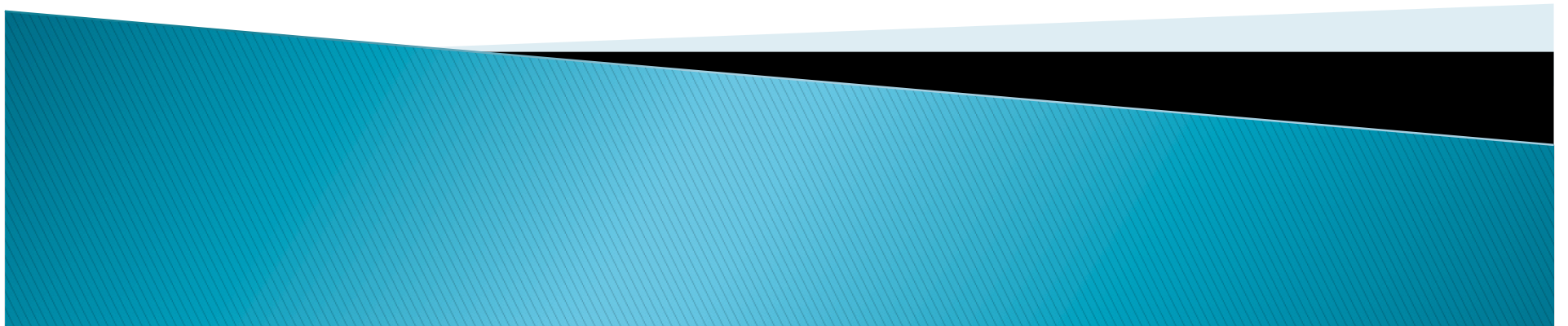


# Fix Me Up: Repairing Access-Control Bugs in Web Applications

Sooel Son

Kathryn S. McKinley      Vitaly Shmatikov  
UT Austin and Microsoft Research



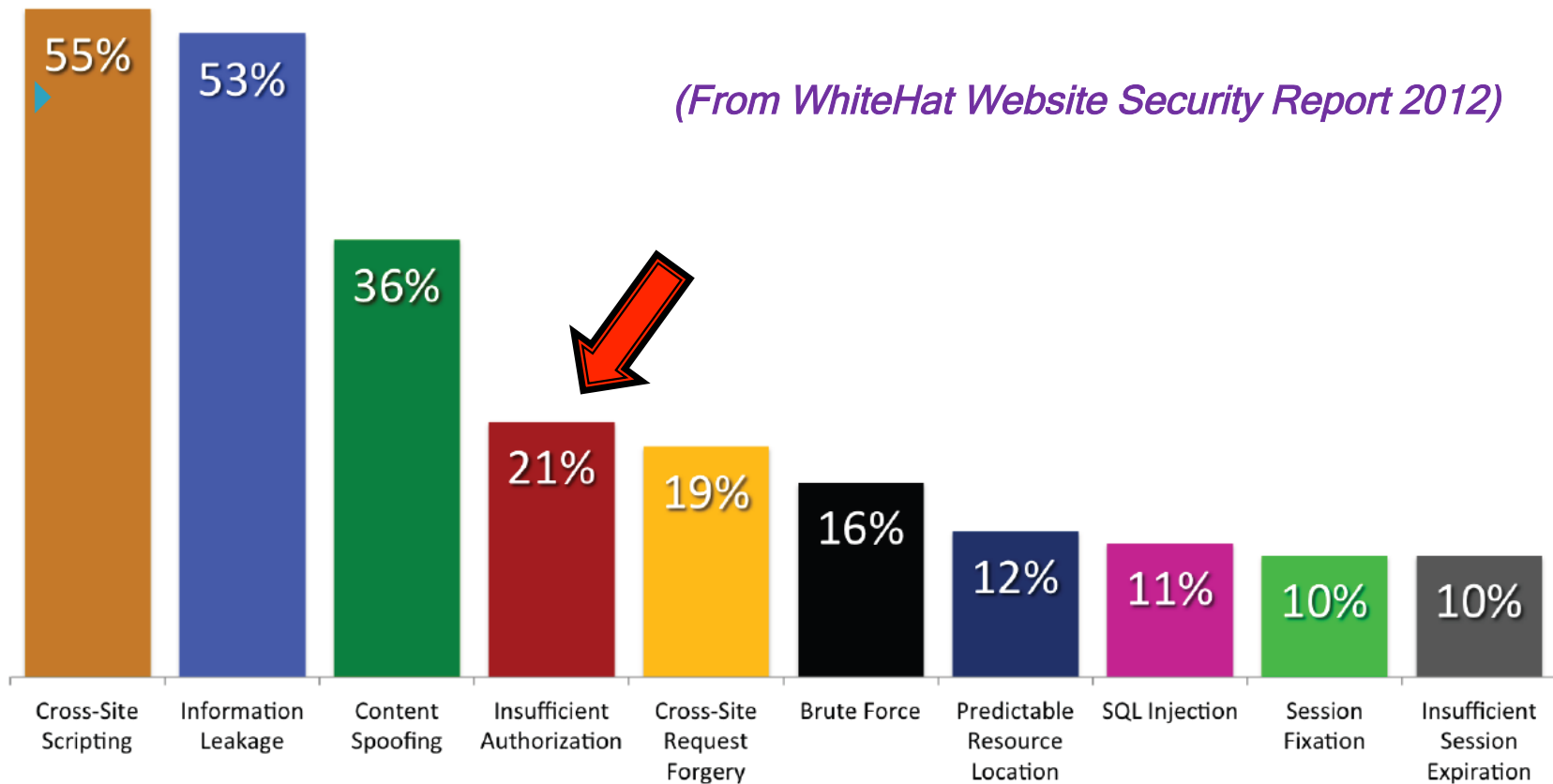
# Access-control bugs

## OWASP Top 10 Application Security Risks - 2010

|                                      | <b>T10</b>                             | <b>OWASP Top 10 Application Security Risks – 2013</b>                                                                                                                                                                                                                                                                                     |
|--------------------------------------|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A1-Injection                         |                                        |                                                                                                                                                                                                                                                                                                                                           |
|                                      |                                        | trusted data is sent to an interpreter. An attacker can trick the interpreter into executing arbitrary code.                                                                                                                                                                                                                              |
|                                      |                                        | Implementation are often not secure. Keys, session tokens, or other sensitive information can be exposed.                                                                                                                                                                                                                                 |
|                                      |                                        | exploit other implementation flaws to assume other users' identities.                                                                                                                                                                                                                                                                     |
| Direct Object References             | Management                             |                                                                                                                                                                                                                                                                                                                                           |
| A5-Cross Site Request Forgery (CSRF) | A3 – Cross-Site Scripting (XSS)        | •XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.                                             |
| A6-Security Misconfiguration         | A4 – Insecure Direct Object References | •A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.                                                      |
| A7-Insecure Cryptographic Storage    | A5 – Security Misconfiguration         | •Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. All these settings should be defined, implemented, and maintained as many are not shipped with secure defaults. This includes keeping all software up to date. |
| A8-Failure to Restrict URL           |                                        |                                                                                                                                                                                                                                                                                                                                           |

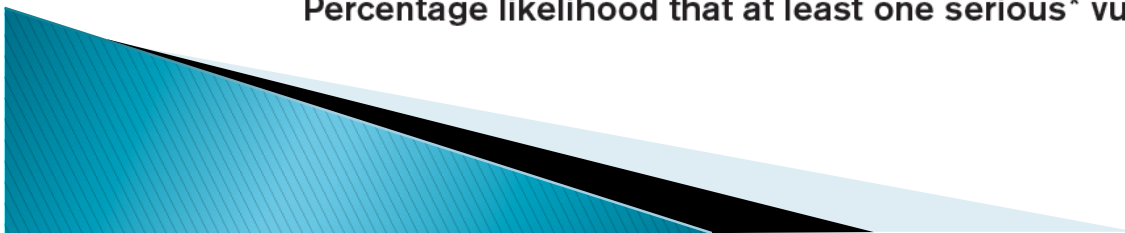
... a developer exposes a reference to an internal implementation object... Without an access control check, attackers can access unauthorized data

# Access-control bugs (2)



**Figure 3. Top Ten Vulnerability Classes (2011)**

Percentage likelihood that at least one serious\* vulnerability will appear in a website



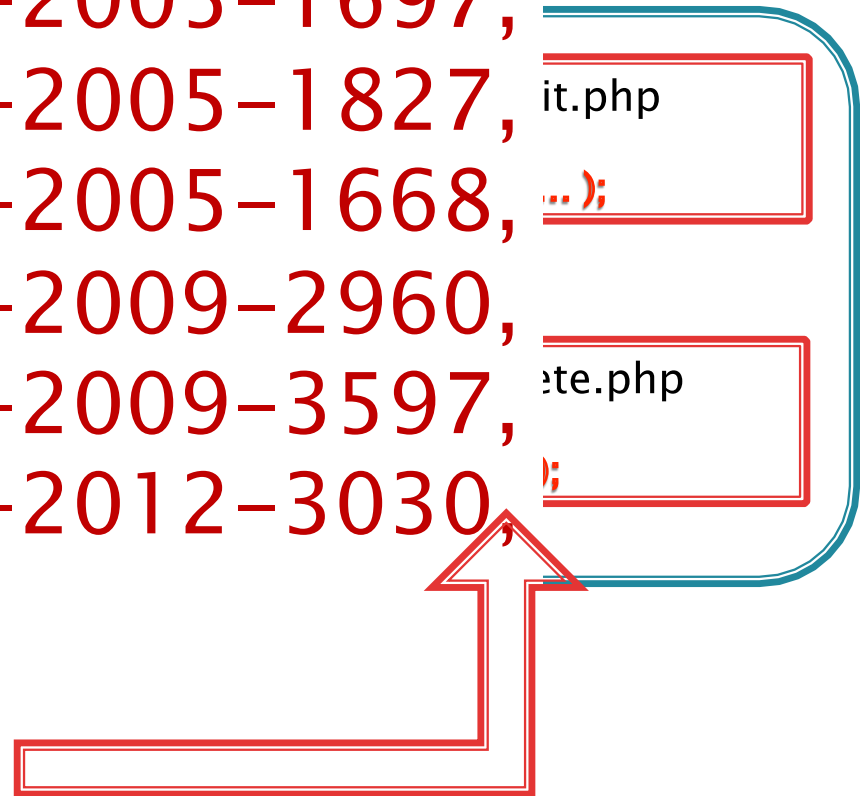
# Forced-browsing attack

CVE-2004-2144, CVE-2004-2257, **ement**  
CVE-2005-1688, CVE-2005-1697,  
CVE-2005-1698, CVE-2005-1827, **it.php**  
CVE-2005-1654, CVE-2005-1668, **... );**  
CVE-2005-1892, CVE-2009-2960,  
CVE-2009-3168, CVE-2009-3597, **delete.php**  
CVE-2011-0316, CVE-2012-3030, **;**  
CVE-2012-6451

.....



[http://host/delete.php?id=victim\\_id](http://host/delete.php?id=victim_id)





A photograph of a multi-story brick building facade. The building features a regular grid of windows and doors. The ground floor has a central entrance with a dark wooden door and a small sign above it. The upper floors have rows of windows, with some doors visible on the second and third floors. The brickwork is a mix of red and brown tones, showing some weathering and discoloration. The text is overlaid in a bright green color.

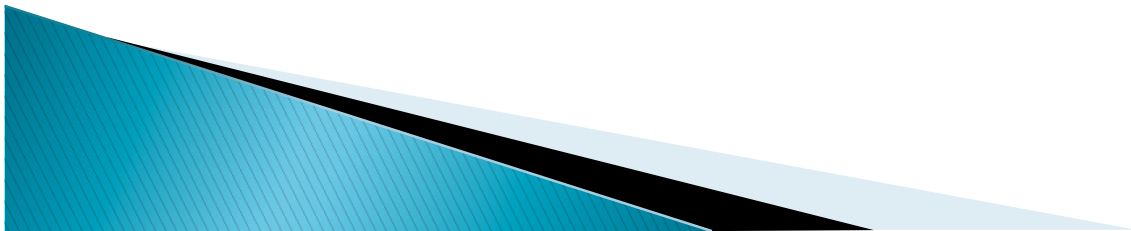
Make sure that every entry is locked with the proper access-control logic

# About FixMeUp

Static **program transformation tool** for finding and fixing access-control bugs in PHP applications

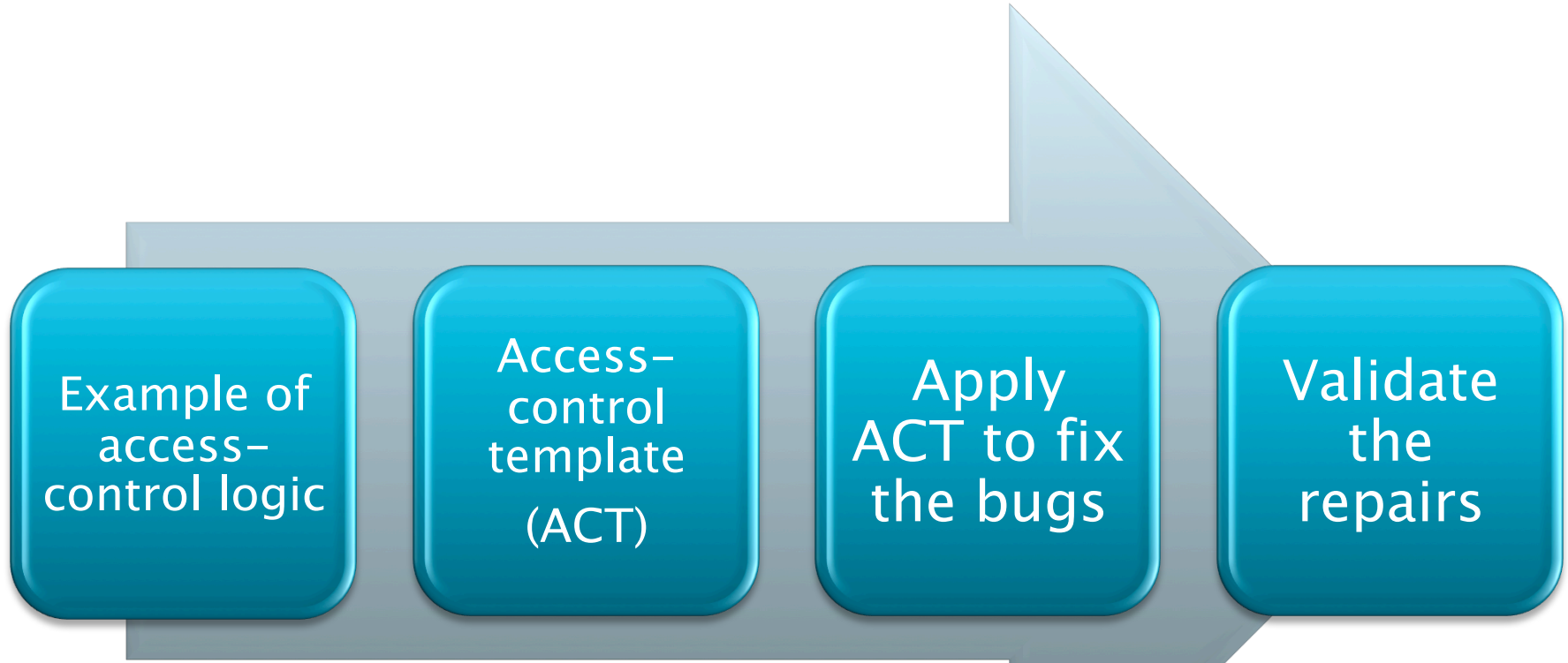
Given an example of correct access control ...

- ▶ 1. Finds calling contexts that do not implement the correct access-control logic
- ▶ 2. Produces candidate **repaired code** that prevents forced-browsing attacks





# FixMeUp workflow

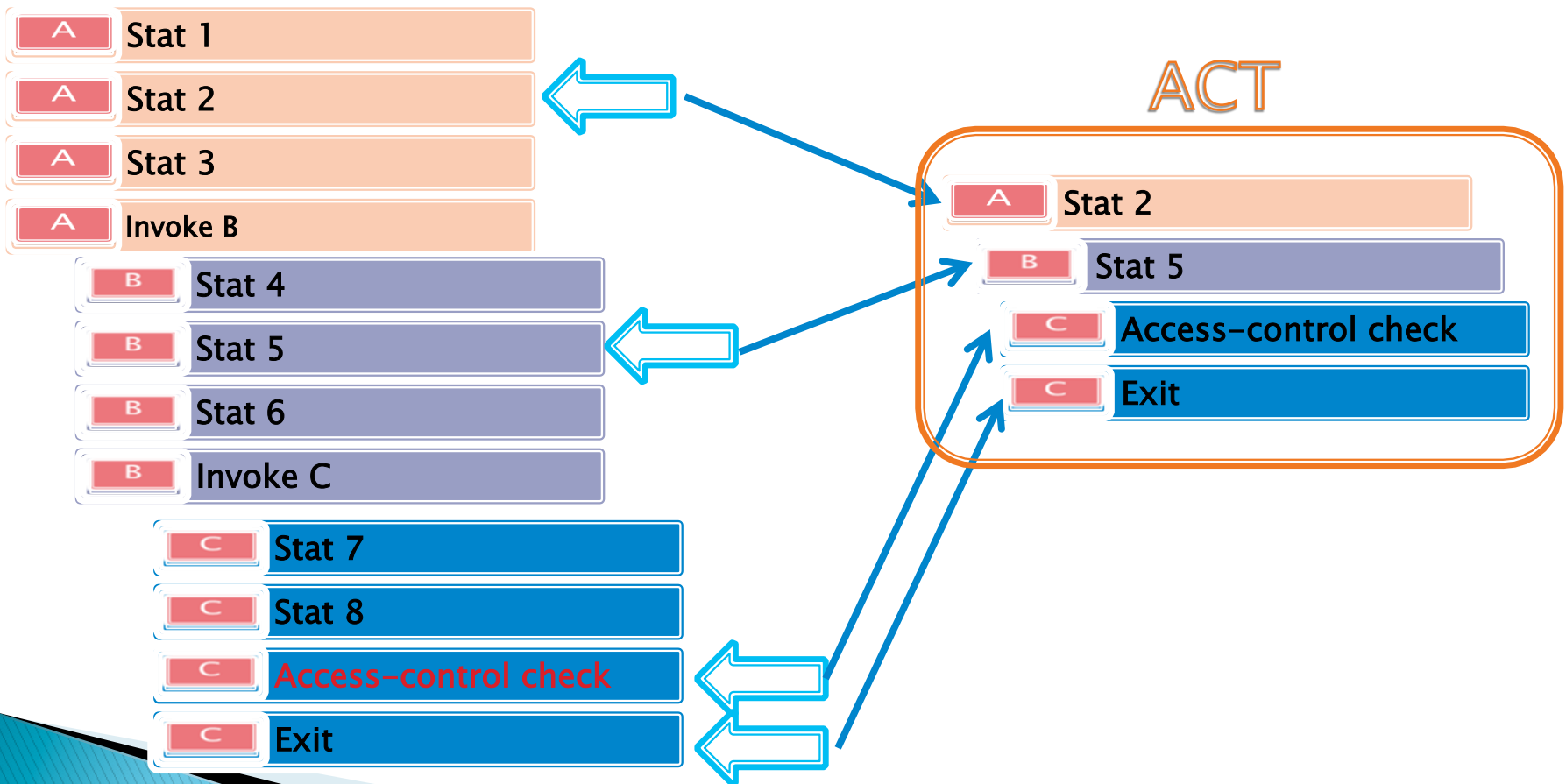


# 1. Example of access-control logic

```
lockSession();
if(!empty($_SESSION['name']) && !empty($_SESSION['pass'])) {
    .....
    $logged = @mysql_affected_rows();
}
if($logged !== 1 && !empty($_COOKIE[COOKIE_USER]) && !empty(
$_COOKIE[COOKIE_PASS])) {
    .....
    $logged = @mysql_affected_rows();
}
if($logged !== 1) {
    unlockSessionAndDestroyAllCokies();
    sleep(5);
    header('Location: '.QUERY_STRING_BLANK.'login');
    die();
}
```

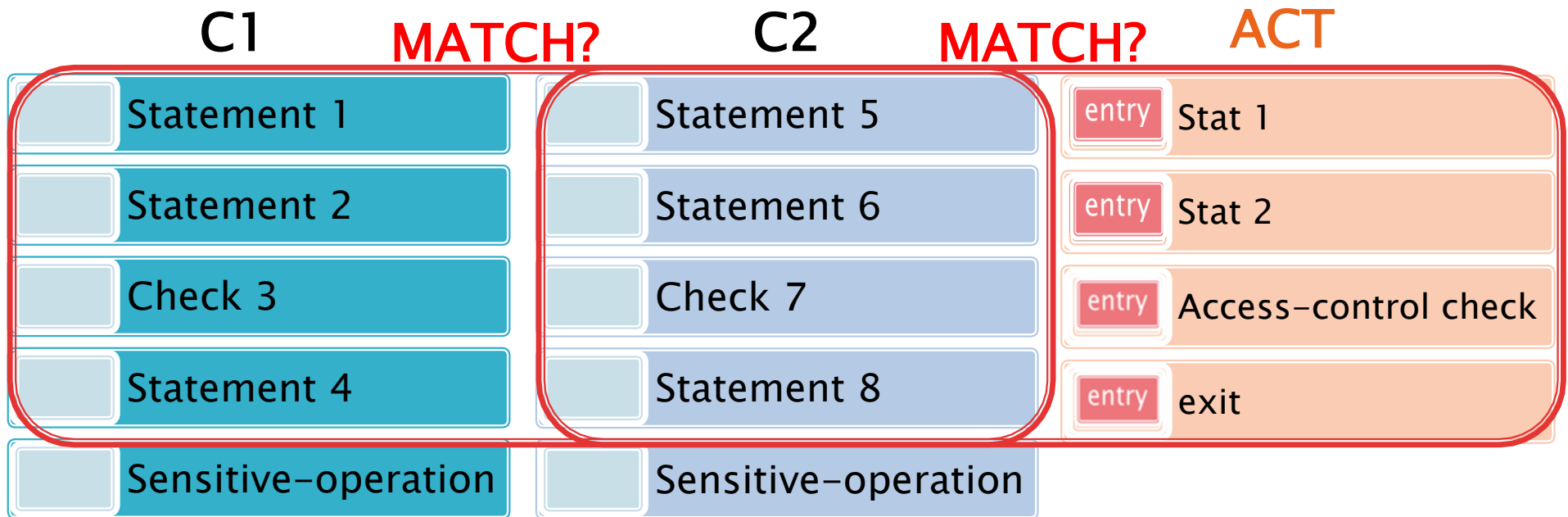
# 2. Access-control template

## ► Compute an ACT

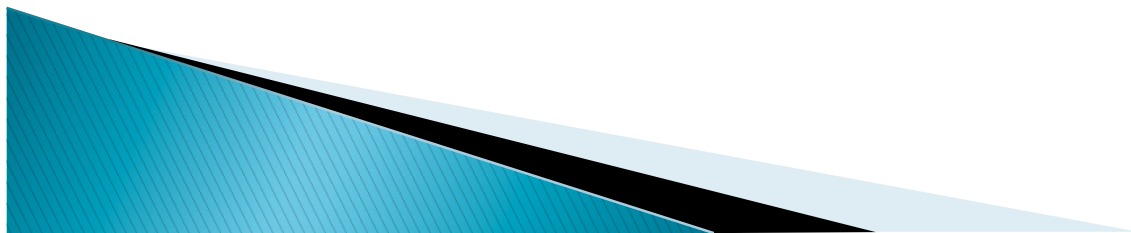




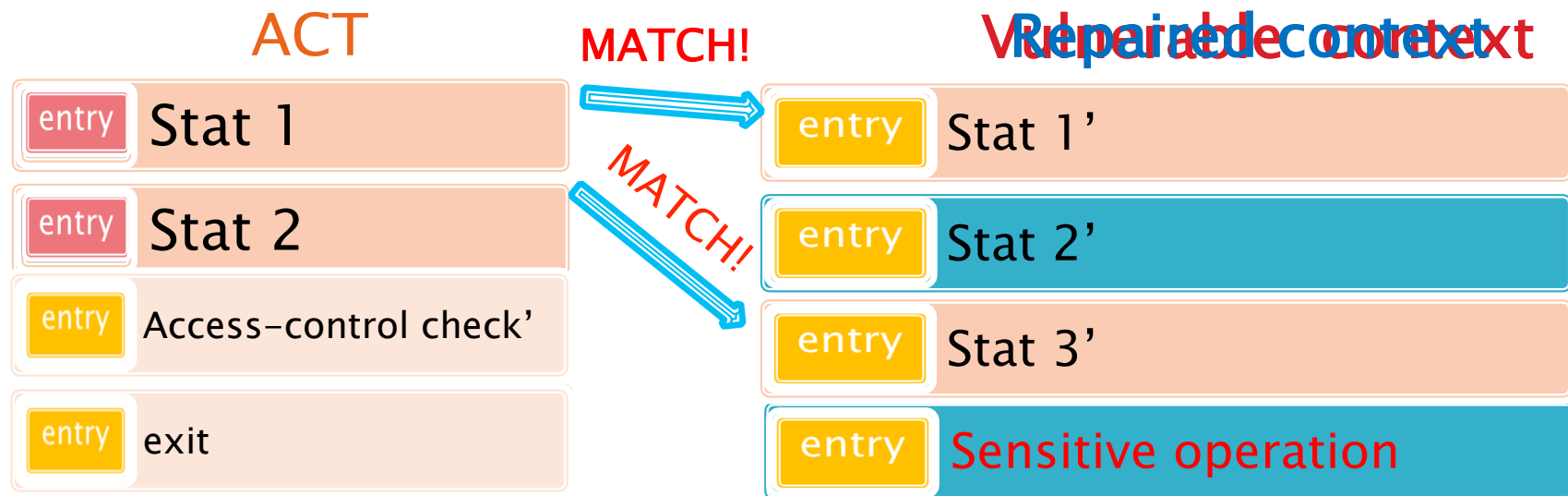
### 3. Apply ACT to fix the bugs



- ▶ Finds vulnerable contexts that do not implement the same logic as the ACT



# 3. Apply ACT to fix the bugs (2)

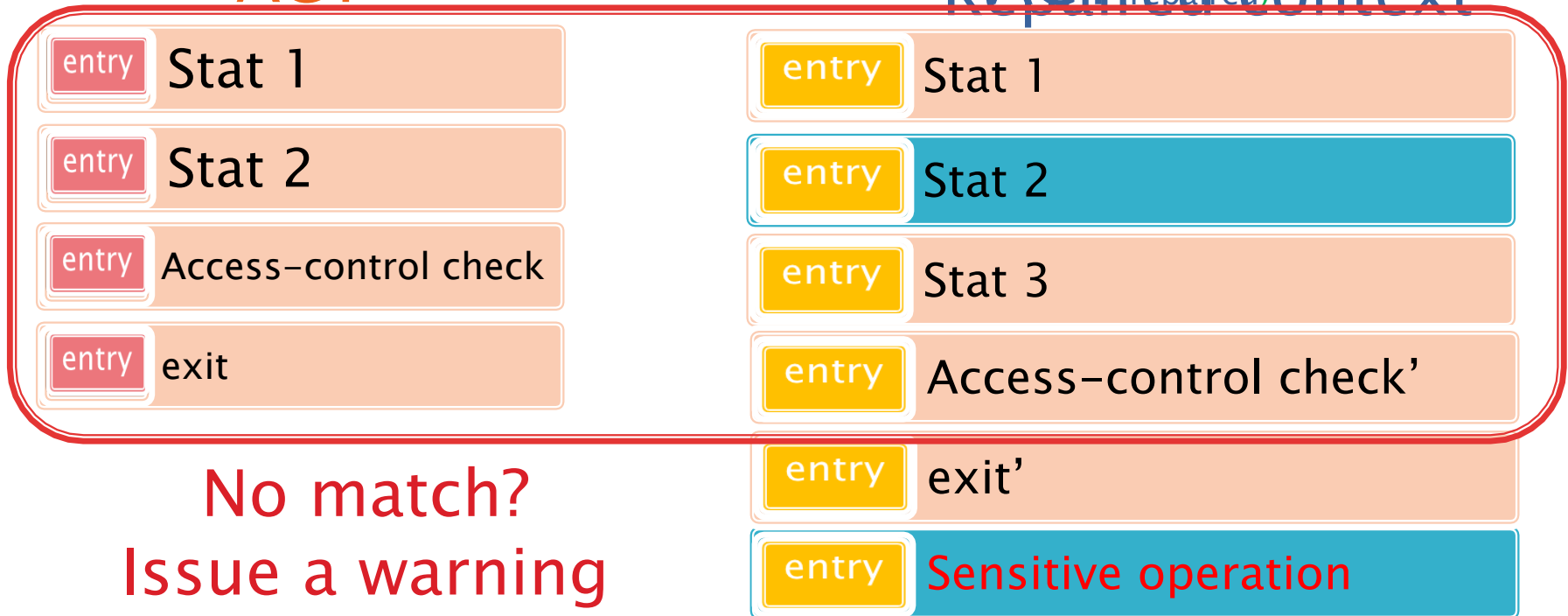


- ▶ Replicates ACT into the vulnerable context while **reusing already existing statements**

# 4. Validate repairs

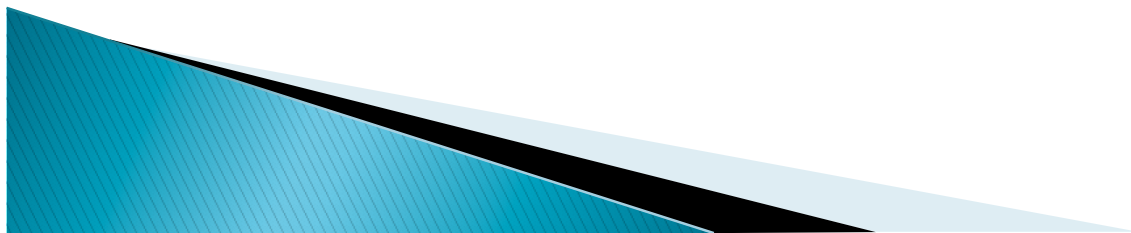
ACT

Repaired context  
Repaired ACT



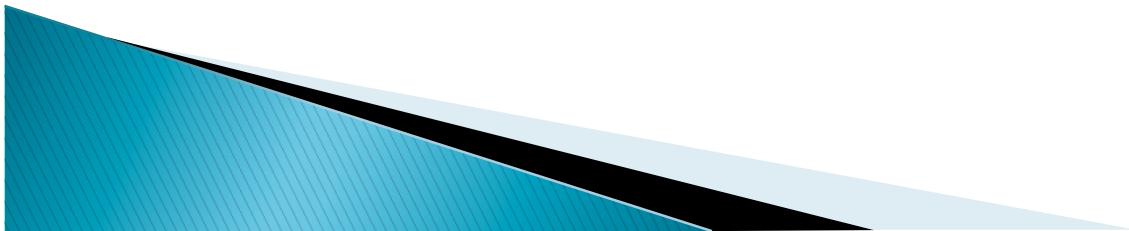
No match?  
Issue a warning

- ▶ Recompute ACT - should be the same as before!



# Evaluation

- ▶ 10 open-source interactive PHP server apps
- ▶ Generated 38 repairs
  - 31 correct
  - 7 in addition to already existing access-control logic
- ▶ 28 partial repairs
  - Reusing existing statements is important!
- ▶ 1 warning
- ▶ 1 unwanted side effect



# Evaluation (2)

```
include('class/common.php') ; // [FixMeUp repair]
$GR_newone = new COMMON( ) ; // [FixMeUp repair]
if (( $ SESSION [ ' no ' ] != 1)) { // [FixMeUp repair]
    $GR_newone ->error( ' Require admin priviledge' , 1 , 'CLOSE' ) ; //
[FixMeUp repair]
}
```

```
include('class/common.php') ; // existing statement
$GR = new COMMON( ) ; // existing statement
if (( $ SESSION [ ' no ' ] )) { // [FixMeUp repair]
    $GR->error( ' Require login procedure' ) ; // [FixMeUp repair]
}
```

.....

```
//@SSO( 'member')
```

```
@fwrite($tmpfs, $saveResult);
```



# Evaluation (3)

- ▶ Warning: after applying the ACT, repaired code does not implement the same logic as the ACT

Program Entry

```
|- include 'conf.php' ;
```

```
session_start(); //existing statement
```

```
...
```

```
if ($confirm==" " ) {
```

```
}else if( $confirm== "yes" ) {
```

```
dbConnect ( ) ; // existing statement
```

```
if ( !verifyuser ( ) ) // [FixMeUp repair]
```

```
{
```

```
header('Location: ./login.php');//[FixMeUp repair]
```

```
exit; // [FixMeUp repair]
```

```
}
```

```
$sql = "DELETE FROM blogdata WHERE postid = $postid";
```

```
$query = mysql_query( $sql ) or die( "Cannot query the database .<br>" .
```

```
mysql_error() );
```

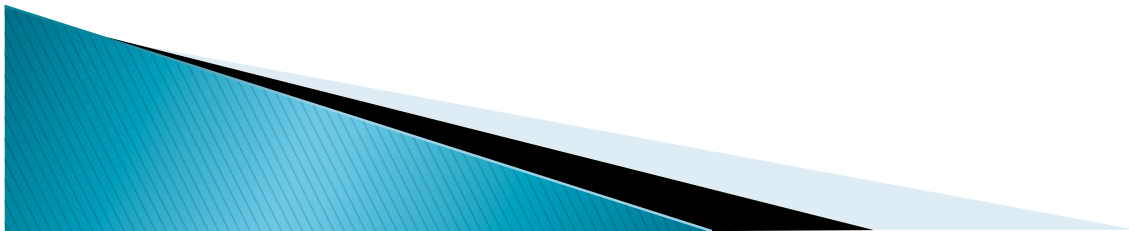
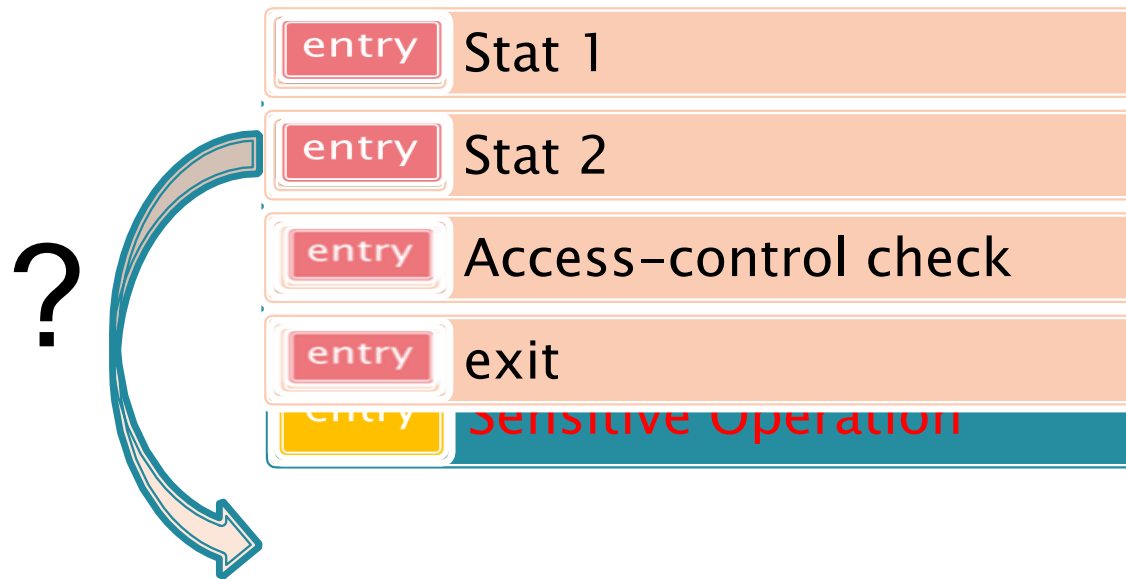
```
.....
```

```
{
```



# Limitations

- ▶ Environmental data dependencies, eval
- ▶ Unwanted side effects



# Avoiding unwanted side effects

- ▶ Use fresh variable names

entry \$local\_var\_1 = session\_id()



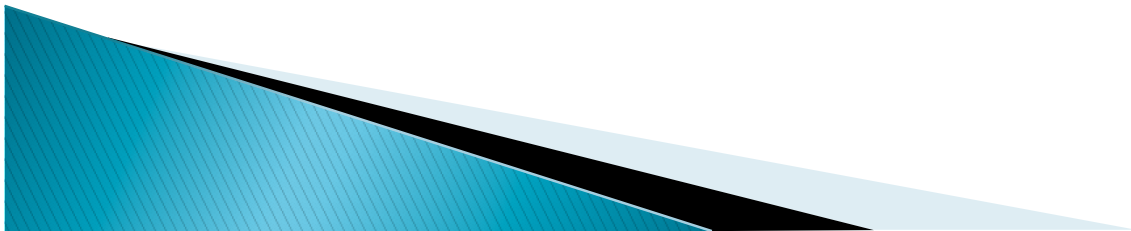
entry \$local\_var\_1\_new = session\_id()

- ▶ Do not replicate already existing statements

|                            |                           |
|----------------------------|---------------------------|
| entry session_start()      | entry session_start()     |
| entry include "a.php";     | entry include "a.php";    |
| entry Access-control check | entry                     |
| entry exit                 | entry Sensitive Operation |

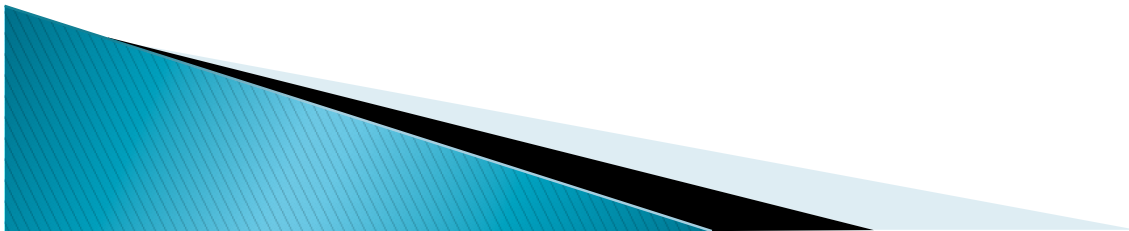
# Related work

- ▶ Static detection of access-control bugs
- ▶ Dynamic detection of access-control bugs
- ▶ Dynamic repair of software bugs



# Conclusion

- ▶ FixMeUp computes code templates for access-control logic from examples
- ▶ Finds and repairs access-control bugs in PHP applications
  - Reuses existing statements
  - Avoids introducing unwanted dependences
- ▶ Successfully repaired 30 access-control bugs in 10 real-world PHP applications





Q & A

Thank you

