Chippy’s Corner Risk Assessment Report

Prepared by:

Mamadu Bah

Date:

July 24, 2025

Table of Contents

Executive Summary

This risk assessment evaluates the cybersecurity posture of Chippy's Corner infrastructure, a simulated enterprise environment composed of Windows 10 workstations, a Windows 10 web server (hosting a XAMPP-based PHP application), a Domain Controller (DC) running Active Directory services, and a pfSense firewall.

The assessment revealed several critical and high-risk vulnerabilities across multiple endpoints. Key issues include the presence of unvalidated user input in a PHP web tool that could enable remote code execution (RCE) and lead to a full system compromise of the endpoint; exposed SMB services, combined with administrative tools such as Sysinternals, that increase the risk of lateral movement; and insecure default configurations, including enabled LLMNR and unrestricted outbound firewall rules.

While the Domain Controller has been secured and is not currently exposed to known vulnerabilities, the supporting infrastructure introduces significant risk to the domain environment as a whole. Multiple endpoints within the network lack hardening, effective logging, or credential protection, making them prime targets for external and internal threat actors.

This report outlines critical findings, provides prioritized recommendations, and proposes a phased response plan to reduce risk exposure and enhance the organization's overall security posture.

Scope

This risk assessment covers the internal enterprise infrastructure of Chippy's Corner staff network operating within the 192.168.110.0/24 subnet. The environment simulates a typical organizational network composed of employee endpoints, administrative servers, and core network services.

The assessment includes the following systems:

- A Windows 10 Pro workstation (Dell laptop) functioning as an internal web server. This device hosts an XAMPP stack with a PHP-based web application used by developers and serves as a critical internal-facing service. It is configured with the internal domain's primary DNS (stamfordlab.local) and has SMB services and Sysinternals utilities installed.

- A Windows 10 Home workstation (Lenovo PC) configured with Splunk for security monitoring. It serves as a general user endpoint within the environment. The system operates with a weak local password policy, has SMB enabled, and exhibits common misconfigurations such as enabled LLMNR and unrestricted outbound firewall settings. Splunk is installed but has not been fully optimized for active detection and response.

- A Domain Controller running on a Windows Server 2022 virtual machine hosted in VMware. This system provides Active Directory Domain Services and manages identity, authentication, and Group Policy within the domain stamfordlab.local. It is assumed to be hardened and operating with standard domain controller protections.

- A pfSense 2.8 virtual appliance running on VirtualBox acts as the environment's firewall and DHCP server. It provides internal routing and network segmentation. This endpoint's firewall is currently enabled and does not allow any inbound traffic to the virtual machine itself.

- A Kali Linux device (Dell laptop) is included in the environment to simulate adversarial activity. This system is used as a red team platform to test vulnerabilities, validate attack paths, and assess overall exposure across the environment.

This report evaluates each system for technical vulnerabilities, configuration weaknesses, and exposure to common enterprise threats. The scope specifically excludes physical security, cloud infrastructure, and third-party integrations.
A full network topology diagram is provided in Appendix A to support this assessment.

Summary of Results

Key Findings

PHP Web Application Allows Remote Code Execution

The internal web server hosts a PHP application that accepts user input without proper validation or sanitization. This flaw allows arbitrary input to be passed to system-level commands such as nslookup, creating a remote code execution (RCE) scenario. An attacker with access to the application could use this vulnerability to execute additional system commands, download malicious payloads, escalate privileges, or establish a persistent foothold on the server. Since the web server runs with local administrative permissions and is connected to the same internal network as other critical assets, a compromise of this system could lead to broader internal access.

Lateral Movement Enabled via SMB and Sysinternals

Both the internal web server and user workstation expose SMB (Server Message Block) services and have Sysinternals tools, such as PsExec, installed. These tools, when combined with misconfigured or over-permissive SMB shares, allow attackers to move laterally from one system to another within the network. If an attacker compromises a single machine (e.g., via phishing or reverse shell), they can use SMB and administrative tools to access other systems, execute commands remotely, and search for credentials or high-value targets such as the Domain Controller. This greatly increases the impact of an initial compromise and places the entire domain at risk.

LLMNR and Outbound Traffic Configuration Enable Credential Theft

Both Windows endpoints have Link-Local Multicast Name Resolution (LLMNR) enabled. LLMNR is a legacy name resolution protocol that is vulnerable to spoofing attacks. Combined with unrestricted outbound network access, this creates a scenario where an attacker can impersonate legitimate hosts on the network, intercept NTLM authentication attempts, and capture hashed credentials. These hashes can then be cracked offline or used directly in pass-the-hash attacks. The lack of outbound firewall restrictions also enables attackers to exfiltrate captured credentials or establish persistent command-and-control communication with

external systems. This type of exposure is commonly exploited during the early phases of internal compromise and credential harvesting.

Key Recommendations

Sanitize Web Input and Restrict PHP Execution Capabilities

All user input in the PHP application hosted on the internal web server must be properly validated and sanitized to prevent command injection. Input functions should be filtered to allow only expected values, and server-side scripts should avoid executing system-level commands altogether. In addition, disable dangerous PHP functions such as exec, shell_exec, and system unless strictly necessary. This change will directly mitigate the remote code execution vulnerability and reduce the server's exposure to arbitrary command abuse, file manipulation, and potential malware delivery.

Disable LLMNR and Restrict Outbound Traffic at the Host Level

LLMNR should be disabled on all Windows endpoints via Group Policy or local registry settings. This protocol is widely known to facilitate NTLM hash theft in internal networks and serves no critical function in modern environments. Additionally, implement host-based firewall rules to control outbound traffic for ports like 445 (SMB), 80/443 (HTTP/S), and DNS (53) to prevent exfiltration and outbound connections to unauthorized destinations. Together, these measures will reduce the risk of credential leakage, command-and-control communication, and unauthorized data transfer.

Reduce Lateral Movement Capabilities and Monitor Admin Tools

SMB services should be restricted to systems where file sharing or remote access is strictly required. Administrative tools such as PsExec and the Sysinternals suite should not be installed on user workstations or exposed servers unless justified and monitored. Additionally, internal firewall rules and Group Policy can be used to isolate critical systems and prevent peer-to-peer access that facilitates lateral movement. Regular auditing of SMB connections and remote tool usage should also be implemented to detect early signs of internal compromise.

Response Plan

Mitigation Prioritization	Vulnerability
Immediate (Imme.)	• Implement strict input validation across all PHP scripts hosted on internal web servers. • Disable high-risk PHP execution functions (exec, system, shell_exec) at the server configuration level. • Enforce LLMNR deactivation across all Windows endpoints using centralized Group Policy. • Apply outbound firewall rules to restrict external access to critical ports (e.g., 445, 53, 80/443) and prevent unauthorized communication.
Short-term (Long.)	• Remove Sysinternals tools from all non-administrative endpoints; restrict usage to jump boxes or administrative systems. • Reconfigure SMB to limit peer-to-peer access; enforce SMB signing and apply host-level access control. • Define and enforce a minimum password policy, including complexity and account lockout thresholds.
Long-term (Short.)	• Deploy Sysmon across endpoints to generate enhanced event telemetry. • Expand and fine-tune Splunk configurations to detect lateral movement, failed logons, and suspicious process behavior. • Conduct a rule-by-rule audit of pfSense firewall configurations to eliminate excessive or overly broad permissions.
Eventual (Evetl.)	• Remove default system configurations and enforce hardened baselines for all systems. • Enable outbound connection logging on endpoints to improve future correlation and detection. • Address low-severity misconfigurations that contribute to technical debt and weaken baseline controls.

Conclusion

The risk assessment of the StamfordLab simulation environment revealed multiple critical and high-risk vulnerabilities that, if left unaddressed, could significantly compromise the confidentiality, integrity, and availability of internal systems. Key findings include the presence of a remote code execution vulnerability within a PHP-based web application, widespread exposure to credential theft through LLMNR and unrestricted outbound communication, and the ability for attackers to move laterally across systems using SMB and administrative tools.

These issues reflect common misconfigurations and oversights that adversaries routinely exploit during targeted attacks and opportunistic breaches. While several mitigations can be implemented immediately, the broader remediation effort will require structured planning, improved monitoring, and the enforcement of security baselines across the environment.

By following the prioritized response plan outlined in this report, StamfordLab can significantly reduce its attack surface, enhance detection capabilities, and move toward a more resilient and defensible network posture. Continued assessments, policy enforcement, and system hardening will be essential to maintaining this posture over time.

This report should serve as both a tactical remediation roadmap and a baseline for future governance and security improvement efforts within the environment.

Appendix

Appendix A – Network Topology Diagram

The diagram below illustrates the structure of the StamfordLab test environment, including all major endpoints, subnets, and the pfSense firewall. It reflects the assessed network scope during this engagement.