IP 127.0.0.1

Internet based penetration test

Produced by:

Name

Date

# 

# Table of contents

Legal notice 1

1. Table of contents 3

2. Management summary 5

3. Description of tested systems 6

4. Vulnerability categorization 7

4.1 Level 5 7

4.2 Level 4 7

4.3 Level 3 7

4.4 Level 2 8

4.5 Level 1 8

5. Identified vulnerabilities 9

5.1 Level 5 vulnerabilities 9

5.1.1 Weak passwords 9

5.1.2 Insufficient Authentication 9

5.1.3 Permanent Cross-site Scripting 10

5.1.4 OS Commanding 11

5.1.5 SSI Injection 11

5.1.6 Use of Vulnerable Software 12

5.2 Level 4 vulnerabilities 13

5.2.1 Weak Password Recovery Validation 13

5.2.2 Insecure Password Storage 14

5.2.3 Credential/Session Prediction 14

5.2.4 Insufficient Session Expiration 15

5.2.5 Session Fixation 15

5.2.6 Dynamic Cross-site Scripting 16

5.2.7 SQL Injection 16

5.2.8 Path Traversal 17

5.3 Level 3 vulnerabilities 18

5.3.1 Insufficient Authorization 18

5.3.2 Directory Indexing 19

5.3.3 Denial of Service 19

5.4 Level 2 vulnerabilities 20

5.4.1 Inconsistent Data Encoding 20

5.4.2 Information Leakage 20

5.4.3 Predictable Resource Location 21

5.4.4 Abuse of Functionality 21

5.4.5 Insufficient Anti-automation 22

5.4.6 Insufficient Error Handling 22

5.5 Level 1 vulnerabilities 23

5.5.1 Insufficient Process Validation 23

6. Appendix A. OSSTMM checklist 24

7. Appendix B. Testing protocols 29

8. Appendix C. OWASP ASVS checklist 31

# Management summary

Management summary

# Description of tested systems

The objective of current test was vulnerability assessment of the given IP address 81.106.xxx.77. Testing was conducted during the time period from 19th February 2008 till 24th February 2008.

Penetration test was based on „Open Source Security Testing Methodology Manual” (OSSTMM) methodology and „Open Web Application Security Project” (OWASP) recommendations.

# Vulnerability categorization

|  |  |  |
| --- | --- | --- |
| Level | Severity | Description |
| 5 | Urgent | Trojan Horses; file read and writes exploit; remote command execution |
| 4 | Critical | Potential Trojan Horses; file read exploit |
| 3 | High | Limited exploit of read; directory browsing; DoS |
| 2 | Medium | Sensitive configuration information can be obtained by hackers |
| 1 | Low | Information can be obtained by hackers on configuration |

## Level 5

Level 5 vulnerabilities provide remote intruders with remote root or remote administrator capabilities. With this level of vulnerability, hackers can compromise the entire host. Level 5 includes vulnerabilities that provide remote hackers full file-system read and write capabilities, remote execution of commands as a root or administrator user. The presence of backdoors and Trojans also qualify as Level 5 vulnerabilities. In case of web applications, full access to sites administration (CMS – content management system, for example) is also considered Level 5 vulnerability.

## Level 4

Level 4 vulnerabilities provide intruders with remote user, but not remote administrator or root user capabilities. Level 4 vulnerabilities give hackers partial access to file-systems (for example, full read access without full write access). Vulnerabilities that expose highly sensitive information qualify as Level 4 vulnerabilities. In case of web applications, access to other user account is also considered Level 4 vulnerability.

## Level 3

Level 3 vulnerabilities provide hackers with access to specific information stored on the host, including security settings. This level of vulnerabilities could result in potential misuse of the host by intruders. Examples of Level 3 vulnerabilities include partial disclosure of file contents, access to certain files on the host, directory browsing, disclosure of filtering rules and security mechanisms, susceptibility to denial of service (DoS) attacks, and unauthorized use of services such as mail relaying.

## Level 2

Level 2 vulnerabilities expose some sensitive information from the host, such as precise versions of services. With this information, hackers could research potential attacks against a host.

## Level 1

Level 1 vulnerabilities expose information, such as open ports, or other sensitive information from the host that does not qualify as Level 2 vulnerability.

# Identified vulnerabilities

## Level 5 vulnerabilities

### Weak passwords

|  |
| --- |
| **Description** |
| Users will often choose easy to guess passwords, possibly found in a dictionary, if choosing weak passwords is permitted by application. Given this scenario, an attacker would try thousands or potentially millions of guesses searching for the valid password. In case of strong passwords attacker would need to try billions of guesses what typically can not be performed in a reasonable amount of time. When a guessed password allows access to the system, the attack has been successful and the attacker is able access the account. |
| **System** |
|  |
| **Sample** |
|  |
| **Recommendations** |
| An adequate password policy should be created after risk assessment. Users should be informed about this policy and aspects of these requirements. In order for the password policy to be effective it has to be technically enforced, by not allowing use of passwords which do not match sites password policy. |

### Insufficient Authentication

|  |
| --- |
| **Description** |
| Insufficient Authentication occurs when a web site permits an attacker to access sensitive content or functionality without having to properly authenticate. Web-based administration tools are a good example of web sites providing access to sensitive functionality. Depending on the specific online resource, these web applications should not be directly accessible without the user required to properly verify their identity. |
| **System** |
|  |
| **Sample** |
|  |
| **Recommendations** |
| To get around setting up authentication, some resources are protected by “hiding” the specific location and not linking the location into the main web site or other public places. However, this approach is nothing more than “Security Through Obscurity”. It’s important to understand that simply because a resource is unknown to an attacker; it still remains accessible directly through a specific URL. The specific URL could be discovered through a Brute Force probing for common file and directory locations (/admin for example), error messages, referrer logs, or perhaps documented in help files. **These resources, whether they are content or functionality driven, should be adequately protected.**  Sometimes resources are not planned to be directly accessed; are not necessary for site functionality or have been set up for temporary use. However, they usually tend to be left accessible for indefinite time and their functionality abused. **Therefore temporary resources should be protected at a production level; unnecessary files removed and direct access allowed only to specified resources.** |