Bug Bounty Report — vulnweb.com

Date: 2025-10-15
Target: vulnweb.com (44.228.249.3 — ec2-44-228-249-3.us-west-2.compute.amazonaws.com)
Reported by: Masud Rana

Executive summary

During passive, non-destructive reconnaissance of **vulnweb.com** on 2025-10-15 multiple security issues were identified that affect confidentiality, availability, and increase attacker reconnaissance effectiveness. **No reachable HTTPS (443 filtered / connection timeout)** — site only responds over cleartext HTTP. (High) **Server version disclosure — Server: nginx/1.19.0** — exact server version exposed. (Medium) **Range handling / Accept-Ranges present** — potential for Range header abuse / Range-request DoS. (Medium) **ETag reveals metadata** (information disclosure/fingerprinting). (Low → Medium) **Missing standard security headers** (HSTS, X-Frame-Options, X-Content-Type-Options, CSP, etc.). (High to Medium) These issues combined mean user traffic is unencrypted, susceptible to MiTM/SSL-strip and session theft, and provide attackers with useful fingerprinting and potential DoS vectors.

Evidence (reproducible outputs taken during triage)

HTTP headers (curl -Is / with redirects):

```
HTTP/1.1 200 OK Server: nginx/1.19.0 Date: Wed, 15 Oct 2025 17:21:17 GMT Content-Type: text/html Content-Length: 4018 Last-Modified: Tue, 28 Jul 2020 09:20:49 GMT Connection: keep-alive ETag: "5f1fedf1-fb2" Accept-Ranges: bytes
```

HTTPS probe (curl -Iv):

```
* connect to 44.228.249.3 port 443 ... failed: Connection timed out * Failed to connect to vulnweb.com port 443 after 135086 ms: Could not connect to server curl: (28) Failed to connect to vulnweb.com port 443 after 135086 ms: Could not connect to server
```

nmap service detection (nmap -sV -p80,443):

```
Host is up (0.29s latency). rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com PORT STATE SERVICE VERSION 80/tcp open http nginx 1.19.0 443/tcp filtered https
```

TCP raw probe with nc (attempted Range request):

```
printf 'GET / HTTP/1.1 Host: vulnweb.com Range: bytes=0-1 ' | nc vulnweb.com 80 vulnweb.com [44.228.249.3] 80 (http) : Connection refused
```

Detailed findings

1) No HTTPS (Port 443 filtered / unreachable) — Severity: High
Description: TLS/HTTPS is not reachable from the testing vantage — connection attempts to port 443 time out and nmap shows *443/tcp filtered*. The site serves only over HTTP on port 80.
Impact: All traffic is cleartext (cookies, credentials, forms, parameters). This enables passive eavesdropping, MiTM attacks, credential/session theft, and SSL-strip style attacks.
Reproduction: `curl -Iv https://vulnweb.com` → connection timed out. `nmap` shows 443 filtered.
Suggested fix: Open/allow 443 at network level, install TLS certificate (Let's Encrypt or CA), configure

nginx to serve HTTPS and redirect HTTP → HTTPS, add HSTS.

2) Server version disclosure — Server: nginx/1.19.0 — Severity: Medium
Description: The Server header returns exact nginx version.
Impact: Facilitates targeted attacks by allowing attackers to query known CVEs for that version.
Reproduction: Observe `Server: nginx/1.19.0` in HTTP headers.
Suggested fix: `server_tokens off;` in nginx config.

3) Byte range support (Accept-Ranges: bytes) — potential Range DoS — Severity: Medium
Description: Accept-Ranges: bytes indicates the server accepts Range requests. Improper handling of many/overlapping ranges can be leveraged for resource exhaustion DoS.
Impact: An attacker can craft many/complex Range requests to increase CPU/memory usage and potentially cause service degradation.
Reproduction: Header Accept-Ranges: bytes seen in response.
Suggested fix: If ranged requests not needed, disable: add `add_header Accept-Ranges none;`. Otherwise use WAF/rate-limit checks to block abusive range headers.

4) ETag information disclosure — Severity: Low → Medium
Description: ETag: "5f1fedf1-fb2" may reveal server-side inode/timestamp/size info or be used for user tracking.
Impact: Fingerprinting, passive tracking, small information leakage that can help attackers.
Suggested fix: Disable or normalize ETags: `etag off;` or set safe ETag behavior.

5) Missing security headers — Severity: Medium → High (contextual)
Description: Responses lack HSTS, Content-Security-Policy, X-Frame-Options, X-Content-Type-Options, Referrer-Policy.
Impact: Increased exposure to clickjacking, MIME sniffing, XSS exploitation or downgrade attacks. Especially critical once HTTPS is enabled (HSTS).
Suggested fix (example headers): add Strict-Transport-Security, X-Frame-Options, X-Content-Type-Options, Referrer-Policy, Content-Security-Policy.

Risk matrix (summary)

Issue	Likelihood	Impact	Priority
No HTTPS (443 filtered)	High	High (confidentiality loss)	P0 / Immediate
Server version disclosure	High	Medium	P2
Accept-Ranges / Range DoS	Medium	Medium (availability)	P2
Missing security headers	High	Medium → High	P1
ETag leakage	Medium	Low → Medium	P3

Remediation — step-by-step (copy-paste ready)

Network / Cloud (AWS) — open and verify 443: aws ec2 authorize-security-group-ingress --group-id <sg-id> --protocol tcp --port 443 --cidr 0.0.0.0/0 sudo ufw allow 443/tcp Obtain TLS certificate (Let's Encrypt) & configure nginx: sudo apt update && sudo apt install certbot python3-certbot-nginx -y sudo certbot --nginx -d vulnweb.com -d www.vulnweb.com Example secure nginx site config (paste into /etc/nginx/sites-available/vulnweb): server { listen 80; server_name vulnweb.com www.vulnweb.com; return 301 https://$host$request_uri; } server { listen 443 ssl http2; server_name vulnweb.com www.vulnweb.com; ssl_certificate /etc/letsencrypt/live/vulnweb.com/fullchain.pem; ssl_certificate_key /etc/letsencrypt/live/vulnweb.com/privkey.pem; ssl_protocols TLSv1.2 TLSv1.3; ssl_prefer_server_ciphers on; ssl_session_cache shared:SSL:10m; add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload" always; add_header X-Frame-Options "DENY" always; add_header X-Content-Type-Options "nosniff" always; add_header Referrer-Policy "no-referrer" always; add_header Content-Security-Policy "default-src 'self';" always; server_tokens off; etag off; add_header Accept-Ranges

none; root /var/www/html; index index.html; location / { try_files $uri $uri/ =404; } } Additional hardening: Upgrade nginx to a supported release Configure WAF / ModSecurity to detect/limit abusive Range requests Enable rate limiting and monitoring

Verification steps (after fixes)

1. curl -Iv https://vulnweb.com -- expect TLS handshake success and 200 2. curl -I http://vulnweb.com -- expect 301 redirect to https 3. curl -I https://vulnweb.com -- expect security headers present and Server tokens hidden 4. nmap -sT -p80,443 vulnweb.com -- expect: 443 open and 80 redirect

Suggested disclosure / remediation timeline

T0 (Immediate): Enable HTTPS and redirect all HTTP→HTTPS; open 443 in cloud/network if accidentally blocked. (P0) T+24h: Add security headers, hide server tokens; disable ETag/Accept-Ranges if unnecessary; deploy WAF rules to limit range abuse. (P1) T+72h: Upgrade nginx to supported stable version; perform full regression testing. (P2) T+7d: Re-run full external scan (nmap, SSL Labs), verify fixes, and follow-up bug bounty submission with remediation evidence.

Example bug bounty submission (concise template)

Title: Missing HTTPS (443 filtered) — site serves only HTTP; information leakage & Range handling issues Summary: vulnweb.com is reachable only over HTTP (port 80). Port 443 is filtered and TLS cannot be established. HTTP responses leak Server: nginx/1.19.0, ETag, and Accept-Ranges: bytes. These issues allow MiTM, session theft, fingerprinting, and potential Range-based DoS. Steps to reproduce: 1. curl -Iv https://vulnweb.com -> connection timed out. 2. curl -Is http://vulnweb.com -> headers show Server: nginx/1.19.0, ETag: "...", Accept-Ranges: bytes. 3. nmap -sV -p80,443 vulnweb.com -> port 80 open (nginx 1.19.0), port 443 filtered. Impact: Attackers can intercept traffic, steal cookies/credentials, fingerprint server stack, and abuse range requests to attempt DoS. Suggested mitigation: Enable HTTPS with valid cert, redirect HTTP->HTTPS, add HSTS and security headers, hide server tokens, disable/normalize ETag, disable Accept-Ranges or protect with WAF, upgrade nginx. Severity: High.

End of report

Prepared by: Masud Rana