German OWASP Day 2025

German OWASP Day 2025

Extract: A PHP Foot-Gun Case Study

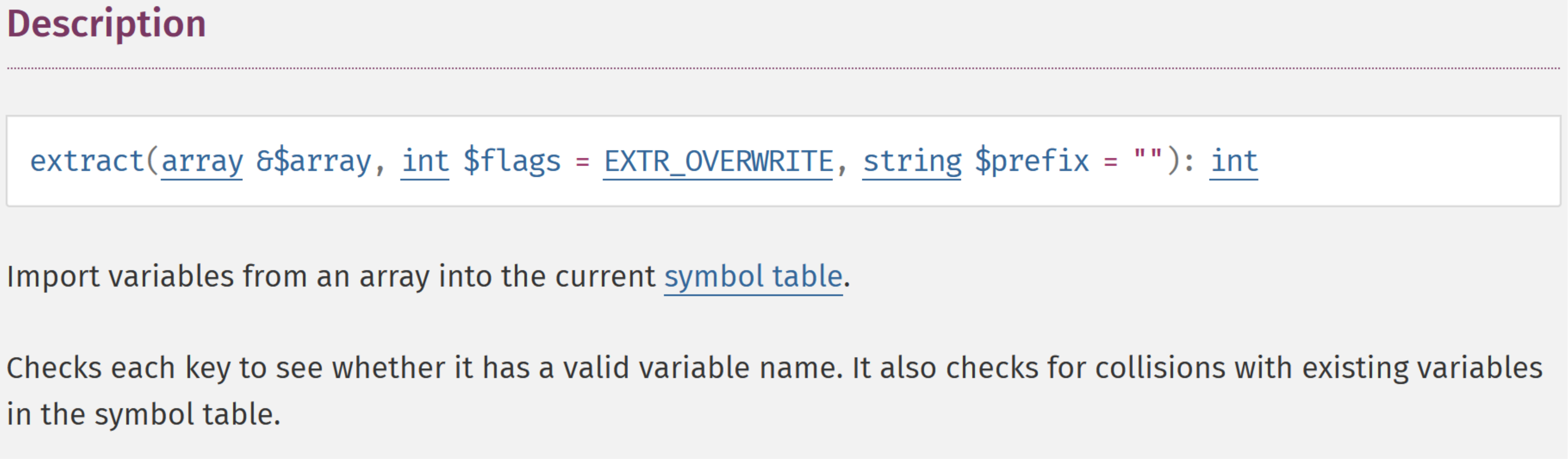
Jannik Hartung, Simon Koch, Martin Johns

- PhD candidate @ IAS / TU Braunschweig
- CTF nerd
- Pwntools maintainer

IAS INSTITUTE FOR APPLICATION SECURITY

Technische Universität Braunschweig

extract



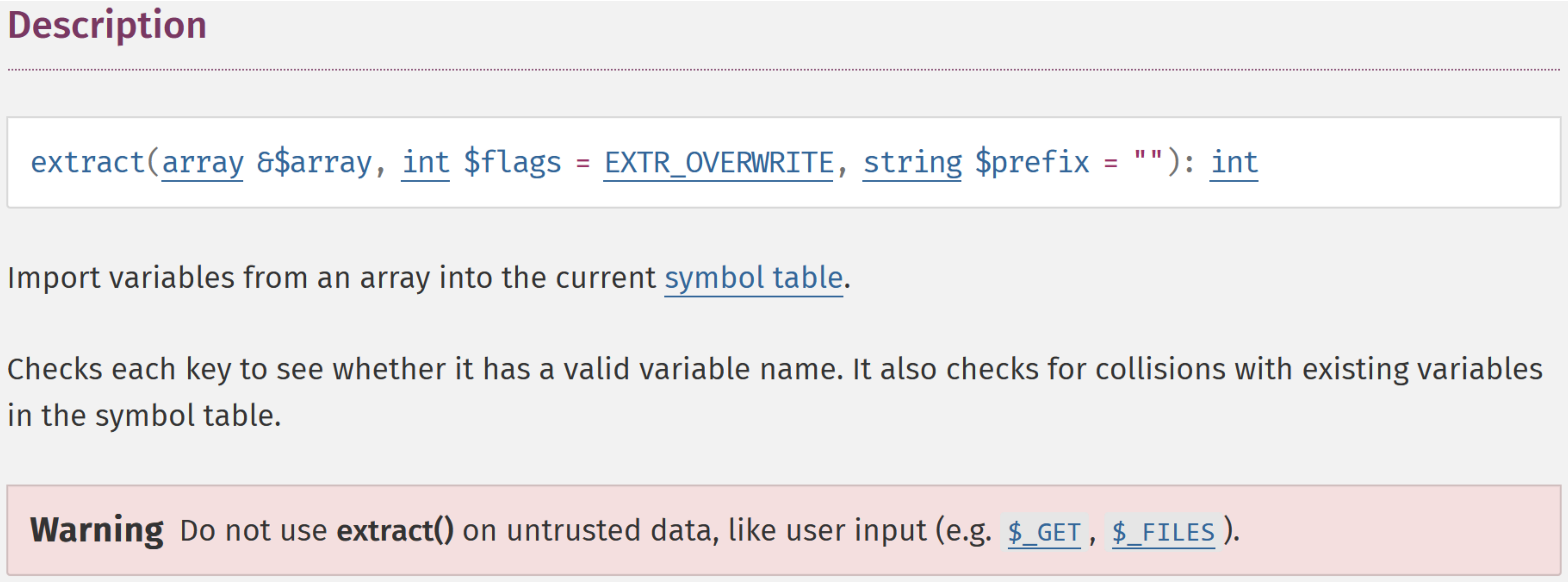
Description

```
extract(array &$array, int $flags = EXTR_OVERWRITE, string $prefix = ""): int
```

Import variables from an array into the current symbol table.

Checks each key to see whether it has a valid variable name. It also checks for collisions with existing variables in the symbol table.

extract



Description

```
extract(array &$array, int $flags = EXTR_OVERWRITE, string $prefix = ""): int
```

Import variables from an array into the current symbol table.

Checks each key to see whether it has a valid variable name. It also checks for collisions with existing variables in the symbol table.

Warning Do not use **extract()** on untrusted data, like user input (e.g. `$_GET`, `$_FILES`).

Using extract

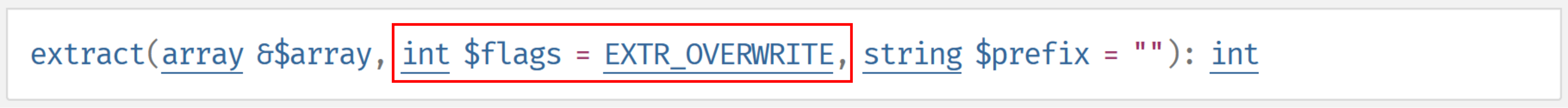
```
$array = ["key1" => "value1"];
echo $array["key1"]; // Output: value1
```

Using extract

```
$array = ["key1" => "value1"];
echo $array["key1"]; // Output: value1

extract($array);
echo $key1; // Output: value1
```

Handling existing variables

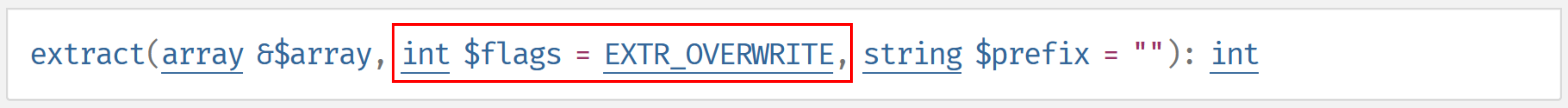


```
extract(array &$array, int $flags = EXTR_OVERWRITE, string $prefix = ""): int
```

```
$key1 = "owasp";
$array = ["key1" => "value1", "key2" => "value2"];

extract($array, EXTR_?);
echo $key1 . " " . $key2;
```

Handling existing variables



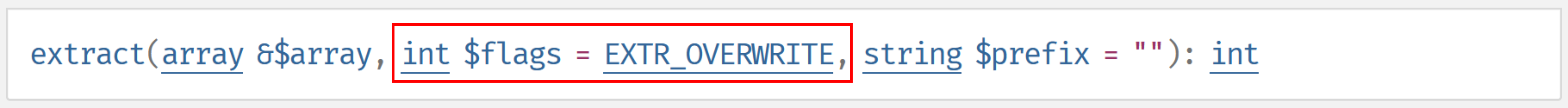
```
extract(array &$array, int $flags = EXTR_OVERWRITE, string $prefix = ""): int
```

```
$key1 = "owasp";
$array = ["key1" => "value1", "key2" => "value2"];

extract($array, EXTR_?);
echo $key1 . " " . $key2;
```

- **EXTR_OVERWRITE** (default): `Output: value1 value2`

Handling existing variables



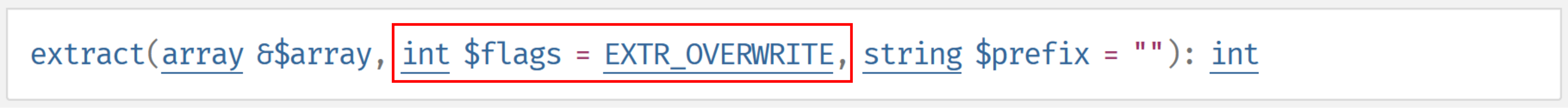
```
extract(array &$array, int $flags = EXTR_OVERWRITE, string $prefix = ""): int
```

```
$key1 = "owasp";
$array = ["key1" => "value1", "key2" => "value2"];

extract($array, EXTR_?);
echo $key1 . " " . $key2;
```

- **EXTR_OVERWRITE** (default): `Output: value1 value2`
- **EXTR_SKIP**: `Output: owasp value2`

Handling existing variables



```
extract(array &$array, int $flags = EXTR_OVERWRITE, string $prefix = ""): int
```

```
$key1 = "owasp";
$array = ["key1" => "value1", "key2" => "value2"];

extract($array, EXTR_?);
echo $key1 . " " . $key2;
```

- **EXTR_OVERWRITE** (default): `Output: value1 value2`
- **EXTR_SKIP**: `Output: owasp value2`
- **EXTR_IF_EXISTS**: `Output: value1`

What‘s the problem?

```php
<?php
$secret = <unknown>;

extract($_GET);
// $_GET["guess"] === $_GET["secret"]
if ($guess === $secret)
  echo "Correct";
else
  echo "Wrong";
```

What‘s the problem?

```
<?php
$secret = <unknown>;

extract($_GET);
// $_GET["guess"] === $_GET["secret"]
if ($guess === $secret)
  echo "Correct";
else
  echo "Wrong";
```

GET /?guess=1

What’s the problem?

```php
<?php
$secret = <unknown>;

extract($_GET);
// $_GET["guess"] === $_GET["secret"]
if ($guess === $secret)
   echo "Correct";
else
   echo "Wrong";
```

```
GET /?guess=1&secret=1
```

User input in PHP

- User-supplied key *and* value: `$_GET, $_POST, $_COOKIES, $_REQUEST`
- User-supplied key: `$_FILES`

User input in PHP

- User-supplied key *and* value: `$_GET, $_POST, $_COOKIES, $_REQUEST`
- User-supplied key: `$_FILES`

- Rich array support:

  GET /?thing[]=1&thing[]=2

  ```
  $_GET["thing"] === [1, 2]
  ```

User input in PHP

- User-supplied key *and* value: `$_GET, $_POST, $_COOKIES, $_REQUEST`
- User-supplied key: `$_FILES`

- Rich array support:

  GET /?thing[]=1&thing[]=2

  ```
  $_GET["thing"] === [1, 2]
  ```

- Even associative:

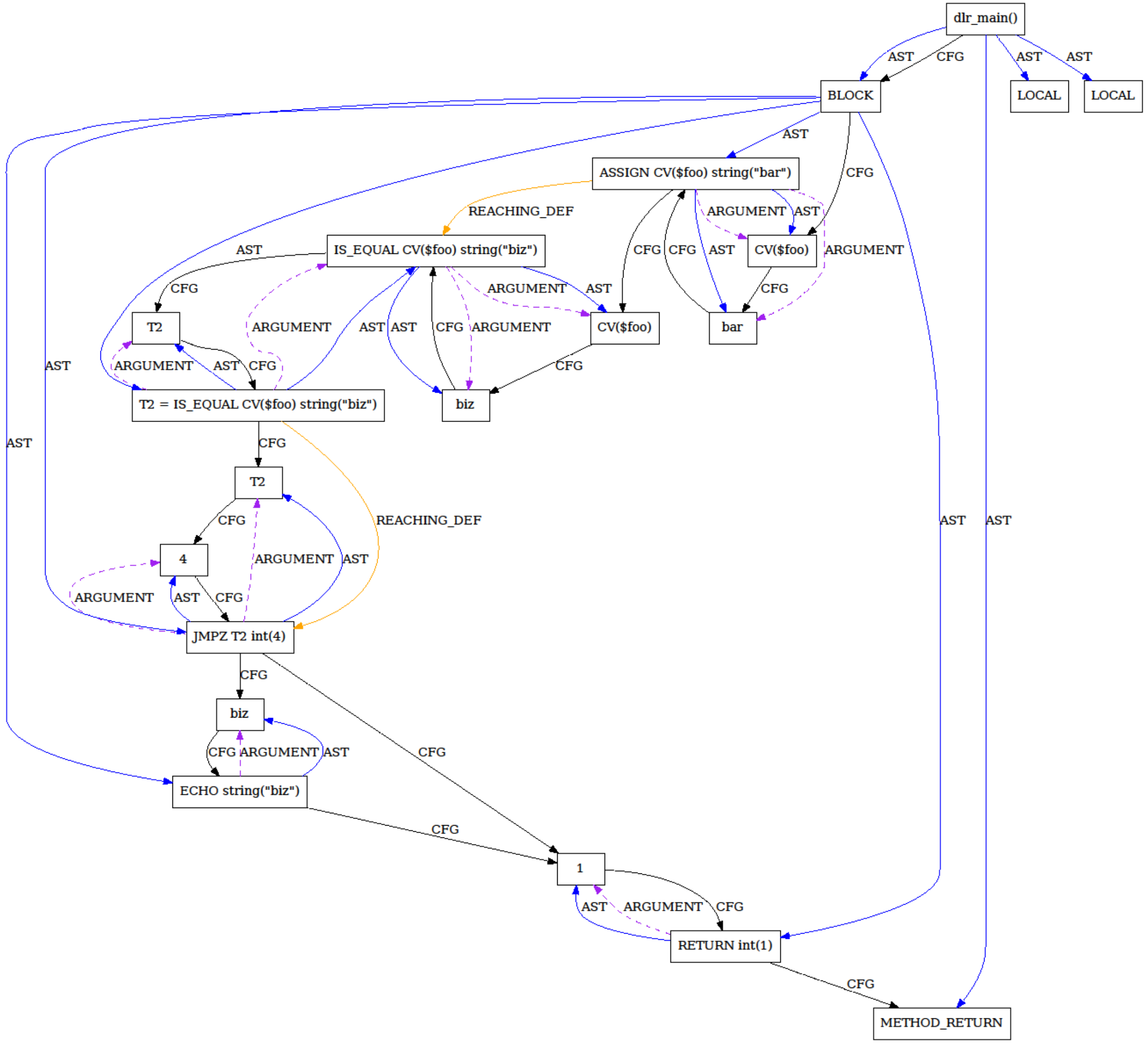
  GET /?config[prefix]=owasp

  ```
  $_GET["config"] === ["prefix" => "owasp"]
  ```

Static Analysis

- PHP Code Property Graph (CPG)

```php
<?php
if($_GET["foo"] == "bar"){
    extract($_GET);
}
echo $bar;
```



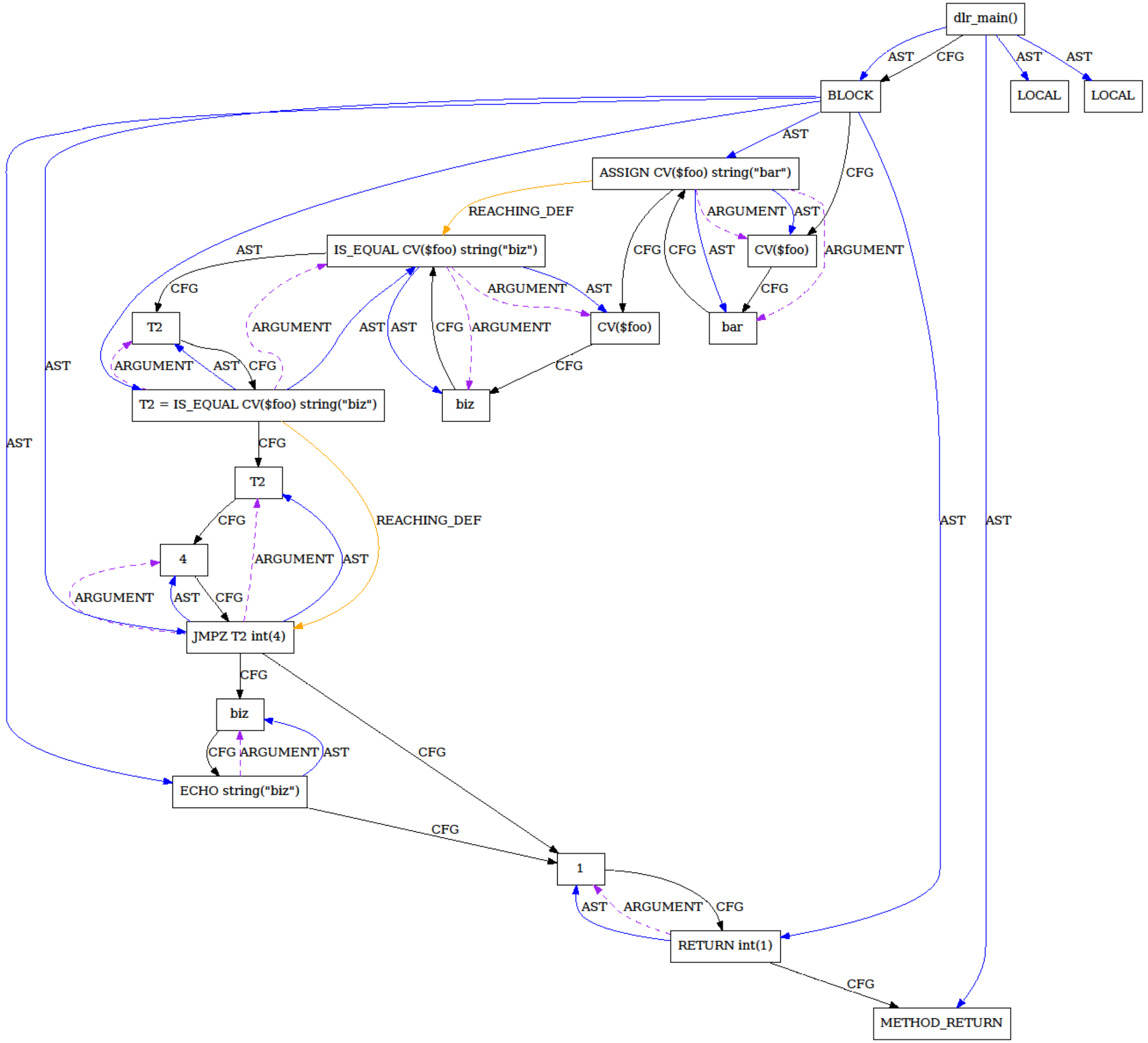
German OWASP Day 2025

Static Analysis

- PHP Code Property Graph (CPG)

1. Find calls to **extract**

```
<?php
if($_GET["foo"] == "bar"){
    extract($_GET);
}
echo $bar;
```

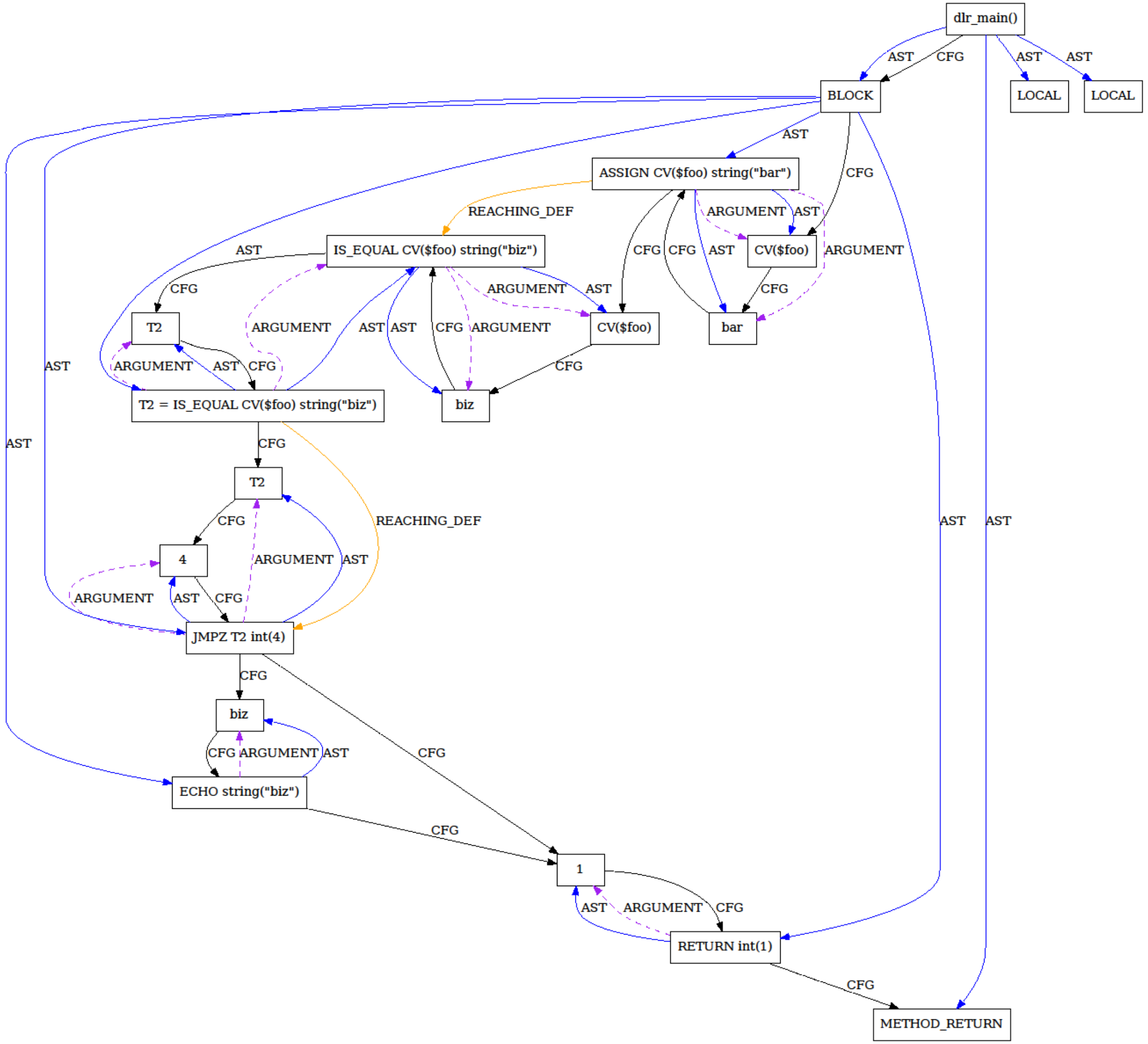


Static Analysis

- PHP Code Property Graph (CPG)

1. Find calls to **extract**
2. Create slices starting at **extract**

```php
<?php
if($_GET["foo"] == "bar"){
    extract($_GET);
}
echo $bar;
```

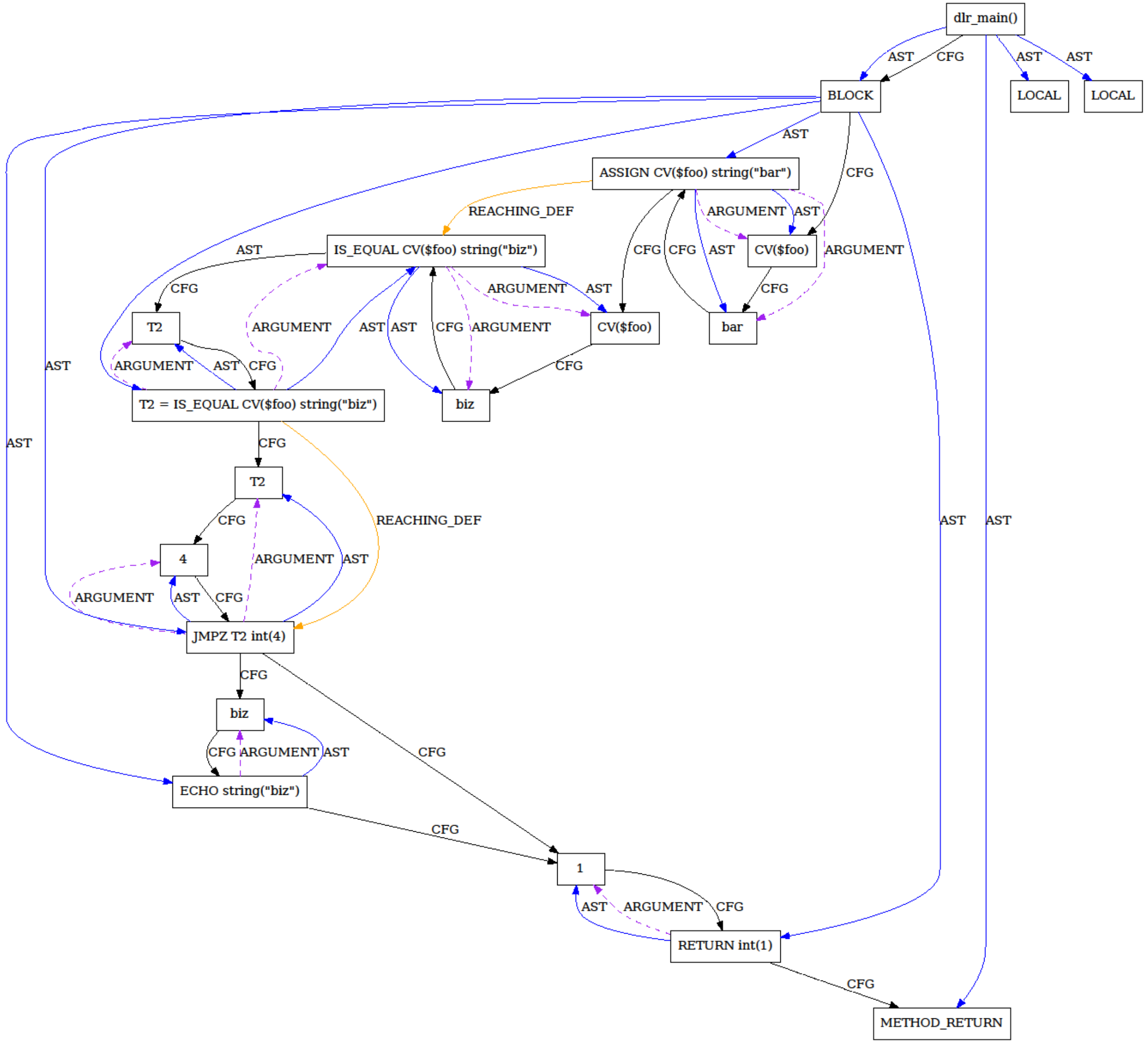


Static Analysis

- PHP Code Property Graph (CPG)

1. Find calls to **extract**
2. Create slices starting at **extract**
3. Test for data flow to user input

```
<?php
if($_GET["foo"] == "bar"){
    extract($_GET);
}
echo $bar;
```



What about the real world?

- 28.158 analyzed CPGs

What about the real world?

- 28.158 analyzed CPGs
- 1.331 repositories used **extract**
  - 4.934 calls to **extract**

What about the real world?

- 28.158 analyzed CPGs
- 1.331 repositories used **extract**
  - 4.934 calls to **extract**
- 43 repositories with user input
  - with 146 **extract** calls

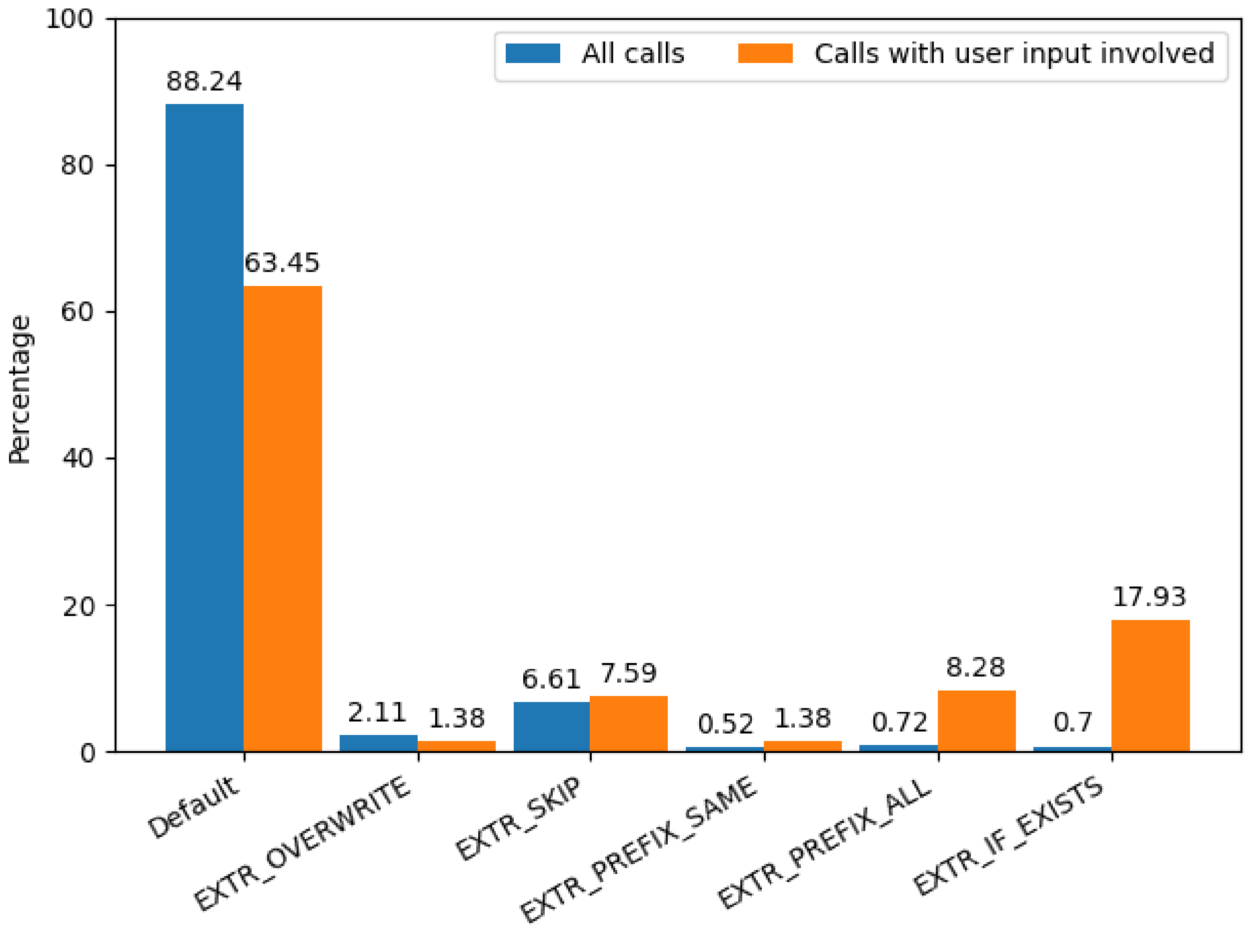
What about the real world?

- 28.158 analyzed CPGs
- 1.331 repositories used **extract**
  - 4.934 calls to **extract**
- 43 repositories with user input
  - with 146 **extract** calls

- 26 vulnerable repositories
  - 117 exploitable calls

What about the real world?

- 28.158 analyzed CPGs
- 1.331 repositories used **extract**
  - 4.934 calls to **extract**
- 43 repositories with user input
  - with 146 **extract** calls
- 26 vulnerable repositories
  - 117 exploitable calls

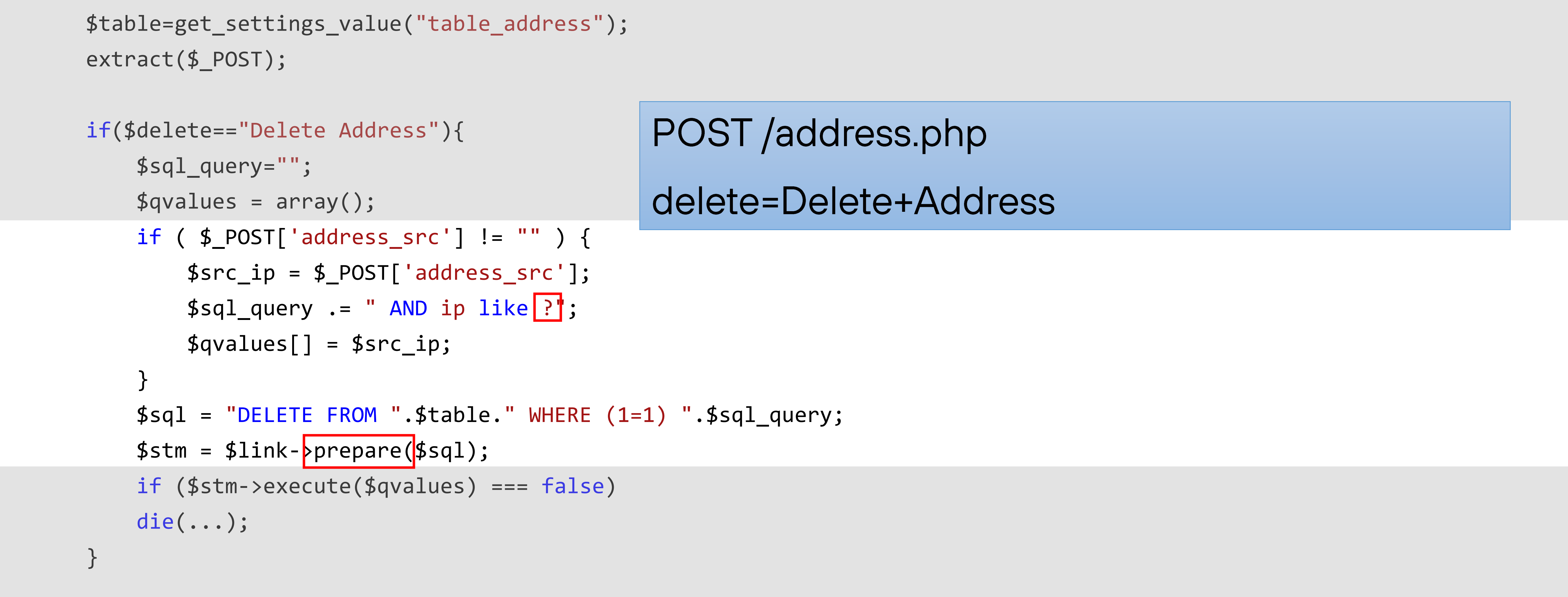


Change arbitrary variables

```
$table=get_settings_value("table_address");
extract($_POST);

if($delete=="Delete Address"){
    $sql_query="";
    $qvalues = array();
    if ( $_POST['address_src'] != "" ) {
        $src_ip = $_POST['address_src'];
        $sql_query .= " AND ip like ?";
        $qvalues[] = $src_ip;
    }
    $sql = "DELETE FROM ".$table." WHERE (1=1) ".$sql_query;
    $stm = $link->prepare($sql);
    if ($stm->execute($qvalues) === false)
    die(...);
}
```

Change arbitrary variables



```
$table=get_settings_value("table_address");
extract($_POST);

if($delete=="Delete Address"){
    $sql_query="";
    $qvalues = array();
    if ( $_POST['address_src'] != "" ) {
        $src_ip = $_POST['address_src'];
        $sql_query .= " AND ip like ?";
        $qvalues[] = $src_ip;
    }
    $sql = "DELETE FROM ".$table." WHERE (1=1) ".$sql_query;
    $stm = $link->prepare($sql);
    if ($stm->execute($qvalues) === false)
    die(...);
}
```

POST /address.php

delete=Delete+Address

Change arbitrary variables

```
$table=get_settings_value("table_address");
extract($_POST);

if($delete=="Delete Address"){
    $sql_query="";
    $qvalues = array();
    if ( $_POST['address_src'] != "" ) {
        $src_ip = $_POST['address_src'];
        $sql_query .= " AND ip like ?";
        $qvalues[] = $src_ip;
    }
    $sql = "DELETE FROM ".$table." WHERE (1=1) ".$sql_query;
    $stm = $link->prepare($sql);
    if ($stm->execute($qvalues) === false)
    die(...);
}
```

```
POST /address.php

delete=Delete+Address
```

Change arbitrary variables

```
$table=get_settings_value("table_address");
extract($_POST);

if($delete=="Delete Address"){
    $sql_query="";
    $qvalues = array();
    if ( $_POST['address_src'] != "" ) {
        $src_ip = $_POST['address_src'];
        $sql_query .= " AND ip like ?";
        $qvalues[] = $src_ip;
    }
    $sql = "DELETE FROM ".$table." WHERE (1=1) ".$sql_query;
    $stm = $link->prepare($sql);
    if ($stm->execute($qvalues) === false)
    die(...);
}
```

```
POST /address.php

delete=Delete+Address&table=users
```

Change arbitrary variables

```
$table=get_settings_value("table_address");
extract($_POST);

if($delete=="Delete Address"){
    $sql_query="";
    $qvalues = array();
    if ( $_POST['address_src'] != "" ) {
        $src_ip = $_POST['address_src'];
        $sql_query .= " AND ip like ?";
        $qvalues[] = $src_ip;
    }
    $sql = "DELETE FROM ".$table." WHERE (1=1) ".$sql_query;
    $stm = $link->prepare($sql);
    if ($stm->execute($qvalues) === false)
    die(...);
}
```

```
POST /address.php

delete=Delete+Address&table=users
```

Change read-only variables

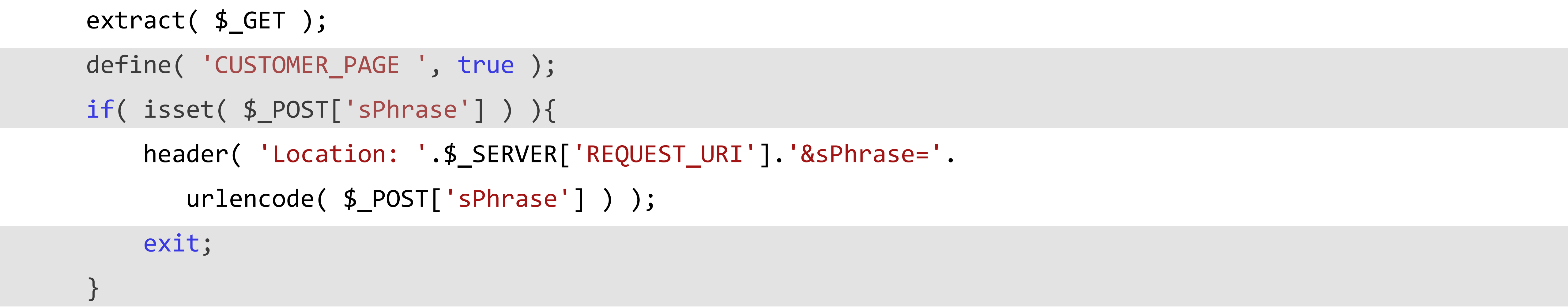
```
extract( $_GET );
define( 'CUSTOMER_PAGE ', true );
if( isset( $_POST['sPhrase'] ) ){
    header( 'Location: '.$_SERVER['REQUEST_URI'].'&sPhrase='.
       urlencode( $_POST['sPhrase'] ) );
    exit;
}
```

Change read-only variables

```
extract( $_GET );
define( 'CUSTOMER_PAGE ', true );
if( isset( $_POST['sPhrase'] ) ){
    header( 'Location: '.$_SERVER['REQUEST_URI'].'&sPhrase='.
       urlencode( $_POST['sPhrase'] ) );
    exit;
}
```

German OWASP Day 2025

Change read-only variables



```php
extract( $_GET );
define( 'CUSTOMER_PAGE ', true );
if( isset( $_POST['sPhrase'] ) ){
    header( 'Location: '.$_SERVER['REQUEST_URI'].'&sPhrase='.
       urlencode( $_POST['sPhrase'] ) );
    exit;
}
```

GET /index.php?_POST[sPhrase]=&_SERVER[REQUEST_URI]=shadyshop.com

Change read-only variables

```
extract( $_GET );
define( 'CUSTOMER_PAGE ', true );
if( isset( $_POST['sPhrase'] ) ){
    header( 'Location: '.$_SERVER['REQUEST_URI'].'&sPhrase='.
        urlencode( $_POST['sPhrase'] ) );
    exit;
}
```

```
GET /index.php?_POST[sPhrase]=&_SERVER[REQUEST_URI]=shadyshop.com
```

Other Superglobals: `$_SERVER, $_ENV, $_SESSION,` …

Persisting changes

```
session_start();
$_SESSION += array('email' => '', 'admin' => '', 'trusted' => 0, 'check' => '', 'posts' => 0);
//...
extract($_REQUEST + array('email' => '', 'topicID' => 0, 'commentID' => 0, 'title' => 0, 'body' => 0, 'delete' => 0));
//...
if($delete && $_SESSION['admin'])
{
  if($delete == 'topic') {
    //...
  } elseif($delete == 'user') {
    query('UPDATE user SET banned = 0 WHERE email = ?', array($email));
  } else {
    //...
  }
}
```

Persisting changes

```php
session_start();
$_SESSION += array('email' => '', 'admin' => '', 'trusted' => 0, 'check' => '', 'posts' => 0);
//...
extract($_REQUEST + array('email' => '', 'topicID' => 0, 'commentID' => 0, 'title' => 0, 'body' => 0, 'delete' => 0));
//...
if($delete && $_SESSION['admin'])
{
  if($delete == 'topic') {
    //...
  } elseif($delete == 'user') {
    query('UPDATE user SET banned = 0 WHERE email = ?', array($email));
  } else {
    //...
  }
}
```

Persisting changes

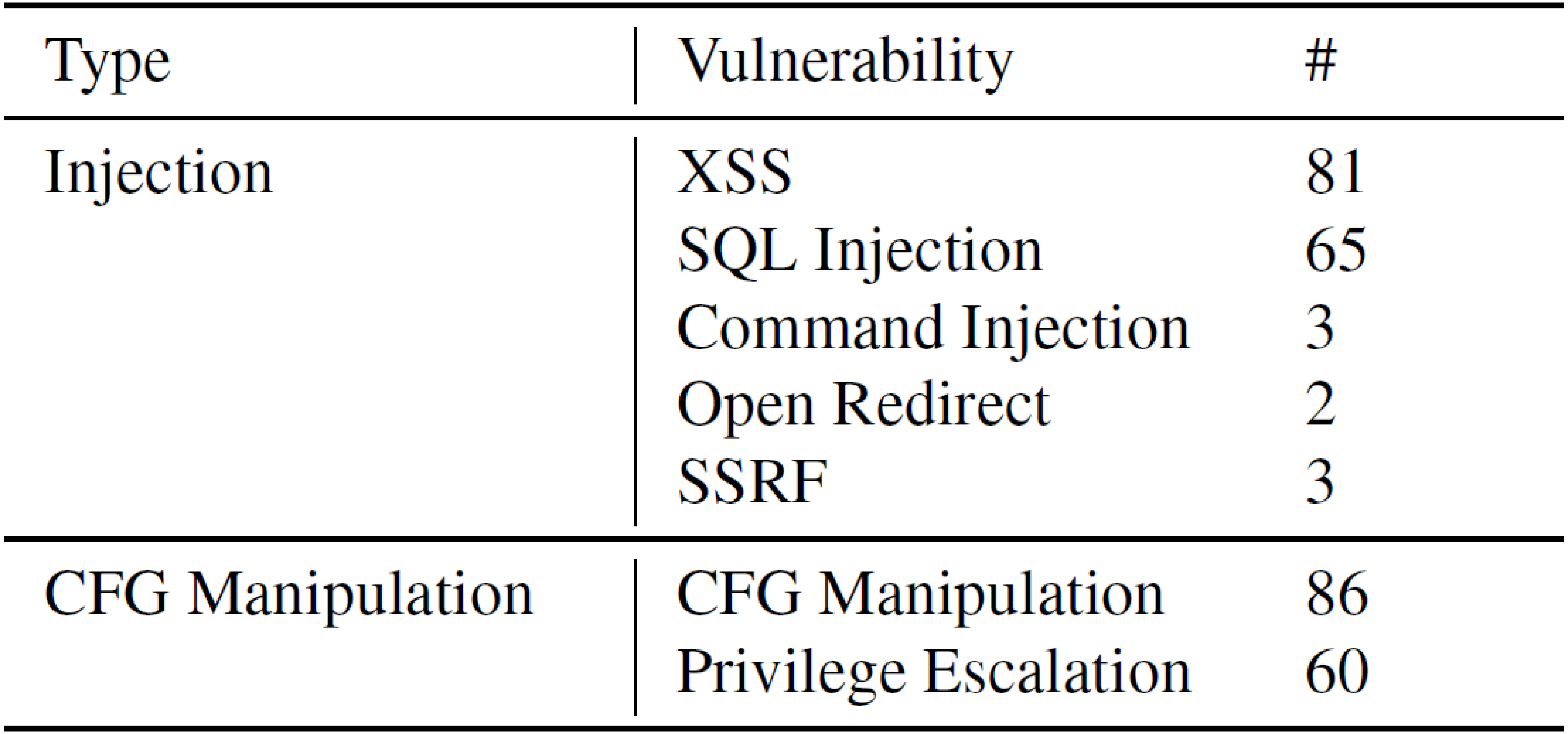
```
session_start();
$_SESSION += array('email' => '', 'admin' => '', 'trusted' => 0, 'check' => '', 'posts' => 0);
//...
extract($_REQUEST + array('email' => '', 'topicID' => 0, 'commentID' => 0, 'title' => 0, 'body' => 0, 'delete' => 0));
//...
if($delete && $_SESSION['admin'])
{
  if($delete == 'topic') {
    //...
  } elseif($delete == 'user') {
    query('UPDATE user SET banned = 0 WHERE email = ?', array($email));
  } else {
    //...
  }
}
```

Persisting changes

```
session_start();
$_SESSION += array('email' => '', 'admin' => '', 'trusted' => 0, 'check' => '', 'posts' => 0);
//...
extract($_REQUEST + array('email' => '', 'topicID' => 0, 'commentID' => 0, 'title' => 0, 'body' => 0, 'delete' => 0));
//...
if($delete && $_SESSION['admin'])
{
  if($delete == 'topic') {
    //...
  } elseif($delete == 'user') {
    query('UPDATE user SET banned = 0 WHERE email = ?', array($email));
  } else {
    //...
  }
}
```

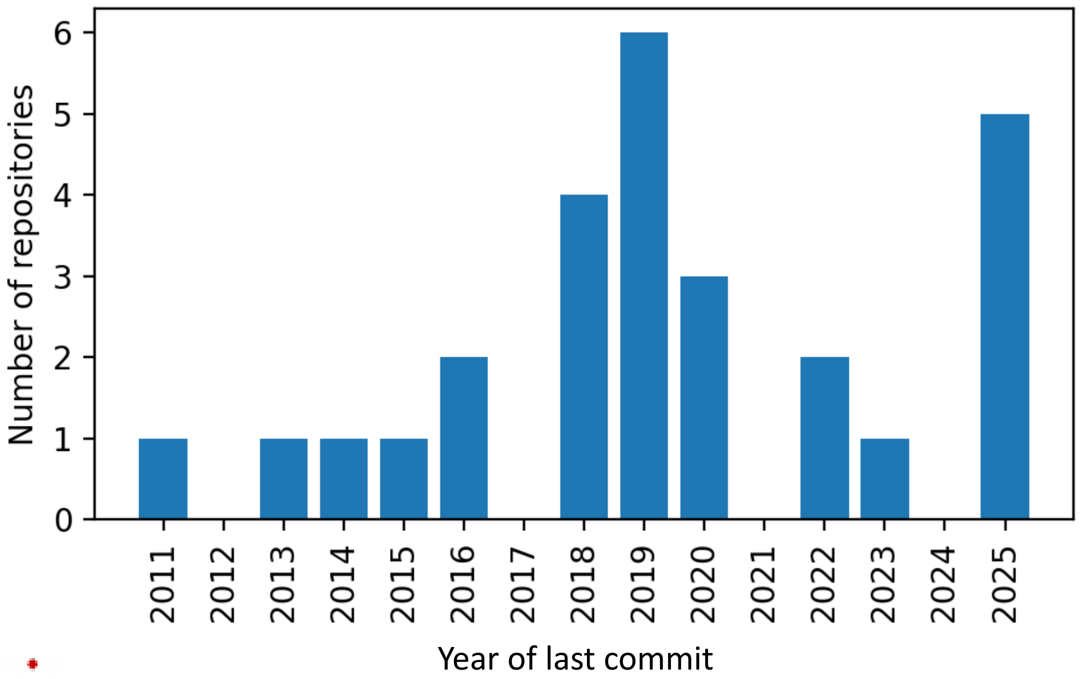
GET /forum.php?delete=user&email=admin@example.com&_SESSION[admin]=1

Overall vulnerabilities



Type	Vulnerability	#
Injection	XSS	81
	SQL Injection	65
	Command Injection	3
	Open Redirect	2
	SSRF	3
CFG Manipulation	CFG Manipulation	86
	Privilege Escalation	60

Nobody does this anymore, right?



snom

VoIP AUSTRALIA

localphone
Call global, pay local

opensips

Telappliant

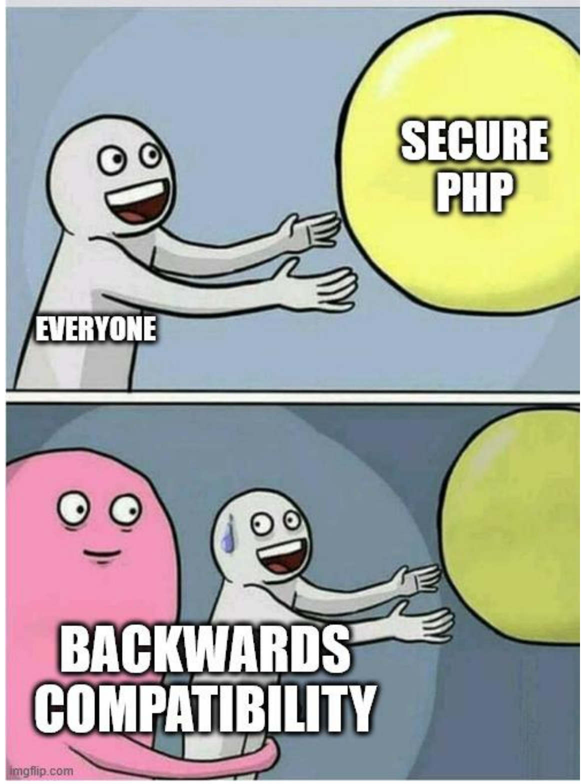
XConnect
Interconnecting Our Digital World™

Mitigations

- Deprecate **extract**

Mitigations

- ~~Deprecate **extract**~~

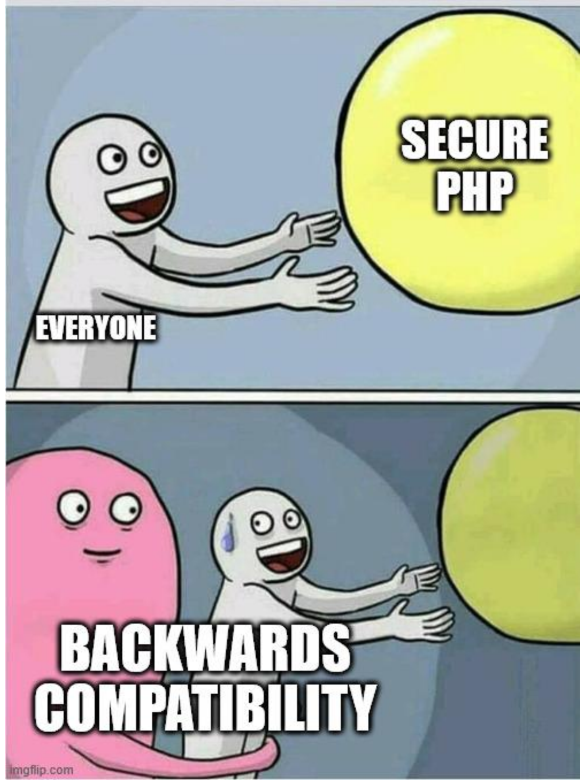


Mitigations

- ~~Deprecate **extract**~~
- Restrict Superglobals

```
extract(["~~_SESSION~~" => ["authenticated" => "1"] ]);
```

  - $GLOBALS already filtered
- Warn on usage in global scope



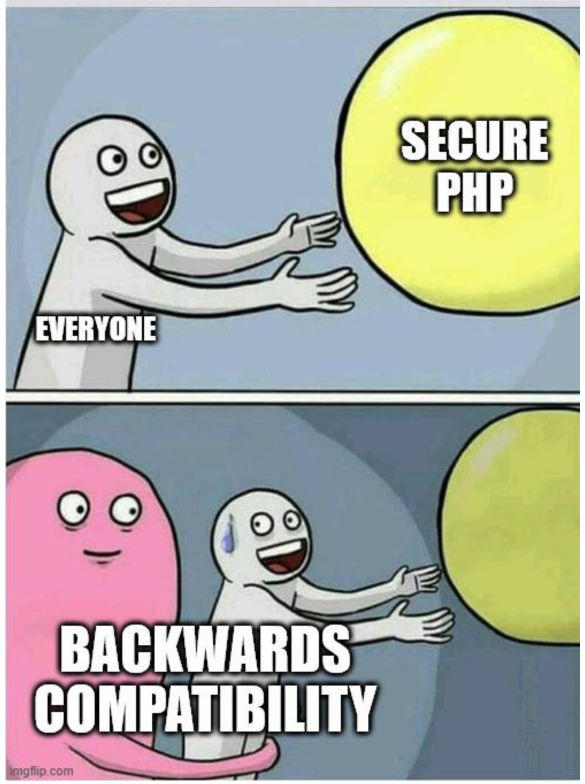
Mitigations

- ~~Deprecate **extract**~~
- Restrict Superglobals

extract([~~"_SESSION"~~ => ["authenticated" => "1"]]);

  - $GLOBALS already filtered
- Warn on usage in global scope

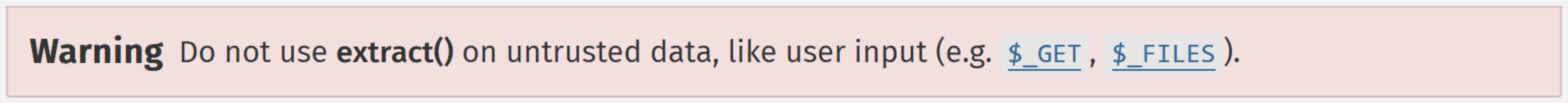
- List expected keys
  - Include default values



Conclusion

Conclusion

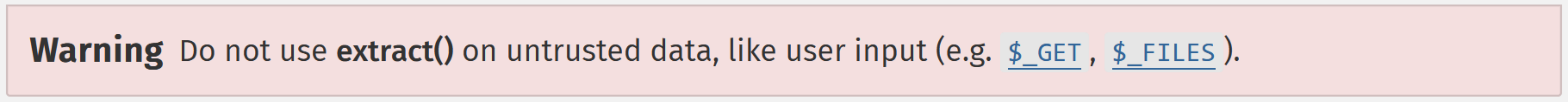
- Do not use extract on untrusted data



Warning Do not use **extract()** on untrusted data, like user input (e.g. `$_GET`, `$_FILES`).

Conclusion

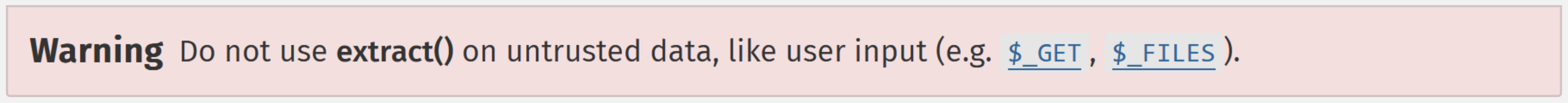
- Do not use extract on untrusted data
- Many ways to exploit untrusted extract calls



Warning Do not use **extract()** on untrusted data, like user input (e.g. `$_GET`, `$_FILES`).

Conclusion

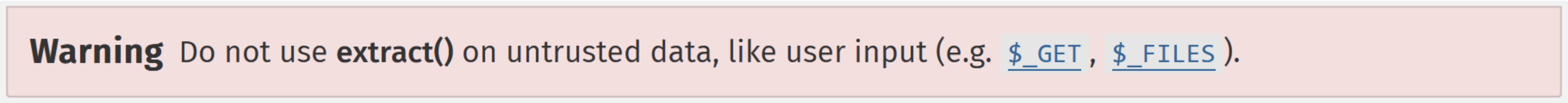
- Do not use extract on untrusted data
- Many ways to exploit untrusted extract calls
- **Protect Superglobals!**



Warning Do not use **extract()** on untrusted data, like user input (e.g. `$_GET`, `$_FILES`).

Conclusion

- Do not use extract on untrusted data
- Many ways to exploit untrusted extract calls
- **Protect Superglobals!**
- Check your code for legacy foot-guns



Warning Do not use **extract()** on untrusted data, like user input (e.g. `$_GET`, `$_FILES`).

German OWASP Day 2025

THANK YOU!