**Ethical Hacking Penetration Test Plan and Findings Report**

**- Submitted by Arun Raj**

**1. Scope Definition and Rules of Engagement**

**Defined Scope**

* **Target Systems**:
  + Public-facing website (https://www.localstartup.com)
  + Internal employee workstations (accessible only via company network/VPN)
* **Testing Boundaries**:
  + Web application (including login, forms, and APIs)
  + Email systems for phishing simulation
  + No access to internal databases unless exposed
* **Excluded**:
  + Denial of Service (DoS/DDoS) attacks
  + Physical attacks or tailgating
  + Social engineering beyond email-based phishing

**Rules of Engagement**

* All testing to be conducted during business hours with prior notice
* Non-destructive testing only (no system alterations or service disruptions)
* Immediate reporting of any critical vulnerabilities found
* Authorization signed by company stakeholders

**Why Scope Definition is Critical**

Scope definition ensures legal and ethical boundaries, prevents unintended service disruption, and aligns expectations between the ethical hacker and the client. It helps prioritize testing efforts, manage risk, and avoid liability.

**2. Reconnaissance Summary**

**Passive Information Gathering**

* **WHOIS Lookup**:
  + Domain: localstartup.com
  + Registrar: Namecheap
  + DNS Hosted by: Cloudflare (potential WAF in place)
* **Subdomain Enumeration**:
  + shop.localstartup.com (main store)
  + admin.localstartup.com (admin panel, protected via login)
  + dev.localstartup.com (potentially unused/staging)
* **Social Media Scraping (LinkedIn & Twitter)**:
  + Identified 6 employees, including CTO and marketing staff
  + Employee titles suggest small IT team
  + Some staff using personal email domains for business correspondence
* **Technology Stack via Shodan & BuiltWith**:
  + Webserver: Apache 2.4.29 (Ubuntu)
  + CMS: WordPress 5.7
  + SSL: Valid Let's Encrypt Certificate
  + No CDN visible beyond Cloudflare

**3. Hypothesized Vulnerabilities**

**1. Outdated WordPress CMS & Plugins**

* WordPress 5.7 is several versions behind the latest (currently 6.x series)
* Plugins observed: Contact Form 7, WooCommerce (older versions known for vulnerabilities)
* Risk: Remote Code Execution or SQL Injection via plugin exploits

**2. Weak Employee Passwords / Credential Stuffing**

* Password reset pages visible for both customer and admin panels
* Public breach records (HaveIBeenPwned) show some company emails involved in leaks
* Small team size suggests likely reuse of passwords across services
* Risk: Unauthorized admin access or lateral movement

**3. Exposed Development Subdomain**

* dev.localstartup.com reachable publicly
* No authentication required; contains references to test user credentials and endpoints
* Risk: Information leakage or unintended access to staging environments

**4. Countermeasures and Prioritization**

| **Vulnerability** | **Mitigation Strategy** | **Priority** | **Risk Level** |
| --- | --- | --- | --- |
| **Outdated WordPress CMS & Plugins** | Implement strict patch management. Update core CMS and all plugins monthly. Use a staging environment for testing updates. | High | Critical |
| **Weak Employee Passwords** | Enforce strong password policies. Implement MFA. Periodically run credential audits using breach monitoring tools. Conduct phishing simulations for awareness. | Medium | High |
| **Exposed Development Subdomain** | Restrict access via .htaccess or IP whitelisting. Move all dev resources behind VPN. Remove hardcoded credentials and sensitive test data. | High | Medium-High |

**5. Executive Summary (For Non-Technical Audience)**

**Overview**

This penetration test was conducted to identify potential weaknesses in your website and employee workstations. We focused on passive data collection and light interaction with your systems to simulate a real-world hacker's approach—without causing damage or disruption.

**Key Findings**

* Your website runs on an outdated version of WordPress, increasing the risk of known exploits.
* Some employee accounts may be using weak or reused passwords, making them vulnerable to hacking.
* A development site is publicly accessible, which could expose sensitive data or internal logic.

**Recommendations**

* Update all software regularly to avoid known security holes.
* Train staff on using strong passwords and implement two-factor authentication.
* Secure development resources by hiding them from the public and using proper access controls.

Taking these steps will significantly improve your security posture and protect customer data, internal operations, and your brand’s reputation.

**Conclusion**

This penetration test revealed three core vulnerabilities that could be exploited by attackers. Addressing them through routine updates, employee education, and infrastructure hardening will help build a stronger security foundation.