



# WAP – Web Application Protection

Ibéria Medeiros, Nuno Neves, and Miguel Correia

nuno@di.fc.ul.pt

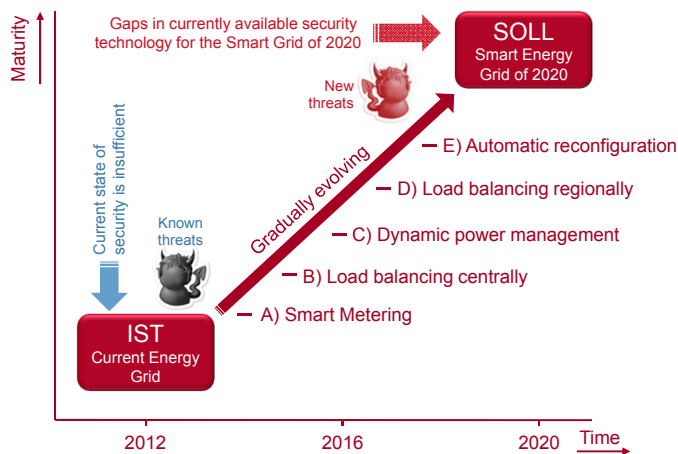


1

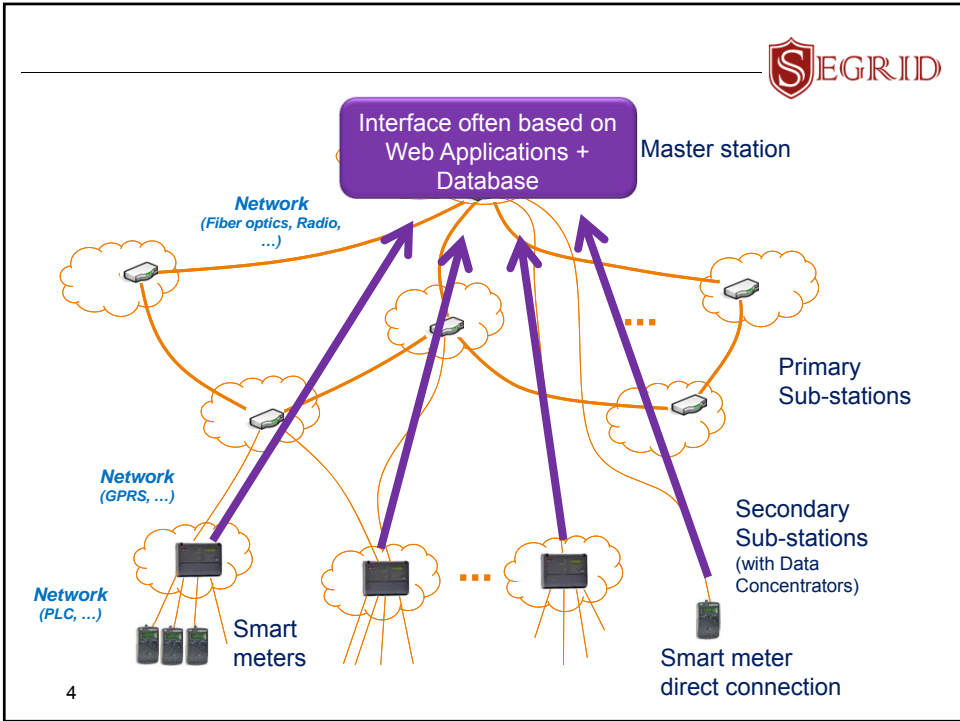
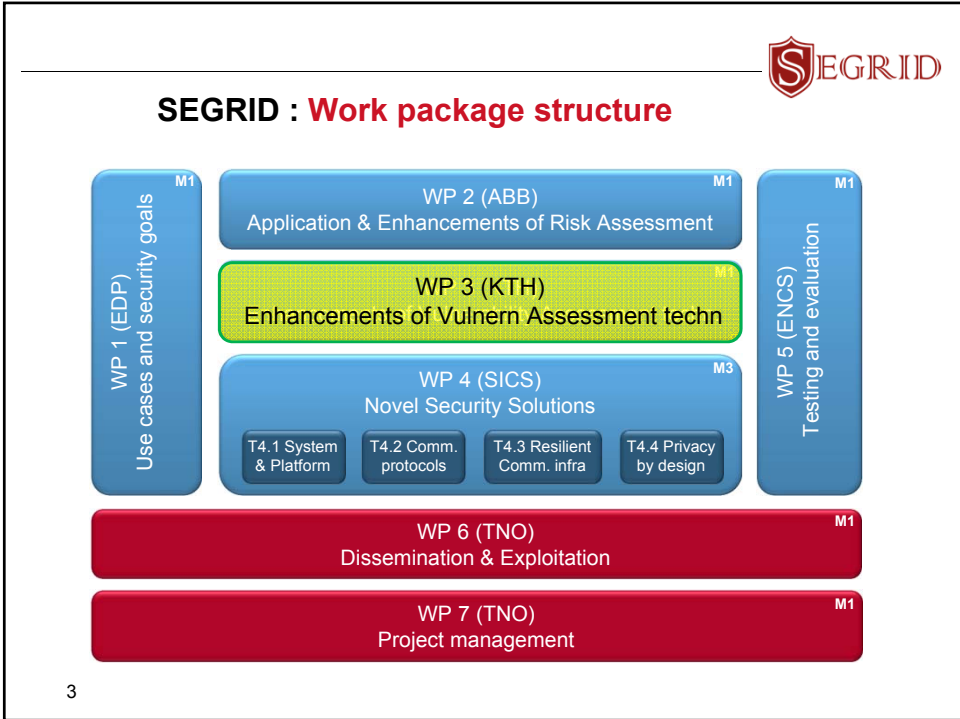



## SEGRID

### › Storyline




2





---



## WAP (*Web Application Protection*)

**Analysis**

- › searches for candidate input validation vulnerabilities in the source code of a PHP web application

**Prediction**

- › predicts if a candidate vulnerability is a false positive or a real vulnerability


**Correction**

- › inserts fixes in the source code to remove the vulnerabilities

**Feedback**

- › reports the real vulnerabilities detected and how they were corrected
- › outputs a corrected version of the web application
- › reports the false positives predicted

5



---

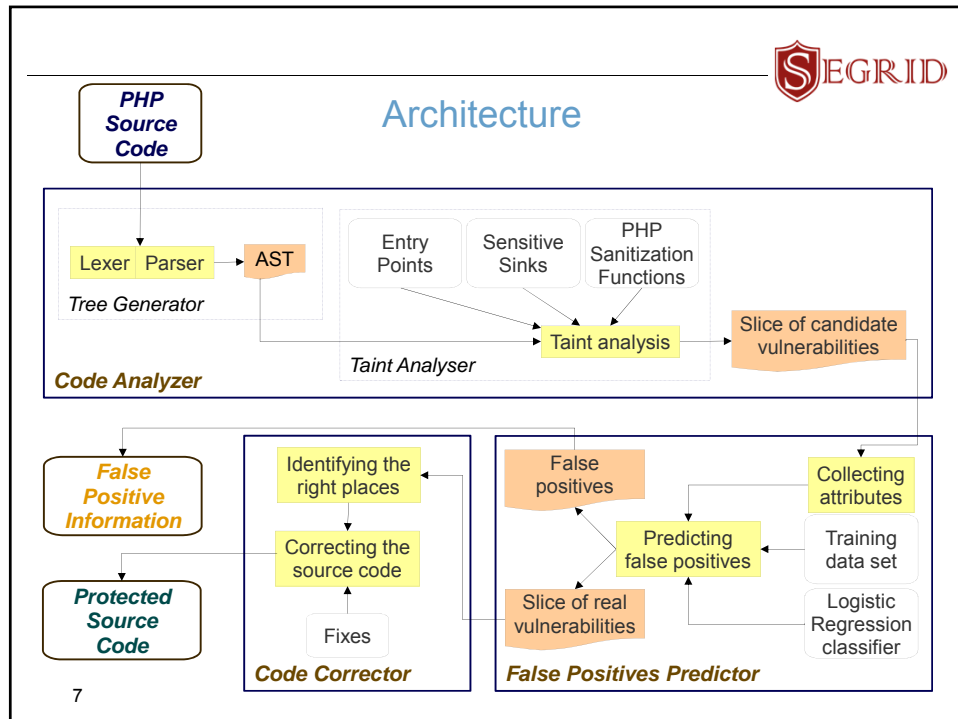
## Vulnerabilities Detected

<p><b>Most exploited:</b></p> <ul style="list-style-type: none"> <li>- <b>SQL Injection</b></li> <li>- <b>Cross Site Scripting (XSS)</b></li> </ul>	<p><b>Others:</b></p> <ul style="list-style-type: none"> <li>- Remote file inclusion</li> <li>- Local file inclusion</li> <li>- Directory Traversal / Path Traversal</li> <li>- Source code disclosure</li> <li>- OS command injection</li> <li>- PHP code injection</li> </ul>
---	---

*Track the user inputs and verify if they reach critical functions, then we could detect vulnerabilities...*

↳ **Focus on PHP**  
**Static analysis ...Taint Analysis**

6



## Working ... output



```

+ + + Type of Analysis: SQLI
  > Summary:
  - Time of analysis: 00:00:01 H
  - Number of vulnerabilities found: 1
  - Number of vulnerable files: 1
  - List of vulnerable files:
    /home/user/example.php

= = = = Vulnerability n.: 1 = = = =
> Vulnerable code:
1: $a = $_GET['user'];
9: $b = $_POST['pass'];
10: $query = "SELECT * FROM users WHERE u = '$a' AND p = '$b'";
11: $r = mysql query($query);

> Corrected code:
1: $a = mysql_real_escape_string($_GET['user']);
9: $b = mysql_real_escape_string($_POST['pass']);

```

8

## Experimental Setting



Web application	Files	Lines of code	Analysis time (s)	Vuln. files	TA vulner. found	False positives	Real vulner.
adminer-1.11.0	45	5,434	27	3	3	0	3
Butterfly insecure	16	2,364	3	5	10	0	10
Butterfly secure	15	2,678	3	3	4	0	4
currentcost	3	270	1	2	4	2	2
dmoz2mysql	6	1,000	2	0	0	0	0
DVWA 1.0.7	310	31,407	15	12	15	8	7
emoncms	76	6,876	6	6	15	3	12
gallery2	644	124,414	27	0	0	0	0
getboo	199	42,123	17	30	64	9	55
Ghost	16	398	2	2	3	0	3
gilbitron-PIP	14	328	1	0	0	0	0
GTD-PHP	62	4,853	10	33	111	0	111
Hexjector 1.0.6	11	1,640	3	0	0	0	0
Hotelmis 0.7	447	76,754	9	2	7	5	2
Lithuanian-7.02.05-v1.6	132	3,790	24	0	0	0	0
Measureit 1.14	2	967	2	1	12	7	5
Mfm 0.13	7	5,859	6	1	8	3	5
Mutilidae 1.3	18	1,623	6	10	19	0	19
Mutilidae 2.3.5	578	102,567	63	7	10	0	10
NeoBill0.9-alpha	620	100,139	6	5	19	0	19
osvsn0.2	4	243	1	0	0	0	0

↳ Tested 45 open source packages with 6.700 files with 1.380.000 lines, where 431 vulnerabilities were found (with at least 43 false positives)

9

## WAP Detection for SQLI / XSS



Just the taint-analysis part of WAP

Full WAP

Webapp	WAP-TA			Pixy				WAP (complete)		
	SQLI	XSS	FP	SQLI	XSS	FP	FN	SQLI	XSS	Corrected
CurrentCost	3	4	2	3	5	3	0	1	4	5
DVWA 1.0.7	4	2	2	4	0	2	2	2	2	4
emoncms	2	6	3	2	3	0	0	2	3	5
Measureit 1.14	1	7	7	1	16	16	0	1	0	1
Mfm-0.13	0	8	3	0	10	8	3	0	5	5
Mutilidae 2.3.5	0	2	0	-	-	-	-	0	2	2
SAMATE	3	11	0	4	11	1	0	3	11	14
Vicnum15	3	1	3	3	1	3	0	0	1	1
Wackopicko	3	5	0	-	-	-	-	3	5	8
ZiPEC 0.32	3	0	1	3	7	8	0	2	0	2
Total	22	46	21	20	53	41	5	14	33	47

WAP-TA detected 5 extra vuln than Pixy

WAP-TA had an accuracy of 69% and Pixy had 44%

WAP had an accuracy of 92.1%

10

### WAP for Several Vuln Classes

Webapp	Detected taint analysis							Detected data mining	Corrected
	SQLI	RFI, LFI DT/PT	SCD	OSCI	XSS	Total	FP		
currentcost	3	0	0	0	4	7	2	5	5
DVWA 1.0.7	4	3	0	6	4	17	8	9	9
emomcms	2	0	0	0	13	15	3	12	12
Measureit 1.14	1	0	0	0	11	12	7	5	5
Mfm 0.13	0	0	0	0	8	8	3	5	5
Mutillidae 2.3.5	0	0	0	2	8	10	0	10	10
OWASP Vicnum	3	0	0	0	1	4	3	1	1
SRD <sup>(1)</sup>	3	6	0	0	11	20	1	19	19
Wackopico	3	2	0	1	5	11	0	11	11
ZiPEC 0.32	3	0	0	0	4	7	1	6	6
<b>Total</b>	<b>22</b>	<b>11</b>	<b>0</b>	<b>9</b>	<b>69</b>	<b>111</b>	<b>28</b>	<b>83</b>	<b>83</b>

A significant number of vulnerabilities were detected

A subset was considered as false positives

The remaining ones were all corrected

11

## Thank you! Any questions?

This was:

# WAP – Web Application Protection

SITE with tool: <http://awap.sourceforge.net/>

Publications:

- I.Medeiros, N.Neves, M.Correia, *Automatic Detection and Correction of Web Application Vulnerabilities using Data Mining to Predict False Positives*, International World Wide Web Conference (WWW), April 2014
- I.Medeiros, N.Neves, M.Correia, *Detecting and Removing Web Application Vulnerabilities with Static Analysis and Data Mining*, under submission to a journal

12