Security of Outsourcing of Software Development

Aziz Ahmad Rais, Rudolf Pecinovsky
University of Economics, Prague, Faculty of Informatics and Statistics,
Department of Information Technologies
W. Churchill Sq. 4, 130 67 Prague 3
arais@seznam.cz, rudolf@pecinovsky.cz

Abstract. Security is crucial for all information systems that provide business service. Many systems do not solve security because they rely on firewalls and network securities or security is ignored because of lock of kills or lock of requirements. When outsourcing software development it can easily happen that security is left out from information system scope because of lack of resources and skills. This article will show most types of security vulnerabilities and describes how to check software against security requirements.

1 Introduction

Security is large topic and it is always difficult to understand some security vulnerability because they are difficult to simulate. Further, most software developers rely on frameworks and during development they are mostly concentrated to think about designing solution and write code. Information system architects are usually also specialized in software architecture and rely on security specialists. This results in that some customers do not employ security specialist and let non-specialist to do such specialized work. These and others causes affect that the software developed through outsourcing have different security vulnerabilities. It is also important to think about the design of the security components of software because It can happen that the some security components are tightly coupled into the information system or developed so that they are not flexible to implement any new changes or update them. All these issues arise questions e.g. how to assure that the system is secure, the data are secured and only authorized people have access to system to perform and execute a business process or manipulate the customer and business data.

1.1 Vulnerabilities and attacks

Client-side Attacks.

Client side attacks usually happen when unauthorized person injects some scripts in the pages of the web site that you visit. These types of attacks are grouped and each type of attack is named. Here will be named a few of them like:

Content Spoofing: this is about injecting a malicious content into a web site, the content can be of type text, URL, html tag etc. for example

```
http://example.org/news?title=some+title+of+the+news
```

the attacker changes the title.

```
http://example.org/pages?pg=http://example.news.org/pr/abcd.html
```

the attacker changes to

```
http://example.org/pages?pg=http://TheAttackersite.news.org/pr/attacker.html
```

Cross-Site Scripting **(XSS, CSS):** It is about injected a script into the trusted web sites. In other words an attacker uses a web application to send malicious code to a different end user. There are three categories of cross site scripting is:

- Reflected Cross-Site Scripting: non-persistent script

  ```
  http://example.org/index.php?user=<script>alert("test 123")</script>
  ```

- **Persistent Cross-Site Scripting**: persistent script

  ```
  <input type="text" name="email" value="aaa@aa.com"> ….
  ```

  injects the script

  ```
  aaa@aa.com"><script>alert(document.cookie)</script> ….
  ```

- **DOM-Based Cross-Site Scripting**: These attacks happened when a website pages is using HTML DOM. Usually the attackers uses the properties or function of objects like Windows, documents (properties: location, URL) and so on.

Cross-Frame Scripting (CFS): This XSS attack by using `iframe` tag of HTML. The CFS can be used for XSS, Cross-Site Request Forgery (CSRF) and Phishing.

- Phishing: The attacker web site is identical to trusted identity. Links to these site can be sent by emails, chats etc. usually it happens on financial websites e.g. banking

Cross-Site Request Forgery: This about that end user to execute unwanted actions on a web application example: clicking on link send by chat/email.

Further similar attacks can happen on technologies like ActionScript, Applets, ActiveX and so on please see for details [2] and [4].

Command attacks

Command attacks happen when an attacker injects your system with malicious code and your system is executing it as system code.

LDAP Injection: This is about when creating an LDAP statement based on user inputs from web site. E.g. user input:

```
<input type="text" size=20 name="userName">
```

Insert the `username</input>` and LDAP query:

```
String ldapSearchQuery = "(cn=" + $userName + ")";
```

hacker can insert * as user name and system can return list of users.

OS Command Injection: This attack can happen when application executes operation system command based on attacker's inputs from web site.

SQL Injection: When attacker uses web sites input fields and access database. Example is building SQL query from user inputs:

```
"SELECT * FROM `userinfo` WHERE `id` = " + uer_id + ";"
```

XPath Injection: XPath addresses part of the XML document. This injection can be used with AJAX calls and when XML is returned.

HTTP Header Injection / Response Splitting: These are vulnerabilities that when user-input data is copied into a response header in an unsafe way. This makes it possible for an attacker to inject newline characters into the header, inject new HTTP headers.

Local File Includes: The attacker uses your local directory structure that is available via web site to brows.

Potential Malicious File Uploads.

Authentication

Your system is vulnerable when there is missing a mechanism locking the user account. This vulnerability gives to an attacker possibility to endlessly try to guess user credentials. Here are a few examples of authentication vulnerabilities that give us idea of to not underestimate the authentication security and think about further attacks and prevent them before they happen.

Different Login Failure Message, Valid vs. Invalid User name or password: If systems provide different messages for different login attempts that help hackers to recognize invalid user and password.

Insufficient Authentication: If web site has not correctly secured the content access, and hacker can access it without authentication.

Weak Password Recovery Validation: is when a web site permits an attacker to illegally obtain, change or recover another user's password and web site allow user to select password that can be easily deduce.

Lack of SSL on Login Pages: If the user password and user id is sent to server as plain text, in other words there is no SSL or other cryptographic methods used.

Auto-complete not Disabled on Password Parameters: The web site should disable the auto complete on user id and password to prevent storing them into browser.

Insufficient Authorization

HTTP Verb Tampering: This happens in combination with XSS, SQL injection to bypass authentication and authorization.

Insufficient authorization can cause that unauthorized people can access to the high-privilege content and logged with low- privilege account.

Session management

System session is usually used for storing information about authorized user's roles and access rights. Based on session server recognize who is the caller how to authorize him. So if system session of the user is not managed correctly it can easily become vulnerability of the system that is used by hackers to attack your system.

Session Fixation: Is about to use the session id of other person to get access without authentication to the system resources.

Failure To Generate New Session ID After Login: When system doesn't invalidate the user session and create new one.

Permissive Session Management: If session duration is long and is not invalidate after leaving the system.

Session Token Passed In URL: Managing session by using only adding token URL is not secure, because there is no way to additional security parameters automatically.

Session Cookie Not Set With Secure Attribute: If Cookies has no additional security attribute that should be encrypted.

Session Cookie Not Set With HttpOnly Attribute: HttpOnly is an additional flag included in a Set-Cookie HTTP response header.

Session Cookie Not Sufficiently Random: Non-random generated cookies can be deduced.

Other attacks

I think it is worth to mention that attacking a system doesn't happen only through scripts injection, but can happen through network and other media too see for details [3].

Here is mentioned only a few network attack that usually are combined with scripting attacks by hackers.

Packet sniffers: Listening to a port where the communication between two parties occurs and sniff the TCP packets. These types of tack can be done by protocol analyzers tools for example Wireshark.

Port scans: There are network tools that allow you to scan what ports are open and what application is listening to that port.

Ping sweeps: Before listening to a port you need to have the IP address of the server, this can be identified by ping sweep. Ping sweep is a technique that allows determining which of a range of IP addresses map to live hosts (computers).

Brute-force attack: A hacker uses weakness of missing lock mechanics on account. An attacker is using to guess the password to get or hacker can use a dictionary of words.

Trust Exploitation: This exploitation is about if your system has trusted relation to a system outside of your LAN. This relation is used by a hacker is used to attack your internal system if the external system is not secure enough.

Port forwarding: This similar to trust exploitation, the attacker uses the publicly available and unsecured service. For example attacker uses the port 22 of unsecured publicly available server and communicates to port 23 of the server in secure zone.

IP Spoofing Attacks: These attacks happen by falsifying the source IP address to appear to be another user. An attacker changes the IP packet header that contains information about the source address and destination. This way an attacker presents him as trusted resource. For this types of attacks hackers usually use protocol analyzers.

2 SOLUTION AND TOOLS

Solution for making your information system secure is security requirements and security testing. Security requirements can be insipid by the above example attacks and in order to execute any security testing you need a security methodology that shows you what to secure in your information system and how to secure the information system, the data and the network.

As security methodology here would be used and references that OSSTMM - Open Source Security Testing Methodology Manual see [1].

One of the important automated security testing is penetration testing, and there are available free software and vulnerability test databases that you can use to find out security holes of your system and then cover the security problem.

Penetration tests: A penetration testing is method of evaluating the limitation of information system or network by simulating attacks. Penetration testing is occasionally called pen-test. A penetration tests can be performed black box, white box or gray box. But the most popular way of penetration testing is black box testing. A third-party pen-tester can conduct penetration testing.

Network and application security scanner tools

Nessus: Nessus is Network security vulnerability scanner as well as web application vulnerability scanner. Nessus scanners can be used throughout an entire enterprise, inside DMZs and across physically separated networks. Nessus supports HTTP, TCP, UPD, SSH, NNTP, FTP, POP2, POP3 or IMAP, SNMP and other protocols and can also perform SYN scan. It supports databases: Oracle, SQL Server, MySQL, DB2, Informix/DRDA and PostgreSQL. Nessus can perform port scanning and verify whether they are open remotely.

Nessus can simulate attacks: SQL Injection, Command Execution, Cross-Site Scripting, HTTP Header Injection, Server Side Includes, Cookie Manipulation, XML Injection and so on. Nessus is highly configurable and can be extended by adding new plugins if the plugin is not part of out-of the box plugins already.

Metasploit: The Metasploit Framework is a free, open source penetration testing solution developed by the open source community and Rapid7 in Ruby programming language. Leverages fully tested and integrated public database of exploits and payloads to perform your security tests. Metasploit integrates Out-of-the-Box with NeXpose and supports scanning of: Servers, Desktops, Web Servers, Databases, Devices.

Security Auditor's Research Assistant (SARA): This is a free of charge Network scanner that supports services (tcp, udp, rpc). SARA provides Plug-in facility for third party apps and it integrates with national vulnerability database.

Nmap: Is network security scanner for discovering of hosts and service on a computer in a network. It runs on Linux, Microsoft Windows, Solaris, HP-UX and BSD. It has the following features:

- Host Discovery – Identifying hosts on a network, for example listing the hosts which respond to pings, or which have a particular port open
- Port Scanning – identifying the open ports on one or more target hosts
- Version Detection –listening to a network services on remote devices to determine the application name and version number.
- OS Detection – Remotely determining the operating system and some hardware characteristics of network devices.

Source Code Security Analyzers

When the application development is outsourced it is highly probable to have a lot of security holes in the application, so it is important to do analysis of source code before any penetration testing. The security of source code is per programming language different that is why it is difficult to provide security check rules for each computer programming language. Most security rules of each programming langue exist and it is important to apply them on source code.

PMD: scans Java source code and looks for potential problems.

C Code Analyzer (CCA): is a free tool for C programming language code analyzing.

Clang Static Analyzer: is a free tool for C and Objective-C programming language code analyzing.

SCA: Is a static code analyzer tool developed by company called Fortify (currently owned by HP). It supports programming languages like ASP.NET, C, C++, C# and other .NET languages, Java, JSP, PL/SQL, T-SQL, VB.NET.

Cryptography

In the beginning of this article were mentioned how to test to find out the application vulnerabilities and weaknesses. But the above solution doesn't solve the problems: confidentiality, privacy, integrity and non-repudiation.

To solve these problems there is need to use cryptography during communication of end-users with information systems.

Cryptography is way how to make the plain readable text difficult to read for unauthorized people. This can be achieved in different ways.

Access control

It is very important that every system has some access control component or it is integrated with system that controls the access to other systems. The access control can be divided into two types of control. Both types of control can be done by one component or system or it can be done by different components or systems.

But the most important rule is to have role based access. It means assign to every user a role and to every role access rights. So after user is authenticated then every time will be checked the authorization.

Application administration

To administrate your application you always need administrator rights. The passwords to user with administrator right, most systems application servers, database servers store in a local files. These passwords should not be stored as plain text and the same rule should be apply on passwords that applications need to access other systems.

3 SUMMARY

This article was trying to point to the security problems outsourcing of software development. At the beginning of this article were summarized most frequently security attack on software and network and later the article described how to proceed to identify the security holes.

To secure and be sure that your system is secure it is important to test your system against known attacks and do not rely on frameworks, network and check the system source code, system in runtime and most important from outside the network where the system is running and as well as from inside.

4 REFERENCES

[1] HERZOG Pete: *OSSTMM 2.1.– The Open Source Security Testing Methodology Manual*. http://isecom.securenetltd.com/osstmm.en.2.1.pdf

[2] OWASP Top 10 – 2010, The Ten Most Critical Web Application Security Risks. http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202010.pdf

[3] RUFI Antoon: *Network Security 1 and 2 Companion Guide. (Cisco Networking Academy)*, Cisco Press 2006. ISBN 1-5871-3162-5

[4] GROSSMAN Jeremiah, HANSEN Rober, PETKOV Petko D., RAGER Anton, FOGIE Seth: *XSS Attacks*, Syngress 2007. ISBN: 978-1-5974-9154-9.

[5] JIROUSEK Radim, IVANEK Jiri, MASA Petr, TOUSEK Jan, VANEK Norbert: *Principy digitální komunikace*. BEN 2007. ISBN 80-7335-084-X.