TESSOLVE
A Hero Electronix Venture

Client logo placeholder

XXX REPORT

Report Details

Title	Xxx Penetration Testing Report
Version	V1.0
Author	
Tester(s)	
Approved by	Client
Classification	Confidential

Recipient

Name	Title	Company

Version Control

Version	Date	Author	Description

TESSOLVE
A Hero Electronix Venture

Table of Contents:

Table List

Graph List

No table of figures entries found.

1. Executive Summary

1.1. Scope Purpose and Duration of Work

In accordance with the contract signed between Tessolve and [CLIENT], the penetration test was performed on [XXX domain and applications] between [DATE] and [DATE]. Domains and applications were tested for ### work hours. Reporting took ### work hours.

The purpose of the test was to [determine sec vulnerabilities, pci compliance, etc].

The scope of the test was limited to [IP address(es) listed/ web application(s) on the IP adress(es) listed] below.

IP Number	Domain
XXX	XXX

Table 1: Tested Systems

1.2. Findings

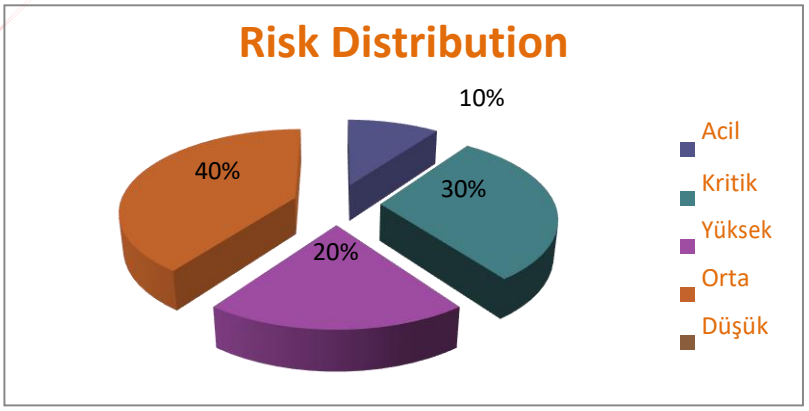
Urgent
Critical
High:
Medium:
Low:

1.3. Social Engineering Statistics

#	
Total emails sent	
Total clicks on the fake page	
Total credentials revealed	

1.4. Risk Distribution

(create a graph like this using Table-3)



Graph 1: Risk Distribution.

2. Methodology

The methodology consisted of steps beginning with the determination of test scope and ending with reporting. These tests were performed by security experts using potential attackers' modes of operation while controlling execution to prevent harm to the systems being tested. The approach included but is not limited to manual and automated vulnerability scans, verification of findings (automated and otherwise). This verification step and manual scanning process eliminated false positives and erroneous outputs, resulting in more efficient tests.

- Determining scope of the test
- Information Gathering / Reconnaissance
- Scanning
- Vulnerability Analysis
- Exploitation
- Post-Exploitation activities
- (Social Engineering - Optional)
- (Other Optional activity e.g. DDoS tests, Firewall gap analysis, log reviews, professional training)
- Reporting

2.1. Determining the Scope

Choose one, delete other

Our first step was determining the scope of the test. Since this was a Blackbox/Whitebox/Graybox (explain) test scope, as agreed with the client.

[SCREENSHOT/LOG]

Our first step was determining the scope of the test. This was a Blackbox test, therefore the target was researched to establish the test scope.

[Full research work, whois data, registrars, scans, etc.]
[SCREENSHOT/LOG]

2.2. Information Gathering

Before directly accessing the target, we researched everything we could locate from third party resources. This included DNS records, previous hacking attempts, job listings, email addresses, etc. This information was used in later tests.

[SCREENSHOT/LOG]

2.2.1. IP Addresses and Domains

Here is a list of the IP addresses and domains gathered using search engines:

```
a.b.c.d              www.host.com
```

2.2.2. Virtual Hosts

Virtual hosts sometimes share the same IP address with other website addresses. An attacker can compromise the server on which the target application runs using a vulnerability found on another website hosted on the same server.

```
a.b.c.d              www.host.com
```

2.2.3. IP Range Information

[SCREENSHOT/LOG]

2.2.4. Detailed DNS Records

DNS records identify URL/IP pairs. DNS servers connect the organization website to outside world. Exploitation of these servers may lead to malicious usage of the organization web and mail servers.

[SCREENSHOT/LOG]

2.2.5. WHOIS Information

'WHOIS' searches provide information regarding the domain name. It may include information such as domain ownership, where and when it was registered, expiration date, email address of the domain manager and the server names assigned to the domain.

[SCREENSHOT/LOG]

2.2.6. Job Advertisements

Job advertisements may reveal information about the organization systems and network. The requirements specified in job adverts sometimes disclose information regarding which programme languages and systems are used, providing attackers with more specific knowledge of the targets.

By analyzing the following job advertisements, we obtained the information listed below.

- PHP

[SCREENSHOT/LOG]

2.2.7. Login Pages Found During Server Analysis

Login pages are the front line of an application's defense against unauthorized access. They also present a surface area of interest to attackers who will try to defeat the defenses to access the functionality and data within the system. This section identifies the URLs and screens of the login pages discovered during analysis.

[SCREENSHOT/LOG]

2.3. Scanning

Various scans were performed to determine and verify vulnerabilities in the target systems.

Expand scans you did and remove scans you didn't do, if you executed a scan not listed here add it to the list and update the template. If you used any tools explain what you used and why. Screenshots/text logs for results

2.3.1. Port Scans

Which tool did you use, explain why. Screenshots/text logs for results

Primarily nmap is used to scan the targets. Besides nmap, tools like strobe, xprobe, amap are used to determine which ports are open, which operating systems are working on targets, and which services are used.

[SCREENSHOT/LOG]

Host (IP)

Open Port: Service:

(tcp/80)	A web server is running on this port.
(tcp/443)	A web server is running on this port through TLSv1.
(tcp/443)	A TLSv1 server answered on this port.

2.3.2. Route Scans

Which tool did you use, explain why. Screenshots/text logs for results

Using tools like *hping, scanrand, traceroute,* the network mapping of targets can be determined. It is also useful for detecting defensive measures like IDS, IPS, UTM, and firewalls.

[SCREENSHOT/LOG]

2.3.3. SNMP Scans

Which tool did you use, explain why. Screenshots/text logs for results

Using *onesixtyone,* SNMP scans were conducted to gain information.

[SCREENSHOT/LOG]

2.3.4. Server Identification

Which tool did you use, explain why. Screenshots/text logs for results

Using tools like *httprint, smtpscan*, detected servers (HTTP, FTP, SMTP, POP, IMAP, etc) from previous scans are listed and classified by their brand/model/operation systems/version numbers.

[SCREENSHOT/LOG]

2.3.5. VPN Identification

Which tool did you use, explain why. Screenshots/text logs for results

Using *ike-scan,* the network was traced for VPN servers.

[SCREENSHOT/LOG]

2.4. Vulnerability Analysis

2.4.1. Scanning Target Systems

Using vulnerability scanners like *nessus, acunetix, etc,* target systems were crosschecked with up-to-date vulnerability databases.

[SCREENSHOT/LOG]

2.4.2. SSL Certificates

SSL certificates used in target systems were scanned to determine the validity of their security. (ssl analyzer)

[SCREENSHOT/LOG]

2.4.3. Password breaking (Optional)

Using tools like *hydra*, *nessus*, a password breaking attack was executed by Brute force / Dictionary attack techniques.

[SCREENSHOT/LOG]

2.4.4. Privilege Escalation Attacks

Attacks where the goal is by-passing access control systems. (Logins, cookies, etc.)

[SCREENSHOT/LOG]

2.4.5. Web Scans

Using tools like *nikto* and *wfuzz,* folders and files hidden from end users were searched.

*: For simplification of the results, false positive items have been removed from the output.

[SCREENSHOT/LOG]

Host (IP) port 80

```
---------------------------------------------------------------------
+ Target IP:
+ Target Hostname:
+ Target Port:
+ Start Time:
---------------------------------------------------------------------
```

2.4.6. Business Logic Flaws

Flaws in application logic are harder to characterize than 'headline' vulnerabilities like SQL injection. In all but the simpliest of applications a large amount of logic is executed at every stage which presents an intricate surface of great interest to attackers.

[SCREENSHOT/LOG]

2.4.7. HTML Source Code Analysis

HTML source codes on the targets were checked to gather useful information.

[SCREENSHOT/LOG]

2.4.8. Testing for CAPTCHA

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a type of challenge-response test used by many web applications to ensure responses are not generated by computers. CAPTCHA implementations are often vulnerable to attacks even if the generated CAPTCHA is unbreakable.

[SCREENSHOT/LOG]

2.4.9. XSS Scans

Cross-Site Scripting vulnerabilities on input fields were checked.

[SCREENSHOT/LOG]

Hostname

```
Affected item
Affected Request
```

2.4.10. SQLi Scans

Possible SQL injection points on target servers were checked.

[SCREENSHOT/LOG]

2.4.11. Input Sanitizations

Various input points in the applications were tested to determine if they could be used for unintended purposes (file upload, file download, read access, etc).

[SCREENSHOT/LOG]

2.4.12. Session Security

Cookie security and the presence of Cross-Site Request Forgery (XSRF) vulnerabilities were tested.

[SCREENSHOT/LOG]

2.5. Social Engineering Approaches

2.5.1. Direct Social Engineering

Exploiting information from employees of the client.

2.5.2. Document Analysis.

Combing through documents found in scans.

2.5.3. Previous Hack Attempts.

Collecting information on previous attacks.

2.5.4. E-mails

Collected email list for SET phishing attacks.

2.6. Exploitation

Collect, list and explain every exploit found in the vulnerability scan steps.

2.7. Post – Exploitation

If necessary list any post exploitation work here (removing traces, deleting logs, removing backdoors put in system)

3. Detailed Information on Findings

3.1. Definition of Risk levels

Risk levels are based upon PCI / DSS standard definitions. The risk levels contained in this report are not the same as risk levels reported by the automated tools in general.

Risk Level	Explanation
	*Trojan horses, Backdoors, file read write vulnerabilities, remote code execution.* *5th level vulnerabilities give attackers remote root/administrator access and full control of the system.*
	*Potential Trojan horses, potential backdoors. File read vulnerability, limited filer write vulnerabilities.* *4ths level gives attacker limited access to controlling the systems. And access to critical confidential data.*
	*Limited read, directory traversal, denial of service.* *3rd level gives attacker access to private data such as security settings and partial file information and/or limited file access. Information gathered from this level vulnerability can potentially be used in harmful ways. Mail relay and DoS vulnerabilities are also classified this level.*
	*Detailed configuration data, service version numbers, installed patches.* *2nd level vulnerabilities discloses sensitive information about systems that can be used as basis for future attacks.*
	*Basic configuration data.* *1st level vulnerabilities (a.k.a. low, a.k.a. informational) vulnerabilities gives basic information for the system.*

Table 2: Risk level definitions.

3.2. Vulnerability List

Vuln. Code	Vulnerability Definition.	Risk Level	Affected count

Table 3: Vulnerabilities According to risk level.

4. Detected Vulnerabilities and Recommendations.

List every vulnerability found using this format

4.1. Host Vulnerabilities

4.1.1. Example XXX Vulnerability.

Risk : risk level

Source : page url, ip address, system name, etc. (i.e. domain name /login.php)

If multiple systems are affected by same vulnerability list all

Explanation : Explanation of vulnerability, including Screenshots

Recommendation : What can client do to solve the problem (i.e upgrade, apply patch)

4.2. Vulnerabilities by IP Numbers

XXXIP	
Name	
Reverse DNS	
Open Services	
Notes	

Vulnerabilities	
Level	
Urgent	
Critical	
Medium	
Low	