AMBERSAIL

Penetration Testing Procedure

This document is a sample penetration testing procedure developed by Ambersail. It forms part of the full PCI DSS policy documentation pack that can be found at https://www.pcipolicypack.com.

Read and use this document in conjunction with sample supporting policies in the accompanying document: *Penetration Testing Policy - Ambersail Sample*.

This document has been released as a guide on what information to include in a Penetration Testing Policy.

Document Reference	Penetration Testing Procedure - Ambersail Sample
Date	1 July 2018
Document Status	Final
Version	3.2
Revision History	3.0 - 30 September 2014 - Initial release for PCI DSS V3. 3.1 – 27 May 2015 – Update to reflect PCI DSS v3.1 changes. 3.2 – 1 Aug 2016: Update to reflect PCI DSS v3.2 changes.

Table of Contents

1. Purpose

This document details the steps required to perform penetration testing on **<COMPANY>'s** Cardholder Data Environment.

This document should be read in conjunction with Penetration Testing Policy.

2. Scope

The test procedures detailed below apply to all network components and application systems within **<COMPANY>**'s Cardholder Data Environment. This includes any wireless devices and connected environments that are directly connected to the Cardholder Data Environment.

3. Roles and Responsibilities

<COMPANY>'s [RESPONSIBLE ROLE] - Responsible for executing and implementing penetration testing.

See P01 – Information Security Policy for team contacts.

4. Procedure

4.1. Key Stages in Performing a Penetration Test

Basic set up

- Confirm 12 month anniversary date (for annual test only – does not apply to adhoc penetration test).
- Identify test targets. See 4.5 below.
- Notify production support teams.
- Notify affected external support teams (for example third party hosting providers).
- Confirm test organisation or individual (4.3).
- Finalise test date.
- Create test authorisation stating targets and test window. Achieve sign off from all directly affected parties.
- Ensure IP addresses of organisation performing external testing are recorded and communicated to affected parties. This will include network security function supporting the IDS / IPS.
- Set up any test accounts required for testing
- Finalise any remote access facilities (for any internal test to be performed from a remote source).
- Finalise any site visit details – for internal testing.

During test

- Monitor testing traffic and any impact on production / test networks.
- Regular update scheduled with Tester to understand progress and report on any significant vulnerabilities identified

Post test

- Confirm testing complete with Tester.
- Revoke any test credentials.
- Revoke any temporary access to internal networks.

- Accept Penetration Test report.
- Internal review to discuss findings.
- Schedule remediation work.
- Schedule retest to confirm vulnerabilities fixed.

4.2. Selection of Testers

<COMPANY> uses Penetration Test providers based on the following selection criteria:

- Ability – based on industry accreditations.
- Reasonable cost – ensuring **<COMPANY>** gets value for money.
- Track record – demonstrable experience with a range of clients.
- Responsiveness – clear and responsive lines of communication before, during and after testing.

External Penetration Tester provider:

Name of Company	Contact Name	Accreditations
<COMPANY> to update	<COMPANY> to update	<COMPANY> to update

-or-

<COMPANY> uses internal staff to perform testing based on the following arrangement:

- Ability – tester(s) must have 3+ year's experience in network or application security.
- Tester(s) must have at least 1+ year's recent experience performing penetration testing. This will be supported by formal training in penetration testing.
- Independence – The tester(s) must be completely independent from development and system maintenance teams to ensure testing is accurate, complete and objective.

Internal Penetration Tester:

Name of Tester	Contact Details	Accreditations / Experience	Independence to Development and Administration Teams
<COMPANY> to update	<COMPANY> to update	<COMPANY> to update	<COMPANY> to update

4.3. Identifying Test Targets

The test targets are those targets the form the Cardholder Data Environment or are directly connected to the Cardholder Data Environment.

There are three categories:

Internal Targets. These targets form the internal perimeter to the cardholder data environment – separating the Cardholder Data Network from non-card processing networks. Also included are the components that are located within the Cardholder Data Environment.

External Targets. These targets form the external perimeter to the cardholder data environment. Normally this will be Internet facing infrastructure that protects the network from the public Internet and is connected to the Cardholder Data Environment.

Segmentation Targets. A subset of the full list of Internal Targets. This particular category of Internal (CDE connected) targets support the segmentation strategy that **<COMPANY>** has adopted to reduce the footprint and scope of its Cardholder Data Environment.

Network level targets – such as Firewalls, servers, hosts – will be specified by unique IP Addresses in IPv4 format. Example format 123.123.123.123.

Application level targets – such as web applications – will be specified by URL or domain name. Example format http://www.myapplication.com.

For an exact breakdown on the Penetration Test targets, please refer to F21 - Penetration Test Targets.

4.4. Establishing Basic Level of Testing

As a basic guide to the types of tests that can be expected to be performed. This list is based on best practice penetration test guidelines, drawn from OWASP and CWE classifications.

More information about OWASP can be found at http://www.owasp.org. More information about CWE can be found at http://nvd.nist.gov/cwe.cfm.

Network Testing

Type of Test	Description
Information Disclosure	Exposure of system information, sensitive or private information, fingerprinting, etc.
Authentication Issues	Failure to properly authenticate users.
Credentials Management	Failure to properly create, store, transmit, or protect passwords and other credentials.
Permissions, Privileges, and Access Control	Failure to enforce permissions or other access restrictions for resources.
Buffer Errors	Buffer overflows and other buffer boundary errors.
Cryptographic Issues	An insecure algorithm or the inappropriate use of one.
Path Traversal	When user-supplied input can be passed through to file access APIs, causing access to files outside of an intended subdirectory.
Format String Vulnerability	The use of attacker-controlled input as the format string parameter in certain functions.
Configuration	A general configuration problem that is not associated with passwords or permissions.
Numeric Errors	Errors that can occur when handling numbers.
OS Command Injections	Allowing user-controlled input to be injected into command lines that are created to invoke other programs, using system() or similar functions.

Type of Test	Description
Race Conditions	The state of a resource can change between the time the resource is checked to when it is accessed.
Resource Management Errors	To consume excess resources, such as memory exhaustion from memory leaks, CPU consumption from infinite loops, disk space consumption, etc.
Link Following	Failure to protect against the use of symbolic or hard links that can point to files that are not intended to be accessed by the application.
Design Error	The initial design causes a vulnerability to exist.

Application Testing

Type of Test	Description
Injection	Hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorisation.
Broken Authentication and Session Management	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens.
Cross-Site Scripting (XSS)	XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
Insecure Direct Object References	Without an access control check or other protection, attackers can manipulate these references to access unauthorised data.
Security Misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform.
Sensitive Data Exposure	Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials.
Missing Function Level Access Control	Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed.
Cross-Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application.
Using Components with Known Vulnerabilities	Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover.
Unvalidated Redirects and Forwards	Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

Wireless Testing

Type of Test	Description
Detect and identify the target wireless network	Identify ESSID.
Test for channels and ESSID	Identify running services.
Map the entirety of the wireless network using GPS	IP address collection of access points and clients. MAC address collection of access points and clients. Test for rogue access points.
Determine encryption controls in use	Conduct WPA or WPA2 specific attacks including acquisition of hashes or other data that can be used to acquire valid credentials.
Analysis of password or account hashes	Conduct offline brute-force or rainbow table lookups to identify weak passwords or poorly implemented encryption.
Bypass MAC Filtering	Determine if filtering is in place.
Access Client LAN	Access core services and try to extract sensitive data from host systems.
LAN-side attacks	Configuration menu access – using browser interface, Telnet, SNMP and FTP. Determine types of authentication methods in place.
Physical Access	Assess physical security.

5. Enforcement

Any employee found to have violated this procedure will be subject to **<COMPANY>** disciplinary procedures, as detailed in the **<COMPANY>** Staff Handbook.

6. Glossary and References

6.1. Glossary

- **See document "P99 - Glossary"**

6.2. References

- P01 – Information Security Policy
- Penetration Testing Policy
- F21 – Penetration Test Targets