CSE551: Advanced Computer Security

17. Clickjacking & XS-Leaks

Seongil Wi

Department of Computer Science and Engineering

HW3

- Write a critique for the two papers:
  - Efficient and Flexible Discovery of PHP Application Vulnerabilities, ***EURO S&P 2017***
  - The Security Lottery: Measuring Client-Side Web Security Inconsistencies, ***USENIX SEC 2022***
- Due: December 2, 11:59PM
- Korean students must write in Korean

Project Final Presentation

- December 9, 11, and 16
- Check the detailed presentation schedule at: https://websec-lab.github.io/courses/2025f-cse551/
- 15 minutes for the presentation + 5 minutes (Q&A and setup for the next team)

- ★ You must submit your slides **by 11:59 PM** **on the day before your presentation**
- ★ Participation Points: Students who ask questions (including both minor and major questions) will receive significant bonus points

Final Report Submission

- Due: December 19, 11:59 PM
- The maximum length of the report is four pages (excluding references).

- Each member must submit their own ZIP file

Class Cancellation Notice

- There will be no classes on Thursday (December 4)
- Due to my business trip

- Instead, the project final presentation will be on December 16 during the final exam period

Final Exam

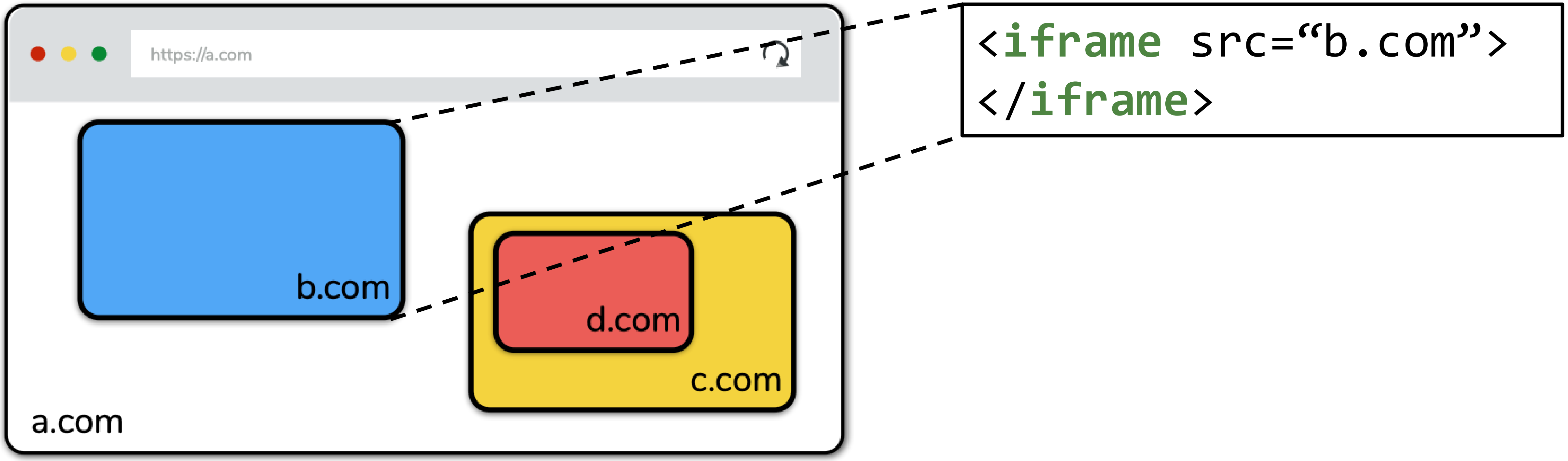
- December 18 (Thursday)
- Class Time (1h 15m)

- Descriptive type questions
  - I will evaluate your understanding of core principles and concepts

Clickjacking

Recap: Browser Execution Model

- Windows may contain frames from different sources
  - –**Frame**: rigid visible division
  - –**iFrame**: floating inline frame



```
<iframe src="b.com">
</iframe>
```

Framing other Websites

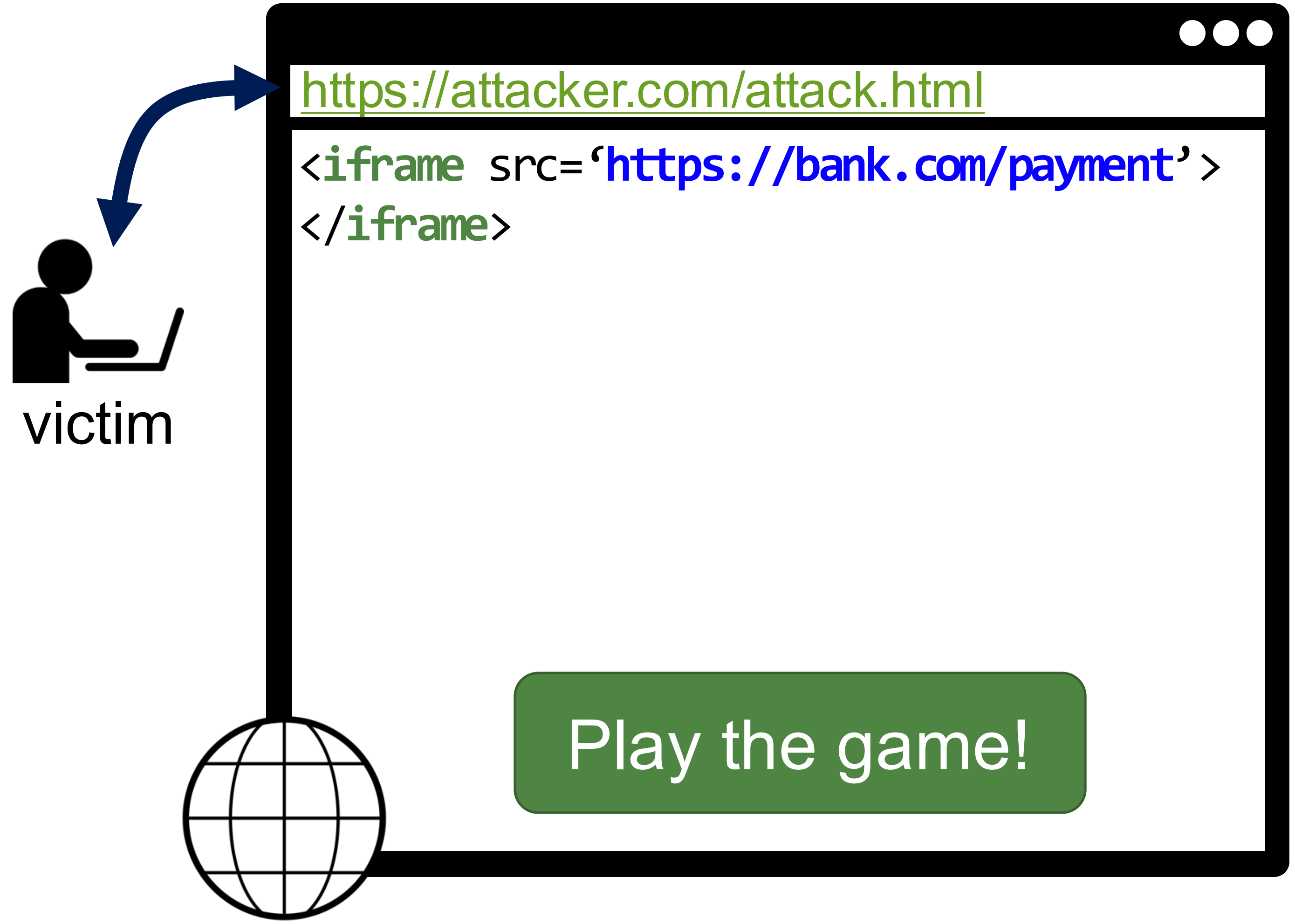
- HTML supports framing of other (cross-origin sites)
  - E.g., iframes
  - Very useful feature for advertisement, like buttons, …

- Embedding site controls most of the frame’s properties
  - How large the frame should be
  - Where the frame is displayed
  - How opaque the frame should be….

What could go wrong?

Clickjacking (UI Redressing)

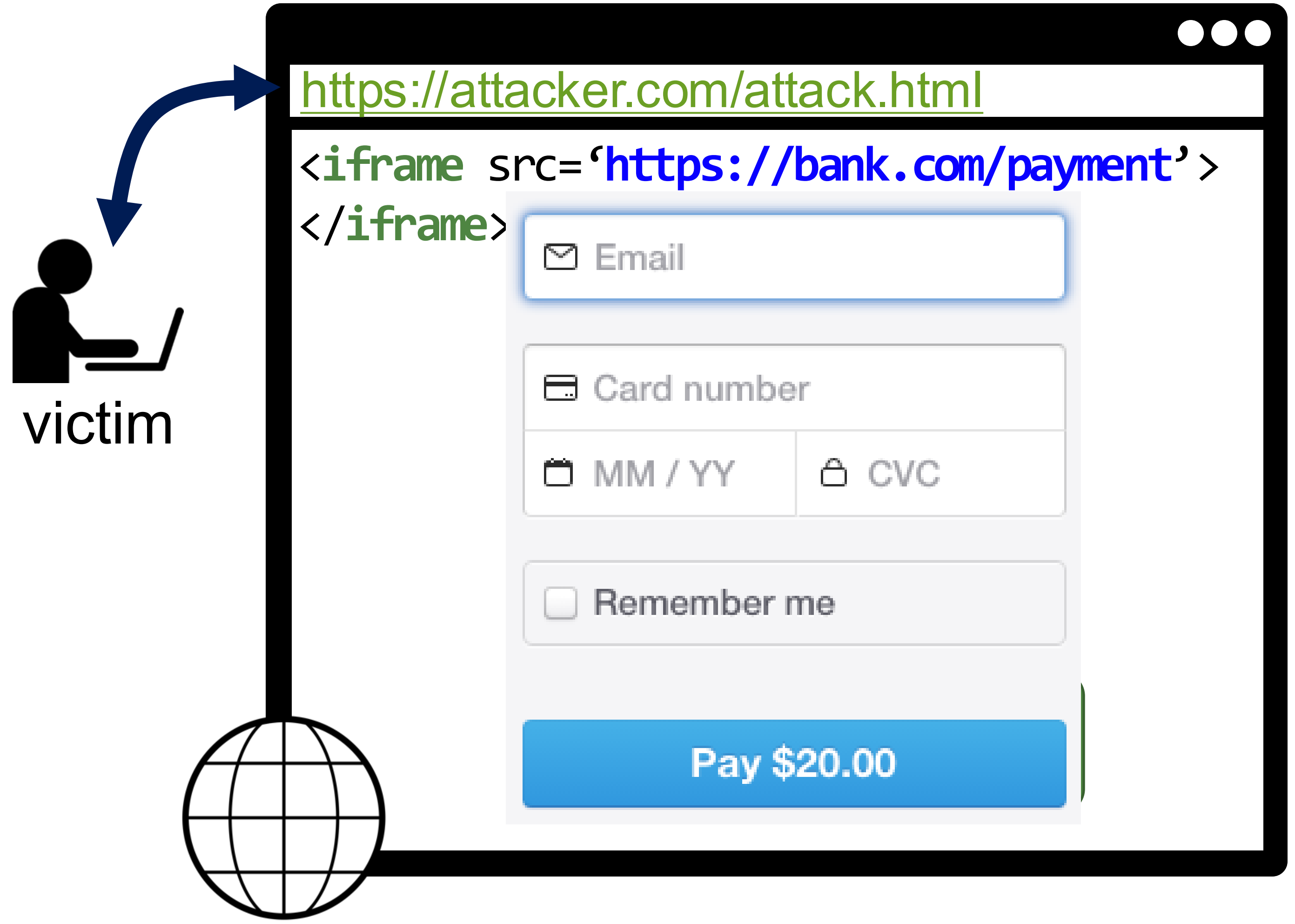
- Attacker **overlays transparent or opaque frames** to trick a user into clicking on a button or link on another page



```
<iframe src='https://bank.com/payment'>
</iframe>
```

Clickjacking (UI Redressing)

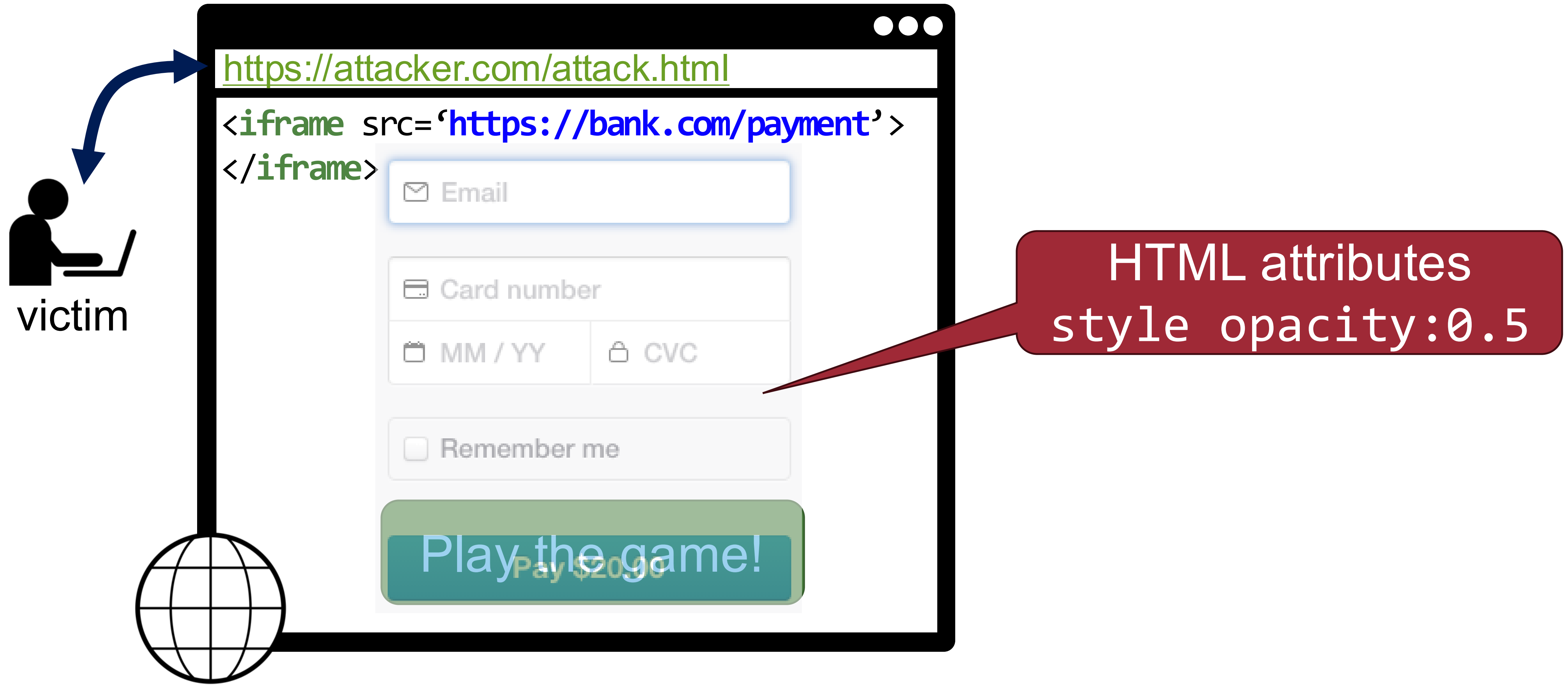
- Attacker **overlays transparent or opaque frames** to trick a user into clicking on a button or link on another page



```
<iframe src='https://bank.com/payment'>
</iframe>
```

Clickjacking (UI Redressing)

- Attacker **overlays transparent or opaque frames** to trick a user into clicking on a button or link on another page

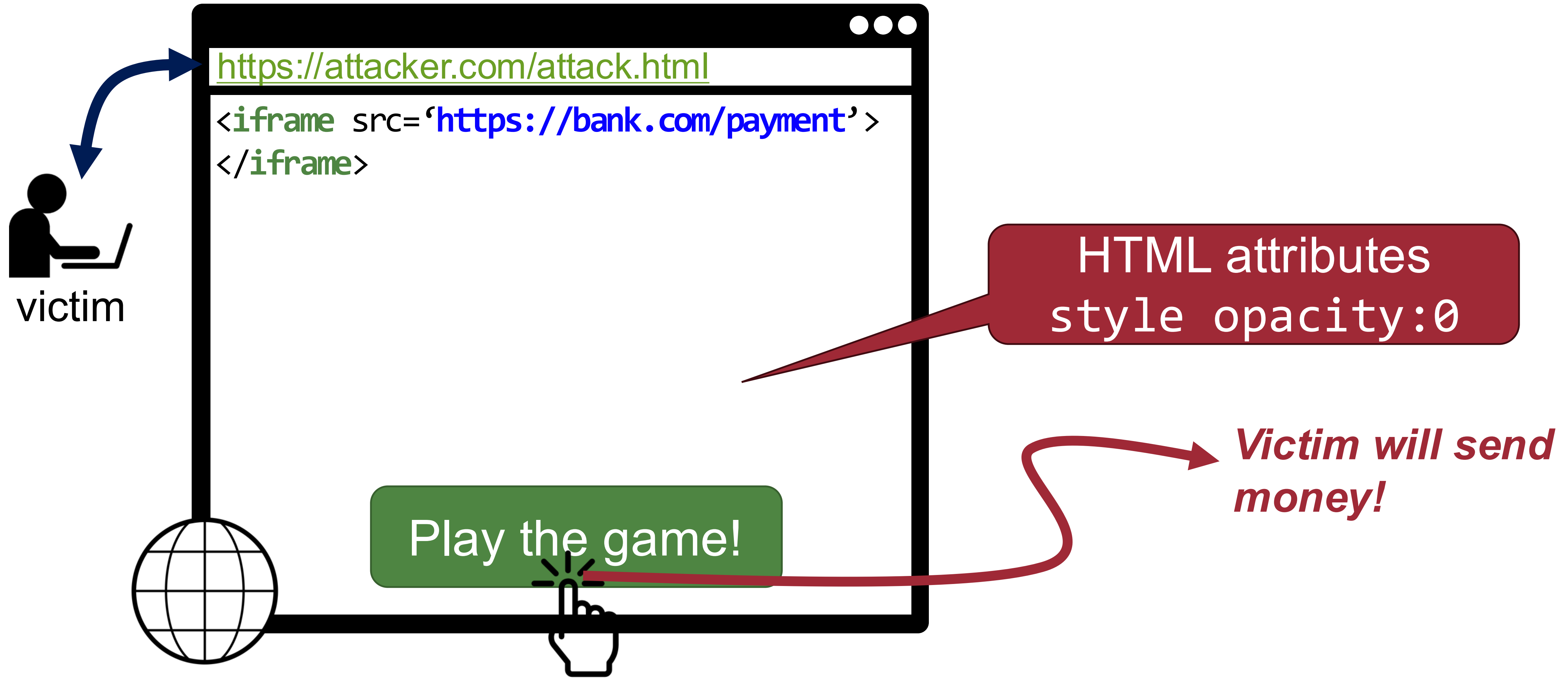


```
<iframe src='https://bank.com/payment'>
</iframe>
```

```
style opacity:0.5
```

Clickjacking (UI Redressing)

- Attacker **overlays transparent or opaque frames** to trick a user into clicking on a button or link on another page

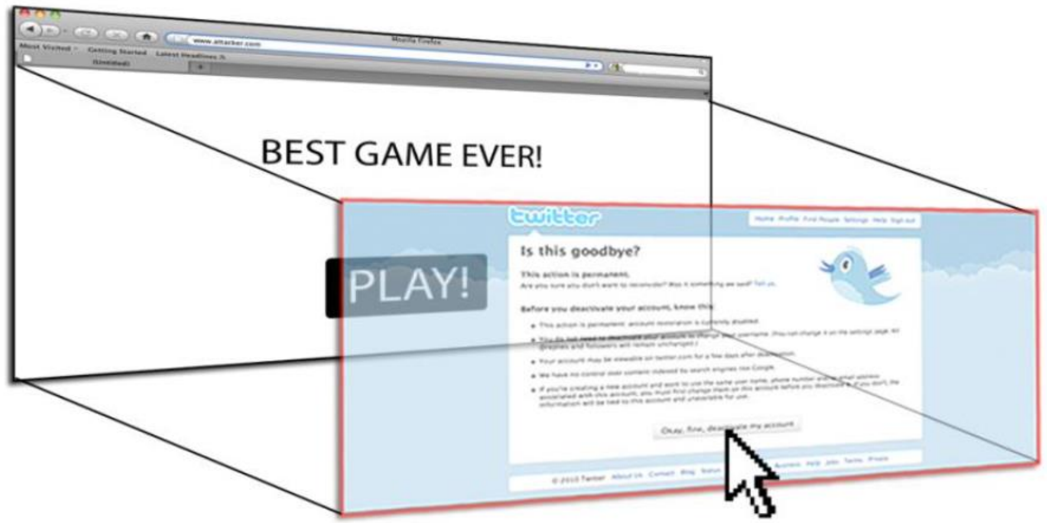


```
<iframe src='https://bank.com/payment'>
</iframe>
```

```
style opacity:0
```

Clickjacking Demo

- Demo (opaque): https://websec-lab.github.io/courses/2025f-cse551/demo/clickjacking_demo1.html
- Demo (half-opaque): https://websec-lab.github.io/courses/2025f-cse551/demo/clickjacking_demo2.html

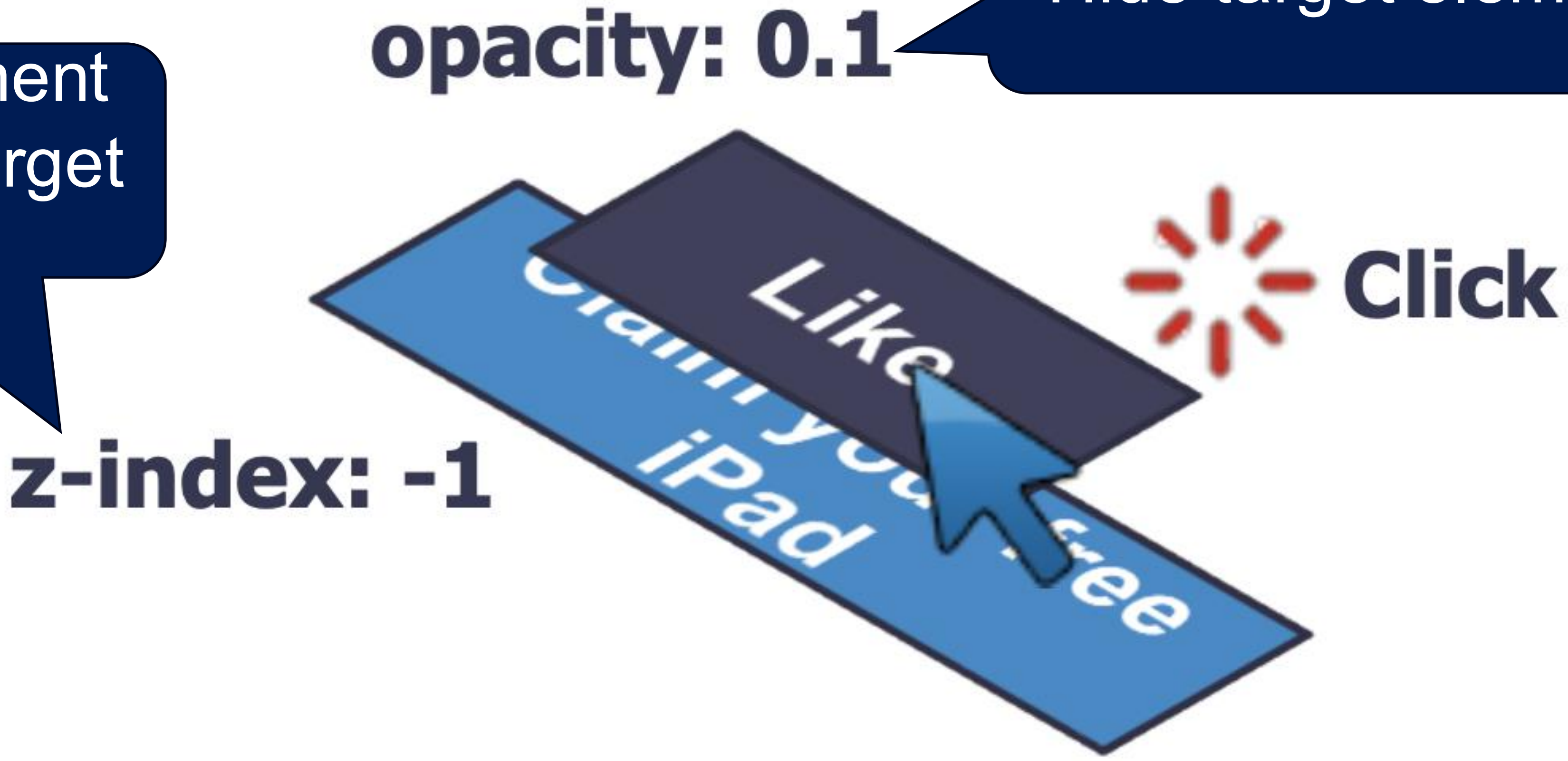


Clickjacking – Hiding the Target Element

- Use CSS `opacity` property and `z-index` property

Hide target element

Make other element float <u>under</u> the target element



Cursor Spoofing

- Use CSS `cursor` property and JavaScript to simulate a fake cursor icon on the screen

Real cursor icon

Fake cursor icon

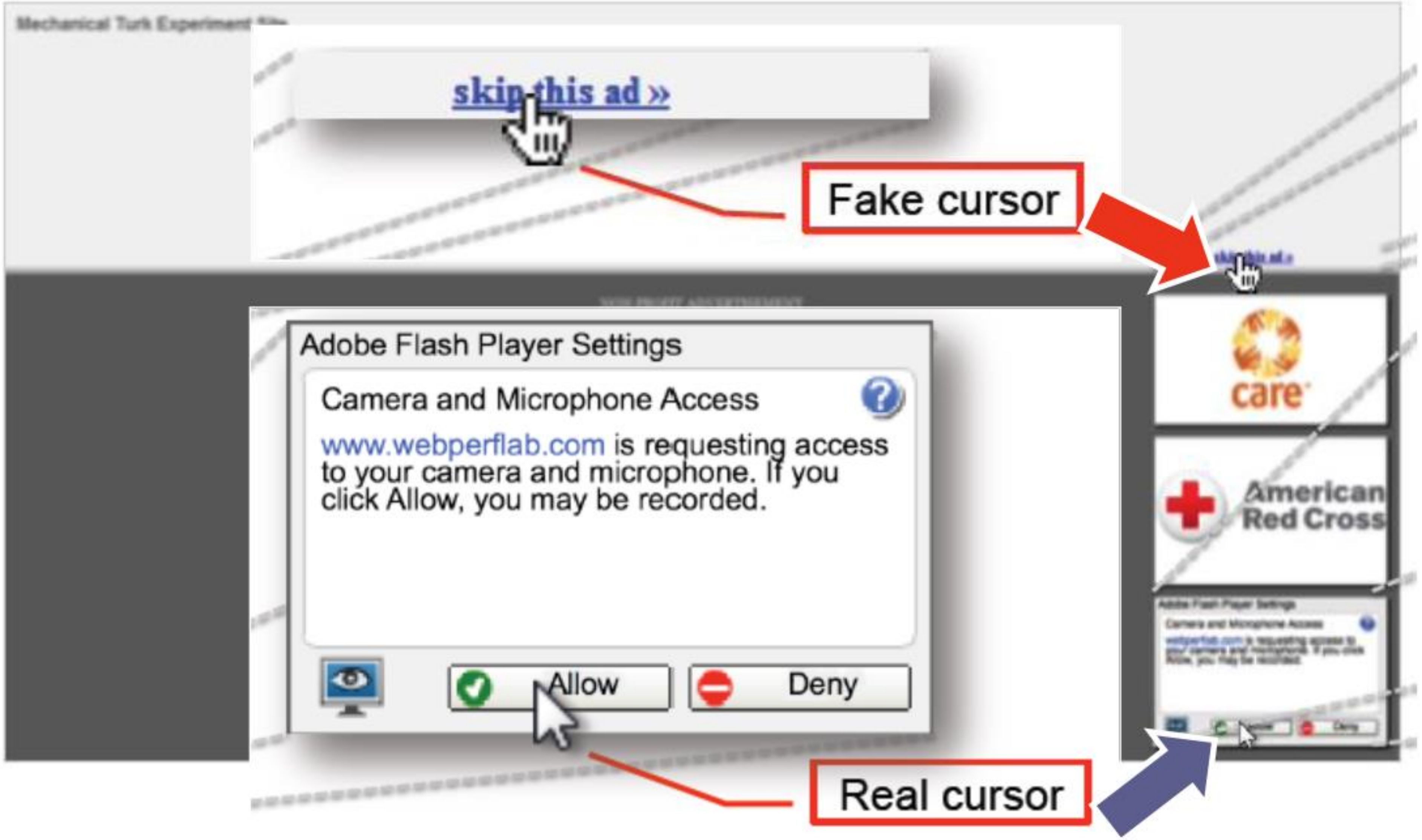
Cursor Spoofing

- Use CSS `cursor` property and JavaScript to simulate a fake cursor icon on the screen

Hide real curson icon by using `cursor: none`

Fake cursor icon

Cursor Spoofing


Mechanical Turk Experiment Site

skip this ad »

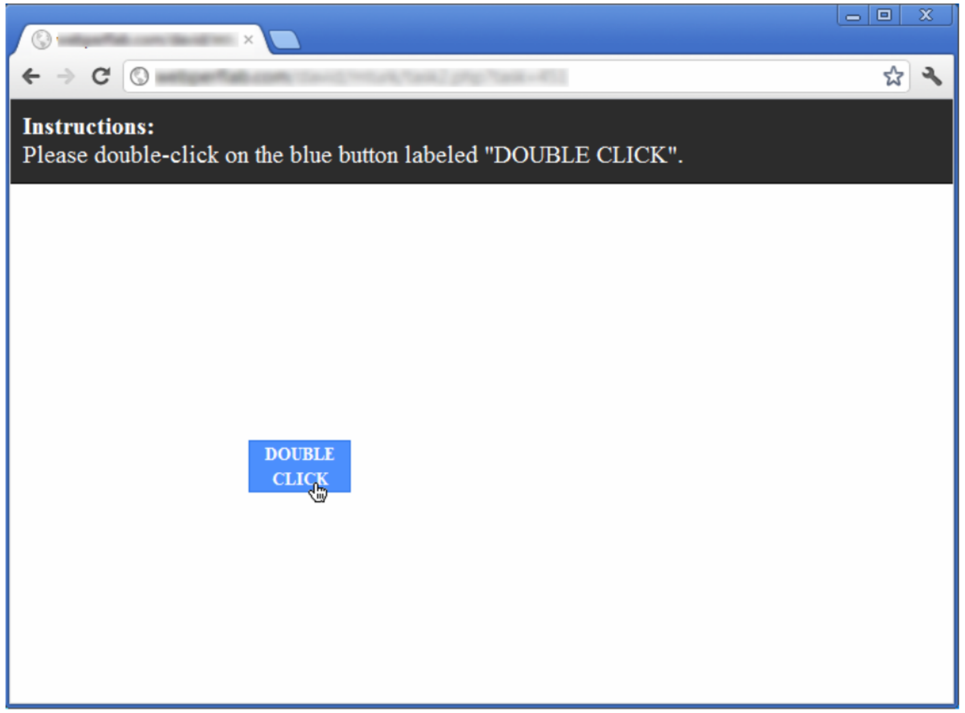
Adobe Flash Player Settings

Camera and Microphone Access

www.webperflab.com is requesting access to your camera and microphone. If you click Allow, you may be recorded.

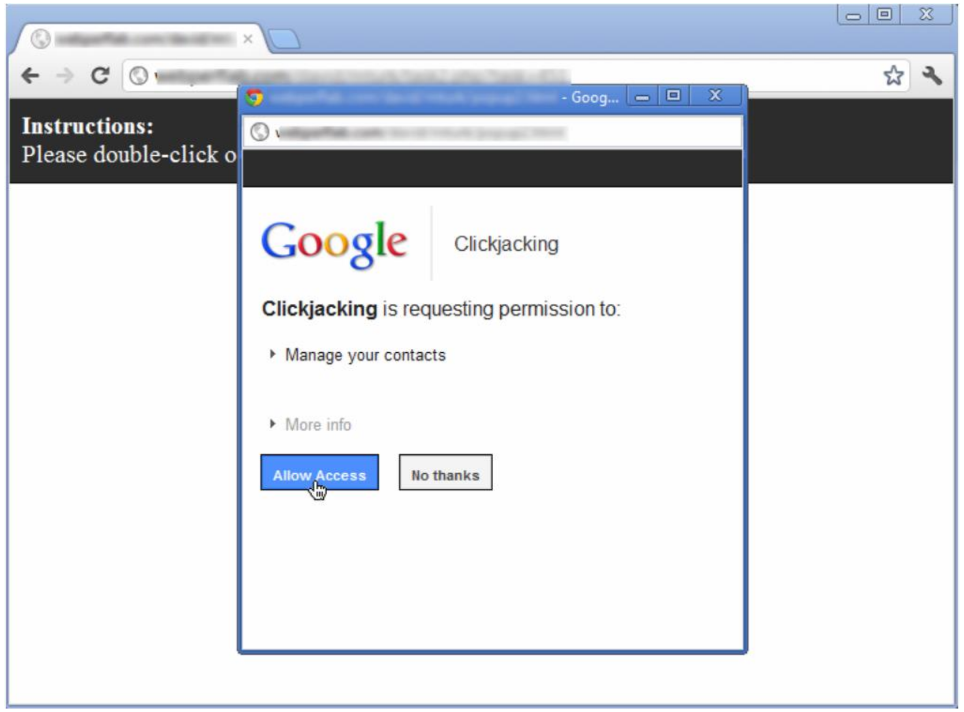
Double-Click Attack

- Bait the user to perform a double-click
  - After the first click, the target window pops up
  - After the second click, permission is allowed



Instructions:
Please double-click on the blue button labeled "DOUBLE CLICK".

First click



Instructions:
Please double-click o

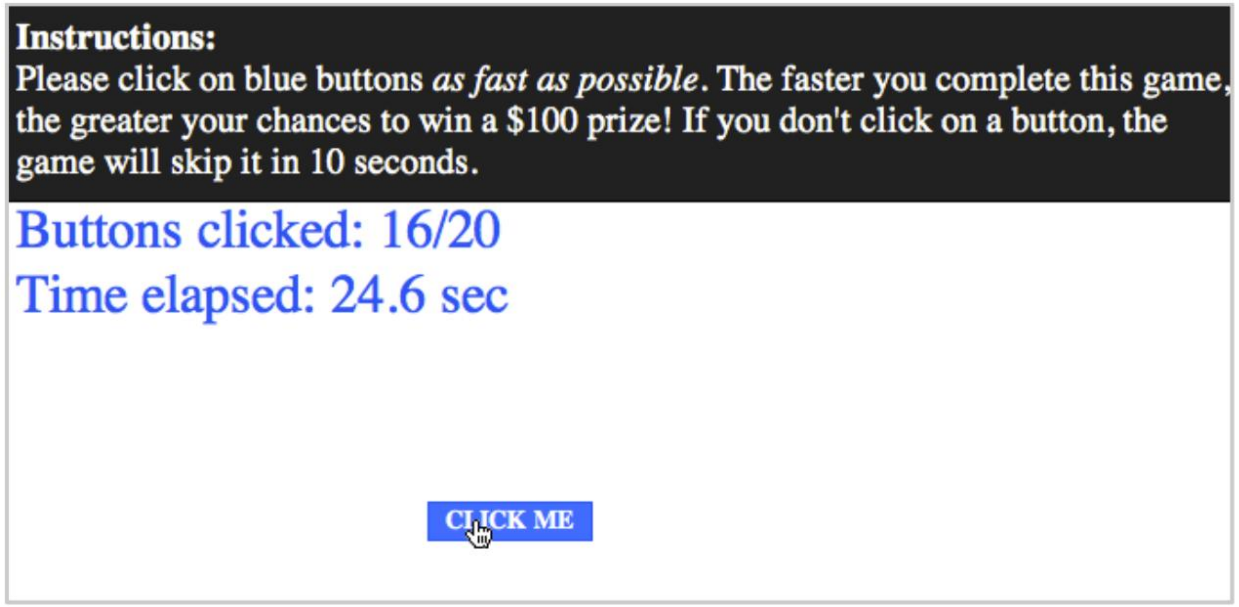
Clickjacking is requesting permission to:

Second click

Permission is allowed

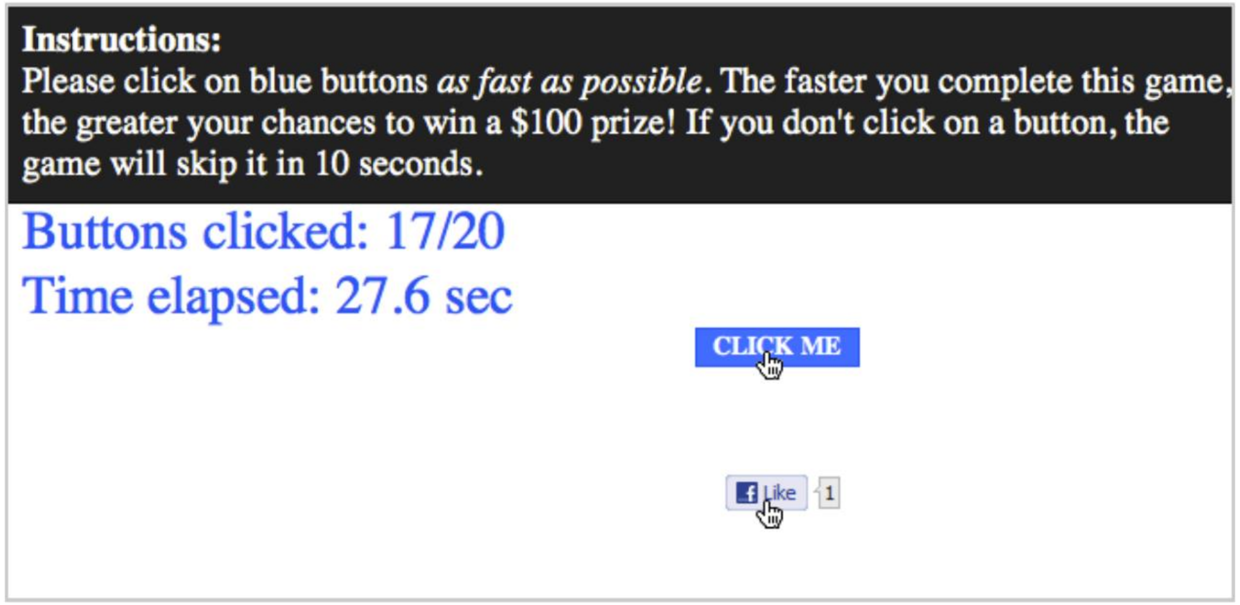
Whack-a-mole Attack

- Ask the user to click as fast as possible, suddenly switch Facebook like button
  - Combine the approaches from the cursor spoofing and double-click attack



Instructions:
Please click on blue buttons *as fast as possible*. The faster you complete this game, the greater your chances to win a $100 prize! If you don't click on a button, the game will skip it in 10 seconds.

Buttons clicked: 16/20
Time elapsed: 24.6 sec

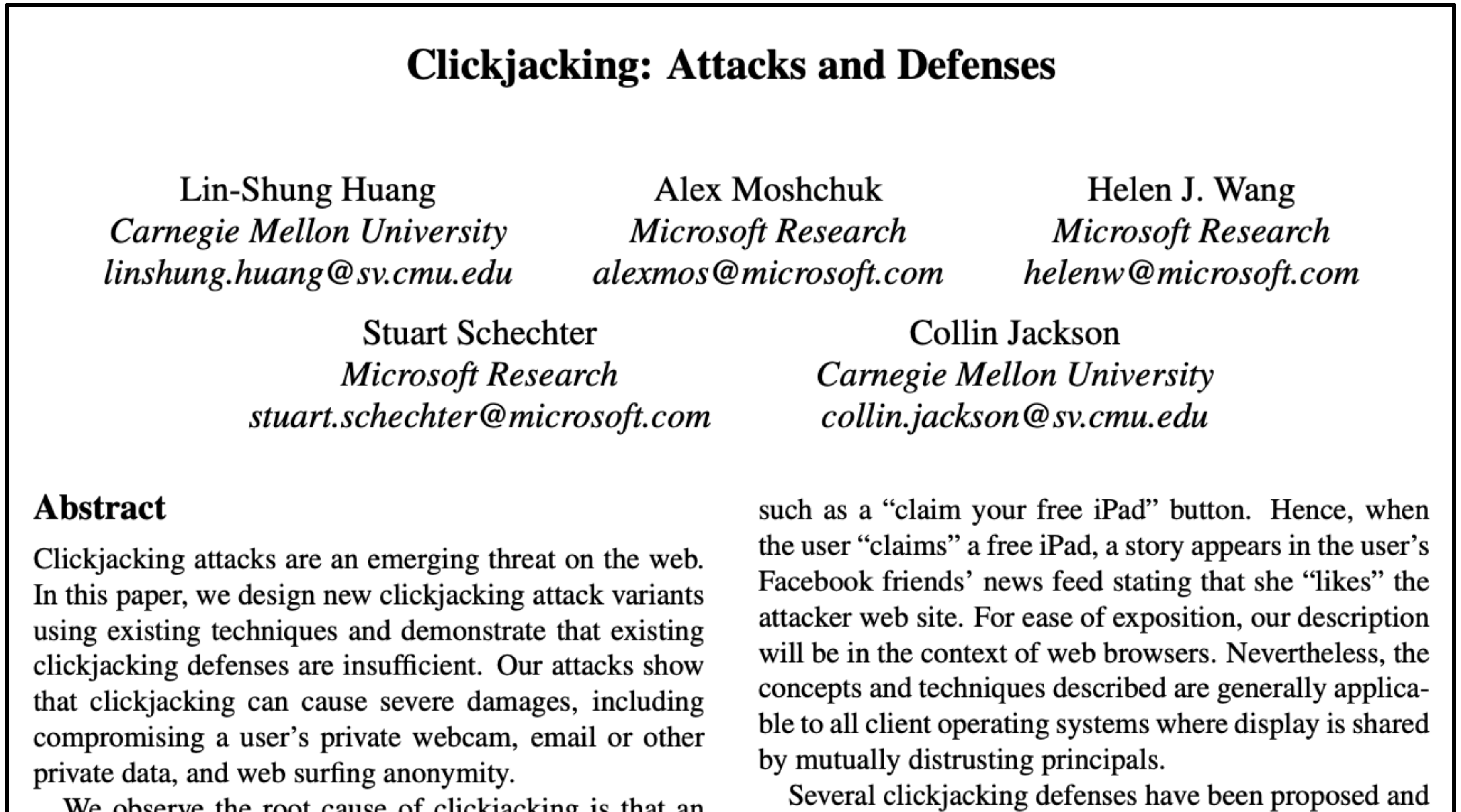


Instructions:
Please click on blue buttons *as fast as possible*. The faster you complete this game, the greater your chances to win a $100 prize! If you don't click on a button, the game will skip it in 10 seconds.

Buttons clicked: 17/20
Time elapsed: 27.6 sec

Clickjacking, *USENIX SEC'12*

- Evaluate the effectiveness of attack techniques



Clickjacking: Attacks and Defenses

Lin-Shung Huang
Carnegie Mellon University
linshung.huang@sv.cmu.edu

Alex Moshchuk
Microsoft Research
alexmos@microsoft.com

Helen J. Wang
Microsoft Research
helenw@microsoft.com

Stuart Schechter
Microsoft Research
stuart.schechter@microsoft.com

Collin Jackson
Carnegie Mellon University
collin.jackson@sv.cmu.edu

Abstract

Clickjacking attacks are an emerging threat on the web. In this paper, we design new clickjacking attack variants using existing techniques and demonstrate that existing clickjacking defenses are insufficient. Our attacks show that clickjacking can cause severe damages, including compromising a user's private webcam, email or other private data, and web surfing anonymity.

We observe the root cause of clickjacking is that an

such as a "claim your free iPad" button. Hence, when the user "claims" a free iPad, a story appears in the user's Facebook friends' news feed stating that she "likes" the attacker web site. For ease of exposition, our description will be in the context of web browsers. Nevertheless, the concepts and techniques described are generally applicable to all client operating systems where display is shared by mutually distrusting principals.

Several clickjacking defenses have been proposed and

Clickjacking, *USENIX SEC'12*

- Evaluate the effectiveness of attack techniques
  - Cursor-spoofing attack: 31/72 (43%)
  - Double-click attack: 43/90 (47%)
  - Whack-a-mole attack: 80/84 (98%)

How to Mitigate Clickjacking?

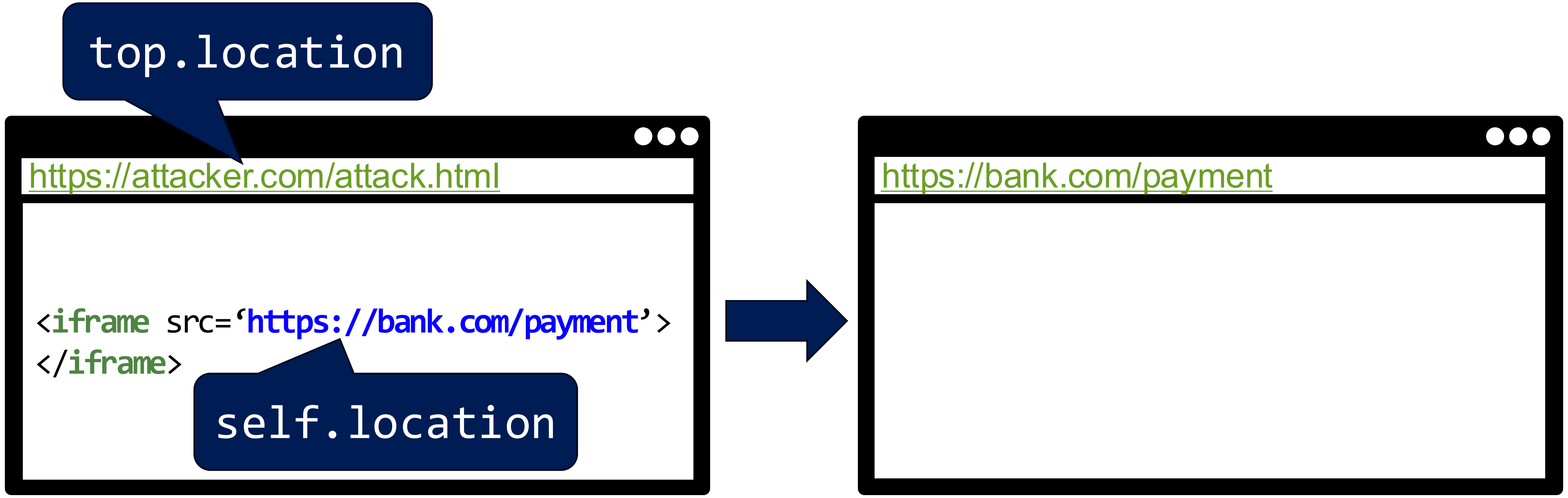
1. Frame busting

Clickjacking Defense: Frame Busting

- Make sure that my website is not loaded in an enclosing frame

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

JS

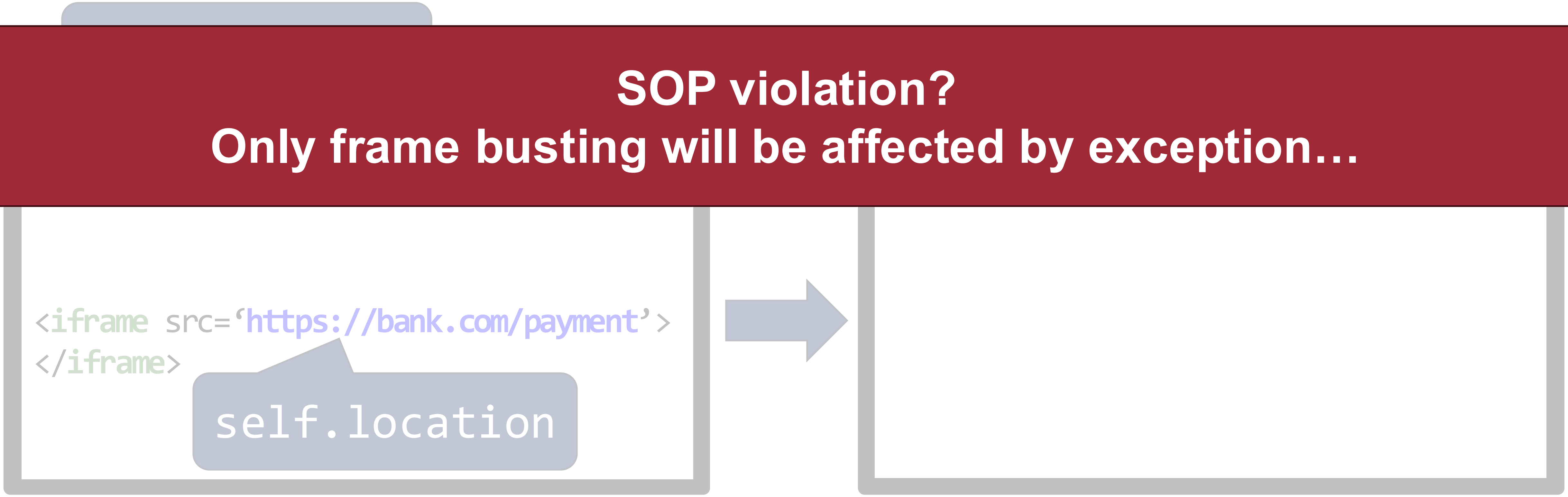

top.location
https://attacker.com/attack.html
<iframe src='https://bank.com/payment'>
</iframe>
self.location
https://bank.com/payment

Clickjacking Defense: Frame Busting

- Make sure that my website is not loaded in an enclosing frame

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

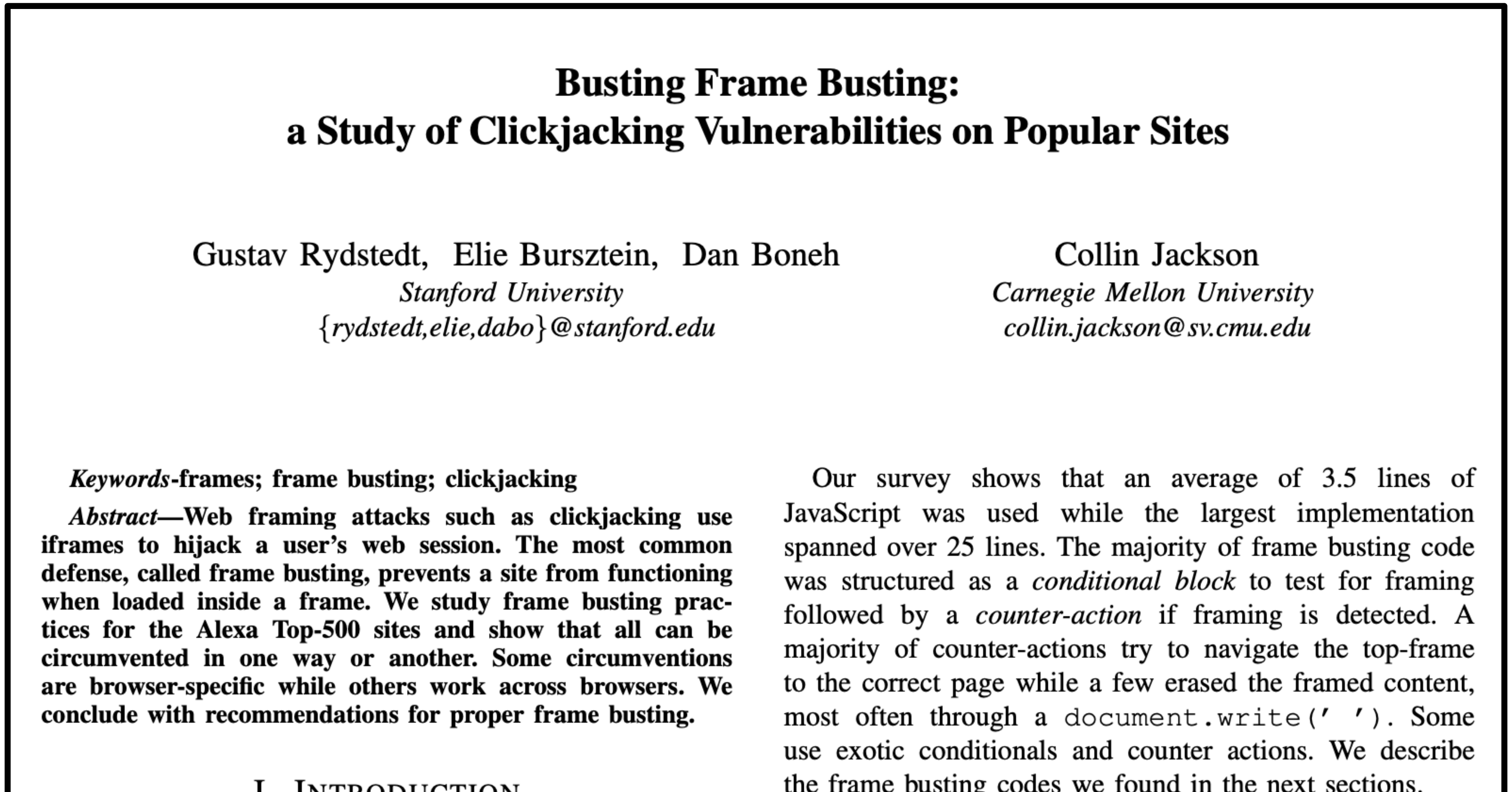
SOP violation?
Only frame busting will be affected by exception…



```
<iframe src='https://bank.com/payment'>
</iframe>
```

Busting Frame Busting, W2SP'10

- Analyze frame busting code from Alexa Top 500 and all US banks



Busting Frame Busting:
a Study of Clickjacking Vulnerabilities on Popular Sites

Gustav Rydstedt, Elie Bursztein, Dan Boneh
Stanford University
{rydstedt,elie,dabo}@stanford.edu

Collin Jackson
Carnegie Mellon University
collin.jackson@sv.cmu.edu

***Keywords*-frames; frame busting; clickjacking**

***Abstract*—Web framing attacks such as clickjacking use iframes to hijack a user's web session. The most common defense, called frame busting, prevents a site from functioning when loaded inside a frame. We study frame busting practices for the Alexa Top-500 sites and show that all can be circumvented in one way or another. Some circumventions are browser-specific while others work across browsers. We conclude with recommendations for proper frame busting.**

I. Introduction

Our survey shows that an average of 3.5 lines of JavaScript was used while the largest implementation spanned over 25 lines. The majority of frame busting code was structured as a *conditional block* to test for framing followed by a *counter-action* if framing is detected. A majority of counter-actions try to navigate the top-frame to the correct page while a few erased the framed content, most often through a `document.write(' ')`. Some use exotic conditionals and counter actions. We describe the frame busting codes we found in the next sections.

Busting Frame Busting, W2SP'10

- Analyze frame busting code from Alexa Top 500 and all US banks

Sites	Framebusting
Top 10	60%
Top 100	37%
Top 500	14%

Busting Frame Busting, W2SP'10

- Analyze frame busting code from Alexa Top 500 and all US banks

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))

Busting Frame Busting, W2SP'10

- Analyze frame busting code from Alexa Top 500 and all US banks

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

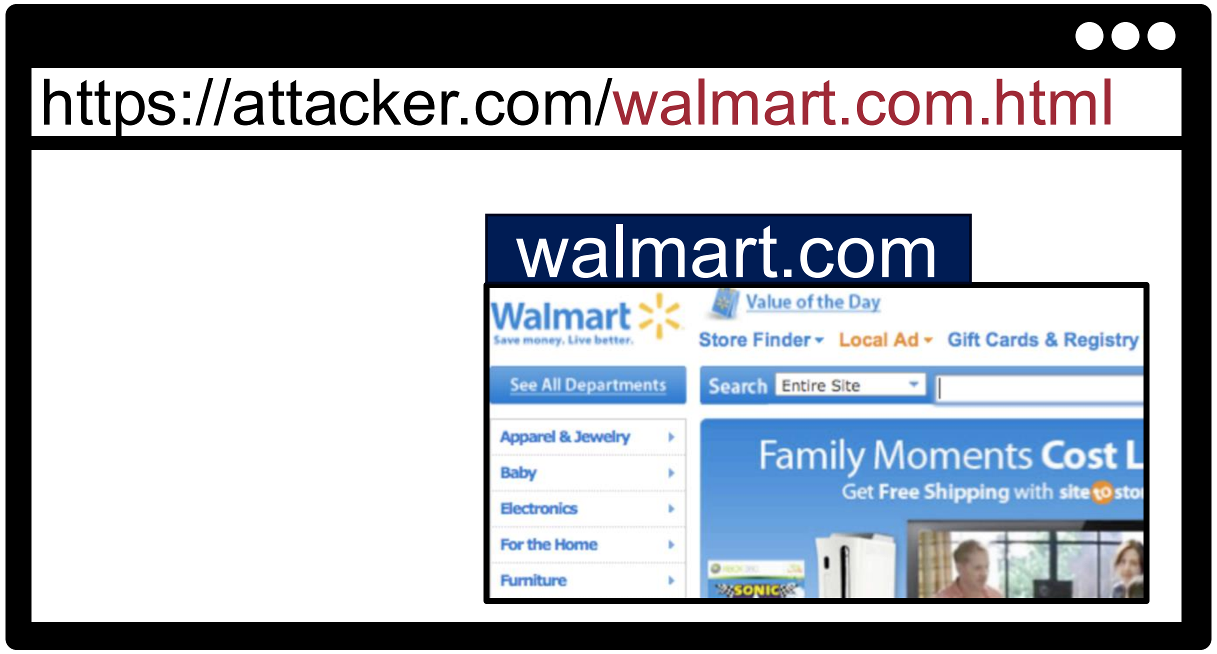
Busting Frame Busting, W2SP'10

- Analyze frame busting code from Alexa Top 500 and all US banks

- Show that all frame busting code is broken

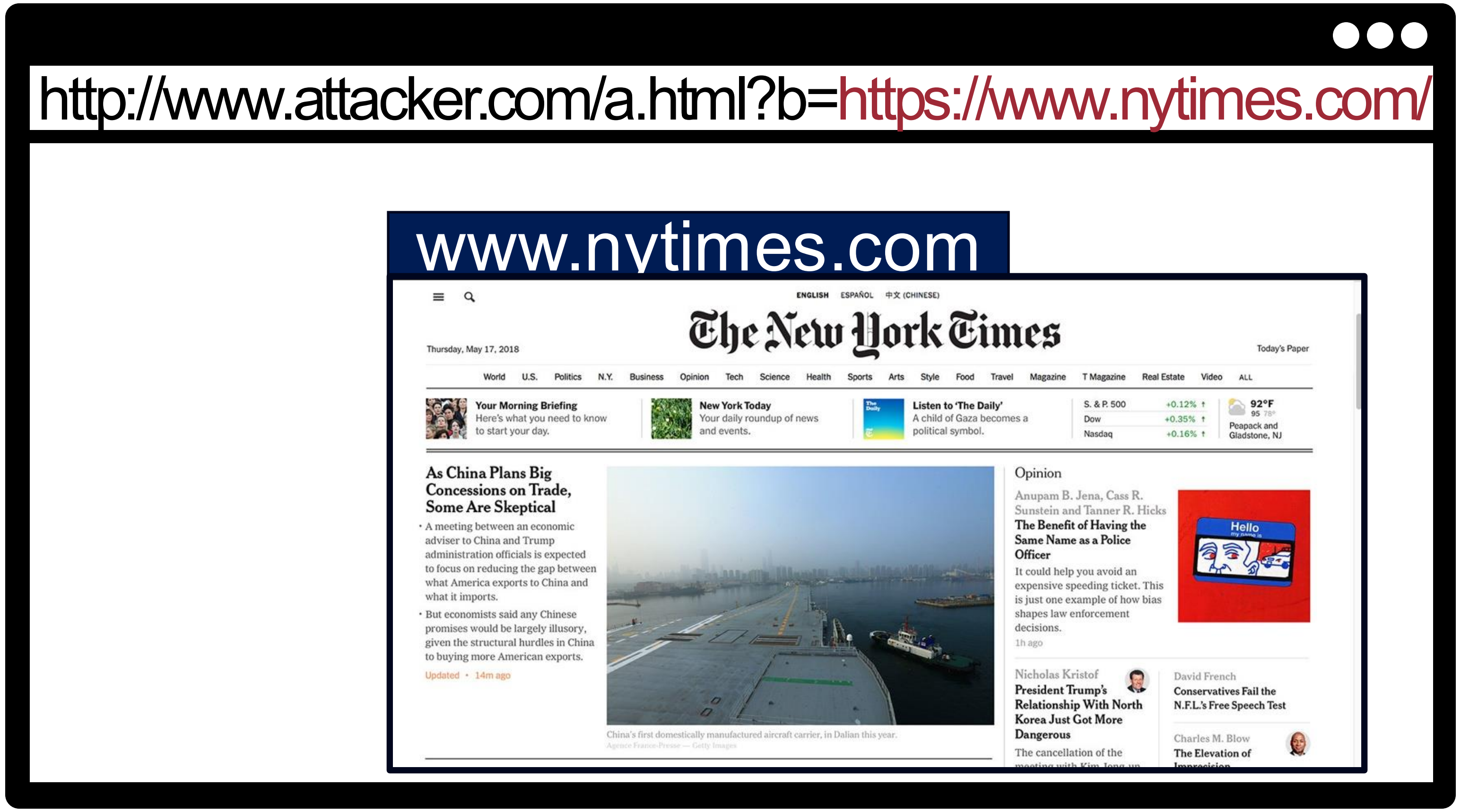
Broken Frame Buster: Walmart

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



Broken Frame Buster: The New York Times

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



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

www.nytimes.com
ENGLISH ESPAÑOL 中文 (CHINESE)
The New York Times
Thursday, May 17, 2018
Today's Paper
World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Style Food Travel Magazine T Magazine Real Estate Video ALL

Your Morning Briefing
Here's what you need to know to start your day.

New York Today
Your daily roundup of news and events.

Listen to 'The Daily'
A child of Gaza becomes a political symbol.

S. & P. 500 +0.12%
Dow +0.35%
Nasdaq +0.16%

92°F 95 78°
Peapack and Gladstone, NJ

As China Plans Big Concessions on Trade, Some Are Skeptical

- A meeting between an economic adviser to China and Trump administration officials is expected to focus on reducing the gap between what America exports to China and what it imports.
- But economists said any Chinese promises would be largely illusory, given the structural hurdles in China to buying more American exports.

Updated • 14m ago

China's first domestically manufactured aircraft carrier, in Dalian this year.

Opinion

Anupam B. Jena, Cass R. Sunstein and Tanner R. Hicks
The Benefit of Having the Same Name as a Police Officer
It could help you avoid an expensive speeding ticket. This is just one example of how bias shapes law enforcement decisions.
1h ago

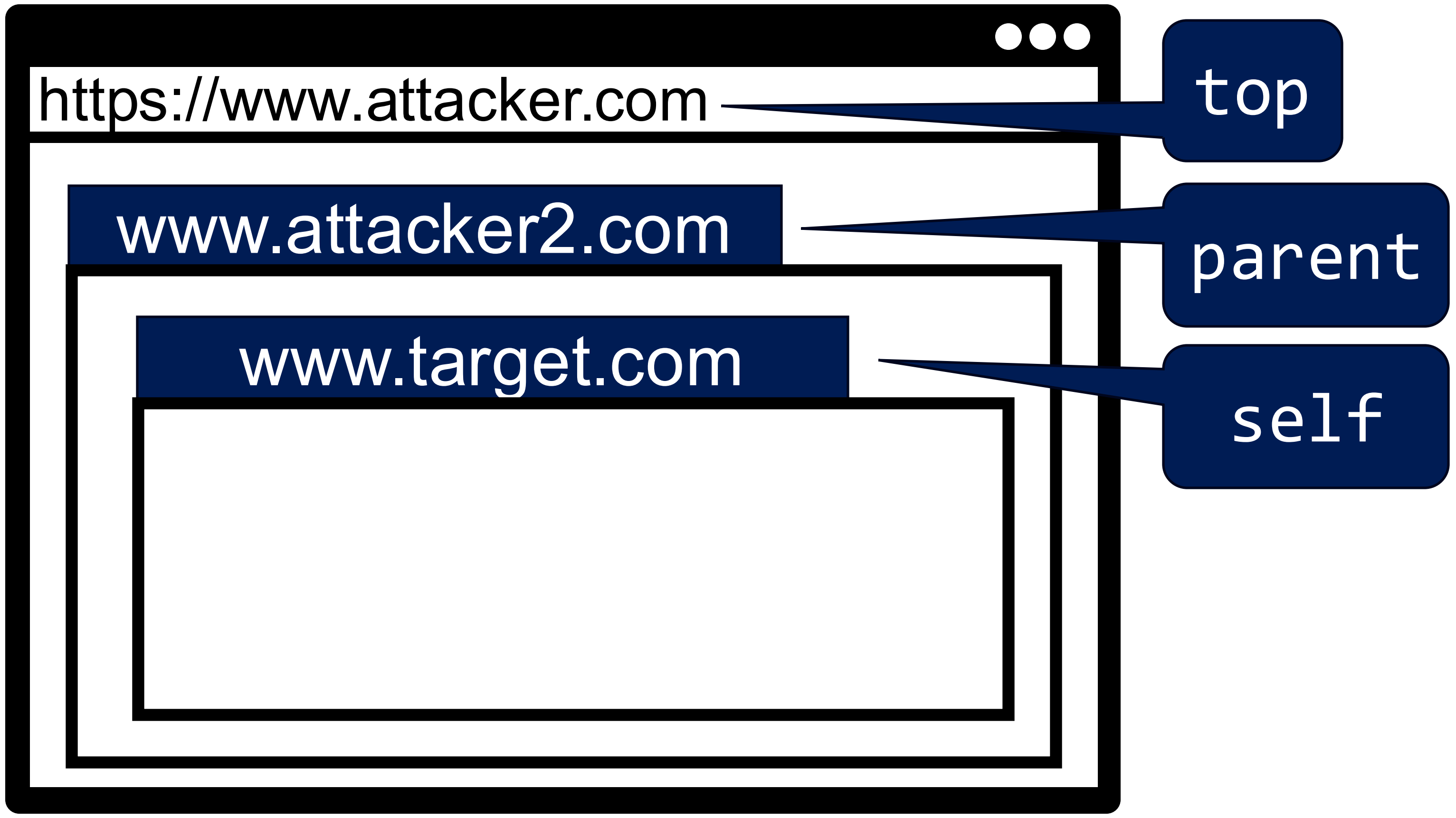
Nicholas Kristof
President Trump's Relationship With North Korea Just Got More Dangerous
The cancellation of the

David French
Conservatives Fail the N.F.L.'s Free Speech Test

Charles M. Blow
The Elevation of

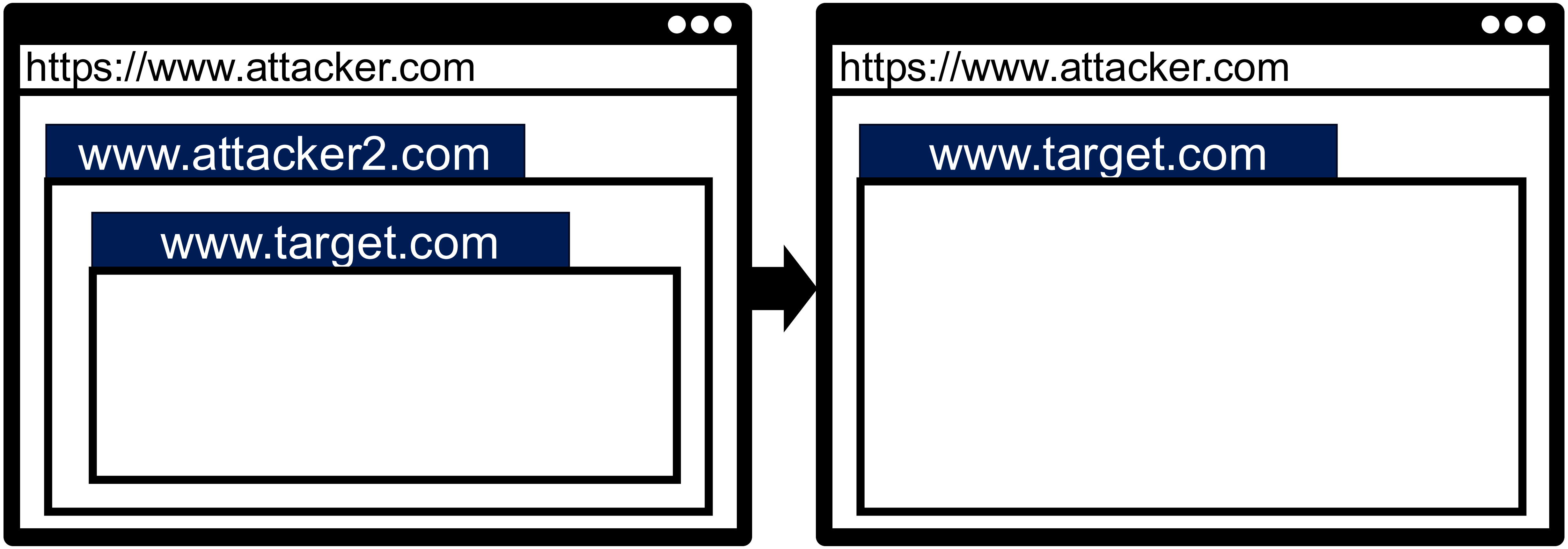
Broken Frame Buster: Double Framing

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



Broken Frame Buster: Double Framing

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



How to Mitigate Clickjacking?

1. Frame busting

2. `X-Frame-Options`

Clickjacking Defense: X-Frame-Options

- Non-standardized (hence the X-), yet widely adopted header
  - Introduced in 2009
  - Has an RFC since 2013 (RFC7034)

- Depending on the browser, two or three options exist
  - **`DENY`**: deny any framing whatsoever
  - **`SAMEORIGIN`**: only allow framing the same origin
    - depending on browser, same origin as top page or as parent page
  - **`ALLOW-FROM`**: <u>single</u> allowed domain (obsolete feature)

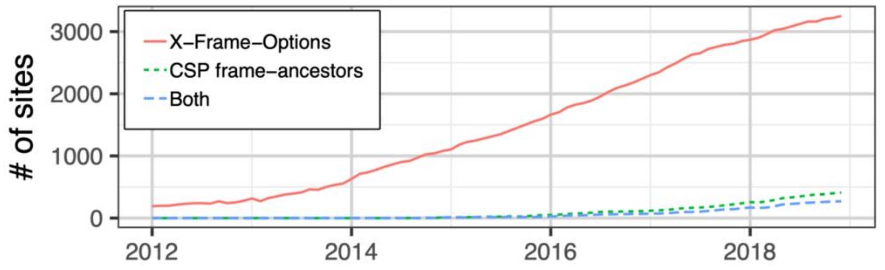
- ~25% adoption on the web in 2017

How to Mitigate Clickjacking?

1. Frame busting
2. `X-Frame-Options`
3. CSP `frame-ancestors`

Clickjacking Defense: CSP's frame-ancestors

- CSP introduced `frame-ancestors` in version 2 (Standard!)
  - Meant to replace non-standardized `X-Frame-Options`
  - Deprecates `X-Frame-Options`
- Determine whether my website may be embedded in another site
  - `'none'`: denies from any host
  - `'self'`: allows only from same origin
  - `http://example.org`: allows specific origin
- As of Sep. 2020, approximately 8.5% of top 10k sites with frame-ancestors
  - Comparison: 37% make use of `X-Frame-Options`



Complex Security Policy?, ***NDSS '20***

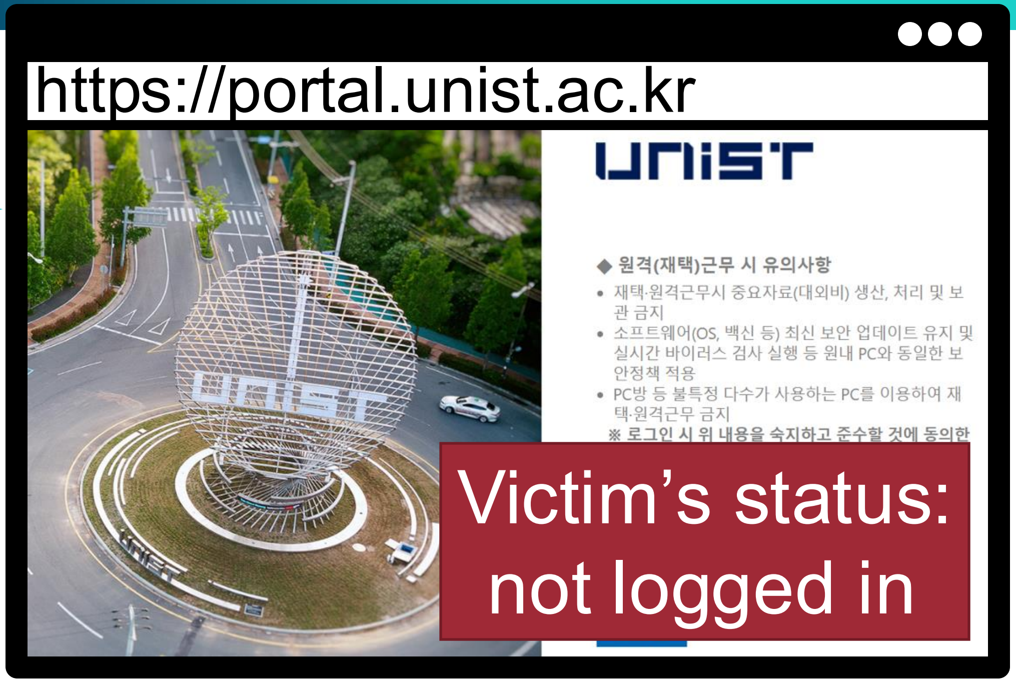
Cross-site Leaks (XS-Leaks)

Cross-Site Leaks (XS-Leaks)

- Steal the state (privacy information) of a victim from cross website
  - Login status (determine if the victim is logged in or not)
  - Visit status (determine if the victim has been visited or not)
  - …

XS-Leaks Example

Attacker's goal:
Determine whether the victim is logged in or not on portal.unist.ac.kr



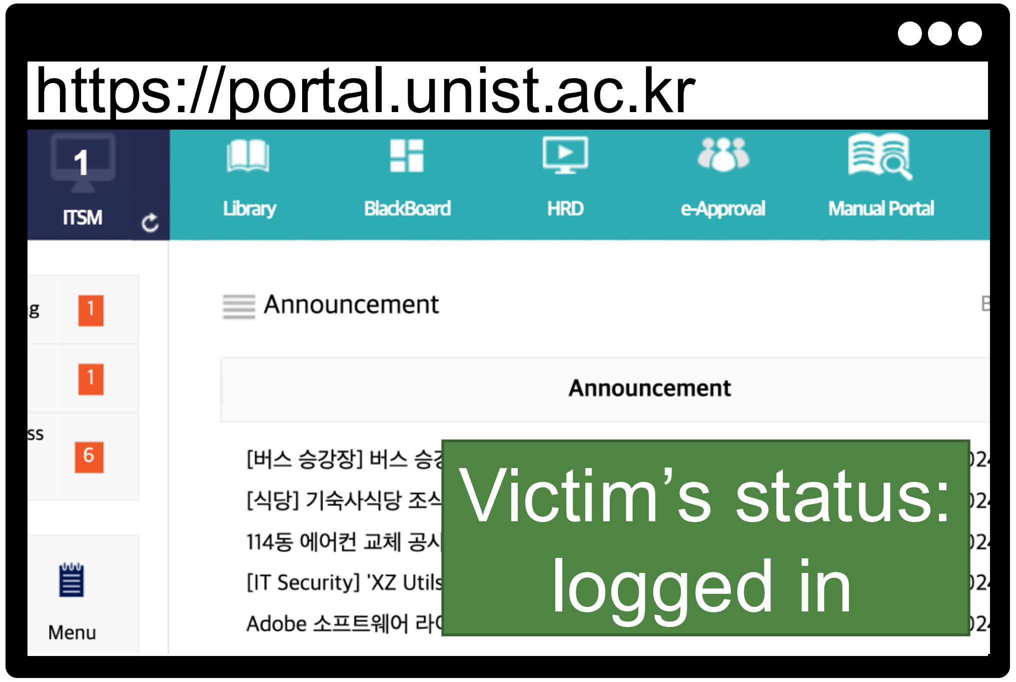
https://portal.unist.ac.kr

◆ 원격(재택)근무 시 유의사항

- 재택·원격근무시 중요자료(대외비) 생산, 처리 및 보관 금지
- 소프트웨어(OS, 백신 등) 최신 보안 업데이트 유지 및 실시간 바이러스 검사 실행 등 원내 PC와 동일한 보안정책 적용
- PC방 등 불특정 다수가 사용하는 PC를 이용하여 재택·원격근무 금지

※ 로그인 시 위 내용을 숙지하고 준수할 것에 동의한

Victim's status: not logged in



https://portal.unist.ac.kr

Announcement

Announcement

[버스 승강장] 버스 승

[식당] 기숙사식당 조

114동 에어컨 교체 공

[IT Security] 'XZ Utils

Adobe 소프트웨어 라

Victim's status: logged in

XS-Leaks Example

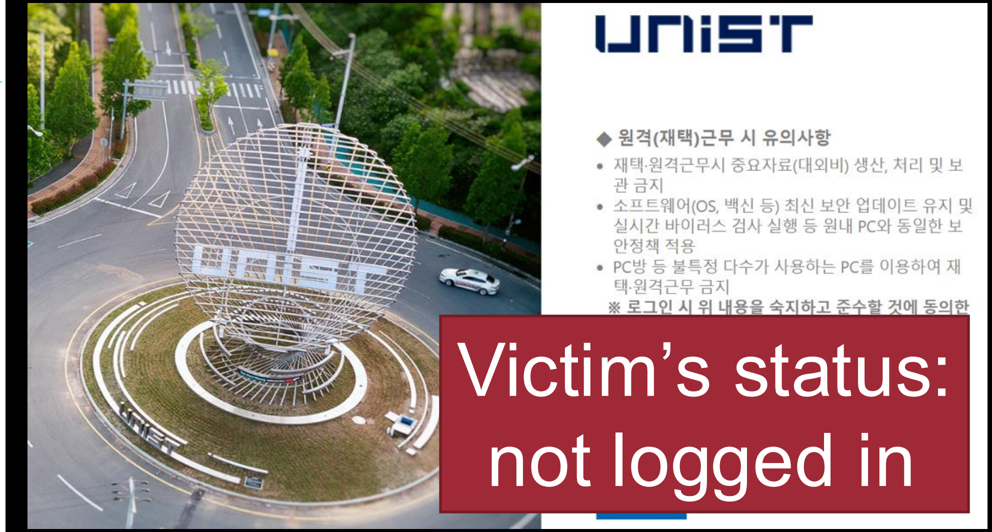
https://www.attacker.com

```
<iframe
  src=https://portal.unist.ac.kr/>
</iframe>
```

```
<script>
  if (“Announcement” in frames[0].content)
    // logged in
  else
    // not logged in
</script>
```

Is it possible?

https://portal.unist.ac.kr

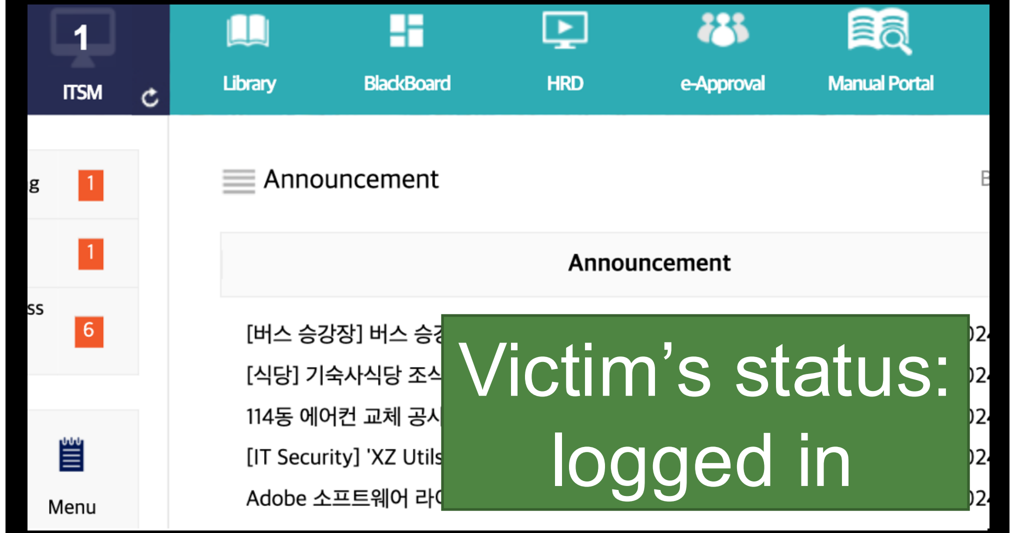


◆ 원격(재택)근무 시 유의사항
- 재택·원격근무시 중요자료(대외비) 생산, 처리 및 보관 금지
- 소프트웨어(OS, 백신 등) 최신 보안 업데이트 유지 및 실시간 바이러스 검사 실행 등 원내 PC와 동일한 보안정책 적용
- PC방 등 불특정 다수가 사용하는 PC를 이용하여 재택·원격근무 금지

※ 로그인 시 위 내용을 숙지하고 준수할 것에 동의한

Victim’s status: not logged in

https://portal.unist.ac.kr



Announcement

Announcement

[버스 승강장] 버스 승
[식당] 기숙사식당 조
114동 에어컨 교체 공시
[IT Security] 'XZ Utils
Adobe 소프트웨어 라

Victim’s status: logged in

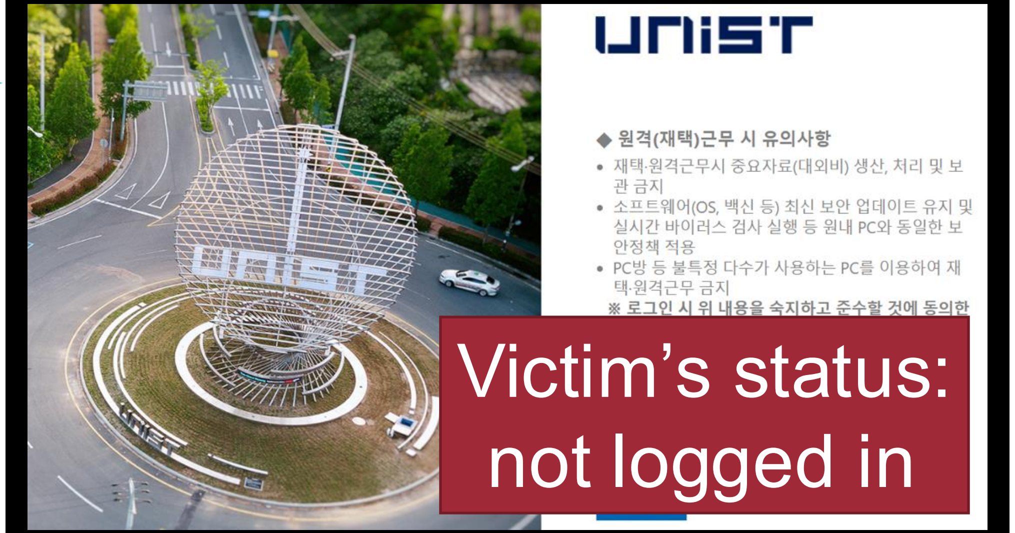
XS-Leaks Example

https://www.attacker.com

```
<iframe
  src=https://portal.unist.ac.kr/>
</iframe>
```

```
<script>
  if (“Announcement” in frames[0].content)
    // logged in
  else
    // not logged in
</script>
```

https://portal.unist.ac.kr

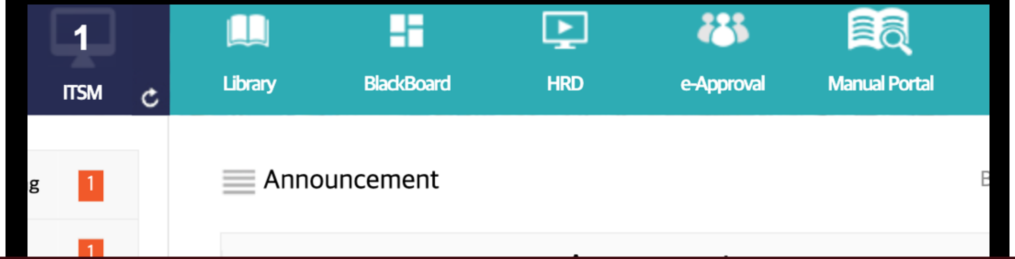


◆ 원격(재택)근무 시 유의사항
- 재택·원격근무시 중요자료(대외비) 생산, 처리 및 보관 금지
- 소프트웨어(OS, 백신 등) 최신 보안 업데이트 유지 및 실시간 바이러스 검사 실행 등 원내 PC와 동일한 보안정책 적용
- PC방 등 불특정 다수가 사용하는 PC를 이용하여 재택·원격근무 금지

※ 로그인 시 위 내용을 숙지하고 준수할 것에 동의한

Victim’s status: not logged in

https://portal.unist.ac.kr

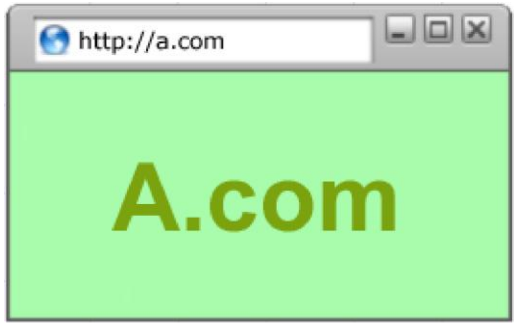


Announcement

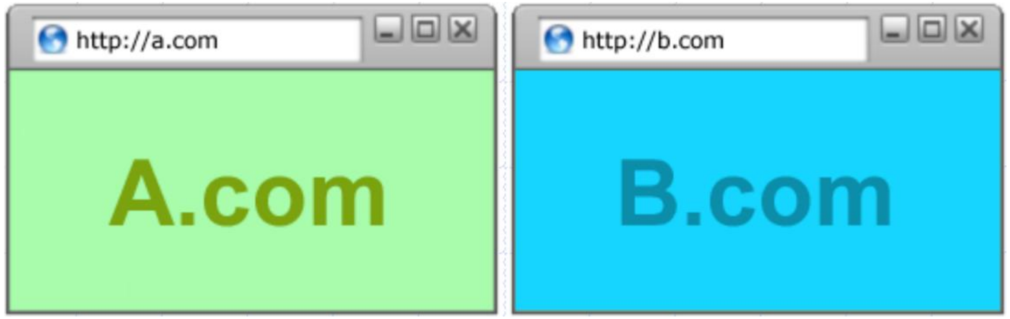
Result in a DOMException due to the same-origin policy

Recall: SOP Goals

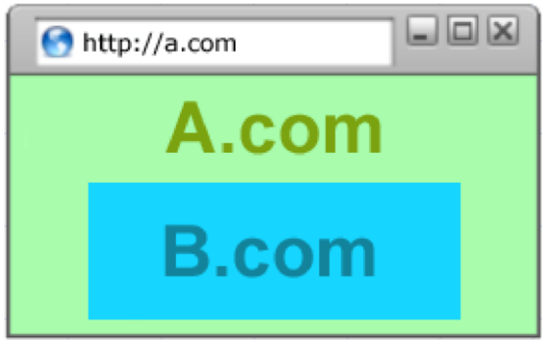
- Safe to visit an evil website



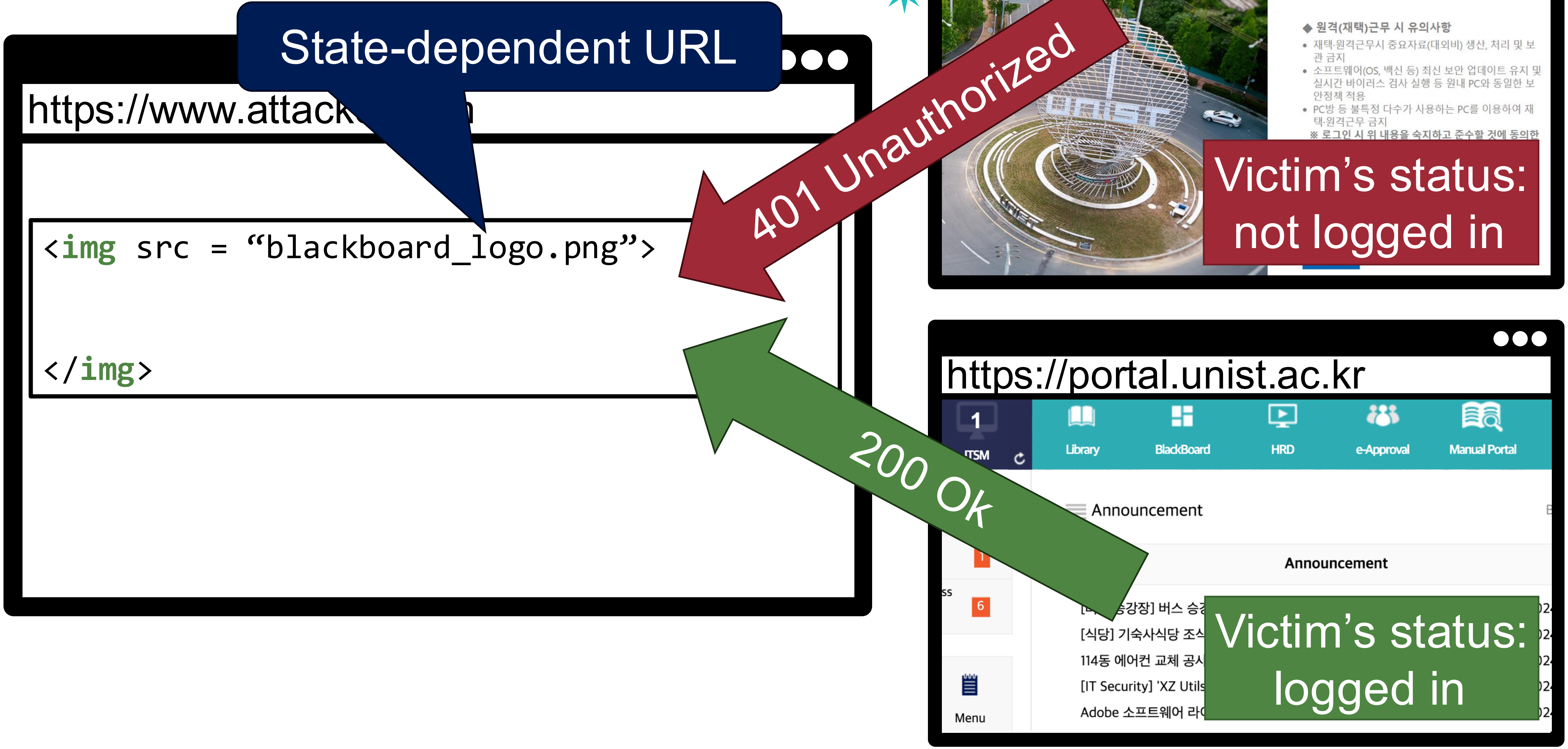
- Safe to visit two pages at the same time
  - Address bar distinguishes them



- Allow safe delegation



XS-Leaks Example



```
<img src = “blackboard_logo.png”>


</img>
```

◆ 원격(재택)근무 시 유의사항
- 재택·원격근무시 중요자료(대외비) 생산, 처리 및 보관 금지
- 소프트웨어(OS, 백신 등) 최신 보안 업데이트 유지 및 실시간 바이러스 검사 실행 등 원내 PC와 동일한 보안정책 적용
- PC방 등 불특정 다수가 사용하는 PC를 이용하여 재택·원격근무 금지

※ 로그인 시 위 내용을 숙지하고 준수할 것에 동의한

Announcement

Announcement

[식당] 기숙사식당 조스

114동 에어컨 교체 공시

[IT Security] 'XZ Utils

Adobe 소프트웨어 라이

XS-Leaks Example

State-dependent URL

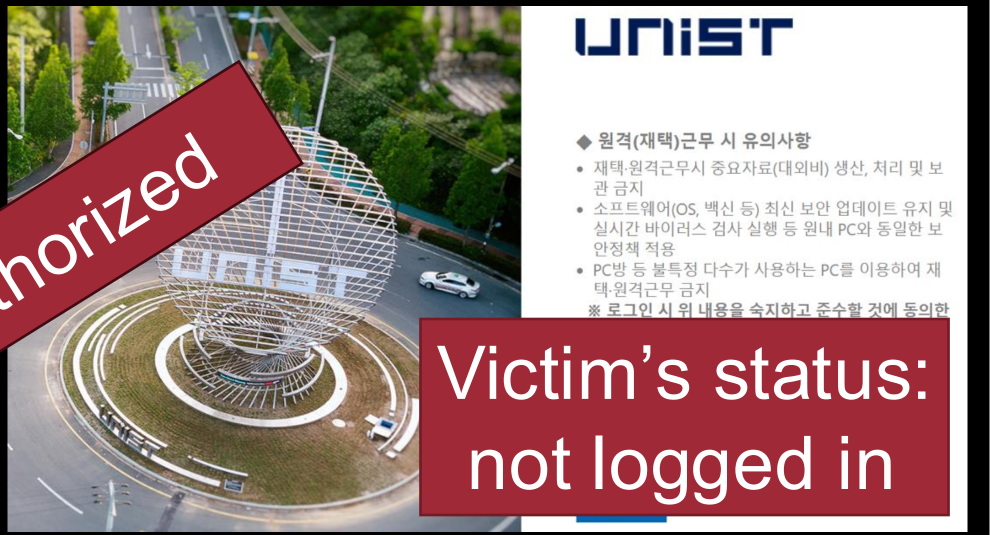
https://www.attack m

```
<img src = “blackboard_logo.png”>
    onload = “alert(‘logged-in’)”
    onerror = “alert(‘not logged-in’)”
</img>
```

Now, the web attacker will know whether the victim is logged in or not

401 Unauthorized

https://portal.unist.ac.kr



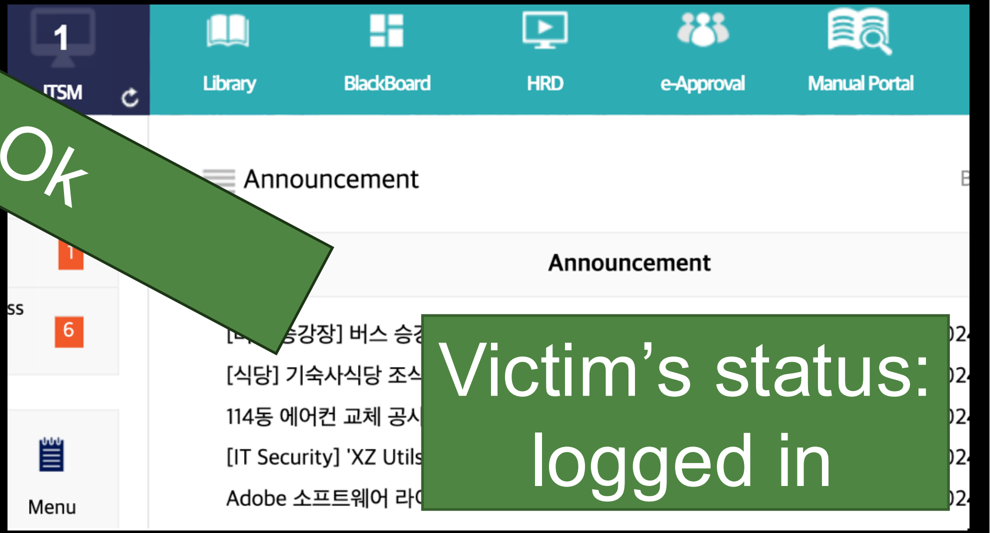
◆ 원격(재택)근무 시 유의사항
- 재택·원격근무시 중요자료(대외비) 생산, 처리 및 보관 금지
- 소프트웨어(OS, 백신 등) 최신 보안 업데이트 유지 및 실시간 바이러스 검사 실행 등 원내 PC와 동일한 보안정책 적용
- PC방 등 불특정 다수가 사용하는 PC를 이용하여 재택·원격근무 금지

※ 로그인 시 위 내용을 숙지하고 준수할 것에 동의한

Victim’s status: not logged in

200 Ok

https://portal.unist.ac.kr



Announcement

Announcement

[ ]승강장] 버스 승
[식당] 기숙사식당 조
114동 에어컨 교체 공
[IT Security] 'XZ Utils
Adobe 소프트웨어 라

Victim’s status: logged in

XS-Leaks Example – gitlab

```
let url = ‘https://git.company.com/profile’
let ref = window.open(url, ‘_blank’)
// wait until pop-up is loaded let
```

XS-Leaks Example – gitlab

```
let url = ‘https://git.company.com/profile’
let ref = window.open(url, ‘_blank’)
// wait until pop-up is loaded let
counted_frames = ref.window.length;
if (counted_frames === 0) {
    // User is logged in
} else if (counted_frames === 3) {
    // User is NOT logged in
}
```

XS-Leaks Example: HotCRP Reviewer Deanonymize

```
<link href=‘https://ndss.hotcrp.com/api.php/review?p=123’
      rel=‘prefetch’
      onload = alert(‘reviewer’)
      onerror = alert(‘not reviewer’) />
```

Chrome

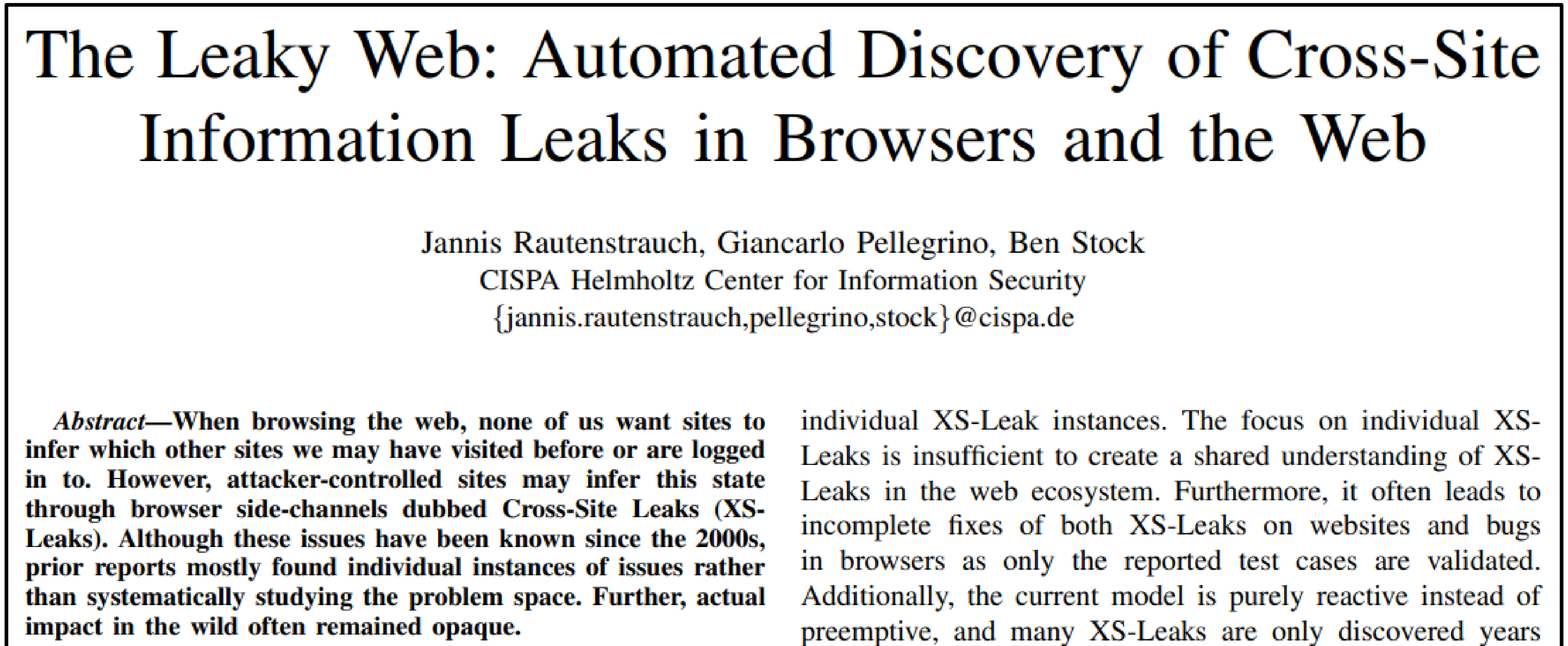
Allow XS-Leaks

Safari Firefox

Prevent XS-Leaks

The Leaky Web, *S&P'2023*

- Detect and characterize 280 observation channels that leak information cross-site in the engines of Chromium, Firefox, and Safari



The Leaky Web: Automated Discovery of Cross-Site Information Leaks in Browsers and the Web

Jannis Rautenstrauch, Giancarlo Pellegrino, Ben Stock
CISPA Helmholtz Center for Information Security
{jannis.rautenstrauch,pellegrino,stock}@cispa.de

***Abstract*—When browsing the web, none of us want sites to infer which other sites we may have visited before or are logged in to. However, attacker-controlled sites may infer this state through browser side-channels dubbed Cross-Site Leaks (XS-Leaks). Although these issues have been known since the 2000s, prior reports mostly found individual instances of issues rather than systematically studying the problem space. Further, actual impact in the wild often remained opaque.**

individual XS-Leak instances. The focus on individual XS-Leaks is insufficient to create a shared understanding of XS-Leaks in the web ecosystem. Furthermore, it often leads to incomplete fixes of both XS-Leaks on websites and bugs in browsers as only the reported test cases are validated. Additionally, the current model is purely reactive instead of preemptive, and many XS-Leaks are only discovered years

How to Mitigate XS-Leaks?

- `SameSite` Cookies
  - Prevent the browser from sending the cookie in cross-site request
  - **Disable state-dependent URL whose responses are based on states saved in cookies**

- `X-Frame-Options` header

- `Cross-Origin-Opener-Policy` (COOP) header: Make sure that my website is not opened on cross website

Question?