Security despite of lazy developers – possible or not?

By

Andrzej Kleśnicki

OWASP

The Open Web Application Security Project

About laziness

Laziness (also called indolence) is a disinclination to activity or exertion despite having the ability to do so.

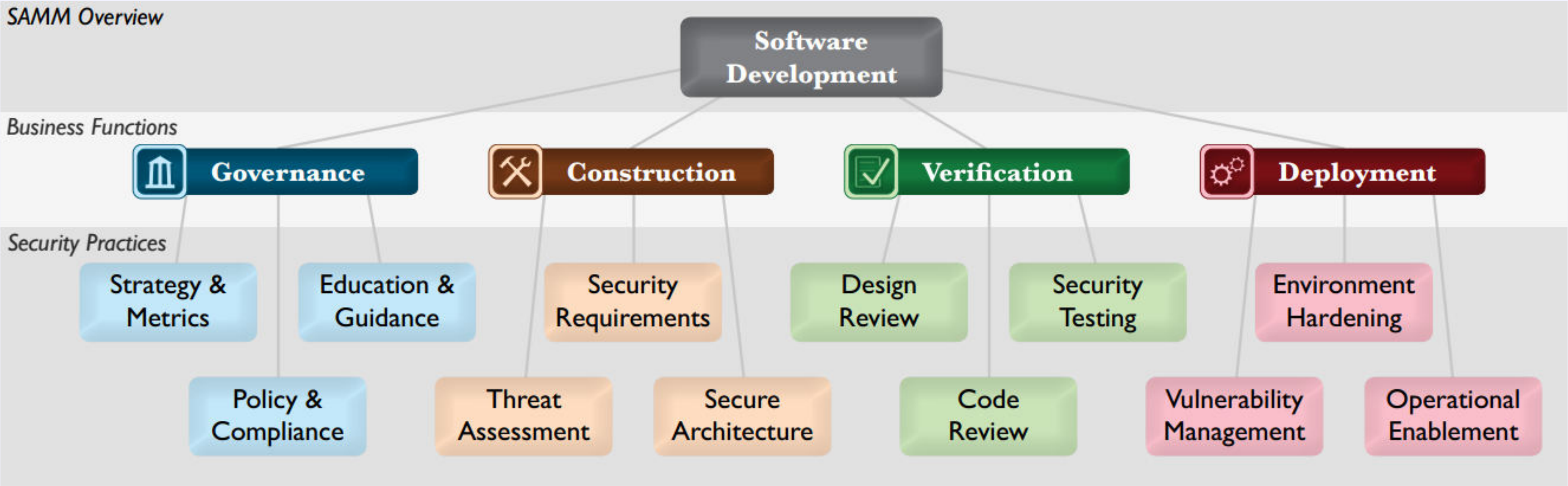
By this definition – developers know how to create secure application – they just do not want to.

True or False?

So why they don’t?

Vulnerable application is not a failure of single developer but whole process of creating solution since first business requirement

Maturity



SAMM Overview

Software Development

Business Functions

Governance

Construction

Verification

Deployment

Security Practices

Strategy & Metrics

Education & Guidance

Policy & Compliance

Security Requirements

Threat Assessment

Secure Architecture

Design Review

Security Testing

Code Review

Environment Hardening

Vulnerability Management

Operational Enablement

The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization. The resources provided by SAMM will aid in:

Reality

- No governance or requirements
- No security design
- Lack of security in QA
- Deployment – fast and cheep

How to live?

- If we can not fight it - how to solve it?

Hygiene

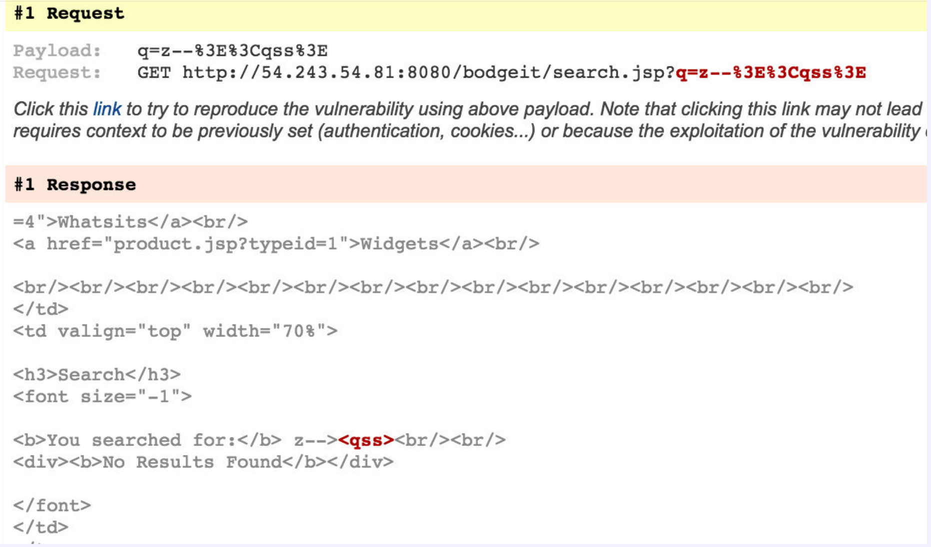
- Secure the environment
- Implement secure configuration
- Learn your surroundings
- Check for known vulnerabilities

- How (examples only):
  - Nmap, OpenVAS, How To, Man pages, Friends ,
  - Qualys ;)

Detect

OWASP
The Open Web Application Security Project

- Perform DAST / SAST



#1 Request

```
Payload:    q=z--%3E%3Cqss%3E
Request:    GET http://54.243.54.81:8080/bodgeit/search.jsp?q=z--%3E%3Cqss%3E
```

Click this link to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability

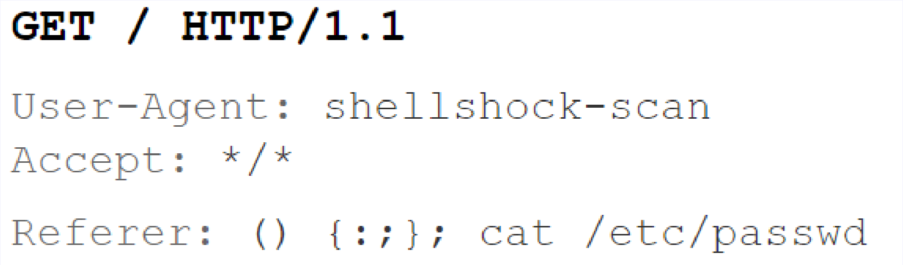
#1 Response

```
=4">Whatsits</a><br/>
<a href="product.jsp?typeid=1">Widgets</a><br/>

<br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/>
</td>
<td valign="top" width="70%">

<h3>Search</h3>
<font size="-1">

<b>You searched for:</b> z--><qss><br/><br/>
<div><b>No Results Found</b></div>

</font>
</td>
```

- How (examples only)
  - https://www.google.pl/search?q=open+source+DAST
  - Qualys?

Protect

- Implement WAF

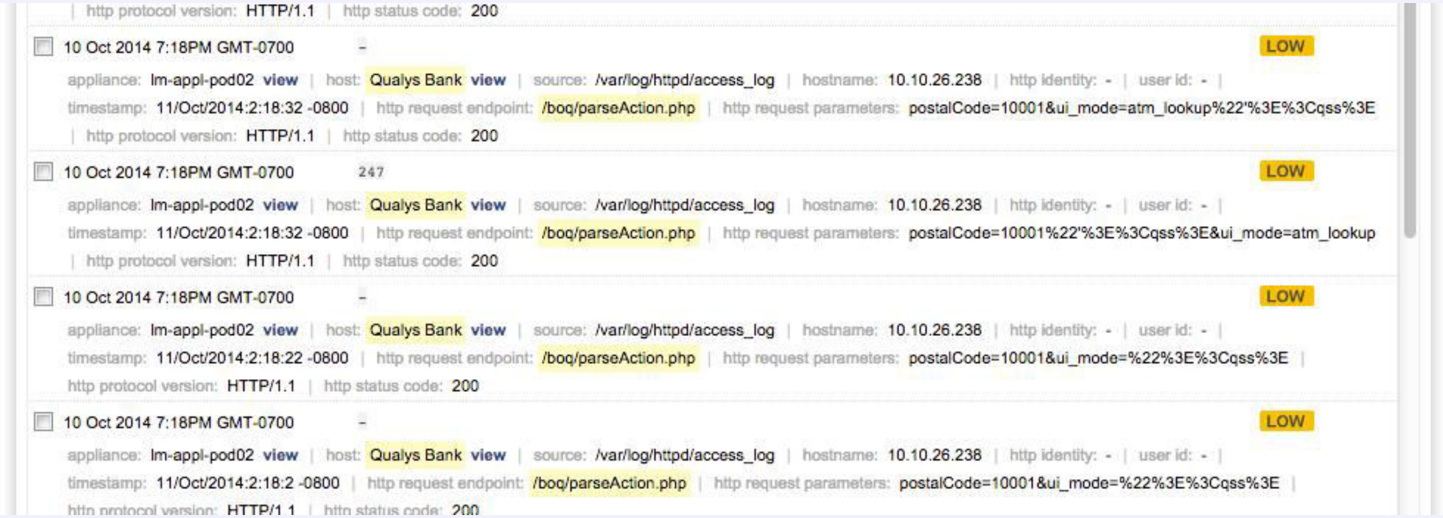


```
GET / HTTP/1.1
User-Agent: shellshock-scan
Accept: */*
Referer: () {:;}; cat /etc/passwd
```

- How (examples only):
  - IronBee
  - Qualys

Monitor

- Look into logs and find suspicious behavior



```
| http protocol version: HTTP/1.1 | http status code: 200
□ 10 Oct 2014 7:18PM GMT-0700        -                                                              LOW
  appliance: lm-appl-pod02 view | host: Qualys Bank view | source: /var/log/httpd/access_log | hostname: 10.10.26.238 | http identity: - | user id: - |
  timestamp: 11/Oct/2014:2:18:32 -0800 | http request endpoint: /boq/parseAction.php | http request parameters: postalCode=10001&ui_mode=atm_lookup%22'%3E%3Cqss%3E
  | http protocol version: HTTP/1.1 | http status code: 200
□ 10 Oct 2014 7:18PM GMT-0700        247                                                            LOW
  appliance: lm-appl-pod02 view | host: Qualys Bank view | source: /var/log/httpd/access_log | hostname: 10.10.26.238 | http identity: - | user id: - |
  timestamp: 11/Oct/2014:2:18:32 -0800 | http request endpoint: /boq/parseAction.php | http request parameters: postalCode=10001%22'%3E%3Cqss%3E&ui_mode=atm_lookup
  | http protocol version: HTTP/1.1 | http status code: 200
□ 10 Oct 2014 7:18PM GMT-0700        -                                                              LOW
  appliance: lm-appl-pod02 view | host: Qualys Bank view | source: /var/log/httpd/access_log | hostname: 10.10.26.238 | http identity: - | user id: - |
  timestamp: 11/Oct/2014:2:18:22 -0800 | http request endpoint: /boq/parseAction.php | http request parameters: postalCode=10001&ui_mode=%22%3E%3Cqss%3E |
  http protocol version: HTTP/1.1 | http status code: 200
□ 10 Oct 2014 7:18PM GMT-0700        -                                                              LOW
  appliance: lm-appl-pod02 view | host: Qualys Bank view | source: /var/log/httpd/access_log | hostname: 10.10.26.238 | http identity: - | user id: - |
  timestamp: 11/Oct/2014:2:18:2 -0800 | http request endpoint: /boq/parseAction.php | http request parameters: postalCode=10001&ui_mode=%22%3E%3Cqss%3E |
  http protocol version: HTTP/1.1 | http status code: 200
```

- How (examples only)
  - By hand?
  - Splunk, Sawmill, Graylog2 etc .
  - Maybe Qualys? :D

Vision

- Scenario 1
  - We detect ShellShock attempt in HTTP request, knowing that host in vulnerable we block it, but also checking in logs to see backlog of 24h requests from same IP
- Scenario 2
  - We detect XSS possible vector using DAST, as result modyfying WAF to protect it right away, tune up Log monitoring to find IOC

ETC…

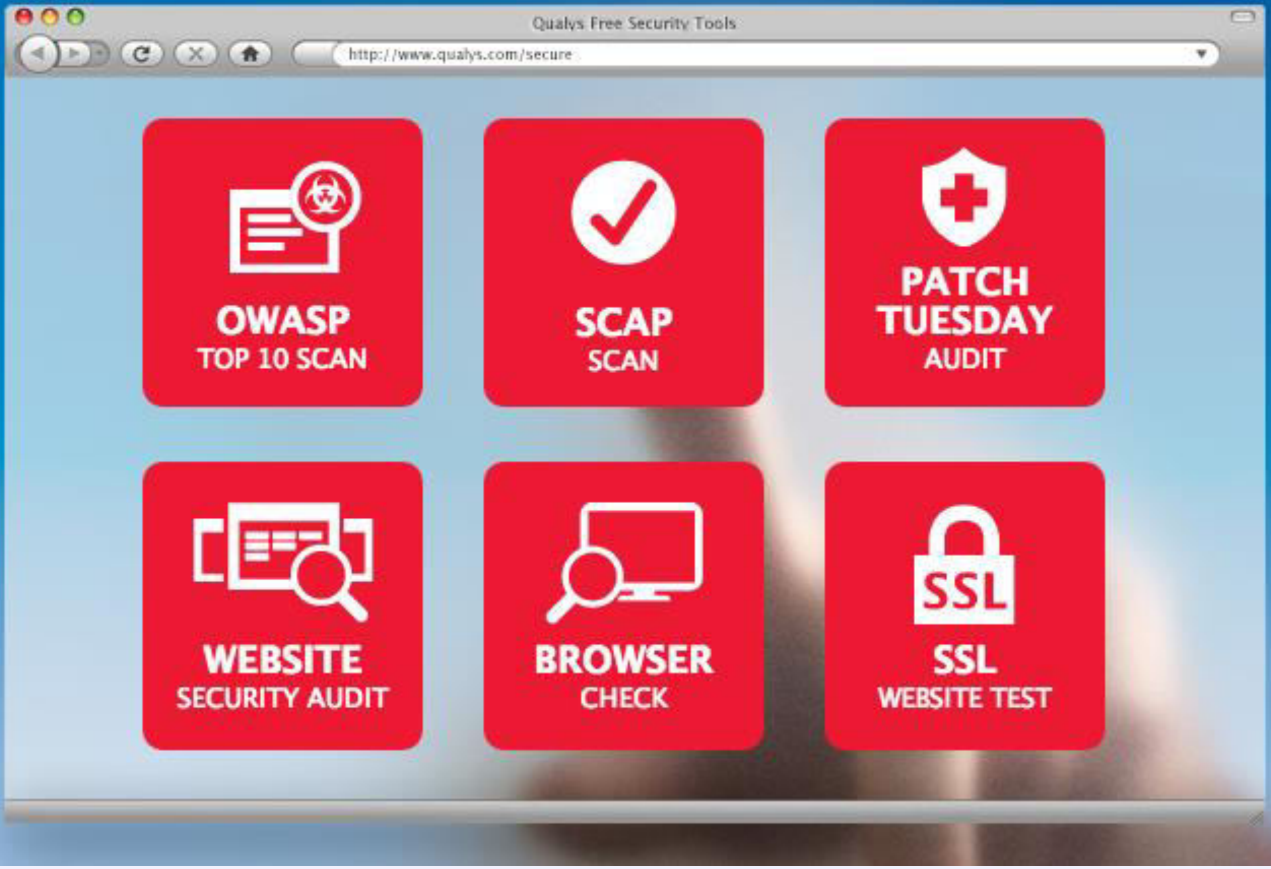
About Me

- Andrzej Kleśnicki
  - @: aklesnicki@qualys.com
  - Security Expert
  - pl.linkedin.com/in/klesnicki/
- Qualys
  - Qualys.com
  - Cool company with cool services
  - OWASP supporter
  - OpenSource contributor

Security at Your Fingertips - FREE

Make sure your systems are secure and compliant.

Select a free tool below. There's nothing to download!



www.qualys.com/secure