Red Team Operations Report

Bah Red Labs
08/13/2025

Executive Summary

Bah Red Labs performed a Red Team engagement on Chippy's Corner domain from July 29, 2025, to July 30, 2025.

The engagement performed by Bah Red Labs employed real-world adversary techniques to target the systems under test. The sequence of activities in this approach involves open source intelligence (OSINT) collection, enumeration, exploitation, and attack in order to perform goal-specific operational impacts.

The goals included:

- Establish administrative-level access on a vulnerable web server.
- Compromise a workstation within the target environment and successfully exfiltrate sensitive data.
- Maintain persistence on compromised hosts to demonstrate long-term unauthorized access capabilities.

Although Red Team engagements are focused on security weaknesses, several positive observations were made:

- Windows Security demonstrated effective endpoint protection capabilities, successfully detecting and blocking the custom executable payload, and in some cases, automatically quarantining or deleting it before execution.

Specific observations for this assessment are outlined in the "Observations and Recommendations" section of this report. The following list is a summary of these observations:

- Observation A: The targeted web application failed to properly sanitize user input, allowing crafted commands to be executed on the underlying web server, leading to remote code execution.
- Observation B: An Nmap scan revealed that the workstation belonging to a high-value target (executive-level user) was running an exposed SSH service.
- Observation C: Weak authentication controls were identified on the same workstation, as a Hydra brute-force attack successfully obtained valid login credentials for SSH access.

A summary of goals and objectives achieved by Bah Red Labs:

- Goal 1 results: Successfully exploited an input validation vulnerability in the target web application to achieve remote code execution. This allowed the team to establish a TCP reverse shell and execute commands from the command line.
- Goal 2 results: Leveraging exposed SSH services on a high-value workstation, the team executed a Hydra brute-force attack to obtain valid credentials. This access enabled the retrieval and exfiltration of sensitive data from the workstation.

- Goal 3 results: After gaining a foothold in the compromised devices, the team deployed the custom-developed C2 to establish persistence in the compromised devices.

Bah Red Labs has provided specific recommendations for reducing the risks imposed by these issues in the “Observations and Recommendations” section of this report.

Bah Red Labs appreciates the opportunity to support Chippy’s Corner with its computer security. We look forward to assisting you and the Chippy’s Corner IT Staff in future endeavors.

TABLE OF CONTENTS

Section

1 Methodology and Goals

Red Team engagements performed by Bah Red Labs employ real-world adversary techniques to target the systems under test. Bah Red Labs uses a red team model that emulates actual adversary tools, techniques, and procedures (TTPs) driven by realistic attack scenarios and operational goals. Unlike a traditional penetration test, the red team approach evaluates the entire security scope of an organization, including people, processes, and technology, under conditions that closely resemble a live attack.

This engagement was conducted using the three primary Red Team phases: Get In, Stay In, and Act.

- Get In: Utilizing an assumed breach model.
- Stay In: Established and maintained access through persistence mechanisms and privilege escalation.
- Act: Conducted actions on objectives, including data collection, lateral movement, and demonstrating operational impacts.

During the engagement, operators followed a careful methodology using vetted tools, proven techniques, and operational discipline to prevent any unintentional disruption or degradation of Chippy's Corner's business operations.

Engagement Goals:

1. Establish administrative-level access on a vulnerable web server.
2. Compromise a workstation within the target environment and successfully exfiltrate sensitive data.
3. Maintain persistence on compromised hosts to demonstrate long-term unauthorized access capabilities.
4. Assess the effectiveness of security controls and identify gaps in detection, prevention, and response.

2 Scenario and Scope

2.1 Scenario

The Red Team engagement was based on the Assumed Breach Model, utilizing internal command and control. The approach of the Assumed Breach Model allows the test to begin quickly and later use access gained from the phishing attack to validate actions.

2.2 Scope

The scope identified by Chippy's Corner is to include the subnet: 192.168.110.0/24.

3 Attack Narrative

The following section outlines the sequence of events and highlights the key points during the engagement.

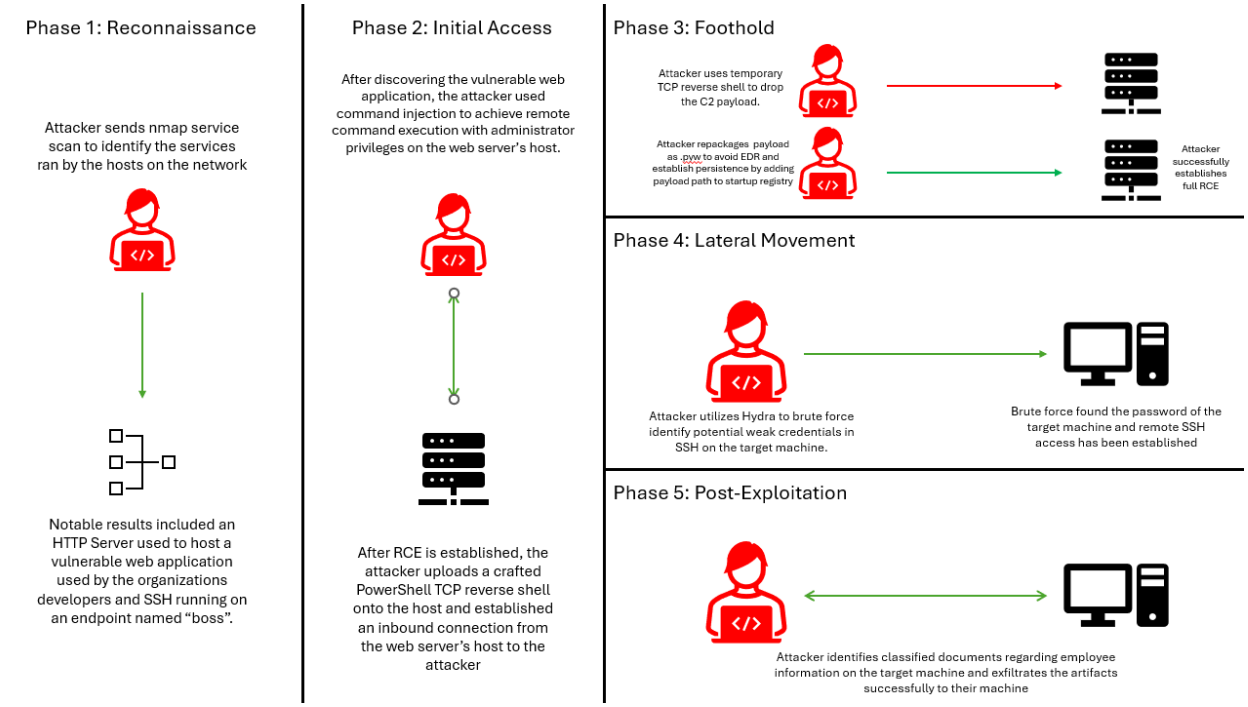

Figure 1: A five-stage attack narrative of the operation.

Here are the following critical steps within the attack narrative:

3.1 Discover the vulnerable webserver

There is an HTTP server running on the DevTools endpoint that hosts a web application allowing developers to run a ping command against a specified IP address. However, by chaining this command

```
Shell
127.0.0.1; whoami /groups
```

We were able to inject additional commands beyond the intended ping. Along with the ping output, the application also executed and displayed the results of the injected command. This

confirms that user input is not being sanitized, enabling remote command execution (RCE) on the XAMPP host. The output also revealed that these commands run in a shell with administrator privileges, providing an attacker with unrestricted control over the system.

3.2 Load and execute initial powershell tcp reverse shell

A custom PowerShell script was created to initiate a reverse TCP shell with administrator privileges. The script was uploaded to the target and executed on the Windows host through the vulnerable web server input field. On the attacker system (Kali Linux), netcat was run to listen for inbound connections from the web server. Once the payload executed, the connection was established, providing the first reverse shell and an initial foothold into the system.

3.3 Use the initial foothold to drop the C2 payload

After establishing an initial foothold on the target environment, we attempted to deliver and execute a C2 payload on the compromised web server. The original plan involved deploying the payload as a compiled executable; however, during testing, Windows Defender flagged and blocked the .exe on execution, preventing the C2 connection from being established.

3.4 Discover an alternate payload execution strategy

Upon further inspection of the web server, we identified that Python was installed on the web server. To evade the EDR detection and avoid displaying a visible console window during execution, we opted to repackage the payload as a .pyw file instead of an executable. This format successfully allowed the script to run silently in the background without triggering the same Defender signature, making it a more suitable option for dropping onto the web server. After dropping the .pyw payload, we successfully moved the payload to the HKCU startup registry key to execute the payload on every startup without notification.

3.5 Brute force success on boss's workstation

Using Hydra, we conducted a brute force attack against the SSH service running on the boss's workstation to guess valid Windows login credentials. Once the correct credentials were discovered, we established SSH access to the target system. With this foothold, we navigated the file system, identified sensitive business data, and successfully exfiltrated the files to the attacker's machine for further analysis.

4 Observations and Recommendations

The following section is intended to discuss specific scenarios that contributed to the compromise. The observations might be individually exploitable, an element of the overall compromise, or serve as a condition that directly impacts the ability to move laterally, escalate privileges, or persist.

4.1 Command Injection on DevTools Endpoint

The DevTools endpoint hosted a web application allowing developers to run ping commands against user-specified IP addresses. Input validation was absent, enabling the injection of additional system commands. By chaining *"127.0.0.1; whoami /groups"* into the ping input, it was possible to execute arbitrary commands in the server's shell environment. Output confirmed that these commands executed with administrator privileges, granting full system control to an attacker. This Remote Command Execution (RCE) vulnerability served as the initial entry point into the network.

4.1.1 Recommendations

- Implement strict input sanitization and parameter validation for all user-supplied data.
- Use allowlists for permitted commands and disallow special shell characters.
- Restrict web application execution privileges to a low-privileged service account.
- Conduct periodic code reviews and security testing for developer tools exposed to production environments.

4.2 Reverse Shell via PowerShell Payload

A custom PowerShell reverse TCP shell was uploaded and executed through the vulnerable web server input field. The payload connected back to the attacker's machine (listening via Netcat) with administrator privileges, providing full control over the host.

4.2.1 Recommendations

- Restrict the ability to run PowerShell scripts, especially from untrusted locations.
- Enable PowerShell Constrained Language Mode and Script Block Logging.
- Implement endpoint protection rules to detect and block reverse shell patterns.
- Monitor outbound connections for anomalous activity to attacker-controlled IPs.

4.3 Evasion via .pyw Payload and Registry Persistence

Upon discovering Python was installed on the target, the payload was repackaged as a .pyw file to evade Defender detection and avoid displaying a console window. The .pyw file was added to the HKCU startup registry key for persistence on reboot.

4.3.1 Recommendations (Optional)

- Monitor for creation/modification of registry run keys via EDR and SIEM rules.
- Restrict execution of scripting interpreters (Python, PowerShell, etc.) unless required for business functions.

4.4 SSH Brute Force on Boss's Workstation

The workstation had an exposed SSH service. Using Hydra, weak Windows login credentials were brute-forced successfully, allowing SSH access. From there, sensitive business data was identified and exfiltrated to the attacker's machine.

4.4.1 Recommendations (Optional)

- Disable SSH on systems where it is not required.
- Enforce strong password policies and multi-factor authentication for remote access. (Password should **NOT** be password)
- Implement account lockout policies after repeated failed login attempts.
- Monitor authentication logs for unusual login patterns.

5 MITRE ATT&CK Mapping

This section maps the Red Team engagement activities to the MITRE ATT&CK framework. The mapping demonstrates the techniques leveraged during the assessment and how the organization's mitigations relate to each observed tactic. Each row provides the procedure or action observed during the engagement, the corresponding MITRE tactic and technique, and additional context or notes on the procedure.

Action Taken	MITRE ATT&CK Tactic	MITRE ATT&CK Technique	MITRE ATT&CK Procedures
Command injection on DevTools endpoint	Initial Access	T1203 – Exploitation for Client Execution	Injected system commands via ping input; executed with administrator privileges on web server
Execution of PowerShell reverse TCP shell	Execution / Persistence	T1059.001 – PowerShell	Uploaded and executed PowerShell reverse shell to establish initial foothold
Dropping Python .pyw payload and adding to HKCU startup	Persistence	T1547.001 – Registry Run Keys / Startup Folder	Repackaged payload as .pyw, added to startup registry key for silent persistence
Brute-force attack on boss's SSH credentials using Hydra	Credential Access	T1110.001 – Brute Force: Password Guessing	Automated guessing of weak Windows login credentials over SSH
Remote SSH access and data exfiltration	Exfiltration / Lateral Movement	T1041 – Exfiltration Over C2 Channel	Logged in using discovered credentials to navigate file system
Evasion of endpoint detection via .pyw format	Defense Evasion	T1036.005 – Masquerading: Renaming/Obfuscating Files or Extensions	Used Python .pyw extension to evade Windows Defender detection
Remote SSH access to	Lateral	T1078 – Valid Accounts	Logged in using discovered

boss’s workstation	Movement		credentials to navigate file system
File system exploration on boss’s workstation	Discovery	T1083 – File and Directory Discovery	Enumerated directories to locate sensitive business files
Staging files for exfiltration	Collection	T1074 – Data Staged	Prepared identified files for exfiltration to attacker system
Data exfiltration	Exfiltration	T1041 – Exfiltration Over C2 Channel	Transferred sensitive files from boss’s workstation to attacker-controlled machine

6 Conclusion

Bah Red Labs performed a Red Team engagement at the request of Chippy's Corner to determine the full impact of a realistic threat. The Bah Red Labs team identified several exploitable vulnerabilities that were leveraged to establish a foothold, escalate privileges, expand access across the domain, and move proprietary information out of the network. Bah Red Labs assesses that an external threat actor could successfully compromise Chippy's Corner systems based on the path demonstrated during the assessment.

No highly specialized exploits or tools were used or required to perform any of the actions described within this report. Bah Red Labs relied primarily on publicly available attack frameworks for nearly all exploitation activities. The technical skill level required to conduct the individual actions ranged from low to intermediate. The required technical capability and the level of access achieved by chaining these vulnerabilities are a cause for concern.

Critical exposures and observations include:

- A command injection vulnerability on the DevTools endpoint allowed remote command execution with administrator privileges.

- Successful evasion of endpoint detection by deploying a .pyw Python payload for persistence through the startup registry key.

- Weak SSH credentials on a key workstation that were brute-forced using Hydra, enabling unauthorized remote access and data exfiltration.

Bah Red Labs operators demonstrated that an adversary with an organized phishing campaign, combined with these identified vulnerabilities, could potentially compromise the **Chippy's Corner** domain and remotely collect sensitive data or observe, disrupt, or deny business operations.

Overall, the Red Team was able to accomplish threat objectives, and it is our hope that the security posture of **Chippy's Corner** systems will be improved as a result of these efforts.