Assignment 04
CVE Comparison

December 11

Prepared by: Danil Vilmont & Jason Mou
Course: ITAS 268 - Assignment 03
Date: December 11, 2025

Table of Contents

Next Page →

CVE Comparison Analysis:

Selected CVE

Field	Details
CVE ID	CVE-2022-42889
Vulnerability Type	OS command
Affected Product	Apache Commons Text
CVSS Score	9.8 (Critical)
Date Published	10/13/2022

Comparison with bWAPP Finding

Similarities

- The attacker sends a harmful or malicious input
- Application inserts that input into a command
- The server executes the OS command

Differences

- Input Location is different:
  - bWAPP: Type the command directly into a text box.
  - CVE-2022-42889: The command is hidden inside a string placeholder that the library evaluates
- Attack Complexity:
  - bWAPP: Very easy to exploit; no special syntax needed.
  - CVE-2022-42889:Require a specific format using Apache's "String Lookup" feature.

Next Page →

Real-World Impact

- This CVE impacted any version of Apache Commons Text starting in v1.5 & patched in 1.10.0.
- This vulnerability allowed attackers to run code by injecting interpolation strings that Apache Commons Text evaluated.
- It also enabled server-side request forgery, which can allow an attacker to make outbound connections to any destination.
- Successful exploitation could result in malware being installed on a server or in credentials (or private data) being stolen.

Discovery & Disclosure Timeline

Event	Date	Details
Discovered	Oct 13, 2022	Found during security testing.
Reported to Vendor	Oct 13, 2022	Sent privately to Apache
Patch Released	Sep. 24, 2022	Apache released Commons Text v1.10.0, disabling the dangerous interpolators by default.
Public Disclosure	Oct 13, 2022	The public disclosure of CVE-2022-42889 (known as "Text4Shell") occurred on October 13, 2022, by the Apache Software Foundation and security researcher Alvaro Muñoz.
CVE Assigned	Oct 13, 2022	Assigned the same day.

Next Page →

Lessons Learned

What can we learn from comparing lab vulnerabilities to production systems? Consider:

- ***Why do these vulnerabilities persist in real software?***
  - Complexity of software
  - Human Error.
  - Legacy Code.
  - Libraries and Dependencies
  - Time and Resource Pressure
  - Limited Awareness
- ***What could have prevented this vulnerability?***
  - Disabling string interpolation by default.
  - Adding input validation.
  - Limiting what the interpolator was allowed to access in the first place.
- ***How does this inform your approach to security?***
  - *This is a reminder to always triple-check which inputs your software can accept and what it can do. Not every function needs to have the scope of everything all at once; sometimes it's better to limit its ability rather than try to restrict the user's ability.*

Report Conclusion

Summary of Findings

#	Vulnerability	Severity	CVSS	Status
1	SQL Injection(Login)	Critical	8.0 (High)	Confirmed
2	XSS - Stored (blog)	Critical	7.6 (High)	Confirmed
3	Directory Traversal	High	7.4 (High)	Confirmed

Next Page →

Overall Risk Assessment

- **Jason Mou**
    - If the application has multiple security weaknesses, including SQL injection and OS command Injection, it would allow attackers to access sensitive data and execute system commands. The vulnerability in bWapp is in the Docker Lab environment. It shows off the importance of proper input validation, secure coding practices, and regular security testing to prevent similar issues in real-world applications.
- **Danil Vilmont**
    - The bWAPP version tested showed a very high-risk security posture, with more than a few critical vulnerabilities, including, but not limited to: SQL Injection in Authentication & input handling, little to no input sanitization on stored blog data (allowing for code execution), and directory traversal, which can allow an attacker to access almost any file on the system. Overall, in a real-world environment, these issues would expose private information, facilitate credential theft, & likely lead to many unhappy customers.

Team Acknowledgment

- By submitting this report, we confirm that all testing was performed ethically on our own local bWAPP Docker installation, and this report represents our original analysis and work.

Team member	**Signature**	**Date**
Member 1	Jason Mou	12/11/2025
Member 2	Danil VIlmont	12/11/2025