come with a variety of *attributes*:
  - **`SameSite`**: do not send this for requests started by a different origin
    - ✔ **`SameSite=Strict`**: Never send this with a cross-origin request
    - ✔ **`SameSite=Lax`**: Don't send this with cross-origin requests, except top-level navigation
    - ✗ **`SameSite=None`**: Send this with any request, even cross-origin

- What's *top-level navigation*?
  - Navigation that changes the URL bar
  - Important: this can only ever be a HTTP **`GET`** request
  - Bob is also leaving the site, so Eve can no longer interact with him

Token storage:

HTTP Cookies

- HTTP cookies come with a variety of *attributes*:
  - **`SameSite`**: do not send this for requests started by a different origin
    - ✔ **`SameSite=Strict`**: Never send this with a cross-origin request
    - ✔ **`SameSite=Lax`**: Don't send this with cross-origin requests, except top-level navigation
    - ✗ **`SameSite=None`**: Send this with any request, even cross-origin

Default before 2021

- What's *top-level navigation*?
  - Navigation that changes the URL bar
  - Important: this can only ever be a HTTP **`GET`** request
  - Bob is also leaving the site, so Eve can no longer interact with him

Token storage:

HTTP Cookies

- HTTP cookies come with a variety of *attributes*:
  - **`SameSite`**: do not send this for requests started by a different origin
    - ✔ **`SameSite=Strict`**: Never send this with a cross-origin request
    - ✔ **`SameSite=Lax`**: Don't send this with cross-origin requests, except top-level navigation
    - ✗ **`SameSite=None`**: Send this with any request, even cross-origin

Default since 2021!

- What's *top-level navigation*?
  - Navigation that changes the URL bar
  - Important: this can only ever be a HTTP **`GET`** request
  - Bob is also leaving the site, so Eve can no longer interact with him

What *can* Eve do?

- Send arbitrary GET requests
  - But: response data is inaccessible!

```
await fetch('https://secure.lawful.org/all_the_dataz.json');
```

- Send arbitrary POST requests

```
<form action="https://secure.lawful.org/add_admin_account" method="POST">
```

- Embed arbitrary JavaScript

```
<script src="https://secure.lawful.org/userdata.js?callback=insecure"></script>
```

- Embed arbitrary web pages as iframes

```
<iframe src="http://twitter.com/home?status=Don't%20Click:%20http://tinyurl.com/amgzs6" scrolling="no"></iframe>
```

What *can* Eve do?

- Navigate Bob to arbitrary URLs

```
window.location = 'https://secure.lawful.org/create_admin_account.php?user=eve&password=evulz';
```

CON recap

- A HTTP **`GET`** request retrieves a resource
- HTTP **`GET`** requests should not modify resources
  - Making the same **`GET`** request multiple times should be safe
- Method functionality is by convention
  - Nothing is stopping you from deleting a file when a **`GET`** request is made…

What *can* Eve do?

- Navigate Bob to arbitrary URLs

```
window.location = 'https://secure.lawful.org/create_admin_account.php?user=eve&password=evulz';
```

- **C**ross-**S**ite **R**equest **F**orgery
  - Significantly harder with the **`SameSite=Lax`** default
  - With the **`None`** default, forging POST forms was possible

- Badly-designed websites might still be vulnerable
  - Never let **`GET`** have side effects!
  - Never trust URL parameters, even from trusted users!

Token storage:

JavaScript `localStorage` ?

- Persistent key/value store in the browser
- Each origin has its own **`localStorage`**
- Not sent anywhere by default
  - JavaScript explicitly reads the token and sends it when necessary
  - Eve's site can't do this, because it has its own **`localStorage`**

- Eve can embed a genuine page in an **`<iframe>`**!
  - The genuine JavaScript runs in the **`<iframe>`**
  - It has the ability to access **`localStorage`**

evil.org - Totally Legitimate Online

evil.org

Search

✗ Your Bank – Trustworthy Online Banking

Welcome to your absolutely legitimate online banking platform.

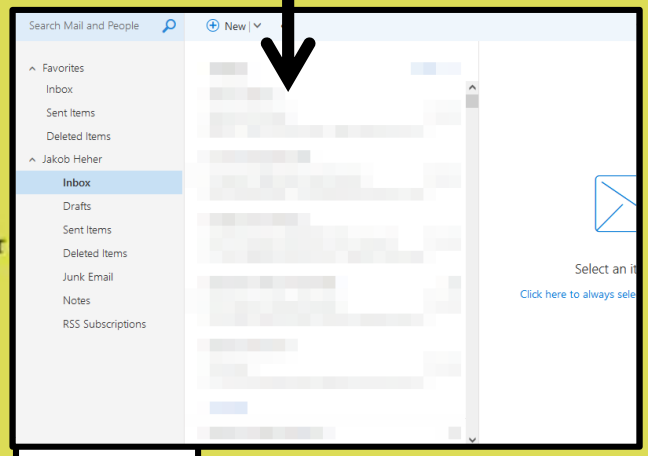
Please enter your secure password below to accept a **$500m donation** from the International Monetary Fund.

Your password: **Secure log in!**

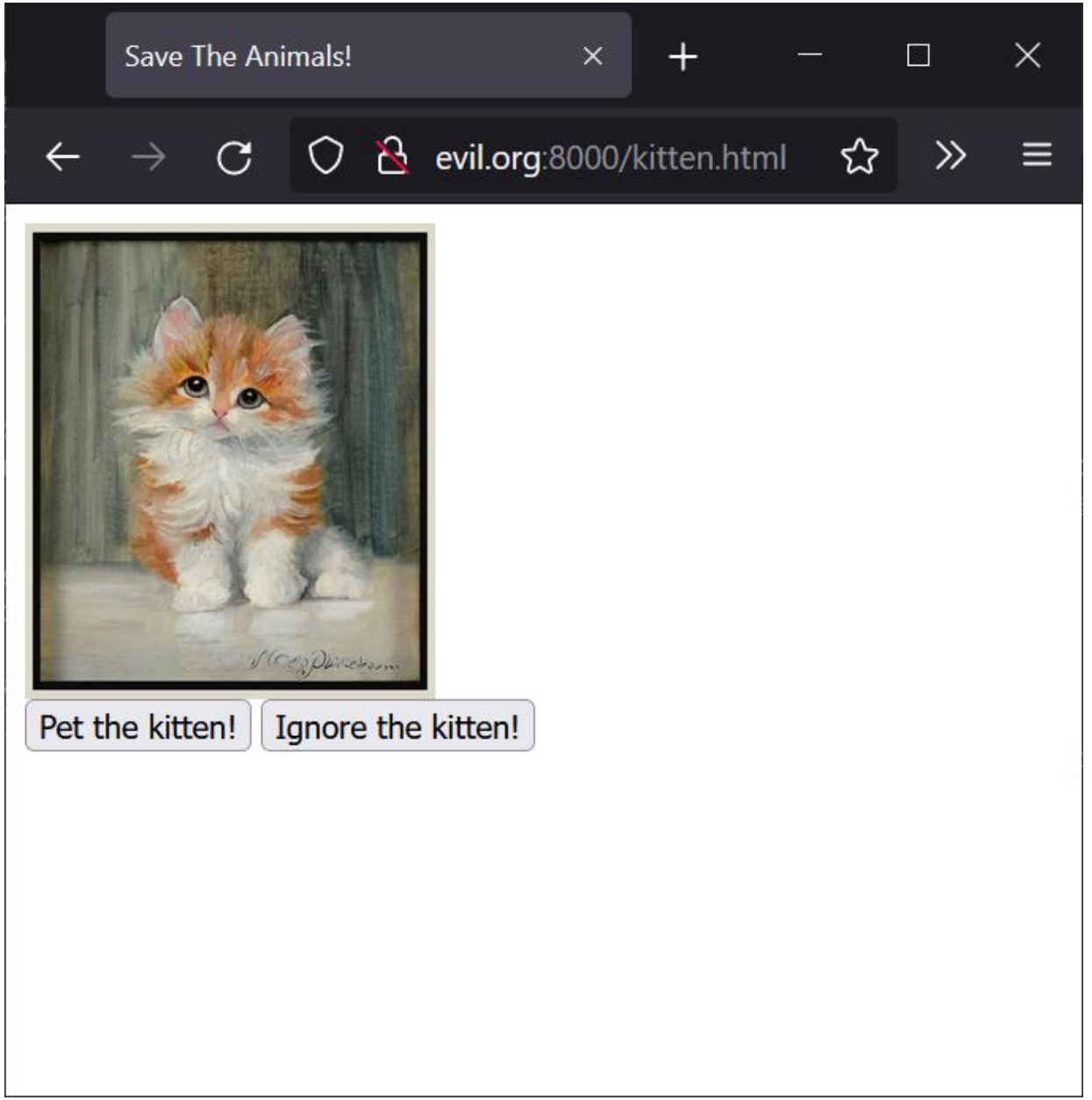
We swear this money actually exists. Look,

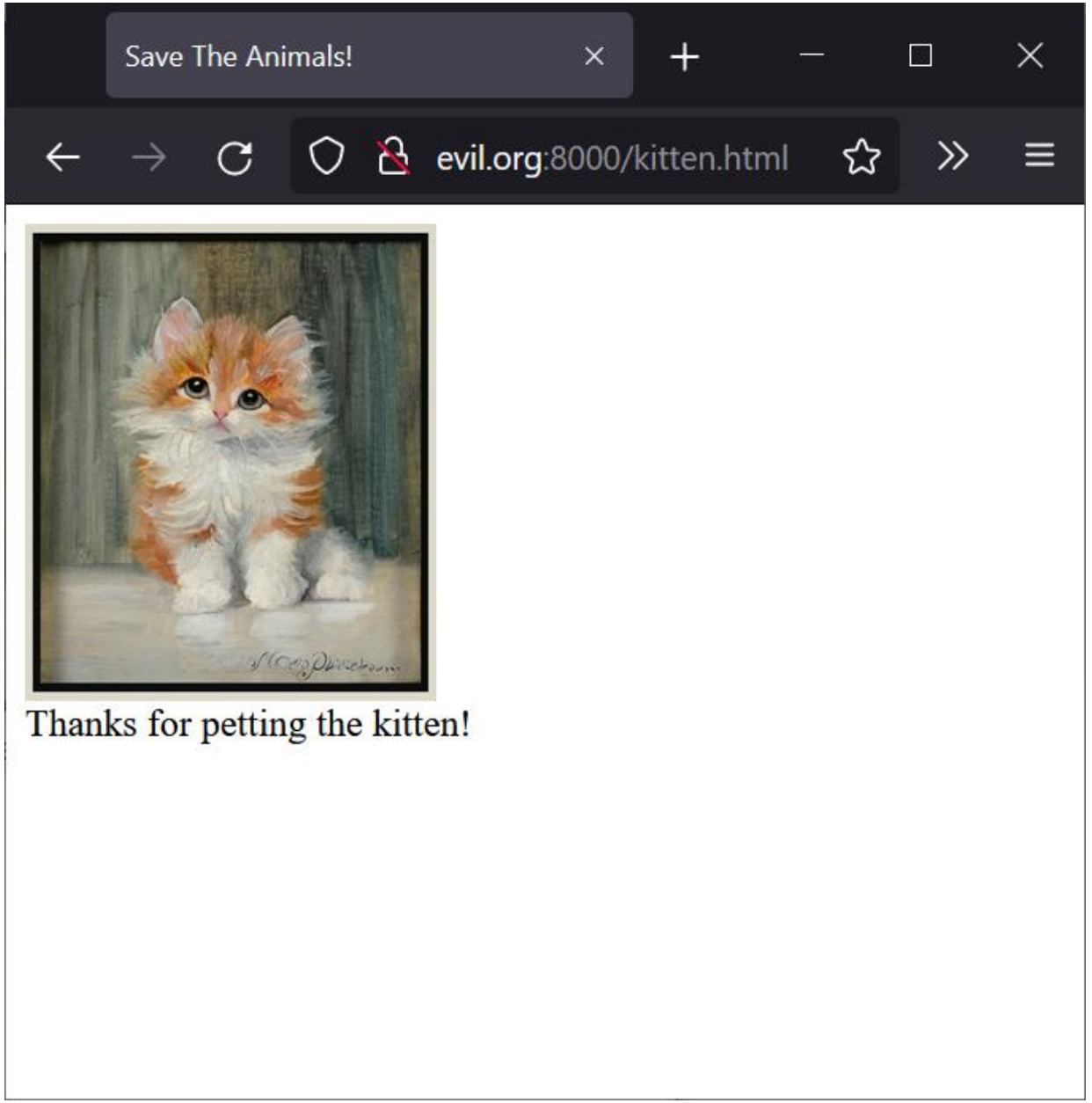
If you do not react within 5 (five) business days, your

no interaction with cross-origin iframe



<iframe>


Save The Animals!
evil.org:8000/kitten.html
Pet the kitten!
Ignore the kitten!


Save The Animals!
evil.org:8000/kitten.html
Thanks for petting the kitten!

C:\Users\jakob.heher\example_web\kitten.html - Notepad++ [Administrator]

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

kitten.html

```
<html>
    <head>
        <title>Save The Animals!</title>
        <style>
        body { position: relative; }
        img { height: 200px; }
        iframe {
            position: absolute; opacity: 0;
            left: -40px;
            top: 145px;
        }
        </style>
    </head>
    <body>
        <img src="kitten.jpg">
        <div><button>Pet the kitten!</button> <button>Ignore the kitten!</button></div>
        <iframe src="http://lawful.org:8000/create_admin_account.html?username=Eve&password=evulz"></iframe>
    </body>
    <script type="text/javascript">
        document.body.addEventListener('click', (e) =>
        {
            console.log(e,e.target);
            if (e.target.closest('iframe'))
                document.getElementsByTagName('div')[0].innerText = 'Thanks for petting the kitten!';
        });
    </script>
</html>
```

Hyper Text Markup Langua | length : 896 lines : 28 | Ln : 8 Col : 43 Pos : 210 | Windows (CR LF) | UTF-8 | INS

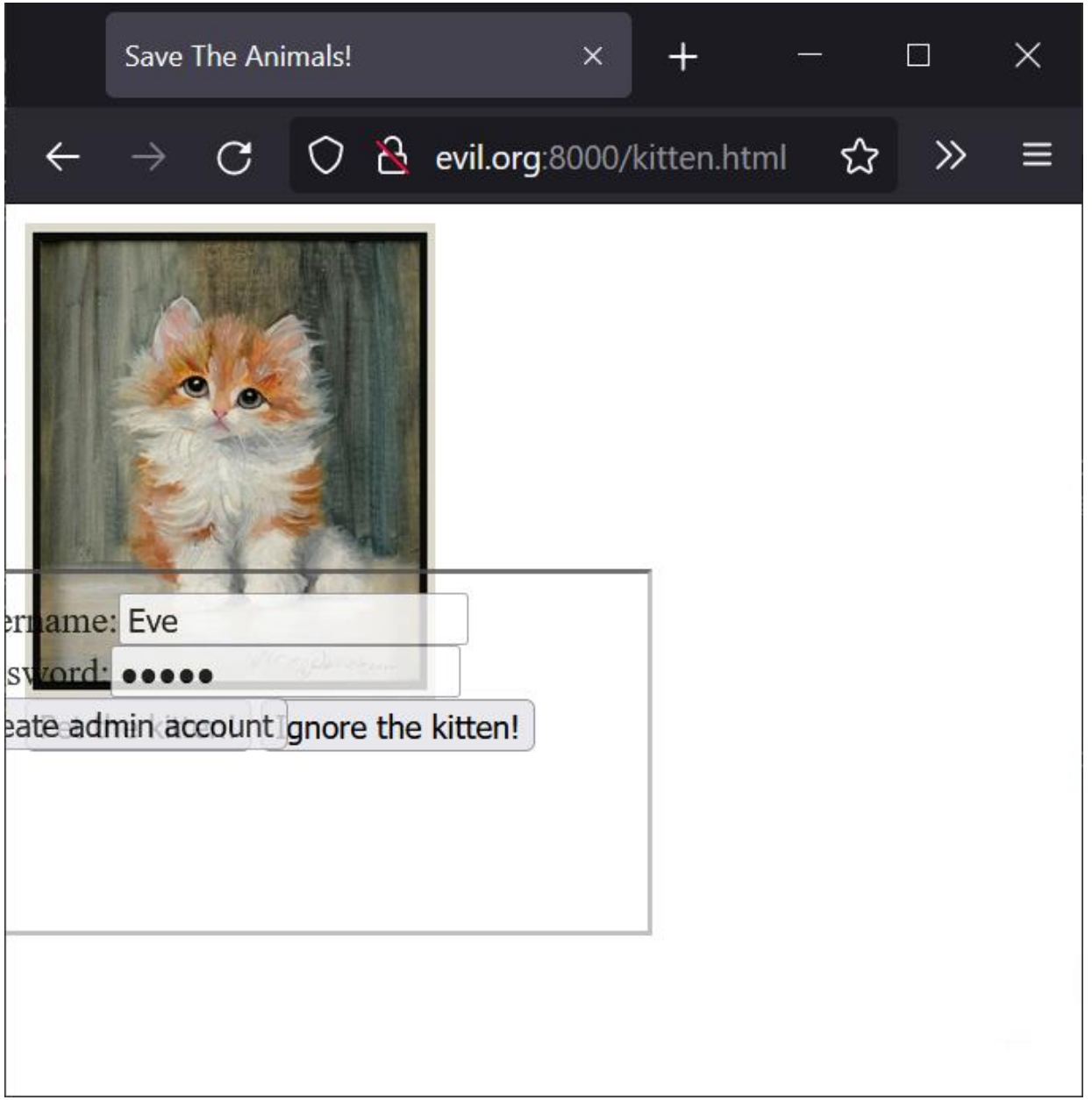
C:\Users\jakob.heher\example_web\kitten.html - Notepad++ [Administrator]

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

kitten.html

```
<html>
    <head>
        <title>Save The Animals!</title>
        <style>
        body { position: relative; }
        img { height: 200px; }
        iframe {
            position: absolute; opacity: 0.8;
            left: -40px;
            top: 145px;
        }
        </style>
    </head>
    <body>
        <img src="kitten.jpg">
        <div><button>Pet the kitten!</button> <button>Ignore the kitten!</button></div>
        <iframe src="http://lawful.org:8000/create_admin_account.html?username=Eve&password=evulz"></iframe>
    </body>
    <script type="text/javascript">
        document.body.addEventListener('click', (e) =>
        {
            console.log(e,e.target);
            if (e.target.closest('iframe'))
                document.getElementsByTagName('div')[0].innerText = 'Thanks for petting the kitten!';
        });
    </script>
</html>
```

Hyper Text Markup Langua | length : 898 lines : 28 | Ln : 8 Col : 45 Pos : 212 | Windows (CR LF) | UTF-8 | INS


Save The Animals!
evil.org:8000/kitten.html
ername: Eve
sword: •••••
eate admin account
gnore the kitten!

What *can* Eve do?

- Embed arbitrary web pages as iframes

```
<iframe src="http://twitter.com/home?status=Don't%20Click:%20http://tinyurl.com/amgzs6" scrolling="no"></iframe>
```

- **Clickjacking**
  - For cookies: significantly harder with the **`SameSite=Lax`** default
  - **`localStorage`** is unprotected – the genuine JavaScript still runs

- Countermeasure: **`X-Frame-Options`** HTTP header
  - Prevents the page from being embedded in an attacker page

Token-based authentication

Token generation

"What's in the token?"

Stateful validation

- The server remembers the token, and can recognize it
- The token itself is just a "meaningless" random string

Stateless validation

- The server doesn't remember the token
- The token can be cryptographically validated somehow

Token storage

"Where does Bob keep the token?"

In the browser, somehow…

- If Eve compromises the token, Eve wins by default
- There are different ways to ask Bob's browser to store information…

Token generation:

Random Session Token

- Server picks a random, “meaningless” session token
- Server remembers that this session token belongs to Bob
- Now Bob is authenticated by this session token

- Potential problems:
  - Tokens must be unpredictable (good randomness!)
  - Tokens must be chosen by the server
    - Don’t just create a session for unknown tokens! (**Session Fixation** attacks)
- Not infinitely scalable…

If we’re talking millions of users…

Token generation:

JSON Web Tokens

- Cryptographically signed *claim* (e.g., “I am Bob”)
- Signed by the server
- Server can verify the claim without needing to remember tokens!
  - Might be a different server! (e.g., Login server signs, front-end verifies)

```
const token = base64urlEncoding(header) + '.' + base64urlEncoding(payload) + '.' + base64urlEncoding(signature)
```

- Impossible to “expire” or “invalidate” this token
  - Need to build expiration into the payload!

Token generation:

JSON Web Tokens

- Server can verify the claim without needing to remember tokens!

```
const token = base64urlEncoding(header) + '.' + base64urlEncoding(payload) + '.' + base64urlEncoding(signature)
```

Header

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Payload

```
{
  "loggedInAs": "admin",
  "iat": 1422779638
}
```

Signature

```
HMAC_SHA256(
  secret,
  base64urlEncoding(header) + '.' +
  base64urlEncoding(payload)
)
```

"HS256" <-> HMAC_SHA256

Token generation:

JSON Web Tokens

- Server can verify the claim without needing to remember tokens!

```
const token = base64urlEncoding(header) + '.' + base64urlEncoding(payload) + '.' + base64urlEncoding(signature)
```

Header

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Payload

```
{
  "loggedInAs": "admin",
  "iat": 1422779638
}
```

Signature

```
HMAC_SHA256(
  secret,
  base64urlEncoding(header) + '.' +
  base64urlEncoding(payload)
)
```

Token generation:

JSON Web Tokens

- Server can verify the claim without needing to remember tokens!

```
const token = base64urlEncoding(header) + '.' + base64urlEncoding(payload) + '.' + base64urlEncoding(signature)
```

- ***Never*** trust the **`alg`** field
  - You should already know what algorithm your tokens use!

- Potential shenanigans:
  - **`alg: "none"`**
  - **`alg: "hs256"`** <-> **`alg: "rs256"`**

Taking the gloves off:

Cross-Origin Resource Sharing

- Sometimes we actually *want to* let cross-origin JavaScript access data!
  - Example: **`timeshare.company.org`** queries **`ical.company.org`**


```
await fetch('http://ical.company.org/employee_time.ics');
```

```
Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource
(Reason: CORS header 'Access-Control-Allow-Origin' missing). [Learn More]
```

Taking the gloves off:

Cross-Origin Resource Sharing

- Sometimes we actually *want to* let cross-origin JavaScript access data!
  - Example: **`timeshare.company.org`** queries **`ical.company.org`**

Not the same origin!

- The **`Access-Control-Allow-Origin`** HTTP header is required
  - Sent by the (potential) "victim" resource

✗ **`Access-Control-Allow-Origin: *`**

This is okay for public data APIs

  - Allows *any* origin (including **`evil.org`**!) to get response data

✓ **`Access-Control-Allow-Origin: https://timeshare.company.org`**

  - Allows *only the specified origin* to get response data
  - Need multiple origins? Check the **`Origin`** header on the incoming request!

Will be **`https://timeshare.company.org`**