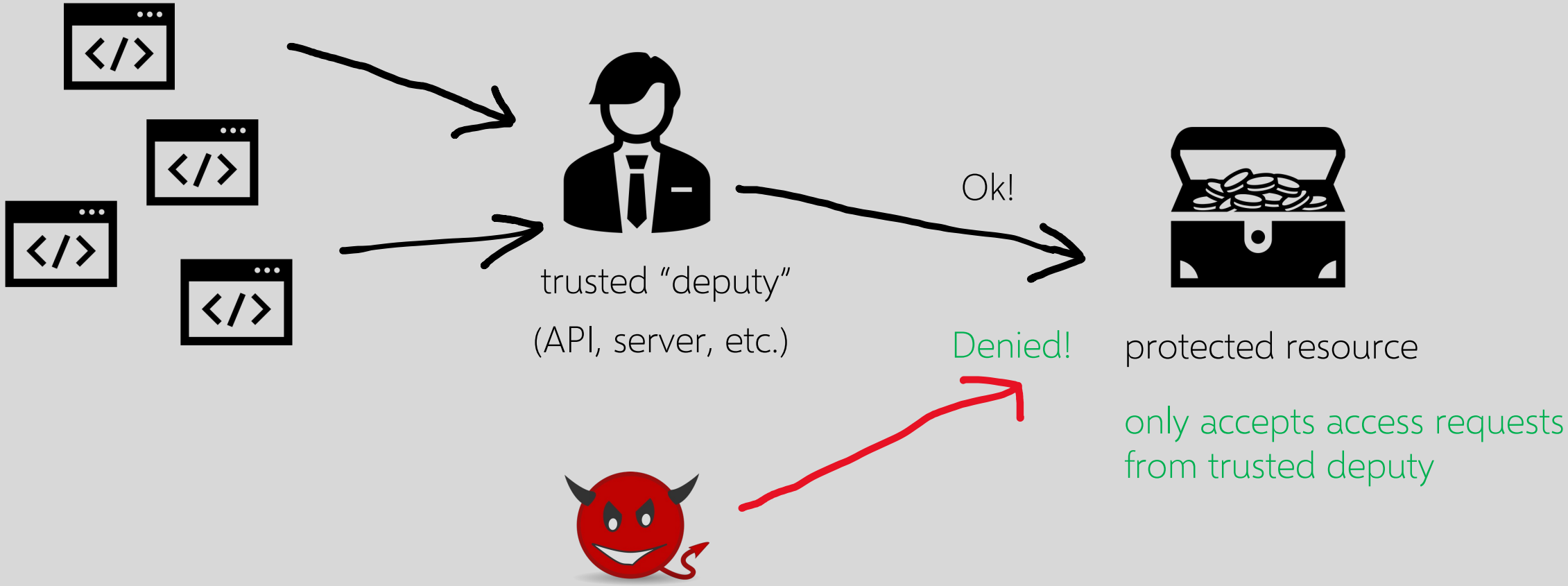
CROSS-SITE REQUEST FORGERY
SERVER-SIDE REQUEST FORGERY
CLICKJACKING

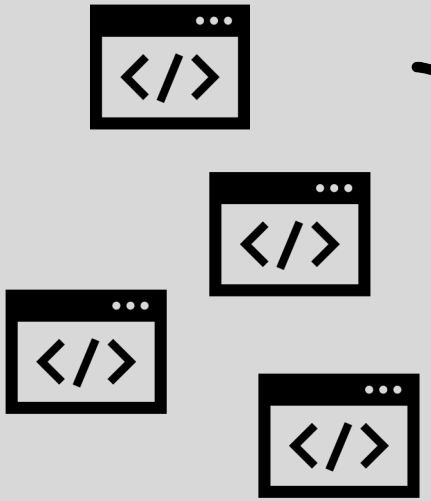
VITALY SHMATIKOV

Delegation and Access Control



Confused Deputy

Hardy. "The Confused Deputy, or why capabilities might have been invented" (1988).



trusted "deputy"

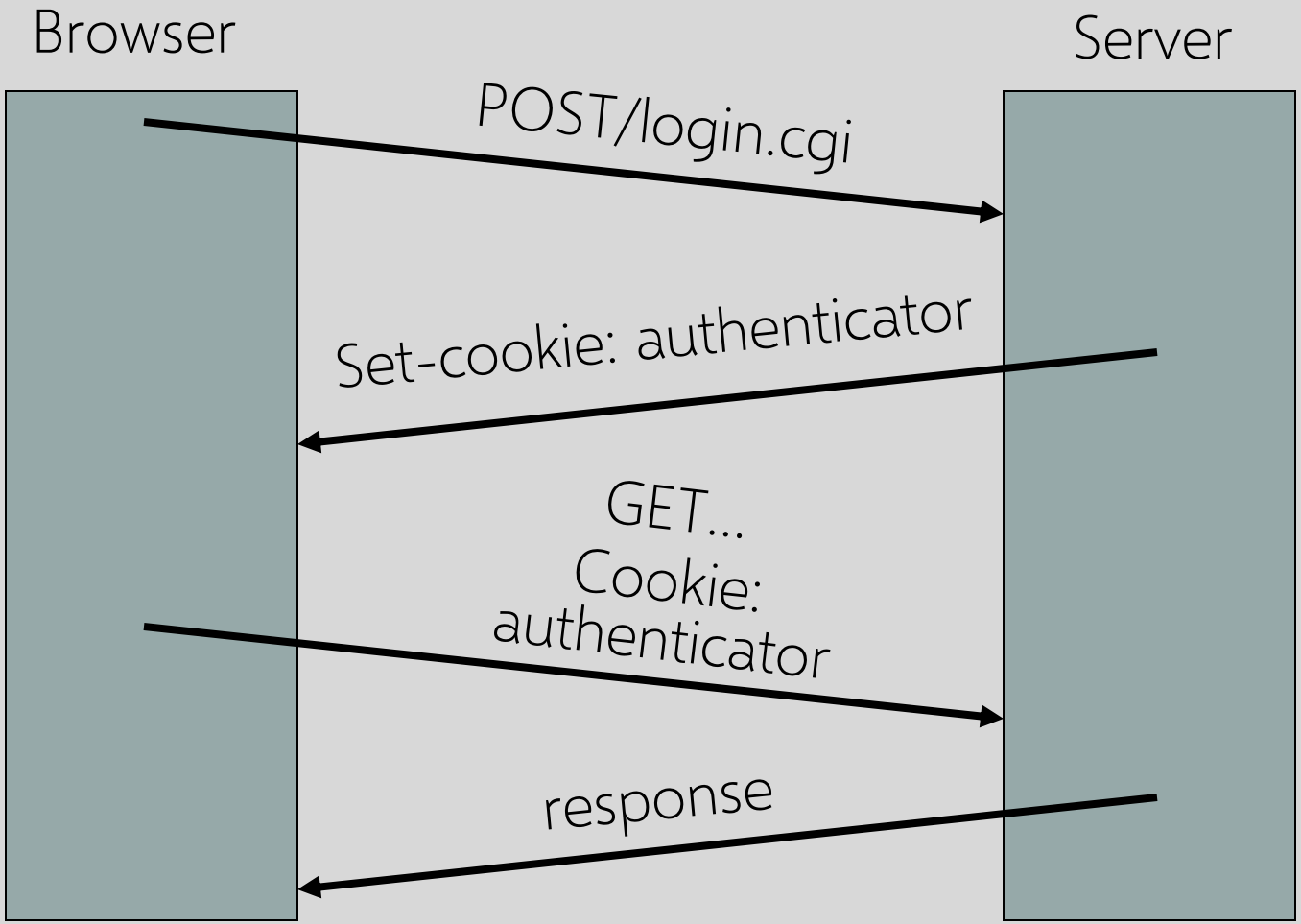
(API, server, etc.)

Ok!

protected resource

only accepts access requests from trusted deputy

Cookies-Based Authentication



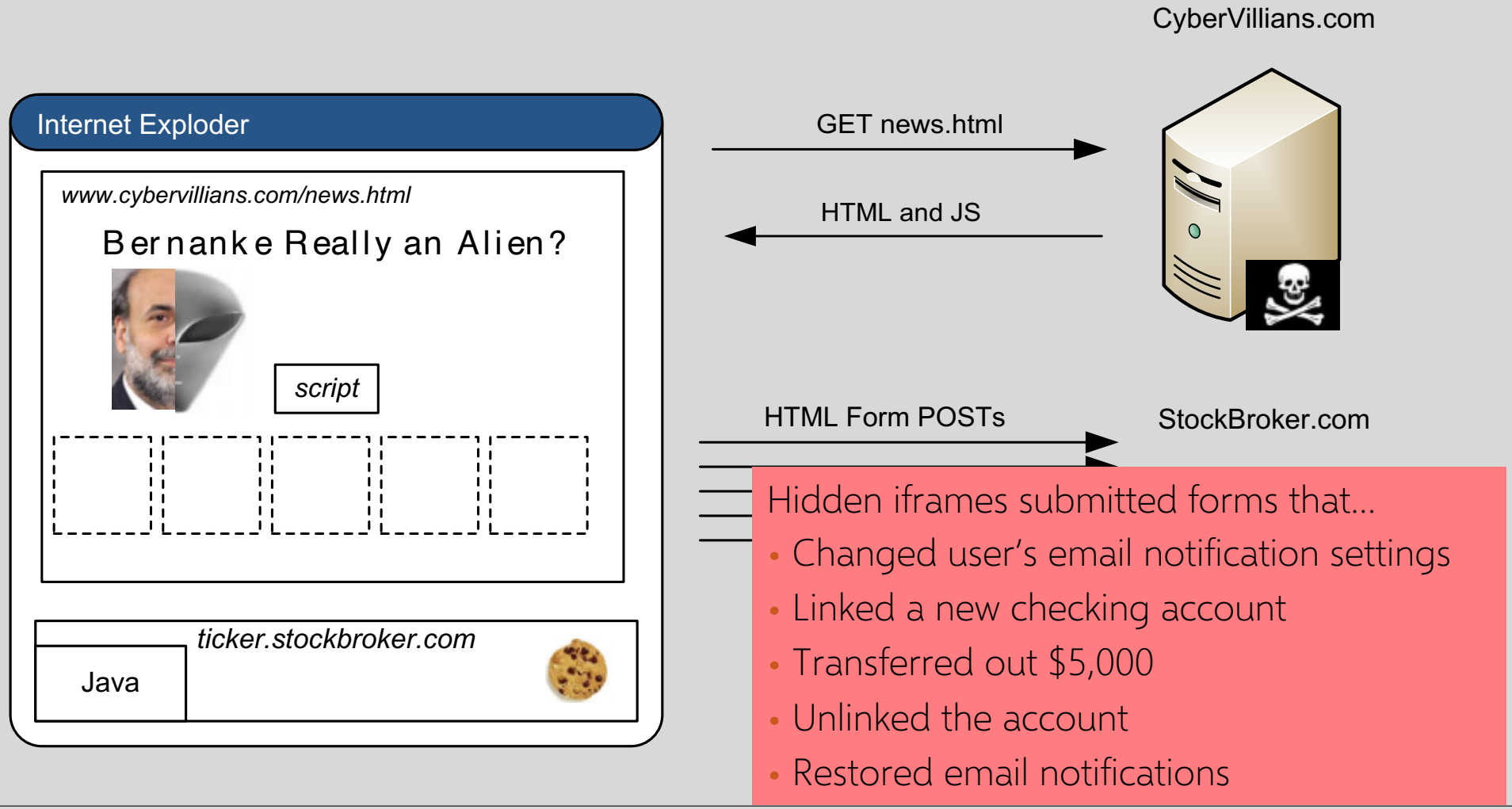
Source: Alex Stamos

XSRF True Story (1)

- User has a Java stock ticker from his broker's website running in his browser
  - Ticker has a cookie to access user's account on the site
- A comment on a public message board on finance.yahoo.com points to "leaked news"
  - TinyURL redirects to cybervillians.com/news.html
- User spends a minute reading a story, gets bored, leaves the news site
- Gets his monthly statement from the broker - $5,000 transferred out of his account!

Source: Alex Stamos

XSRF True Story (2)



Hidden iframes submitted forms that…

- Changed user's email notification settings
- Linked a new checking account
- Transferred out $5,000
- Unlinked the account
- Restored email notifications

Browser Sandbox Redux

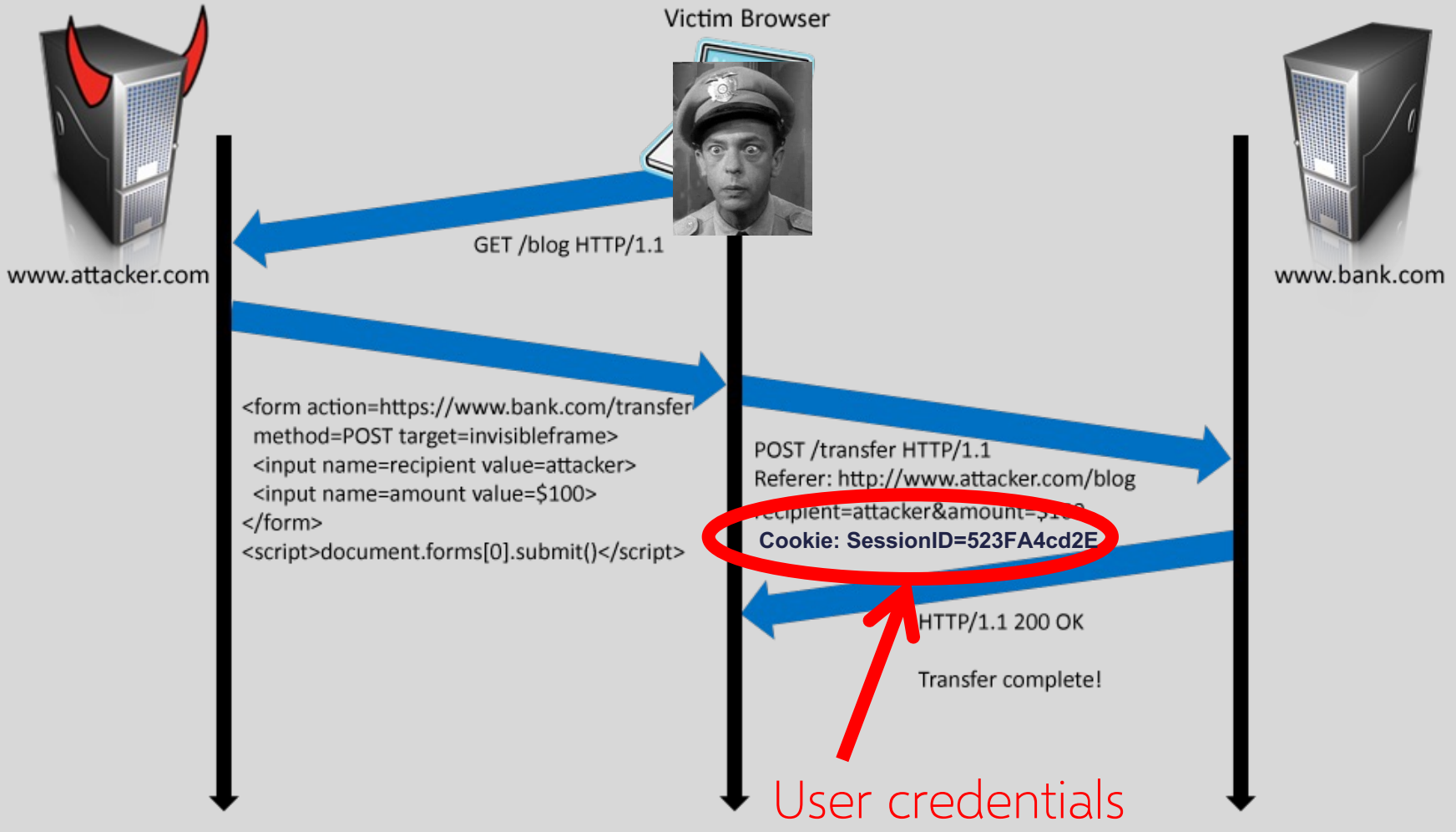
Based on the same origin policy (SOP)

Active content (scripts) can send anywhere

- Except for some ports such as SMTP

Can only read response from the same origin

Cross-Site Request Forgery



Cross-Site Request Forgery

User logs into bank.com, forgets to sign off

- Session cookie remains in browser state

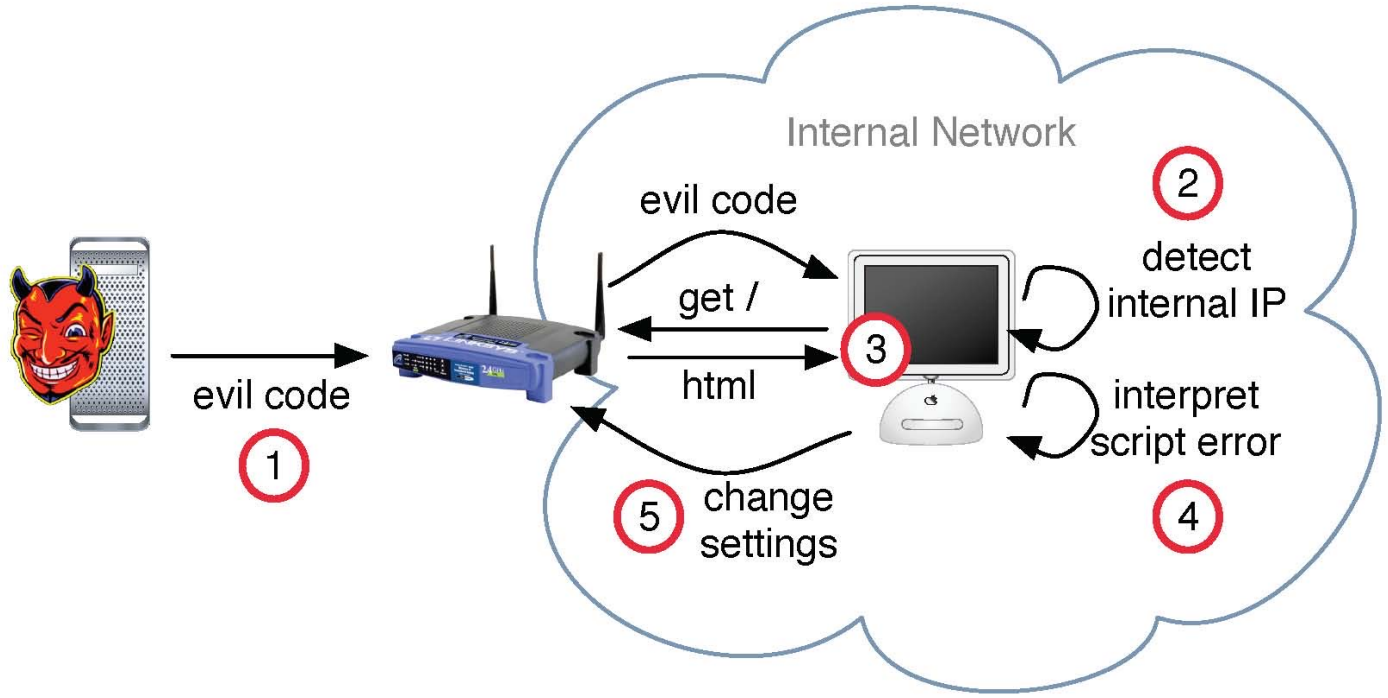
User then visits a malicious website containing

```
<form name=BillPayForm
action=http://bank.com/BillPay.php>
<input name=recipient  value=badguy> …
<script> document.BillPayForm.submit(); </script>
```

Browser submits the form + cookie, payment request fulfilled!

Cookie authentication is not sufficient when side effects can happen!

Drive-By Pharming



Stamm et al. "Drive-By Pharming" (2006)

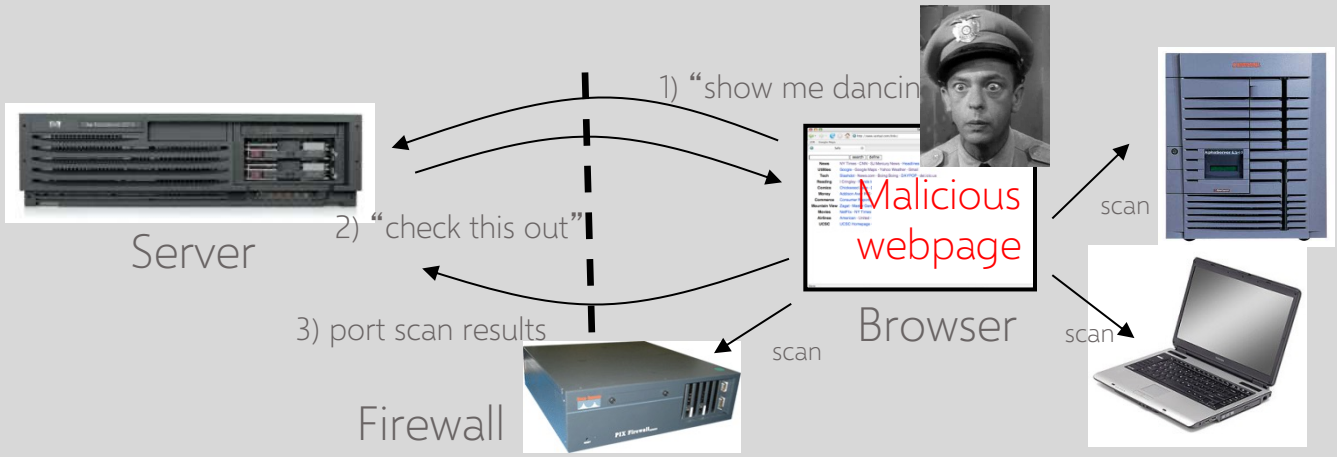
User is tricked into visiting a malicious site

Malicious script detects victim's address

- Socket back to malicious host, read socket's address

Next step: reprogram the router

Finding the Router



Script from a malicious site can scan local network without violating the same origin policy!

- Pretend to fetch an image from an IP address
- Detect success using onError

```
<IMG  SRC=192.168.0.1 onError = do()>
```

Determine router type by the image it serves

Basic JavaScript function, triggered when error occurs loading a document or an image… can have a handler

Sample JavaScript Code

```
<html><body><img id="test" style="display: none">
<script>
    var test = document.getElementById('test');
    var start = new Date();
    test.onerror = function() {
          var end = new Date();
          alert("Total time: " + (end - start));
     }
     test.src = "http://www.example.com/page.html";
</script>
</body></html>
```

When response header indicates that page is not an image, the browser stops and notifies JavaScript via the onError handle

Reprogramming the Router

Log into router

- 50% of home users use a broadband router with default or no password (2006 statistics)

  <script src="http://admin:password@192.168.0.1"></script>
- Or post a forged form to update the router config (cross-site request forgery)

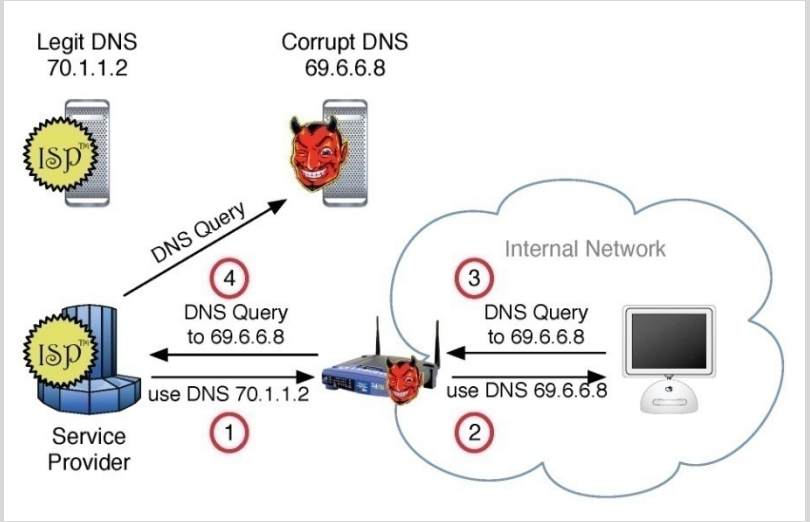
Replace DNS server address with address of an attacker-controlled DNS server

Risks of Drive-By Pharming

Completely 0wn the victim's Internet connection

Undetectable phishing: user goes to a financial site, attacker's DNS gives IP of attacker's site

Subvert anti-virus updates, etc.



Web attacker becomes a network attacker (more powerful!)

XSRF Defenses

Secret validation token

```
<input type=hidden value=23a3af01b>
```

Referer validation

```
Referer:
http://www.facebook.com/home.php
```

Custom HTTP header

```
X-Requested-By: XMLHttpRequest
```

Add Secret Token to Forms

Hash of user ID

<input type=hidden value=23a3af01b>

- Can be forged by attacker

Session ID

- If attacker has access to HTML or URL of the page (how?), can learn session ID

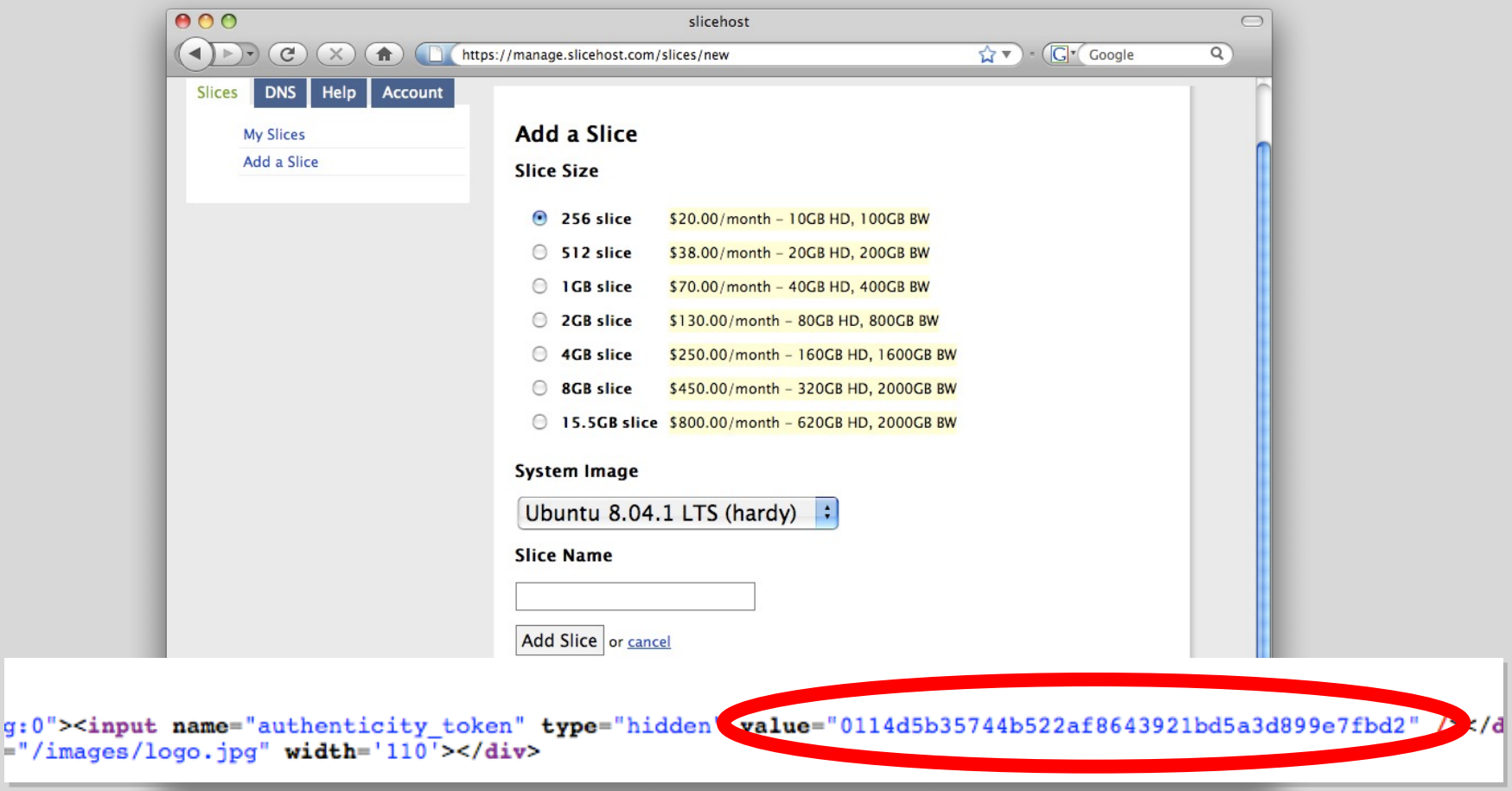
Session-independent nonce – Trac

- Can be overwritten by subdomains, network attackers

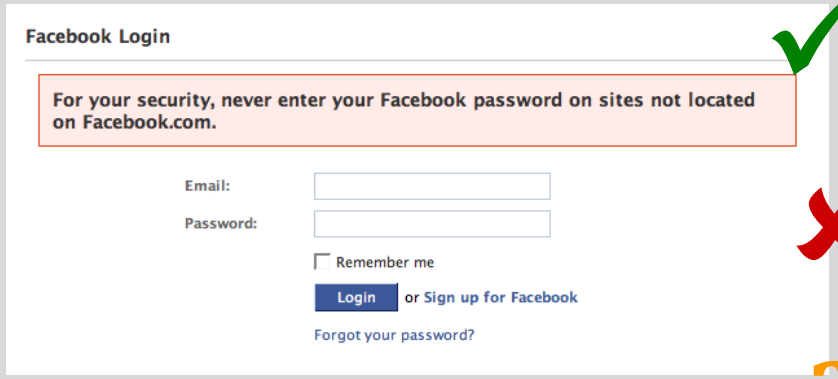
Need to bind session ID to the token

- CSRFx, CSRFGuard - manage state table at the server
- Keyed HMAC of session ID – no extra state!

Secret Token: Example



Referer Validation



Referer:
http://www.facebook.com/home.php

Referer:
http://www.evil.com/attack.html

Referer:

Lenient referer checking – header is optional

Strict referer checking – header is required

Why Not Always Strict Checking?

The referer header might be suppressed

- Stripped by the organization's network filter
  - For example, http://intranet.corp.apple.com/projects/iphone/competitors.html
- Stripped by the local machine
- Stripped by the browser for HTTPS → HTTP transitions
- User preference in browser
- Buggy browser

Web applications can't afford to block these users

Referer header rarely suppressed over HTTPS

Custom Header Forces Pre-Flight Check

XMLHttpRequest is for same-origin requests

For XMLHttpRequest to other origins, browser performs a "pre-flight" CORS check to see if the destination is willing to receive the request

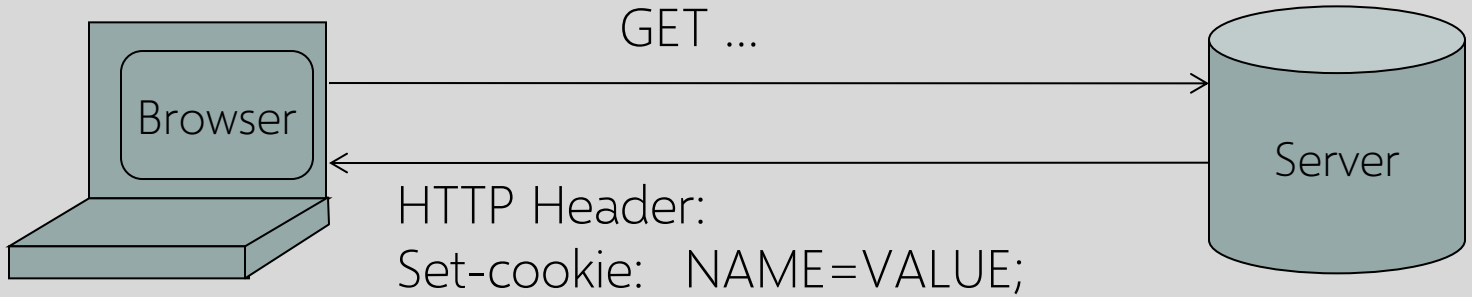
- … but typical GETs and POSTs don't require pre-flight check even if XMLHttpRequest

Adding a custom header to XMLHttpRequest forces pre-flight check because sites can only send custom headers to themselves, not other origins

Use **X-Requested-By** or **X-Requested-With**

```
X-Requested-By: XMLHttpRequest
```

SameSite Cookies



strict = cookie won't be sent even if user follows normal link

lax = cookie won't be sent with XSRF-prone methods like POST

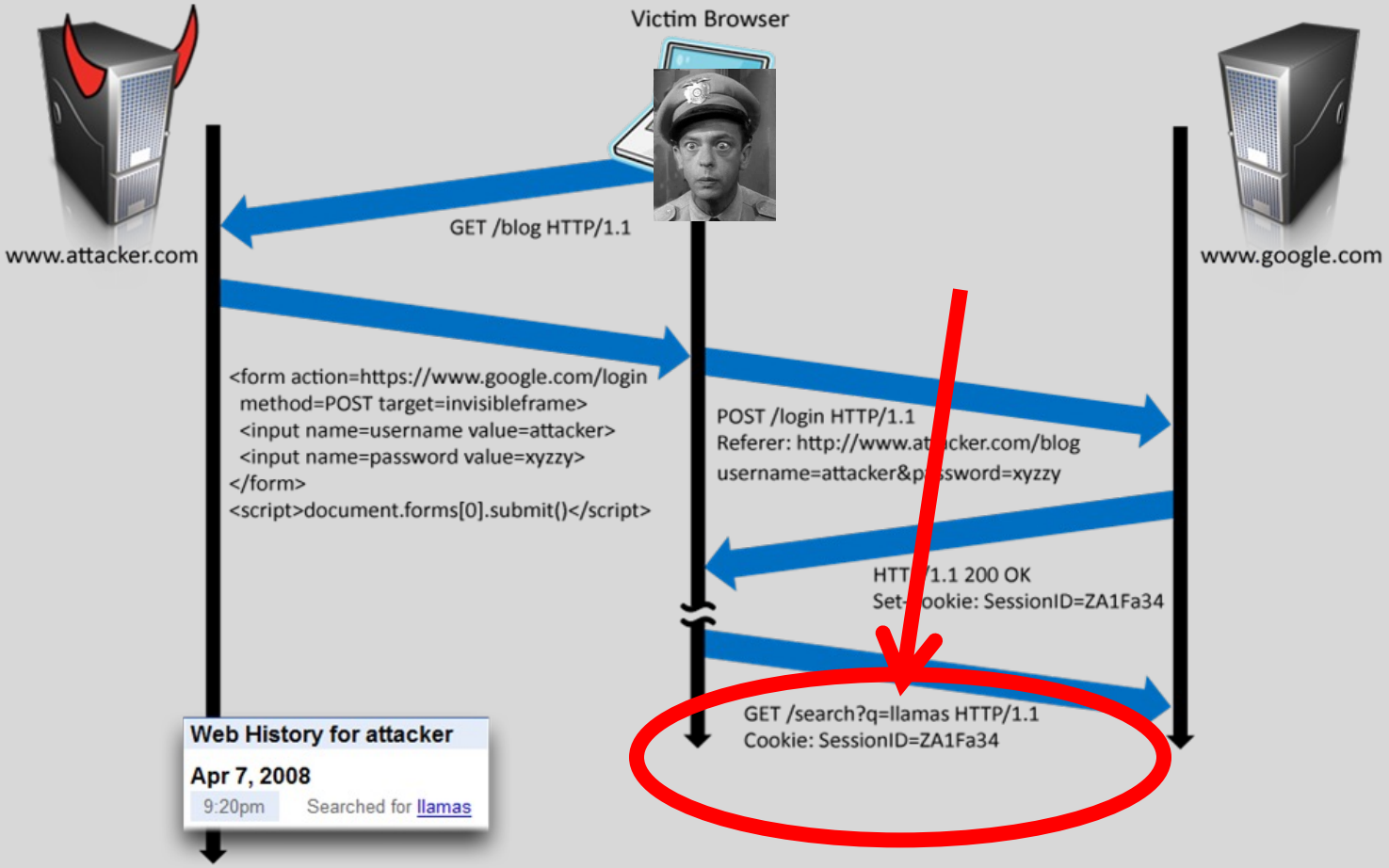
Broader View of XSRF

Abuse of cross-site data export

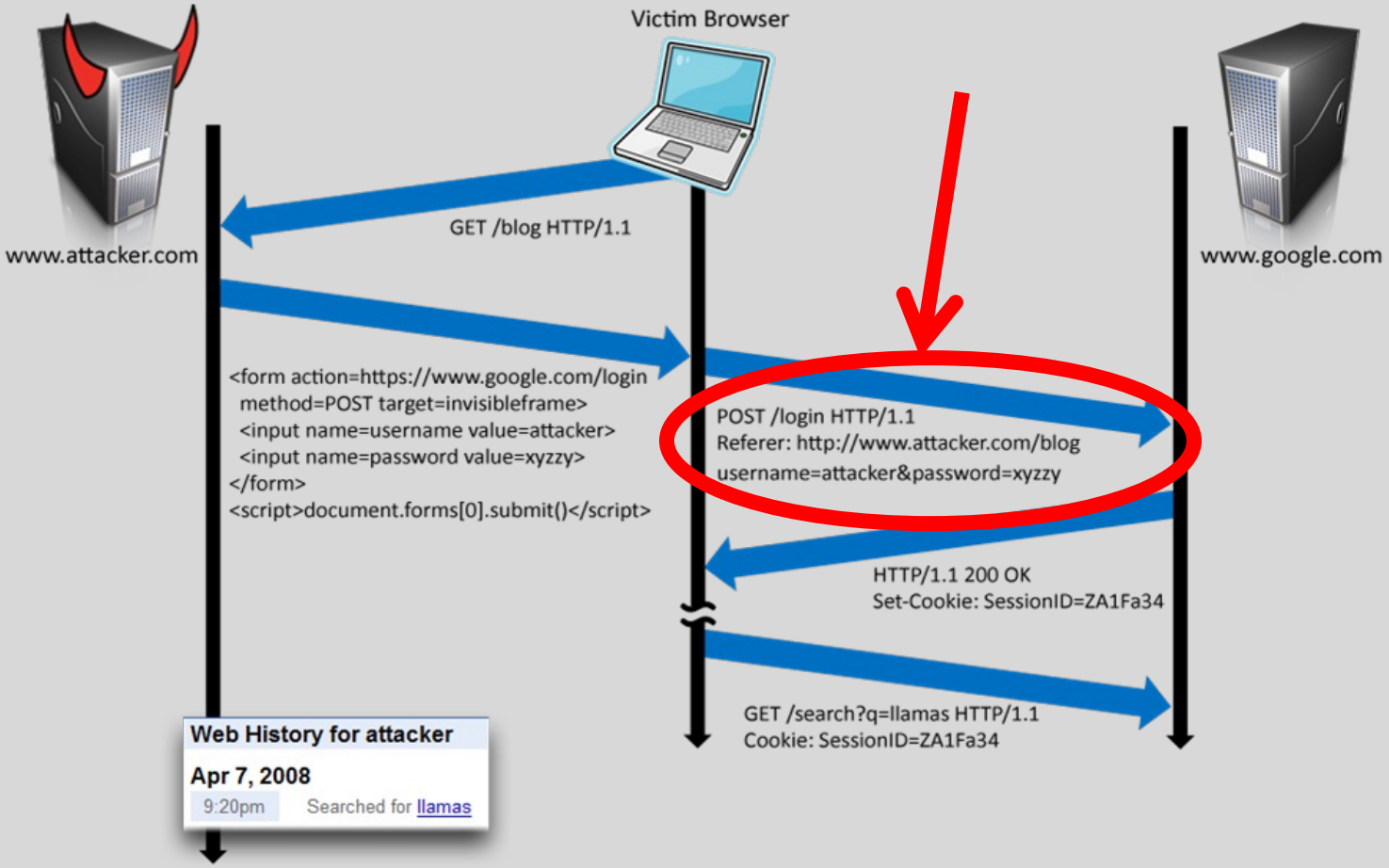
- SOP does not control data export
- Malicious webpage can initiate requests from the user's browser to an honest server
- Server thinks requests are part of the established session between the browser and the server

Many reasons for XSRF attacks, not just "session riding"

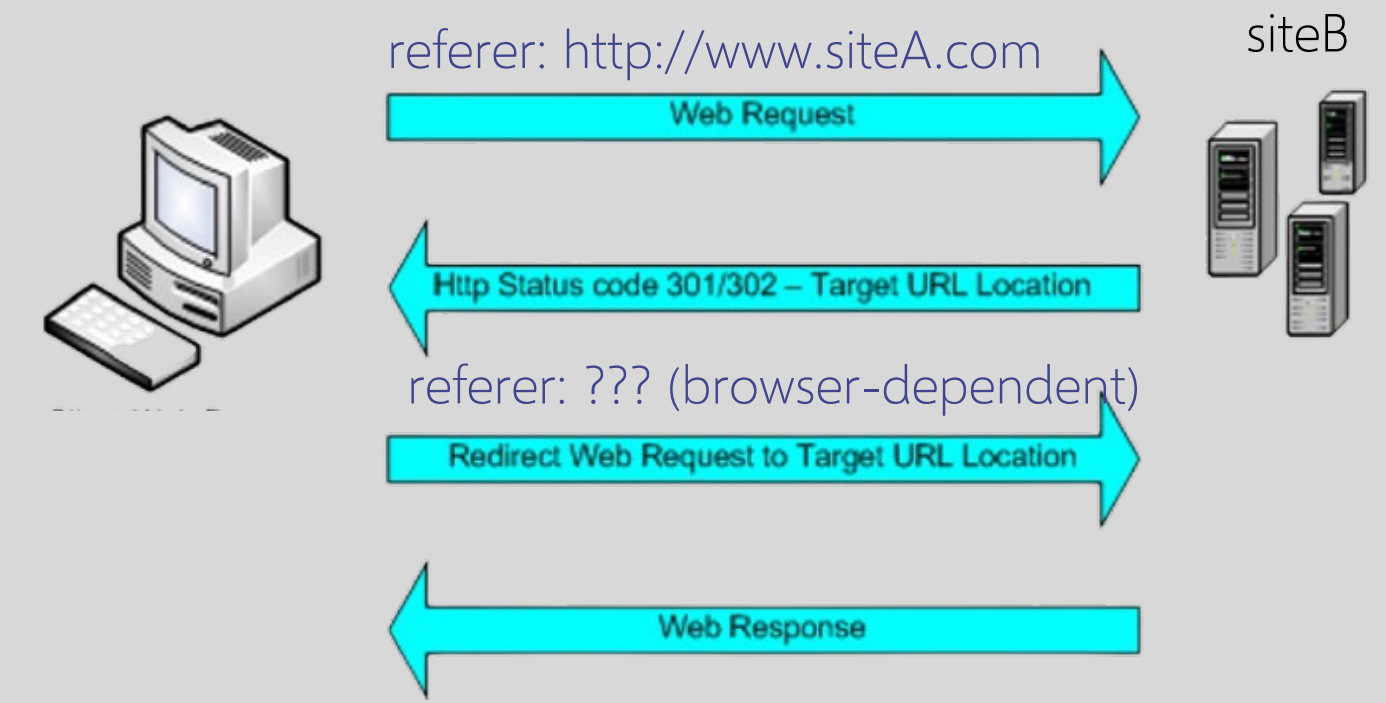
Login XSRF


Victim Browser
www.attacker.com
www.google.com
GET /blog HTTP/1.1
<form action=https://www.google.com/login
method=POST target=invisibleframe>
<input name=username value=attacker>
<input name=password value=xyzzy>
</form>
<script>document.forms[0].submit()</script>
POST /login HTTP/1.1
Referer: http://www.attacker.com/blog
username=attacker&password=xyzzy
HTTP/1.1 200 OK
Set-Cookie: SessionID=ZA1Fa34
GET /search?q=llamas HTTP/1.1
Cookie: SessionID=ZA1Fa34
Web History for attacker
Apr 7, 2008
9:20pm
Searched for llamas

Referer Header Should Help, Right?



Laundering Referer Header



Identity Misbinding Attacks

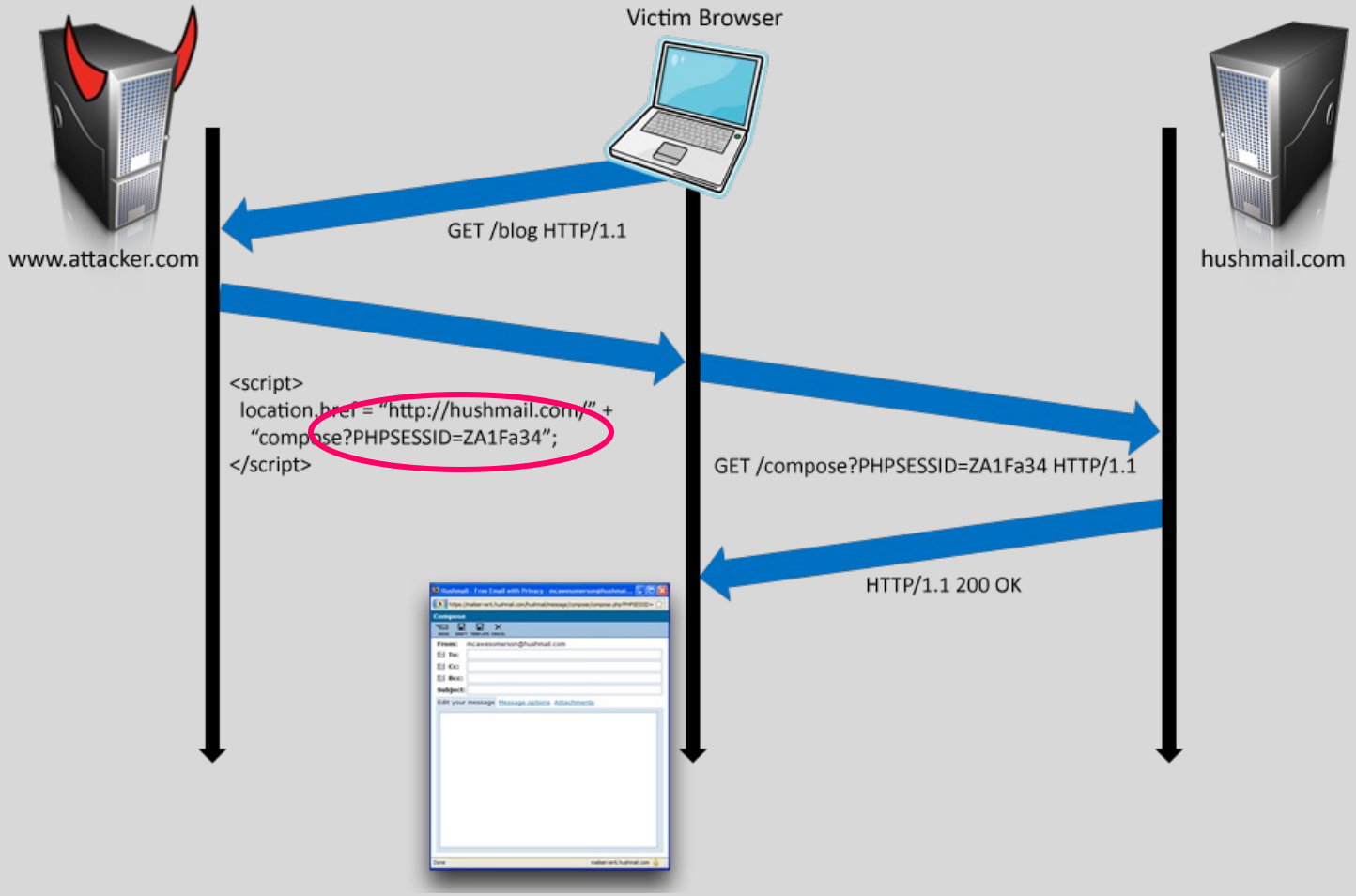
User's browser logs into website, but the session is associated with the attacker

- Capture user's private information (Web searches, sent email, etc.)
- Present user with malicious content

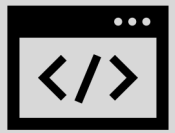
Many examples

- Login XSRF
- OpenID
- PHP cookieless authentication

PHP Cookieless Authentication



Confused Deputies Are Everywhere



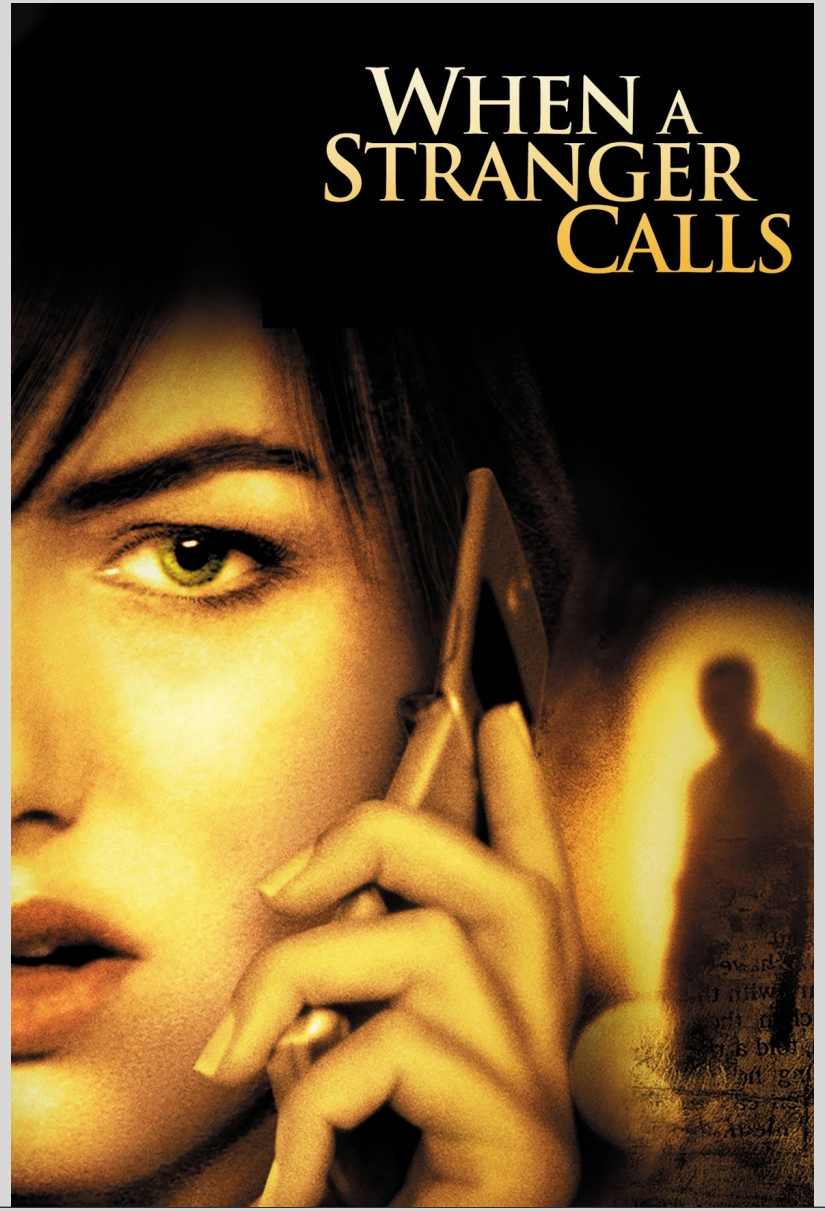
trusted “deputy”

(API, server, etc.)

Ok!

protected resource

only accepts access requests from trusted deputy



A girl is babysitting children in their home

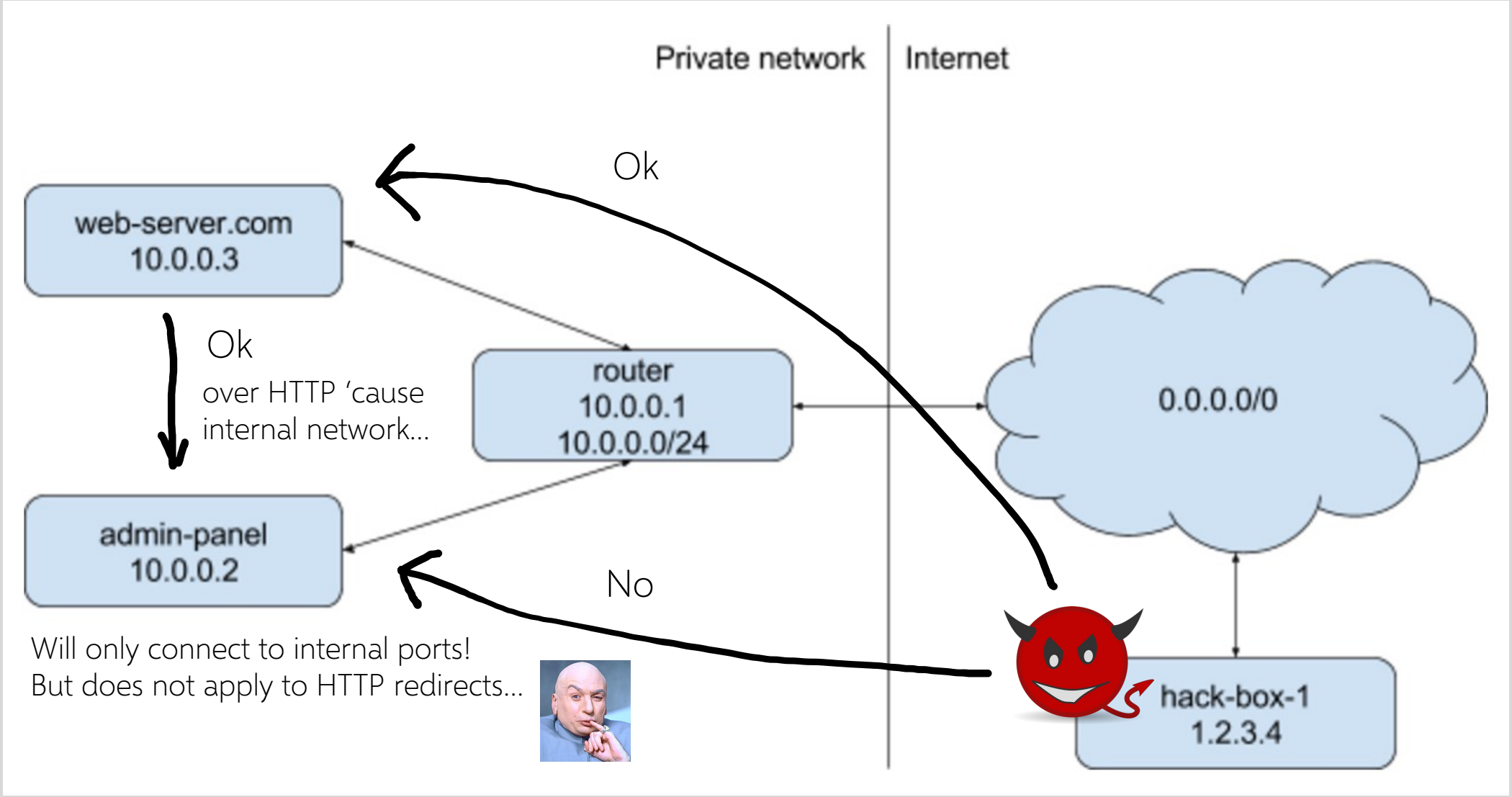
When the children are asleep upstairs, she is getting ominous calls asking her if she has checked on them

The babysitter calls the police

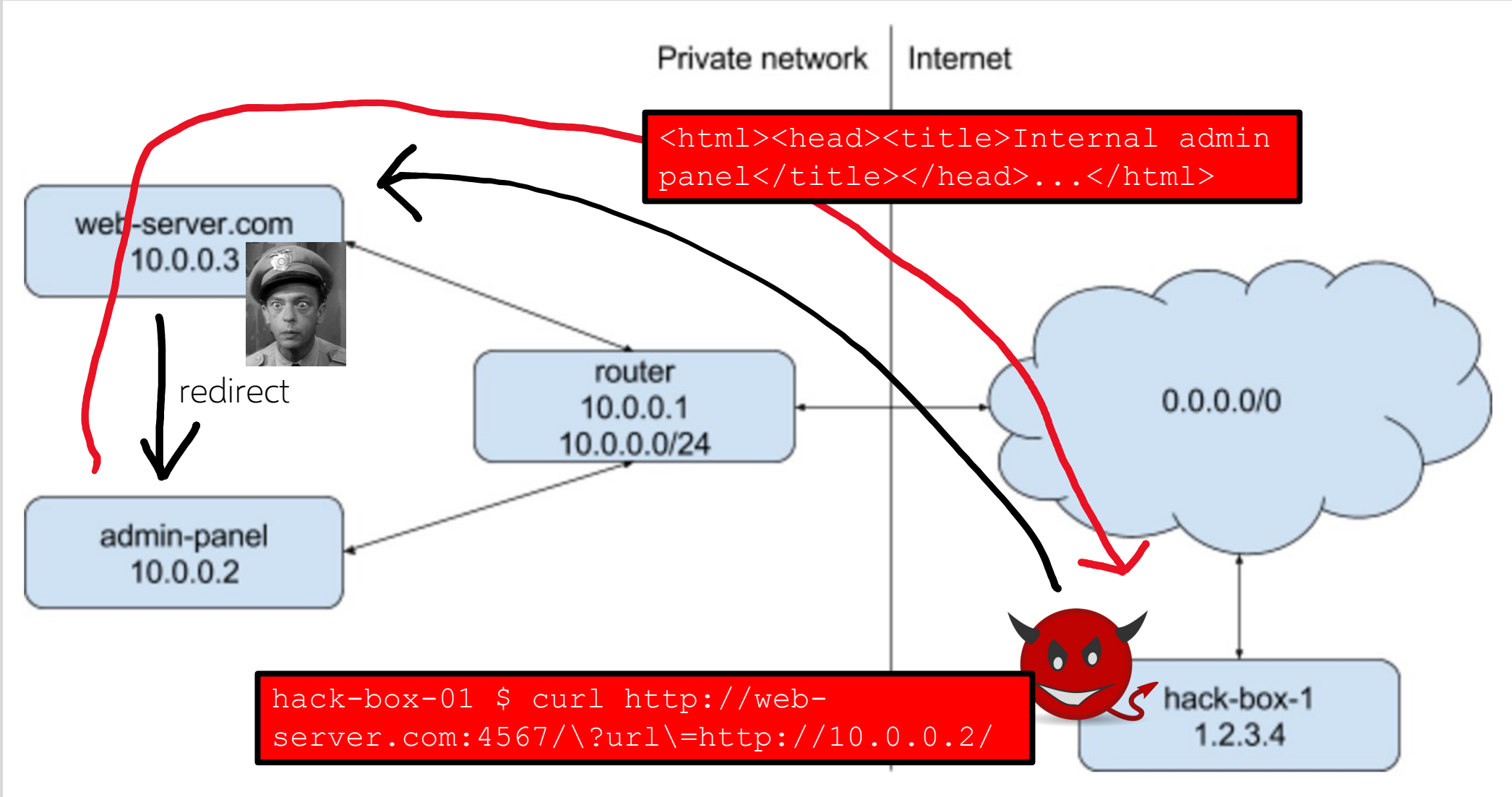
The police trace the call and tell her ...

THE CALL IS COMING FROM INSIDE THE HOUSE

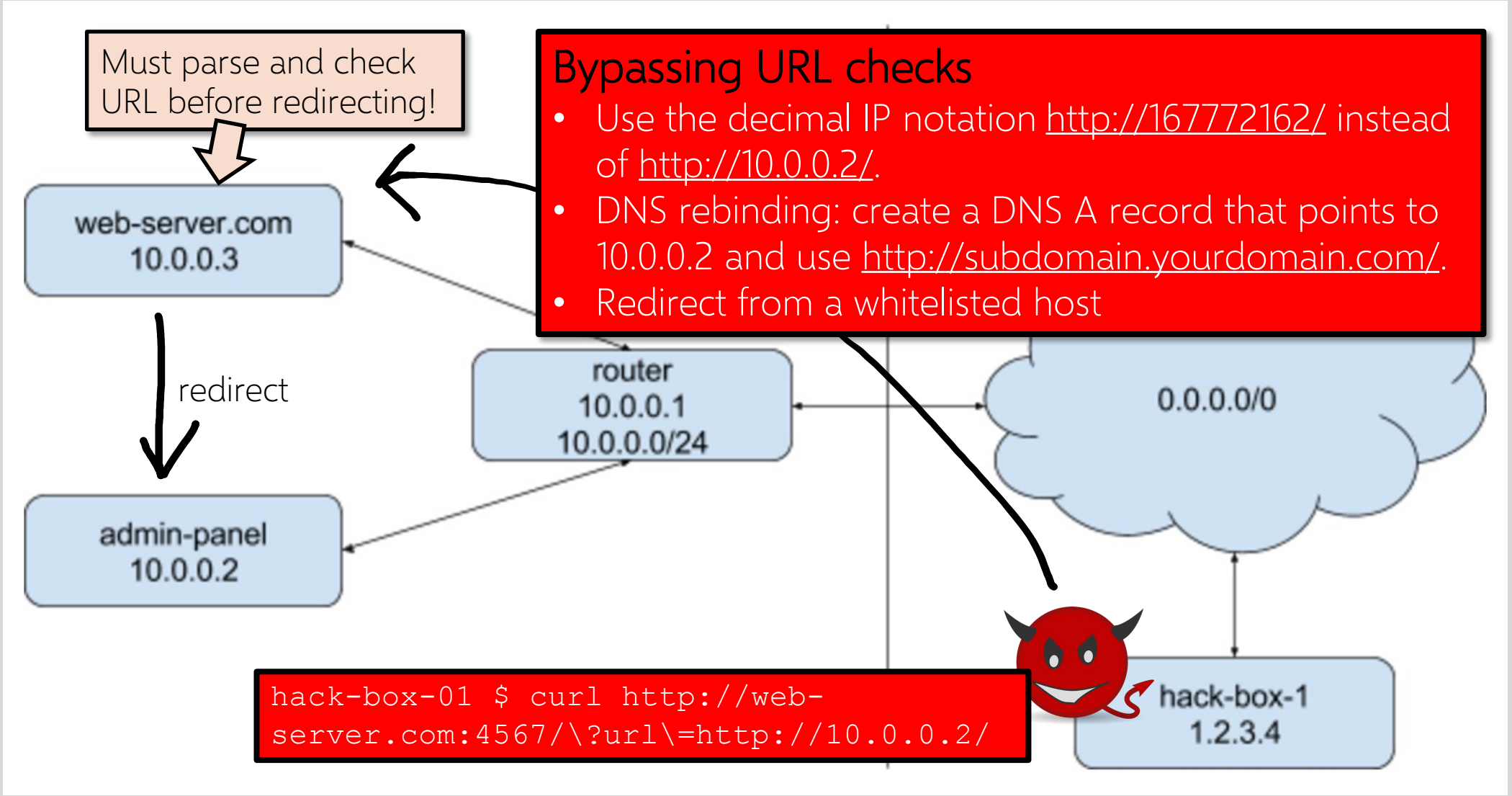
This is what server-side request forgery all about



https://www.hackerone.com/application-security/how-server-side-request-forgery-ssrf



https://www.hackerone.com/application-security/how-server-side-request-forgery-ssrf



https://www.hackerone.com/application-security/how-server-side-request-forgery-ssrf

Any Indirect Access Is Prone to SSRF!

Webhooks

- Look for services that make HTTP requests when certain events happen. In most webhook features, the end user can choose their own endpoint and hostname. Try to send HTTP requests to internal services.

PDF generators

- Try injecting <iframe>, <img>, <base> or <script> elements or CSS url() functions pointing to internal services.

Document parsers

- Try to discover how the document is parsed. In case it's an XML document, use the PDF generator approach. For all other documents, see if there's a way to reference external resources and let the server make requests to an internal service.

Link expansion

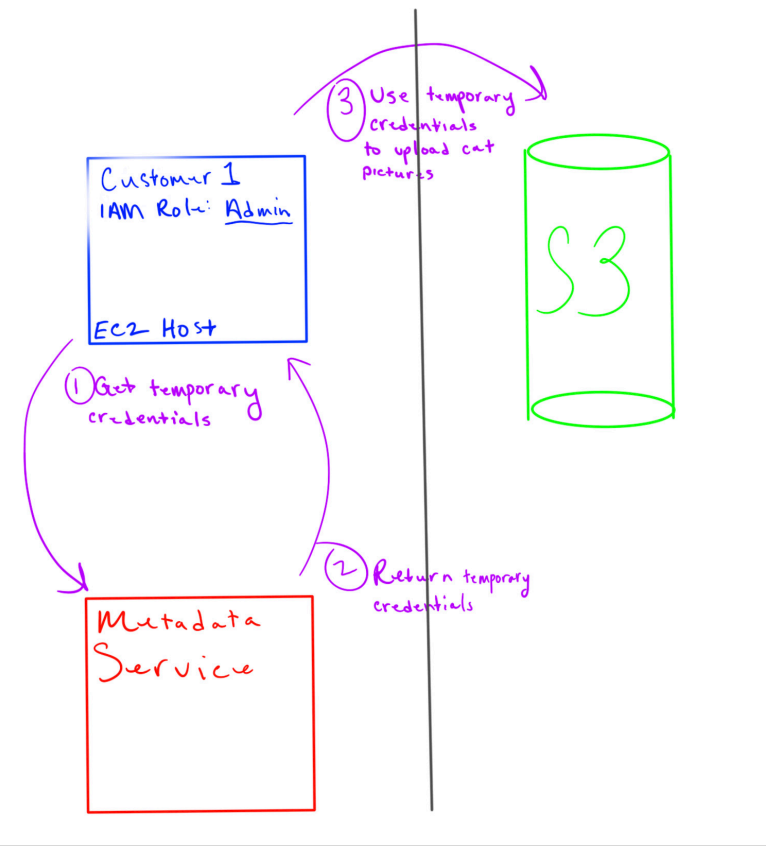
- Example: Twitter link expansion

File uploads

- Instead of uploading a file, try sending a URL and see if it downloads the content of the URL.

https://www.hackerone.com/application-security/how-server-side-request-forgery-ssrf

AWS Metadata Service

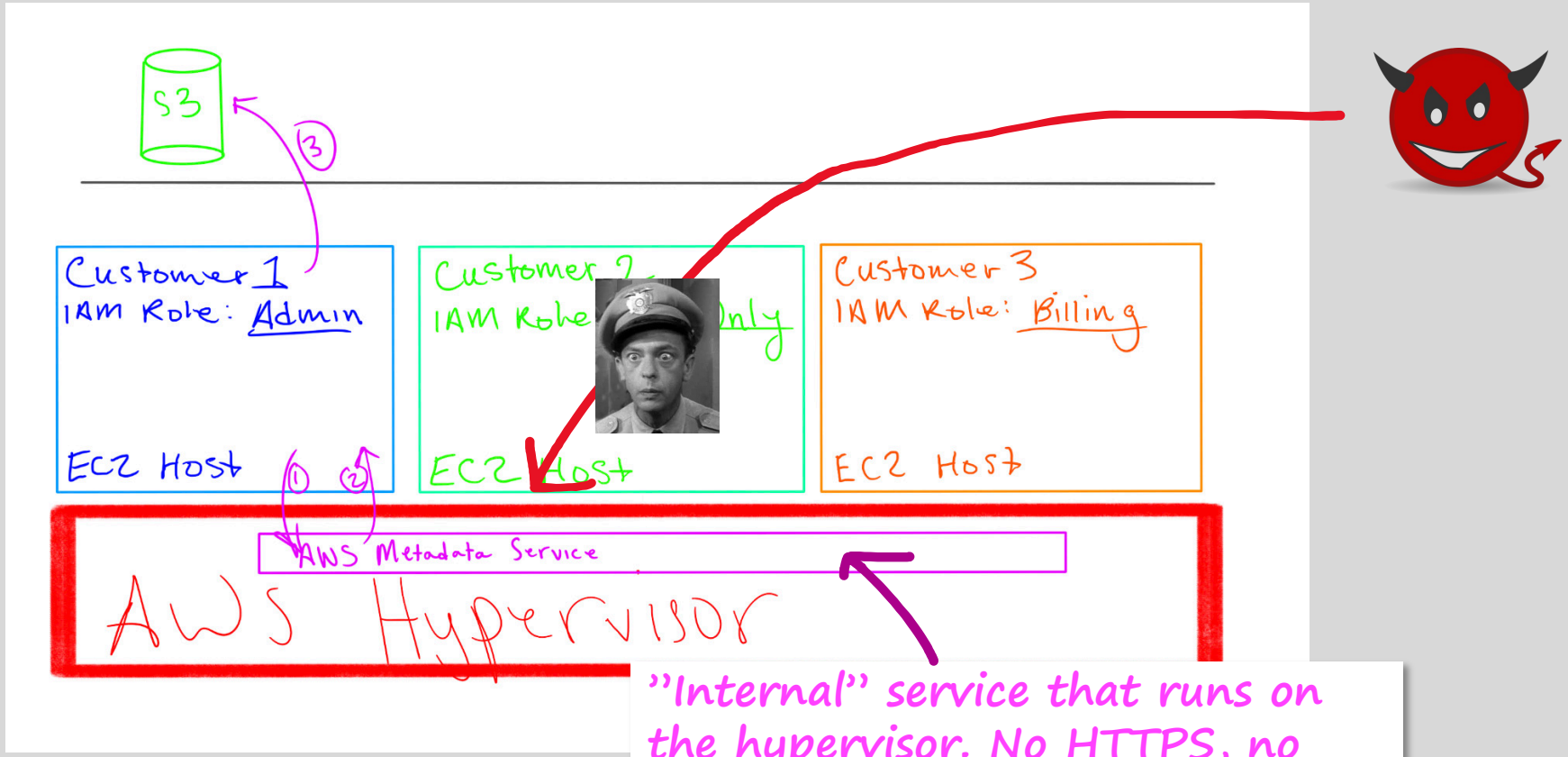


Good design!
Instead of directly handling IAM keys to access S3, use metadata service to obtain temporary credentials

https://ejj.io/blog/capital-one

Much more about
VM security later...

Multiple VMs on Same Hypervisor



"Internal" service that runs on the hypervisor. No HTTPS, no authentication, no authorization.

https://ejj.io/blog/capital-one

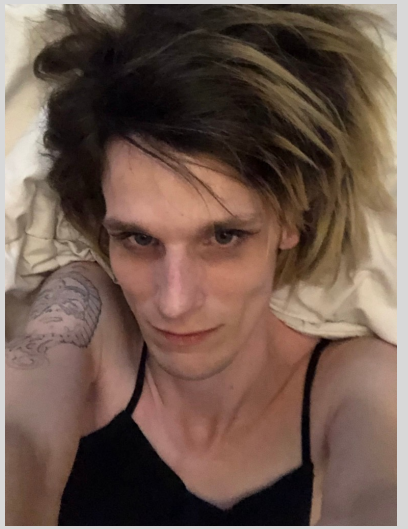
A hacker gained access to 100 million Capital One credit card applications and accounts

Capital One data breach: Arrest after details of 106m people stolen

Capital One fined $80 million for 2019 hack of 100 million credit card applications

Understanding the 2019 Capital One Attack

... a vulnerability in the WAF [Web Application Firewall] allowed a "Server Side Request Forgery" (SSRF) attack where the attacker manipulates a vulnerable web server to make new http requests on its behalf to access resources that the attacker should not have direct access to. The resource in this case was the AWS metadata service.



Paige Thompson,
the Capital One hacker

https://securityboulevard.com/2020/12/understanding-the-2019-capital-one-attack/

At Least 30,000 U.S. Organizations Newly Hacked Via Holes in Microsoft's Email Software

Early Jan 2021: four previously unknown ("zero-day") vulnerabilities in the Microsoft Exchange server

- #1: use SSRF to login into an administrator's account without authentication
- #2: gain ability to execute code (via insecure deserialization + stolen credentials)
- #3, #4: inject malicious code into any path on the server

Feb 23: Microsoft gives early warning and "proof-of-concept" attack code to its security partners via Microsoft Active Protections Program

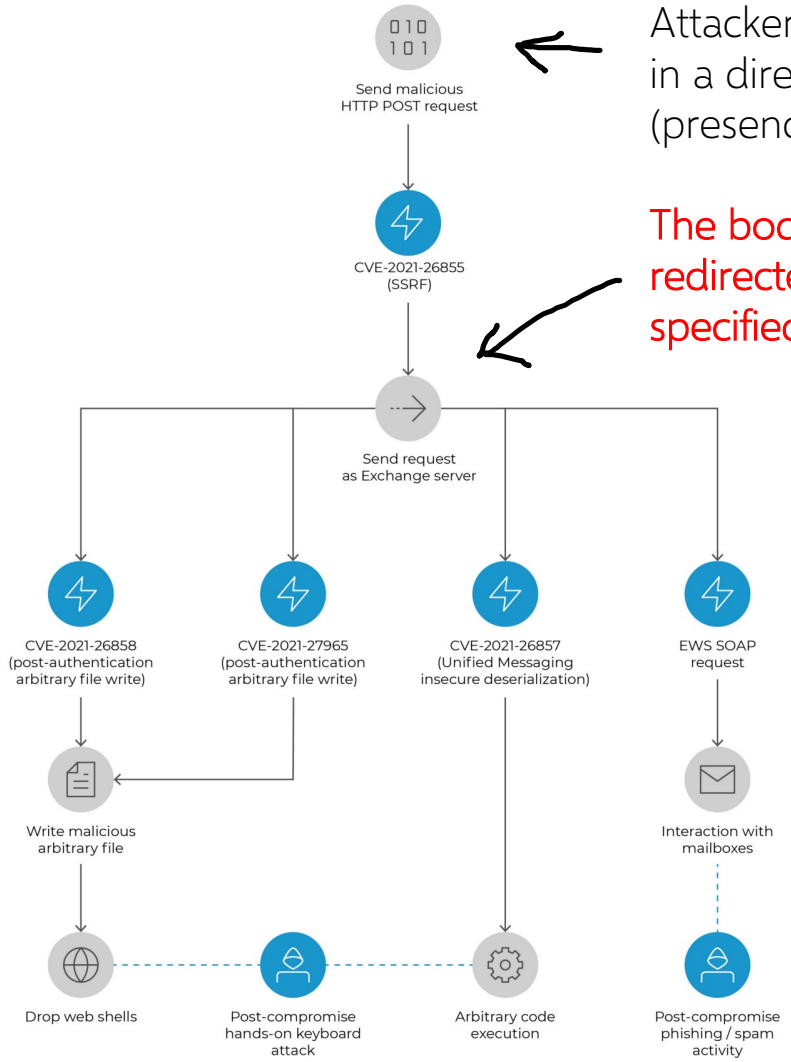
Feb 27-28: wave of attacks, attackers install backdoors to return later

Mar 2: Microsoft pushes patches to Exchange software

250,000 Exchange servers compromised worldwide

Headline:
Krebs on Security

Exploiting SSRF in MS Exchange



Attacker sends a POST request for a static file in a directory readable without authentication (presence of the file not required)

The body of the POST request will be redirected to any internal service specified in the X-BEResource cookie

The service thinks the request is coming from the mail-server account

Attacker then uses other vulnerabilities to overwrite files and inject malicious code

https://bi-zone.medium.com/hunting-down-ms-exchange-attacks-part-1-proxylogon-cve-2021-26855-26858-27065-26857-6e885c5f197c

South Africa's mobile fraud problem – fleecing millions from accounts

Staff Writer 1 September 2020

Mobile users in South Africa are very often subscribed to mobile services without their consent… South Africans are mostly at risk from a very basic fraudulent mobile activity, clickjacking.
"Clickjacking is a type of mobile-based fraud that is more than five years old and could be blocked very quickly if local market players took this threat seriously."

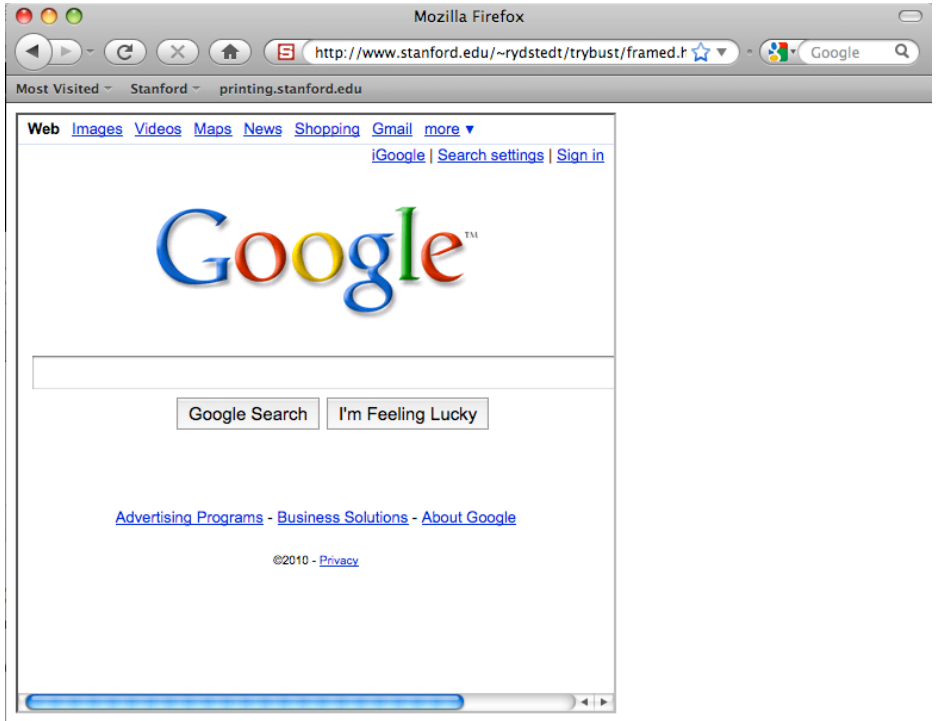
https://www.evina.com/press-releases/south-africa-has-a-massive-mobile-fraud-problem/

Clickjacking (UI Redressing)

Attacker overlays multiple transparent or opaque frames to trick a user into clicking

Clicks meant for the visible page are hijacked and routed to another, invisible page

Hansen and Grossman (2008)



It's All About the iFrame

- Any site can frame any other site

  <iframe

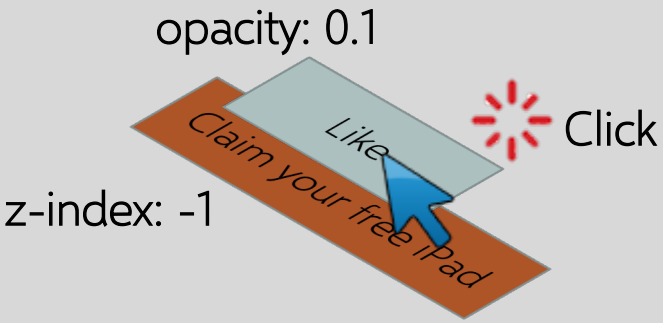
  src="http://www.google.com/..."></iframe>

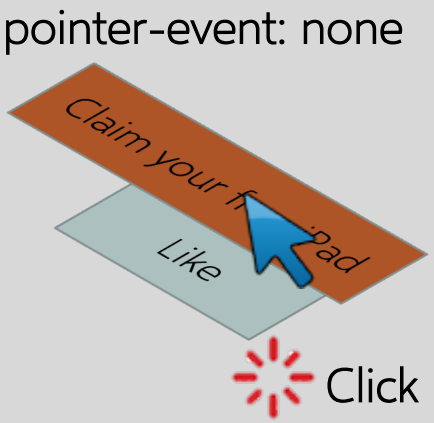
- HTML attributes: style, opacity
  - Opacity defines visibility percentage of the iframe
    - 1.0: completely visible
    - 0.0: completely invisible

Hiding the Target Element

Use CSS `opacity` property and `z-index` property to hide target element and make other element float under the target element

Use CSS `pointer-events: none` property to cover other element over the target element

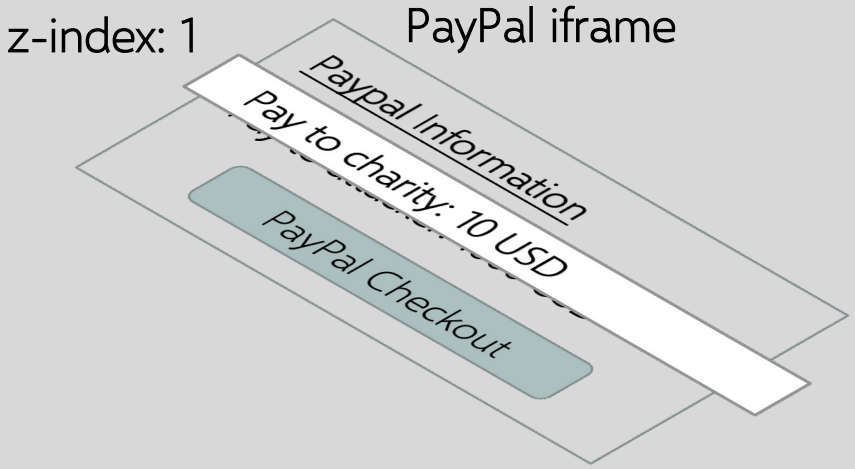




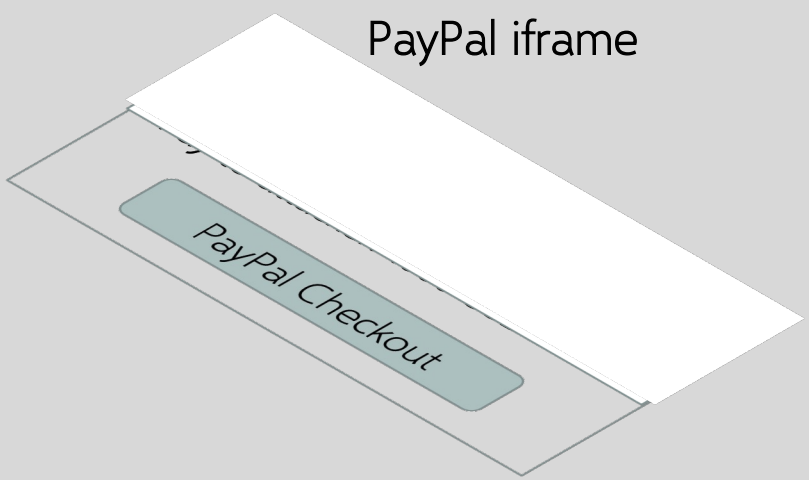
"Clickjacking: Attacks and Defenses"

Partial Overlays and Cropping

Overlay other elements onto an iframe using CSS `z-index` property or Flash Window Mode `wmode=direct` property



Wrap target element in a new iframe and choose CSS position offset properties

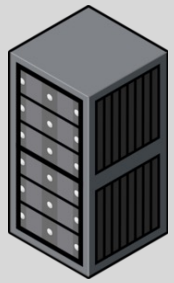


"Clickjacking: Attacks and Defenses"

How to Block Framing with CSP

web browser

example.com



HTTP response from server

HTTP/1.1 200 OK

...

Content-Security-Policy: frame-ancestors 'none';

...

<iframe src='example.com'>
will cause an error

frame-ancestors 'self' ;
means only example.com
can frame page

Frame Busting

I am a page owner

All I need to do is make sure that my web page is not loaded in an enclosing frame … Clickjacking: solved!

- Does not work for FB “Like” buttons and such, but Ok

How hard can this be?

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

If My Frame Is Not On Top, Move It To Top

Conditional Statements
if (top != self)
if (top.location != self.location)
if (top.location != location)
if (parent.frames.length > 0)
if (window != top)
if (window.top !== window.self)
if (window.self != window.top)
if (parent && parent != window)
if (parent && parent.frames && parent.frames.length>0)
if((self.parent&& !(self.parent===self))&& (self.parent.frames.length!=0))

Counter-Action Statements
top.location = self.location
top.location.href = document.location.href
top.location.href = self.location.href
top.location.replace(self.location)
top.location.href = window.location.href
top.location.replace(document.location)
top.location.href = window.location.href
top.location.href = "URL"
document.write('')
top.location = location
top.location.replace(document.location)
top.location.replace('URL')
top.location.href = document.location
top.location.replace(window.location.href)
top.location.href = location.href
self.parent.location = document.location
parent.location.href = self.document.location
top.location.href = self.location
top.location = window.location
top.location.replace(window.location.pathname)

What About My Own iFrames?

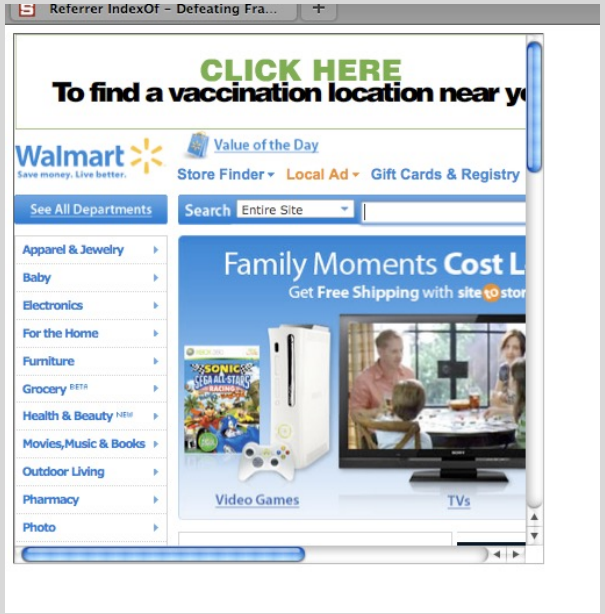
Check: is the enclosing frame one of my own?

How hard can this be?

Survey of by Rydstedt et al. of several hundred top websites …

… all frame busting code is broken!

http://www.attacker.com/walmart.com.html




```
if (top.location != location) {
  if(document.referer &&
   document.referer.indexOf("walmart.com") == -1)
   {
      top.location.replace(document.location.href);
   }
}
```

Checks if the URL contains walmart.com

http://www.attacker.com/a.html?b=https://www.nytimes.com/

The New York Times

```
if (window.self != window.top &&
    !document.referer.match(
    /https?:\/\/[^?\/]+\.nytimes\.com\//))
{
    self.location = top.location;
}
```

Checks if the URL ends with nytimes.com

http://usbank.attacker.com/

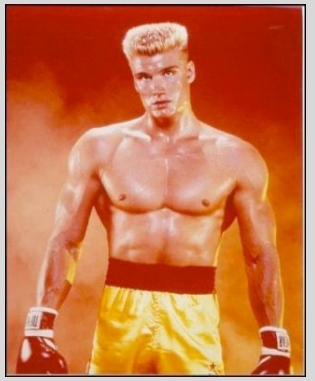
Norwegian State House Bank
http://www.husbanken.no

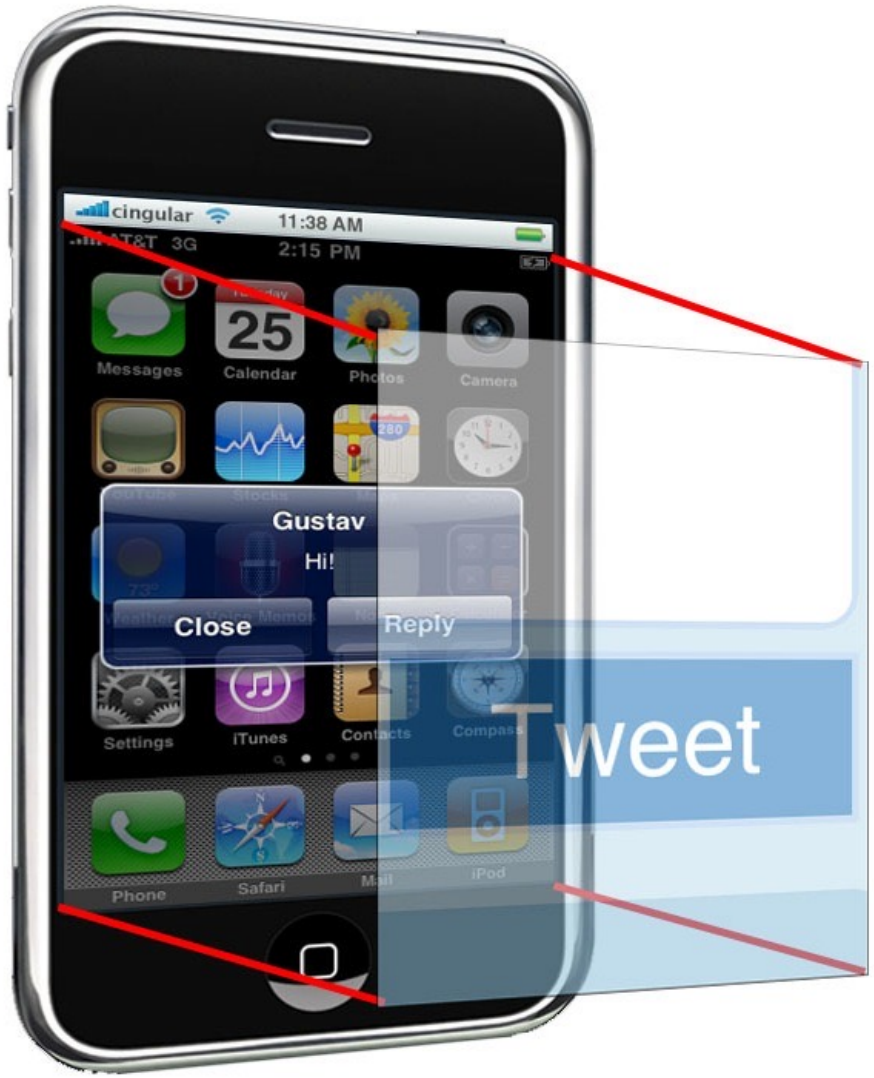

```
if (self != top) {
   var domain = getDomain(document.referer);
   var okDomains = /usbank|localhost|usbnet/;
   var matchDomain = domain.search(okDomains);


   if (matchDomain == -1) {
       // frame bust
    }
}
```

Checks if the domain name contains usbank, localhost, or usbnet

Bank of Moscow
http://www.rusbank.org





Tap-jacking

User visits a gaming website…

- Can zoom, auto scroll
- Website zooms buttons in a transparent frame so they cover entire screen
- … hides or fakes URL bar
- … imitates a known app to trick user into clicking
  - Ex: display incoming text message screen, but frame Twitter

CONFUSED DEPUTIES ARE EVERYWHERE