REPORT ON PAST APACHE 2.4 FLAWS AND SECURITY BUGS

This report outlines significant past security vulnerabilities and flaws that have been identified and addressed within the Apache HTTP Server 2.4 branch. It details the nature of these issues and the corresponding fixes implemented by the Apache Software Foundation.

INTRODUCTION

The Apache HTTP Server is one of the most widely used web servers globally. Like any complex software, it has experienced security vulnerabilities over time. This document focuses on historical issues within the 2.4.x series that have been patched, providing insight into how these weaknesses were exploited and subsequently remedied.

KEY PAST VULNERABILITIES AND FIXES

1. CVE-2017-9788: HTTP Request Smuggling

Vulnerability Description:
This vulnerability (CVE-2017-9788) affected Apache HTTP Server 2.4.25 and earlier. It concerned a flaw in how the server processed HTTP requests, particularly those with conflicting `Content-Length` and `Transfer-Encoding` headers. An attacker could craft a request that was interpreted differently by the front-end server (e.g., a load balancer or proxy) and the back-end Apache server. This discrepancy allowed the attacker to 'smuggle' a second, malicious HTTP request within the body of the first one, potentially leading to unauthorized access, cache poisoning, or the execution of unintended actions.

Fix Implemented:
The issue was addressed in Apache HTTP Server version 2.4.26. The fix involved enhancing the request parsing logic to strictly enforce RFC specifications regarding HTTP headers. Specifically, Apache was updated to reject ambiguous requests that contained both `Content-Length` and `Transfer-Encoding` headers or to process them in a standardized, secure manner that prevented desynchronization between

servers. This ensured that all compliant servers and proxies would interpret the request consistently.

2. CVE-2019-0211: Race Condition in Scoreboard Image Handling

Vulnerability Description:
Affecting Apache HTTP Server 2.4.35 and earlier, this vulnerability (CVE-2019-0211) was a race condition. The `scoreboard` is a shared memory segment used by Apache processes to track worker status. In certain configurations, a specially crafted request could trigger a race condition when writing to the scoreboard image. If an attacker could influence the timing of these writes, they might be able to overwrite arbitrary files on the server, potentially leading to privilege escalation or denial of service.

Fix Implemented:
The fix, introduced in Apache HTTP Server version 2.4.37, involved making the scoreboard updates more robust and less susceptible to race conditions. This was achieved by ensuring that writes to the scoreboard were atomic or properly synchronized, preventing multiple processes from corrupting the data structure. The update also included stricter validation and handling of shared memory operations to eliminate the window of vulnerability.

3. CVE-2021-41773 & CVE-2021-42340: Path Traversal and Information Disclosure

Vulnerability Description:
These vulnerabilities (CVE-2021-41773 and CVE-2021-42340) were discovered in Apache HTTP Server version 2.4.49. They were path traversal flaws that, when `mod_proxy` was used with certain configurations, allowed remote attackers to access files outside of the intended document root. For CVE-2021-41773, it was possible to map URLs to files outside of the configured `DocumentRoot` using a path traversal sequence (e.g., `../`). CVE-2021-42340 extended this by allowing similar access through a misconfiguration in `mod_proxy`'s handling of specific URL paths.

Fix Implemented:
Both vulnerabilities were addressed in Apache HTTP Server version 2.4.50. The primary fix involved strengthening the validation of file path requests, especially when `mod_proxy` was involved. The server's internal functions responsible for resolving URL paths to file system paths were updated to more strictly adhere to the configured access controls and to prevent the interpretation of path traversal

sequences like `..` from escaping the intended directories. This ensured that only files within the explicitly allowed directories could be served.

CONCLUSION

The vulnerabilities discussed above highlight the importance of timely patching and secure configuration for web server software. The Apache Software Foundation has a consistent record of addressing security concerns promptly. Users are always encouraged to keep their Apache HTTP Server instances updated to the latest stable version to benefit from these security enhancements and mitigations.

David Keith Jr "mrdj" ✓