



Link Mountain, LLC PO Box 182 Port Sanilac Michigan, 48469

---

Date:  
Author:

Link Mountain, LLC  
PO Box 182  
Port Sanilac, MI 48469  
[www.LinkMountain.com](http://www.LinkMountain.com)

# Internal Penetration Test Report

---

For

[ClientName]

**CONFIDENTIAL-SENSITIVE**

Page 1

This document contains proprietary and confidential information of a highly sensitive nature. Reproduction or distribution without the express written permission of Link Mountain, LLC or the Client named above is strictly prohibited.

## Contents

<b>Executive Summary</b>	<b>3</b>
<b>Recommendations</b>	<b>3</b>
<b>Scope of Testing</b>	<b>3</b>
<b>Testing Details</b>	<b>4</b>
Network Discovery	4
Tools Used in Network Discovery	4
Summary Table – Network Discovery	4
Passive Discovery – Nslookup, TraceRoute, Zone Transfer requests	4
Port Scanning – TCP Syn Scan	5
Port Scanning – UDP Ports	5
Vulnerability Scanning	5
Scanners Used	5
Summary of Scanning Results	5
Penetration Testing	5
Objectives	5
Coverage	5
Tools Used in Penetration Testing	5
Network Test Coverage: NetBios enumeration	6
Network Test Coverage: LDAP	6
Network Test Coverage: SNMP Enumeration	6
Network Test Coverage: Open administrative interfaces	6
Network Test Coverage: Authentication attacks	6
Network Test Coverage: Application information disclosure	6
Web Application Test Coverage: Information Disclosure – Robots.txt, Hidden HTML fields, Comments, etc	6
Web Application Test Coverage: Session Handling	7
Web Application Test Coverage: Encryption	7
Web Application Test Coverage: Authentication – Logon Logic, Password Recovery, Account Creation	7
Web Application Test Coverage: Access Controls – Logical Faults, Path Traversal, Predictability, etc	7
Web Application Test Coverage: Reflection Issues – XSS, Response Splitting, etc	7
Web Application Test Coverage: Input Validation Issues – SQL Injection, Command Injection, etc	7
Web Application Test Coverage: Error Handling	8
Web Application Test Coverage: Configuration Issues – Directory Enumeration, Open Admin Interfaces, etc	8
Test Items Discovered	8
Item 1: Short Description	8
Item 2: Short Description	8
Item 3: Short Description	8
Item 4: Short Description	9
Item 5: Short Description	9

## Executive Summary

The first objective of this internal penetration test was to fully examine the systems under test to identify vulnerabilities that could allow an attacker to compromise the confidentiality, integrity or availability of those systems. Our second objective was to safeguard the stability of the [CLIENTNAME] systems under test. Our third objective was to prove exploitability by pursuing vulnerabilities to the point of compromise. The priority of these objectives dictated that vulnerabilities were not necessarily pursued to the point of full exploitation and compromise. Full exploitation was not pursued if the vulnerability appeared to be systemic, or if remediation was mandatory for PCI compliance, or if exploitation would have jeopardized either full test coverage or the stability of the systems under test.

Full details of our findings are found in the Testing Details section of the report; the following is an executive level summary of issues found:

Finding Summary here...

## Recommendations

LinkMountain recommends that all of the vulnerabilities be remediated, and a follow up test be conducted to verify remediation. If the test is conducted for PCI-DSS compliance, a remediation test is required (PCI-DSS 11.3.a: Verify that noted vulnerabilities were corrected and testing repeated.).

Further recommendations here...

## Scope of Testing

The following [CLIENTNAME] hosts were in scope and included in this penetration test:

Scope statement here...

## Testing Details

### Network Discovery

#### Tools Used in Network Discovery

NMap and Wireshark were used for all port scanning and packet capture. Manual inspection was also conducted using Trace Route and DNS Zone transfer requests to network name servers. The following table summarizes the results. Details are listed below.

#### Summary Table – Network Discovery

IP Address	Description	Discovery Method	
		TCP Scan	UDP Scan

#### Passive Discovery – Nslookup, Traceroute, Zone Transfer requests

Manual NSLookup and Traceroute requests were conducted to learn as much as possible about the target network. DNS Zone transfer requests were issued to any discovered name servers.

**CONFIDENTIAL-SENSITIVE**

Finding Summary here...

### Port Scanning – TCP Syn Scan

The target hosts were scanned for open ports with NMap. The open ports discovered are noted in the table above.

### Port Scanning – UDP Ports

The target hosts were scanned for open UDP ports using NMap. Ports that responded are noted in the table above.

## Vulnerability Scanning

### Scanners Used

Nessus, Nikto and Link Mountain custom scanners were used to scan the target hosts for known vulnerabilities.

### Summary of Scanning Results

Full details of the vulnerability scan are in the attached report. The following summary describes the major items that were of importance to the penetration test or that require remediation for PCI-DSS compliance:

Finding Summary here...

## Penetration Testing

### Objectives

Link Mountain conducted an internal penetration test of the network using a scenario of a compromised workstation with access to the card holder environment segment of the internal network. No domain or host user accounts were provided. The goal of this phase was to attempt to gain unauthorized access to any of the hosts or to otherwise gain access to sensitive data. The first objective was maximum test coverage; the second objective was safeguarding the stability of the systems under test, and the last objective was proof of exploitability. The priority of these objectives dictated that vulnerabilities were not necessarily pursued to the point of full exploitation and compromise. Full exploitation was not pursued if the vulnerability appeared to be systemic, or if remediation was mandatory for PCI compliance, or if exploitation would have jeopardized either full test coverage or the stability of the systems under test.

### Coverage

Link Mountain tested both the network and application layers. Any web applications that were previously tested externally were not retested. Any new applications or web interfaces discovered were tested without credentials.

### Tools Used in Penetration Testing

Link Mountain utilized Wire Shark, NMap, Netcat, Web Scarab, Nikto and manual testing procedures in this phase of the engagement. Net-snmp v 5.5 was used for SNMP enumeration and MIB walking. Native Windows NET commands were used in attempting connections to and enumerating network shares in Windows environments. Microsoft's LDP.EXE was used to enumerate LDAP trees in Windows environments. Web Scarab was used to spider web applications, Nikto was used to test for common directories, resources and configuration issues relating to web servers, Netcat was used for manual inspection of service banners and custom perl scripts were used for automated fuzzing of web application parameters and dictionary attacks. In addition, extensive manual inspection and manipulation was used.

**CONFIDENTIAL-SENSITIVE**

Page 5

This document contains proprietary and confidential information of a highly sensitive nature. Reproduction or distribution without the express written permission of Link Mountain, LLC or the Client named above is strictly prohibited.

#### **Network Test Coverage: NetBios enumeration**

Native Windows Net API commands were used from the testing host to identify and attempt connection to NetBIOS shares.

Finding Summary here...

#### **Network Test Coverage: LDAP**

Scanning tools were used to identify LDAP servers and, if found, Link Mountain attempted to enumerate LDAP trees.

Finding Summary here...

#### **Network Test Coverage: SNMP Enumeration**

Scanning tools were used to identify running SNMP agents and guess community strings. MIB walking was attempted for any SNMP agents found using default or easily guessable community strings.

Finding Summary here...

#### **Network Test Coverage: Open administrative interfaces**

Ports identified by NMap or scanners were manually tested using a browser or Netcat to manually confirm banner information and to look for open administrative interfaces. Directories discovered by Nikto or other scanning were also examined.

Finding Summary here...

#### **Network Test Coverage: Authentication attacks**

If authentication mechanisms were discovered, and a potential user list was also discovered or could be guessed, Link Mountain conducted either brute force or dictionary attacks.

Finding Summary here...

#### **Network Test Coverage: Application information disclosure**

Banners and pages returned by web applications and other services were checked for sensitive information leaks.

Finding Summary here...

#### **Web Application Test Coverage: Information Disclosure – Robots.txt, Hidden HTML fields, Comments, etc**

Web servers and web application roots were tested for the existence of Robots.txt files, and if found, were examined for sensitive data. All pages served by web applications were examined for sensitive information disclosed in HTML comments. Pages were inspected for 'hidden' fields to determine if sensitive information was disclosed.

Finding Summary here...

### Web Application Test Coverage: Session Handling

Session tokens were evaluated for predictability. GET requests were examined to ensure that session tokens were not passed in query strings. Applications were inspected for mixed use of encrypted and unencrypted transport, to insure that cookies containing session tokens are not sent over clear text channels.

Finding Summary here...

### Web Application Test Coverage: Encryption

Testing included verifying the enforcement of appropriate transport encryption, either SSLv3 or TLSv1. Cookies were examined for sensitive information encrypted with weak encryption or encoding schemes.

Finding Summary here...

### Web Application Test Coverage: Authentication – Logon Logic, Password Recovery, Account Creation

Logon logic was assessed to determine if authentication was enforced for access to all functionality and files, and that authentication could not be successfully bypassed. Testing also included verification of account lockout functionality, and examined password recovery and account creation logic for errors or weakness or disclosure of user accounts.

Finding Summary here...

### Web Application Test Coverage: Access Controls – Logical Faults, Path Traversal, Predictability, etc

Web servers and applications were extensively tested for the existence of common directory names that might have been missed in access control logic or have listable content. Traversal techniques were used to determine if web server or operating system faults could be used to bypass application access controls. Access control logic was tested by requesting resources from unexpected application states, particularly after error states were encountered, and manipulation of parameters and parameter names, host header and referrer values.

Finding Summary here...

### Web Application Test Coverage: Reflection Issues – XSS, Response Splitting, etc

Web applications were extensively 'fuzzed', and the results were inspected for evidence of unsanitized values reflected back to the client. Header values and cookies were manually manipulated in a similar manner to test for reflection.

Finding Summary here...

### Web Application Test Coverage: Input Validation Issues – SQL Injection, Command Injection, etc

Web applications were extensively 'fuzzed', and the results were inspected for evidence of unsanitized input values reaching application or other server side code. Header values, parameter names and cookies were manually manipulated in a similar manner to test for input validation.

Finding Summary here...

### Web Application Test Coverage: Error Handling

All responses received from web servers and applications during testing were inspected for evidence of improper error handling at the application, database and web server layers.

Finding Summary here...

### Web Application Test Coverage: Configuration Issues – Directory Enumeration, Open Admin Interfaces, etc

Nikto was used to find listable directories, common administrative interfaces and configuration errors. Information gained from HTML comments, error messages, Robots.txt files and other sources was manually examined for any information that could help identify code libraries in use. Default interfaces for web servers and any identified third party components were tested for secure configuration and patch levels.

Finding Summary here...

## Test Items Discovered

### *Item 1: Short Description*

---

**Full Description:**

**Severity:**

**Remediation:**

**Testing Notes:**

### *Item 2: Short Description*

---

**Full Description:**

**Severity:**

**Remediation:**

**Testing Notes:**

### *Item 3: Short Description*

---

**Full Description:**

**Severity:**

**Remediation:**

**Testing Notes:**

**CONFIDENTIAL-SENSITIVE**

*Item 4: Short Description*

---

**Full Description:**

**Severity:**

**Remediation:**

**Testing Notes:**

*Item 5: Short Description*

---

**Full Description:**

**Severity:**

**Remediation:**

**Testing Notes:**