Logan Gillis, Aiden Habboub, Yaz Abu-Zaid

Bug Bounty Hunting: Vulnerability Discovery

What is HackerOne Bug Bounty?

- **Crowdsourced Security Platform**
- **Bug Bounty Programs**
- **Vulnerability Reporting Workflow**
- **Ethical Hacking Framework**
- **Used by Major Companies**

Neon

- Open-Source Database Company
- Separation of Storage & Compute
- Key Features
  - Autoscaling
  - Instant branching (for quick testing environments)
  - Bottomless storage (no storage limits)
- Security-Focused
- Innovation-Driven

HTML Injection Overview

Why it matters?

- Stored/DOM-based XSS can lead to user-level compromise
- Elevating a known vulnerability
- Validating Neon’s security posture

Scope of Testing

- Focused on authenticated pages and user-controlled inputs
- Operated within the constraints of Neon’s Bug Bounty Disclosure

HTML Injection Setup and Testing

Environment Setup

- MacOs system using OWASP ZAP for traffic interception/vulnerability scanning
- Test accounts created using HackerOne alias
- ZAP configured for both passive and active scanning

Testing

- Mapped application endpoints/user flows
- Injected known XSS payloads into user-editable fields
- Checked for payload persistence and JS execution in DOM

Results

Findings

- No successful payloads were executed via script injection
- All were escaped before any execution

Takeaways

- Neon's current sanitization appears effective against common XSS escalation
- Proper output encoding and input filtering are present
- User inputs are handled with appropriate escaping

Next Steps

- Explore other input contexts or chained vulnerabilities
- Test across different browsers
- Review JS source maps or API responses for hidden/overlooked parameters

Misconfigurations

CORS: What It Is, Why It's Important, And How To Exploit It

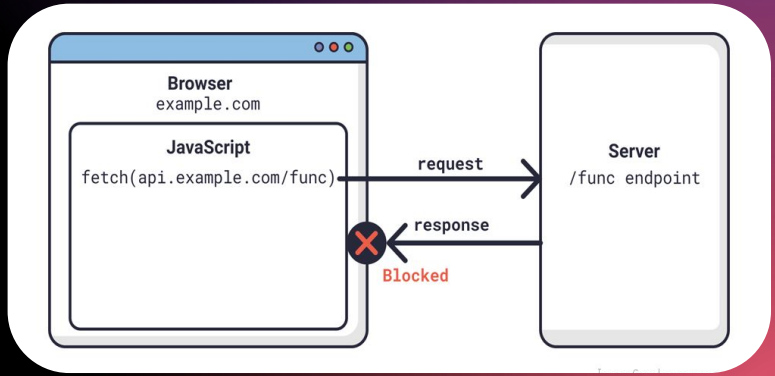
What Is CORS & Why It Matters

Why it matters?

- Stops malicious pages access to authenticated API responses
- Complements HTTPS & server-side auth
- Misconfig risk: wildcard * or reflecting arbitrary Origin with credentials

Scope of Testing

- Focused on account accessible API endpoints
- Ignored public facing API



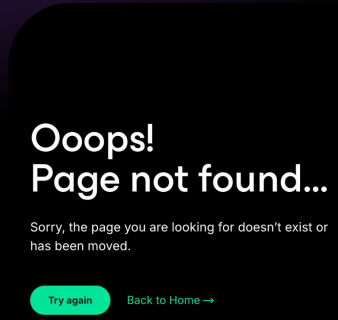
Testing For CORS Misconfigurations

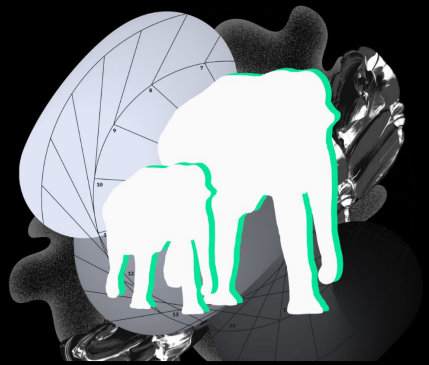
Tools

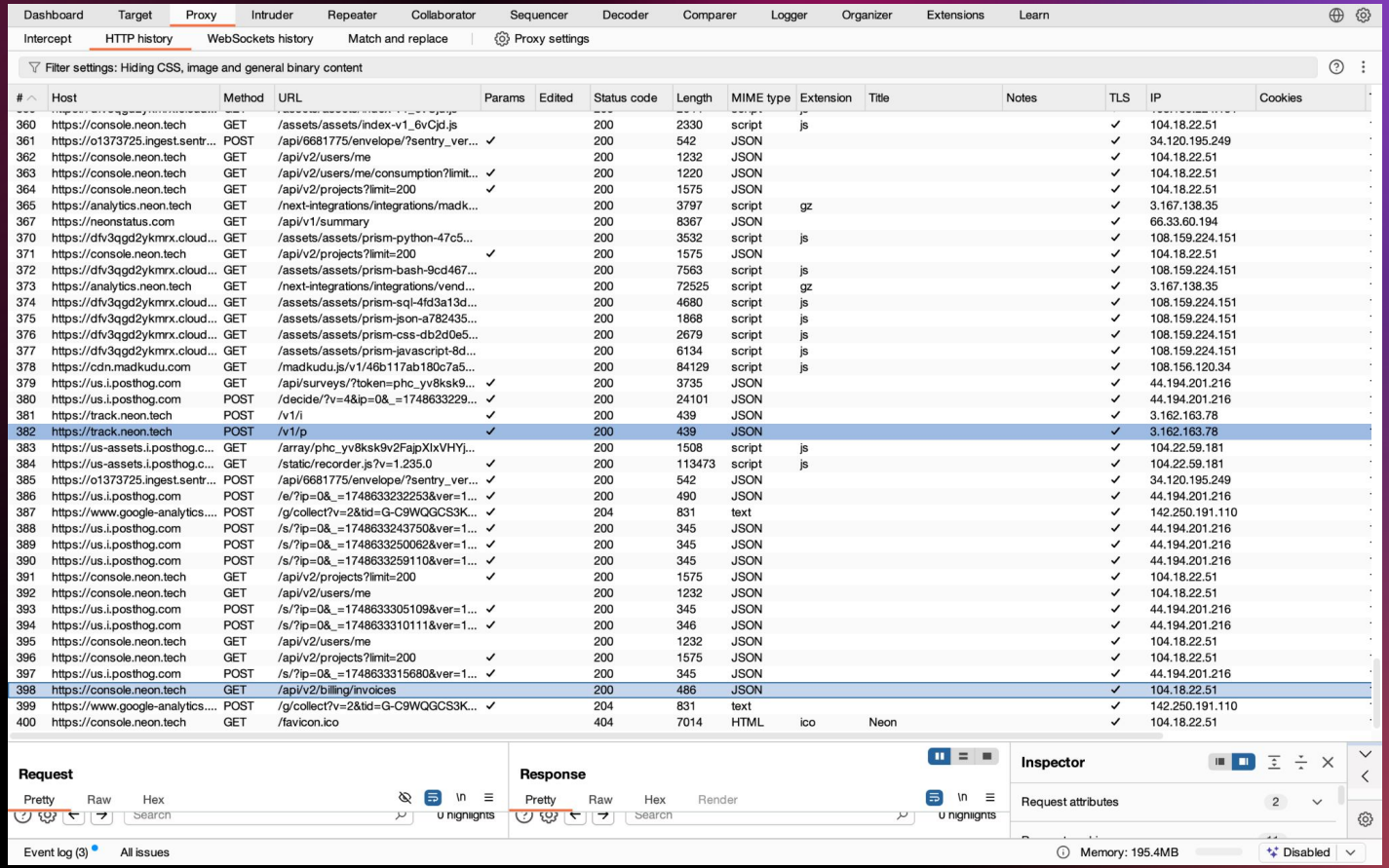
- Curl: quick header check
- Burp Browser: capture a API request
- Burp Repeater: edits the contents of a request and repeat it

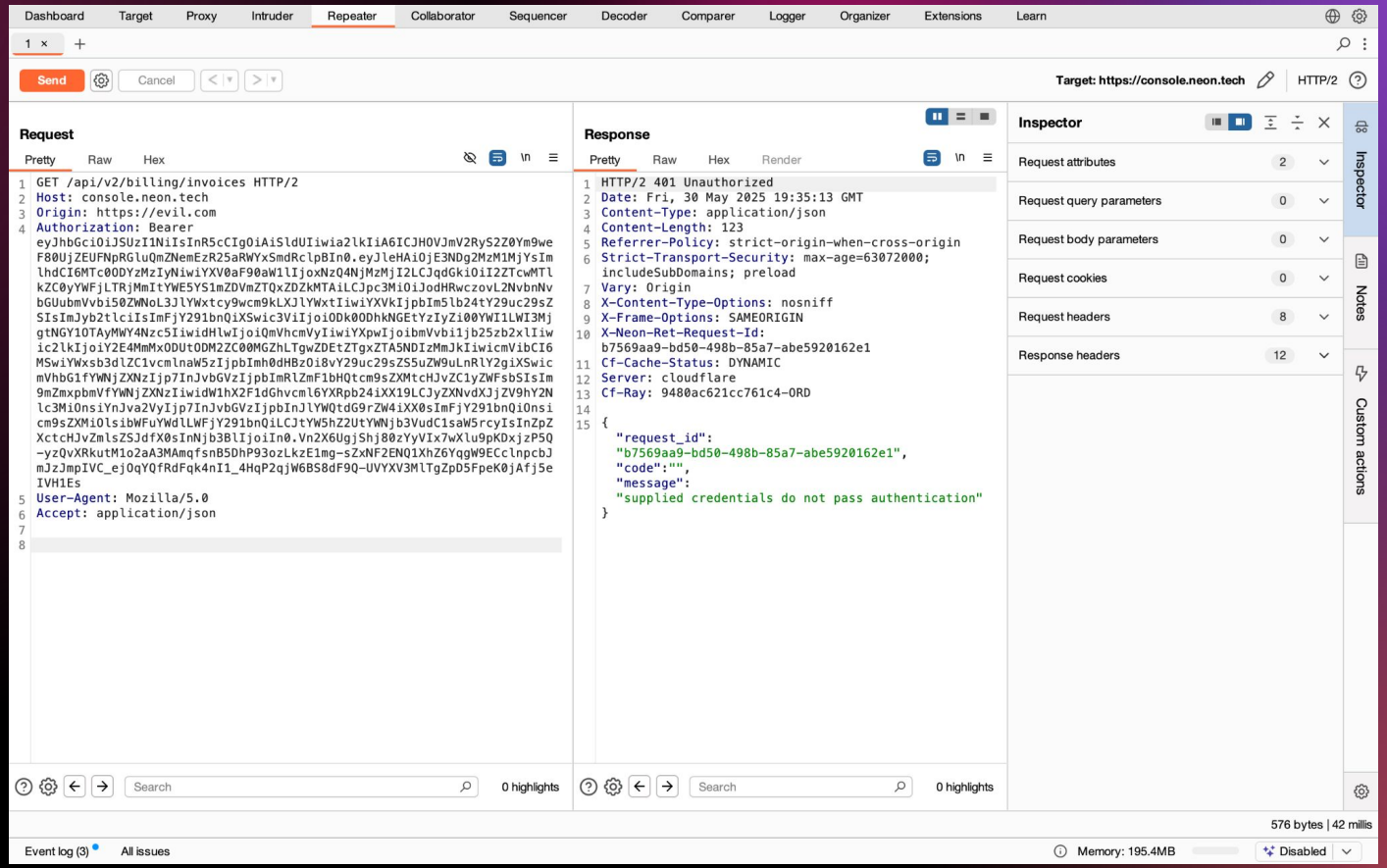
Execution

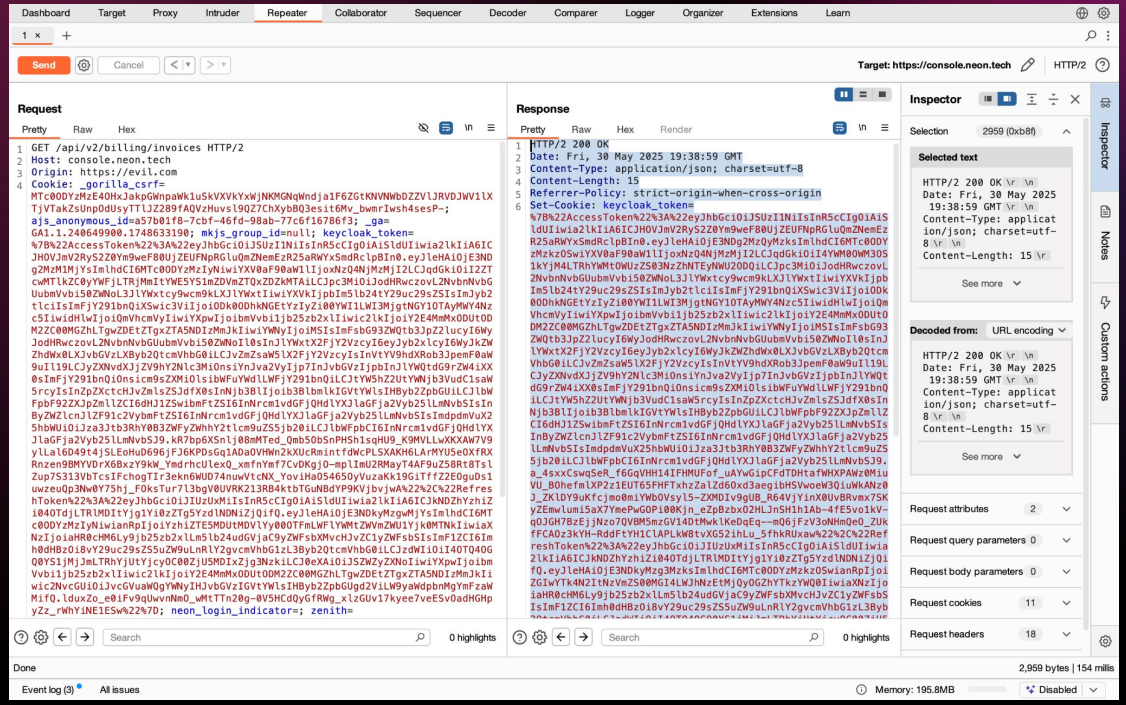
1. Capture a request to an API endpoint
2. Inspect it in Burp Suite
3. Send to repeater to alter request
4. Resend with foreign origin site






Dashboard
Target
Proxy
Intruder
Repeater
Collaborator
Sequencer
Decoder
Comparer
Logger
Organizer
Extensions
Learn
Intercept
HTTP history
WebSockets history
Match and replace
Proxy settings
Filter settings: Hiding CSS, image and general binary content
Host Method URL Params Edited Status code Length MIME type Extension Title Notes TLS IP Cookies
360 https://console.neon.tech GET /assets/assets/index-v1_6vCjd.js 200 2330 script js ✓ 104.18.22.51
361 https://o1373725.ingest.sentr... POST /api/6681775/envelope/?sentry_ver... ✓ 200 542 JSON ✓ 34.120.195.249
362 https://console.neon.tech GET /api/v2/users/me 200 1232 JSON ✓ 104.18.22.51
363 https://console.neon.tech GET /api/v2/users/me/consumption?limit... ✓ 200 1220 JSON ✓ 104.18.22.51
364 https://console.neon.tech GET /api/v2/projects?limit=200 ✓ 200 1575 JSON ✓ 104.18.22.51
365 https://analytics.neon.tech GET /next-integrations/integrations/madk... 200 3797 script gz ✓ 3.167.138.35
367 https://neonstatus.com GET /api/v1/summary 200 8367 JSON ✓ 66.33.60.194
370 https://dfv3qgd2ykmrx.cloud... GET /assets/assets/prism-python-47c5... 200 3532 script js ✓ 108.159.224.151
371 https://console.neon.tech GET /api/v2/projects?limit=200 ✓ 200 1575 JSON ✓ 104.18.22.51
372 https://dfv3qgd2ykmrx.cloud... GET /assets/assets/prism-bash-9cd467... 200 7563 script js ✓ 108.159.224.151
373 https://analytics.neon.tech GET /next-integrations/integrations/vend... 200 72525 script gz ✓ 3.167.138.35
374 https://dfv3qgd2ykmrx.cloud... GET /assets/assets/prism-sql-4fd3a13d... 200 4680 script js ✓ 108.159.224.151
375 https://dfv3qgd2ykmrx.cloud... GET /assets/assets/prism-json-a782435... 200 1868 script js ✓ 108.159.224.151
376 https://dfv3qgd2ykmrx.cloud... GET /assets/assets/prism-css-db2d0e5... 200 2679 script js ✓ 108.159.224.151
377 https://dfv3qgd2ykmrx.cloud... GET /assets/assets/prism-javascript-8d... 200 6134 script js ✓ 108.159.224.151
378 https://cdn.madkudu.com GET /madkudu.js/v1/46b117ab180c7a5... 200 84129 script js ✓ 108.156.120.34
379 https://us.i.posthog.com GET /api/surveys/?token=phc_yv8ksk9... ✓ 200 3735 JSON ✓ 44.194.201.216
380 https://us.i.posthog.com POST /decide/?v=4&ip=0&_=1748633229... ✓ 200 24101 JSON ✓ 44.194.201.216
381 https://track.neon.tech POST /v1/i ✓ 200 439 JSON ✓ 3.162.163.78
382 https://track.neon.tech POST /v1/p ✓ 200 439 JSON ✓ 3.162.163.78
383 https://us-assets.i.posthog.c... GET /array/phc_yv8ksk9v2FajpXlxVHYj... 200 1508 script js ✓ 104.22.59.181
384 https://us-assets.i.posthog.c... GET /static/recorder.js?v=1.235.0 ✓ 200 113473 script js ✓ 104.22.59.181
385 https://o1373725.ingest.sentr... POST /api/6681775/envelope/?sentry_ver... ✓ 200 542 JSON ✓ 34.120.195.249
386 https://us.i.posthog.com POST /e/?ip=0&_=1748633232253&ver=1... ✓ 200 490 JSON ✓ 44.194.201.216
387 https://www.google-analytics.... POST /g/collect?v=2&tid=G-C9WQGCS3K... ✓ 204 831 text ✓ 142.250.191.110
388 https://us.i.posthog.com POST /s/?ip=0&_=1748633243750&ver=1... ✓ 200 345 JSON ✓ 44.194.201.216
389 https://us.i.posthog.com POST /s/?ip=0&_=1748633250062&ver=1... ✓ 200 345 JSON ✓ 44.194.201.216
390 https://us.i.posthog.com POST /s/?ip=0&_=1748633259110&ver=1... ✓ 200 345 JSON ✓ 44.194.201.216
391 https://console.neon.tech GET /api/v2/projects?limit=200 ✓ 200 1575 JSON ✓ 104.18.22.51
392 https://console.neon.tech GET /api/v2/users/me 200 1232 JSON ✓ 104.18.22.51
393 https://us.i.posthog.com POST /s/?ip=0&_=1748633305109&ver=1... ✓ 200 345 JSON ✓ 44.194.201.216
394 https://us.i.posthog.com POST /s/?ip=0&_=1748633310111&ver=1... ✓ 200 346 JSON ✓ 44.194.201.216
395 https://console.neon.tech GET /api/v2/users/me 200 1232 JSON ✓ 104.18.22.51
396 https://console.neon.tech GET /api/v2/projects?limit=200 ✓ 200 1575 JSON ✓ 104.18.22.51
397 https://us.i.posthog.com POST /s/?ip=0&_=1748633315680&ver=1... ✓ 200 345 JSON ✓ 44.194.201.216
398 https://console.neon.tech GET /api/v2/billing/invoices 200 486 JSON ✓ 104.18.22.51
399 https://www.google-analytics.... POST /g/collect?v=2&tid=G-C9WQGCS3K... ✓ 204 831 text ✓ 142.250.191.110
400 https://console.neon.tech GET /favicon.ico 404 7014 HTML ico Neon ✓ 104.18.22.51
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Request attributes 2
Search 0 highlights
Search 0 highlights
Event log (3)
All issues
Memory: 195.4MB
Disabled


Dashboard
Target
Proxy
Intruder
Repeater
Collaborator
Sequencer
Decoder
Comparer
Logger
Organizer
Extensions
Learn
1 ×
Send
Cancel
Target: https://console.neon.tech
HTTP/2
Request
Pretty
Raw
Hex
GET /api/v2/billing/invoices HTTP/2
Host: console.neon.tech
Origin: https://evil.com
Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJHOVJmV2RyS2Z0Ym9weF80UjZEUFNpRGluQmZNemEzR25aRWYxSmdRclpBIn0.eyJleHAiOjE3NDg2MzM1MjYsImlhdCI6MTc0ODYzMzIyNiwiYXV0aF90aW1lIjoxNzQ4NjMzMjI2LCJqdGkiOiI2ZTcwMTlkZC0yYWFjLTRjMmItYWE5YS1mZDVmZTQxZDZkMTAiLCJpc3MiOiJodHRwczovL2NvbnNvbGUubmVvbi50ZWNoL3JlYWxtcy9wcm9kLXJlYWxtIiwiYXVkIjpbIm5lb24tY29uc29sZSIsImJyb2tlciIsImFjY291bnQiXSwic3ViIjoiODk0ODhkNGEtYzIyZi00YWI1LWI3MjgtNGY1OTAyMWY4Nzc5IiwidHlwIjoiQmVhcmVyIiwiYXpwIjoibmVvbi1jb25zb2xlIiwic2lkIjoiY2E4MmMxODUtODM2ZC00MGZhLTgwZDEtZTgxZTA5NDIzMmJkIiwicmVibCI6MSwiYWxsb3dlZC1vcmlnaW5zIjpbImh0dHBzOi8vY29uc29sZS5uZW9uLnRlY2giXSwicmVhbG1fYWNjZXNzIjp7InJvbGVzIjpbImRlZmF1bHQtcm9sZXMtcHJvZC1yZWFsbSIsIm9mZmxpbmVfYWNjZXNzIiwidW1hX2F1dGhvcml6YXRpb24iXX19LCJyZXNvdXJjZV9hY2Nlc3MiOnsiYnJva2VyIjp7InJvbGVzIjpbInJlYWQtdG9rZW4iXX0sImFjY291bnQiOnsicm9sZXMiOlsibWFuYWdlLWFjY291bnQsLCJtYW5hZ2UtYWNjb3VudC1saW5rcyIsInZpZXctcHJvZmlsZSJdfX0sInNjb3BlIjoiIn0.Vn2X6UgjShj80zYyVIx7wXlu9pKDxjzP5Q-yzQvXRkutM1o2aA3MAmqfsnB5DhP93ozLkzE1mg-sZxNF2ENQ1XhZ6YqgW9ECclnpcbJmJzJmpIVC_ejOqYQfRdFqk4nI1_4HqP2qjW6BS8dF9Q-UVYXV3MlTgZpD5FpeK0jAfj5eIVH1Es
User-Agent: Mozilla/5.0
Accept: application/json
Search
0 highlights
Response
Pretty
Raw
Hex
Render
HTTP/2 401 Unauthorized
Date: Fri, 30 May 2025 19:35:13 GMT
Content-Type: application/json
Content-Length: 123
Referrer-Policy: strict-origin-when-cross-origin
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
Vary: Origin
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Neon-Ret-Request-Id: b7569aa9-bd50-498b-85a7-abe5920162e1
Cf-Cache-Status: DYNAMIC
Server: cloudflare
Cf-Ray: 9480ac621cc761c4-ORD
{
"request_id": "b7569aa9-bd50-498b-85a7-abe5920162e1",
"code":"",
"message": "supplied credentials do not pass authentication"
}
Search
0 highlights
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 0
Request cookies 0
Request headers 8
Response headers 12
Inspector
Notes
Custom actions
576 bytes | 42 millis
Event log (3)
All issues
Memory: 195.4MB
Disabled



Response

Pretty Raw Hex Render

```
InByZWZlcnJlZF91c2VybmFtZSI6InNrcm1vdGFjQHdlYXJlaGFja2Vyb25lLmNvbSIsImdpdmVuX25hbWUiOiJza3Jtb3RhY0B3ZWFyZWhhY2tlcm9uZS5jb20iLCJlbWFpbCI6InNrcm1vdGFjQHdlYXJlaGFja2Vyb25lLmNvbSJ9.a_4sxxCswqSeR_f6GqVHH14IFHMUFof_uAYwGipCFdTDHtafWHXPAWz0MiuVU_BOhefmlXP2z1EUT65FHFTxhzZalZd6Oxd3aegibHSVwoeW3QiuWkANz0J_ZKlDY9uKfcjmo0miYWbOVsyl5-ZXMDIv9gUB_R64VjYinX0UvBRvmx7SKyZEmwlumi5aX7YmePwGOPi00Kjn_eZpBzbxO2HLJnSH1h1Ab-4fE5vo1kV-qOJGH7BzEjjNzo7QVBM5mzGV14DtMwklKeDqEq--mQ6jFzV3oNHmQeO_ZUkfFCAOz3kYH-RddFtYH1ClAPLkW8tvXG52ihLu_5fhkRUxaw%22%2C%22RefreshToken%22%3A%22eyJhbGciOiJIUzUxMiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJkNDZhYzhiZi04OTdjLTRlMDItYjg1Yi0zZTg5YzdlNDNiZjQifQ.eyJleHAiOjE3NDkyMzg3MzksImlhdCI6MTc0ODYzMzkzOSwianRpIjoiZGIwYTk4N2ItNzVmZS00MGI4LWJhNzEtMjQyOGZhYTkzYWQ0IiwiaXNzIjoiaHR0cHM6Ly9jb25zb2xlLm5lb24udGVjaC9yZWFsbXMvcHJvZC1yZWFsbSIsImF1ZCI6Imh0dHBzOi8vY29uc29sZS5uZW9uLnRlY2gvcmVhbG1zL3Byb2QtcmVhbG0iLCJzdWIiOiI4OTQ4OGQ0YS1jMjJmLTRhYjUtYjcyOC00ZjU5MDIxZjg3NzkiLCJ0eXAiOiJSZWZyZXNoIiwiYXpwIjoibWVvbi1jb25zb2xlIiwic2lkIjoiY2E4MmMxODUtODM2ZC00MGZhLTgwZDEtZTgxZTA5NDIzMmJkIiwic2NvcGUiOiJvcGVuaWQgYWNyIHJvbGVzIGVtYWlsIHByb2ZpbGUgd2ViLW9yaWdpbnMgYmFzaWMifQ.epzluDOU2neC5wYzkKTrIOSx5qDkdvy750ZK9cIXXaRi_x4Vi3x79w8D8UnE4FCRhvs8mZ-P4ucf40tArS4Lkw%22%7D
; Path=/; Max-Age=31104000; HttpOnly; Secure; SameSite=Lax
Set-Cookie: neon_login_indicator=; Path=/;
Domain=neon.tech; Max-Age=604800; Secure
Strict-Transport-Security: max-age=63072000;
includeSubDomains; preload
Vary: Origin
Vary: Cookie
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Neon-Ret-Request-Id: 573631f8-2d05-4cc9-bd00-d2180fe566ea
Cf-Cache-Status: DYNAMIC
Server: cloudflare
Cf-Ray: 9480b1ec29f434ad-ORD

{
  "invoices":[
  ]
}
```

Takeaways

- CORS = JS-level gatekeeper, not network blocker
- Always pair with strong server-side auth/authorization
- Best practice: whitelist exact origins, never * with credentials, validate Origin server-side



Work Summary

Low-Impact Authorization/Authentication Issues

- Insecure direct object references (IDOR) with low-privilege data

Subdomain Takeovers

- Enumerate unused subdomains (especially under `neon.build`)
  Check for CNAMEs pointing to decommissioned services

Verbose Error Messages in APIs

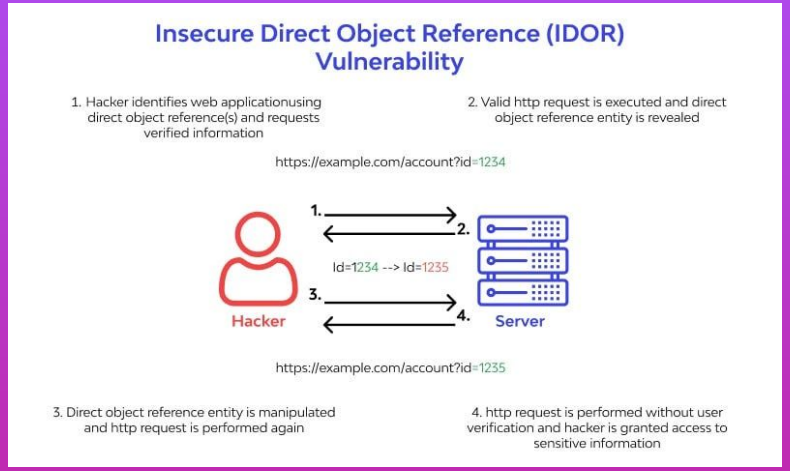
- Send malformed JSON or unexpected data types
- Trigger deserialization/parsing errors data types
- Explore `/api/v2/` for error leaks with internal structure details

Indirect Object References (IDOR) Testing Overview

01 Testes 3 main API categories

02 Used Burp Suite for traffic analysis

03 Cross-account access attempts



What Are Indirect Object References?

- Vulnerability allowing unauthorized access through ID manipulation
- Occurs when applications lack proper access control verification
- Attackers modify URLs or parameters to access restricted data
- Common in APIs, databases, and file systems

Target APIs Identified

- User Authentication (`/api/v2/users/me`)
- Project Management (`/api/v2/projects`)
- Consumption Data (`/api/v2/users/me/consumption`)

What is Burp Suite?

Burp or Burp Suite is a set of tools used for penetration testing of web applications. It is developed by the company named Portswigger. It is the most popular tool among professional web app security researchers and bug bounty hunters.

Testing Methods

→ Project ID manipulation

→ User ID modification

→ SQL Parameter Injection

→ Cross-account session testing

Methodology

Used Burp Suite's Interceptor to grab traffic from neon.tech

Step 01

Observed the traffic and note key GET and POST requests

Step 02

Analyze request structure, and send intercepted request to Burp Suite's repeater function

Step 03

Create Two accounts on the service

These will be used to test inter-account information access

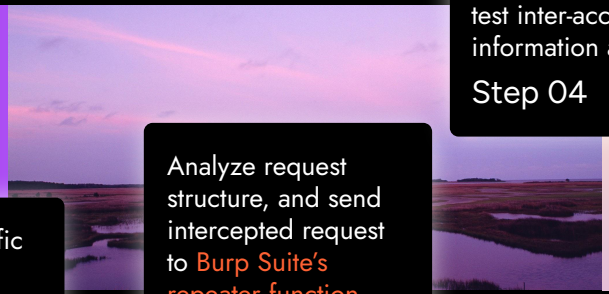
Step 04

Manipulate requests Authentication data, Project Endpoints, Database/Branch Operations,

Step 05

Observe response from the site, note headers and response info

Step 06



Results

Attack Vectors

- Project Access
- User Information
- Consumption data

Results

- No IDOR vulnerability endpoints
- Broad Error Messages

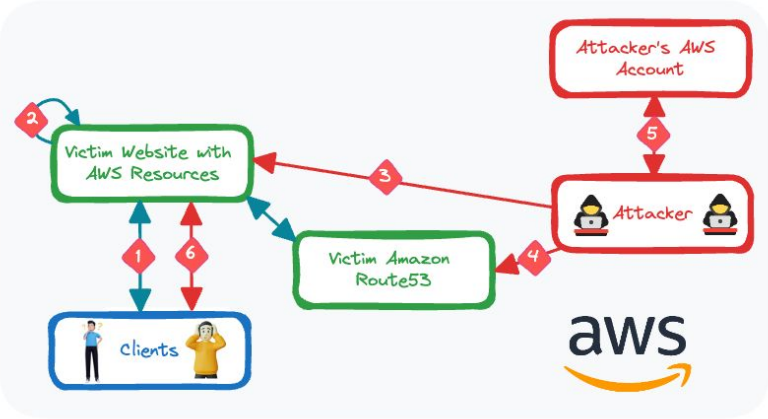
{

"request_id":"6dfb81f7-10c2-4c51-a807-16899462233f","code":"","message":"supplied credentials do not pass authentication"

}

Bug Bounty Attack 2

Subdomain Takeovers



What is a Subdomain Takeover

- Subdomain
  - A subdomain is a **separate section of a website** that uses a unique URL, but resides under your main domain name
- Takeover
  - Take control over abandoned/Inactive subdomains
  - Uses Canonical names to gain access

What is a Canonical Name (CNAME)

- These names take place of the original hosting URL. For example, if you want to host a website at www.example.com, and it is originally hosted on example1proj.vercel.app. The CNAME allows a record to be created, and this record allows the DNS to resolve the original URL to the desired URL

Subdomain Takeover Vulnerabilities

- Access to main domain cookies
- Cross-Site Scripting
- Circumvent security policies
- Gather sensitive information

Testing Methodology

→ Reconnaissance

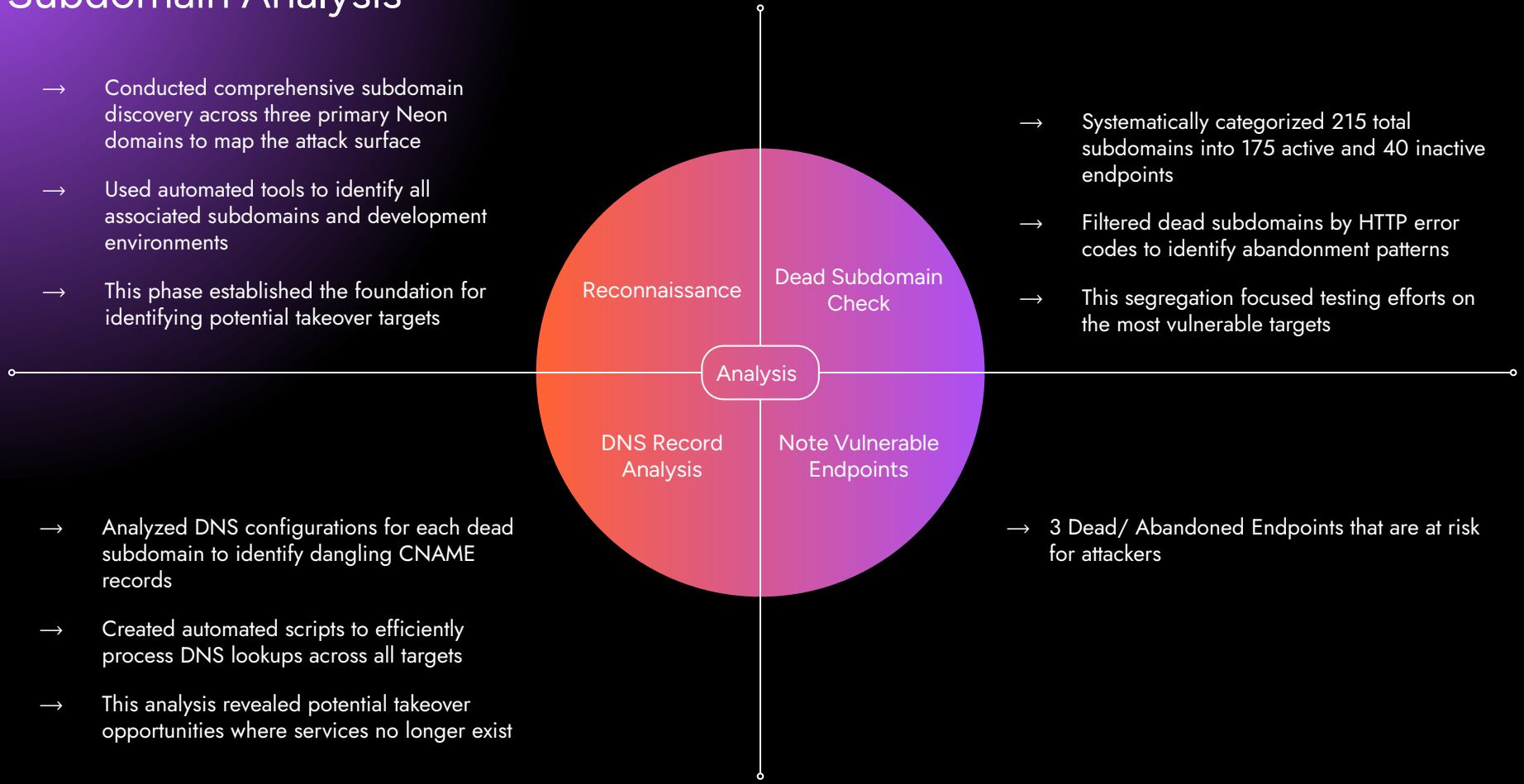
→ Check for Dead Subdomain Connections

→ DNS Record Analysis

→ Note Vulnerable enpoints

Subdomain Analysis

- → Conducted comprehensive subdomain discovery across three primary Neon domains to map the attack surface
- → Used automated tools to identify all associated subdomains and development environments
- → This phase established the foundation for identifying potential takeover targets



- → Systematically categorized 215 total subdomains into 175 active and 40 inactive endpoints
- → Filtered dead subdomains by HTTP error codes to identify abandonment patterns
- → This segregation focused testing efforts on the most vulnerable targets

- → Analyzed DNS configurations for each dead subdomain to identify dangling CNAME records
- → Created automated scripts to efficiently process DNS lookups across all targets
- → This analysis revealed potential takeover opportunities where services no longer exist

- → 3 Dead/ Abandoned Endpoints that are at risk for attackers

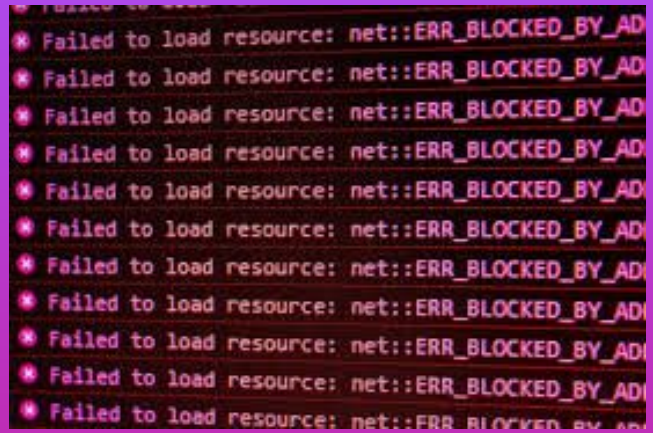
What this Script does

- **Reads dead subdomains** from dead-subdomains.txt file
- **Cleans domain names** by removing https://, URL paths, and status codes from httpx output
- **Queries DNS records** for each cleaned subdomain using dig command
  - **A Records**: Shows IP addresses
  - **CNAME Records**: Shows if domain redirects to another
  - **All Records**: Comprehensive
- **Identifies dangling CNAMEs** that point to external services
- **Saves results** to dns-records.txt for analysis of potential subdomain takeover vulnerabilities

```
echo "=== DNS Record Analysis for Dead Subdomains ==="
echo "Date: $(date)"
echo ""
while read subdomain; do
    # Remove https:// and any path/parameters to get clean
domain
    clean_domain=$(echo "$subdomain" | sed 's|https\?://||' | sed
's|/.||' | sed 's|\[.||')
   echo "=== Checking $clean_domain ==="
    echo "A Record:"
    dig +short A $clean_domain
    echo "CNAME Record:"
    dig +short CNAME $clean_domain
    echo "All Records:"
    dig +short $clean_domain
    echo "---"
    echo ""
done < dead-subdomains.txt > dns-records.txt
echo "DNS analysis complete. Results saved to
dns-records.txt"
```

Bug Bounty Attack 3

Verbose Error Messages In APIs



What Are Verbose Error Messages In APIs

- APIs that return detailed error information about internal system architecture
- Error messages containing stack traces, database schemas, file paths, or configuration details
- Common in development environments that accidentally reach production

Why this is important

Information disclosure can reveal:

- Database structure and table names
- Internal file paths and directory structure
- Framework versions and dependencies
- Server configuration details
- Business logic and validation rules

Real Worl Examples

- Netflix API leak exposed internal architecture through error messages
- GitHub API historically leaked repository existence through error variations
- Stripe API error messages revealed payment processing internals

Testing Methodology & Tools - API Error Exploitation

Objective: Identify improperly handled or overly descriptive error responses in Neon's `/api/v2/` endpoints.

Goal: Discover Errors that leak:

- Stack traces
- Internal file paths
- Function/class names
- Framework or tech stack info

Approach

Fuzzing malformed inputs: Send intentionally broken or incorrectly typed JSON payloads.

Trigger edge-case failures: Use unexpected characters, types (e.g., array instead of object), or missing fields.

Force parsing or deserialization errors: Attempt to break backend handling logic.

Example:

{ "user_id": ["not", "a", "string"] }
{ "incomplete_payload":

Tools Used

Key Tools

Burp Suite (or Postman): Manual request crafting, parameter tampering, intercepting responses.

ffuf or Intruder (Burp): Fuzzing input fields with malformed payloads.

httpx (ProjectDiscovery): Confirm which endpoints are live and responsive.

jq or custom Python scripts: To parse and filter responses for error patterns

Setup

- Use the unique header X-Bug-Bounty:HackerOne-username to ensure compliance with testing rules.
- Target only in-scope endpoints, especially /api/v2/.

Why It’s Important

Security Implications

Verbose errors can expose internal logic, architecture, and stack traces.

These details help attackers:

- Map internal services
- Identify backend frameworks and vulnerable libraries
- Craft targeted exploits (e.g., RCE, privilege escalation)

Value to Neon

Reducing verbosity prevents reconnaissance by bad actors.

Aligns with secure coding practices and OWASP API Security Top 10 (API6:2023 - Unrestricted Access to Sensitive Business Flows).

AI & How To Write A Report

Vulnerability Report & Latest In The Field

Standards For Vulnerability Reports

Contents

- Concise Summary Title
- Summary
- Steps to Reproduce
- Expected vs. Actual Behavior
- Impact & CVSS Rating
- Supporting Material:

Takeaways

- Reports are Information so ***be direct, clear, and concise***
- The report is useless if reviewers cannot reproduce the vulnerability using your directions

Common Vulnerability Scoring System - CVSS Scores

What it is

- Industry-standard way to rate the severity of a vulnerability
- Helps triage teams prioritize fixes by standardizing impact

Base Score (0.0–10.0)

- *Exploitability*: how easy is it to exploit?
  - Attack vector
  - Attack complexity
  - Privileges required
  - User interaction
- *Impact*: what gets affected if exploited?
  - Confidentiality
  - Integrity
  - Availability

Severity Rating	CVSS 3.1 Score	Description
CRITICAL	9.0 - 10	Exploitation of the vulnerability allows an attacker administrative-level access to systems and/or high-level data that would catastrophically impact the organization. Vulnerabilities marked CRITICAL require immediate attention and must be fixed without delay, especially if they occur in a production environment.
HIGH	7.0 - 8.9	Exploitation of the vulnerability makes it possible to access high-value data. However, there are certain pre-requisites that need to be met for the attack to be successful. These vulnerabilities should be reviewed and remedied wherever possible.
MEDIUM	4.0 - 6.9	Exploitation of the vulnerability might depend on external factors or other conditions that are difficult to achieve, like requiring user privileges for a successful exploitation. These are moderate security issues that require some effort to successfully impact the environment.
LOW	0.1 - 3.9	Vulnerabilities in the low range typically have very little impact on an organization's business. Exploitation of such vulnerabilities usually requires local or physical system access and depends on conditions that are very difficult to achieve practically.
INFORMA-TIONAL	0.0	These vulnerabilities represent significantly less risk and are informational in nature. These items can be remediated to increase security.

CVSS Vector String

What it is

- Compact individual metrics that make up a CVSS Base Score.

Example:

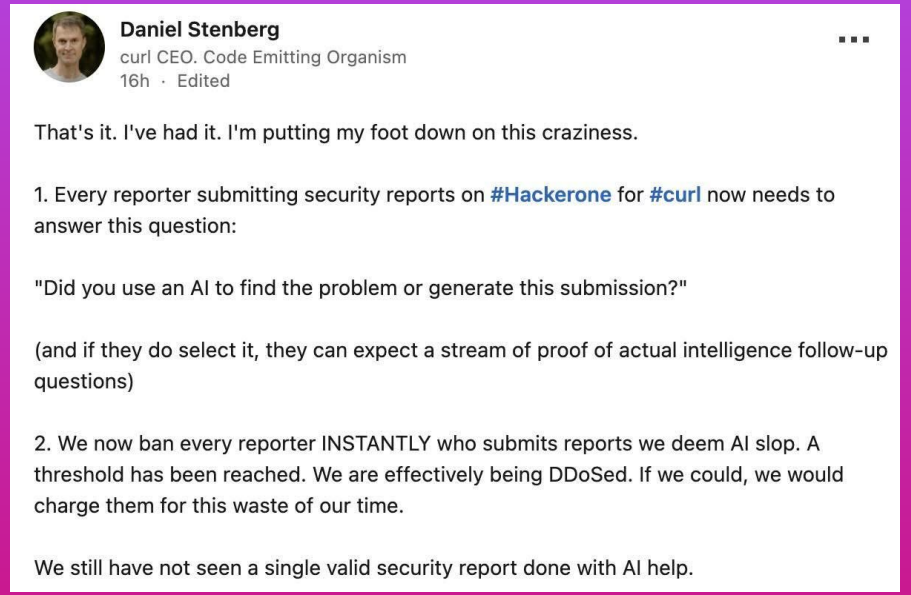
- `AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N`

Metric	Value	Meaning
`AV:N`	**Attack Vector: Network**	Can be exploited remotely over the internet
`AC:L`	**Attack Complexity: Low**	No special conditions required; easy to exploit
`PR:N`	**Privileges Required: None**	Attacker doesn't need to be logged in
`UI:N`	**User Interaction: None**	No action from the user is needed
`S:U`	**Scope: Unchanged**	Exploit only impacts the same security domain
`C:H`	**Confidentiality: High**	Major data exposure is possible
`I:H`	**Integrity: High**	Data could be manipulated or altered
`A:N`	**Availability: None**	System availability is not affected

Bans On AI Bugs reports Begin

“Curl project founder snaps over deluge of time-sucking AI slop bug reports”

The misconception that vulnerability threat assessments to identify bugs is easy because it accessible has made a convenient target for unorganized AI automation



Daniel Stenberg
curl CEO. Code Emitting Organism
16h · Edited

That's it. I've had it. I'm putting my foot down on this craziness.

1. Every reporter submitting security reports on #Hackerone for #curl now needs to answer this question:

"Did you use an AI to find the problem or generate this submission?"

(and if they do select it, they can expect a stream of proof of actual intelligence follow-up questions)

2. We now ban every reporter INSTANTLY who submits reports we deem AI slop. A threshold has been reached. We are effectively being DDoSed. If we could, we would charge them for this waste of our time.

We still have not seen a single valid security report done with AI help.

https://www.theregister.com/2025/05/07/curl_ai_bug_reports/

Citations

- https://www.infosecinstitute.com/resources/vulnerabilities/how-to-write-a-vulnerability-report/
- https://owasp.org/www-community/vulnerabilities/Vulnerability_template
- https://docs.hackerone.com/en/articles/8475116-quality-reports?utm_source=chatgpt.com
- https://www.threads.com/@nixcraft/post/DJROwcupBat/media
- https://www.theregister.com/2025/05/07/curl_ai_bug_reports/
- https://www.phoenix.com/blog/cvss-4-0-is-here/
- https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html
- https://www.geeksforgeeks.org/what-is-burp-suite/
- https://owasp.org/www-project-api-security/
- https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final
- https://github.com/projectdiscovery/subfinder
- https://github.com/projectdiscovery/httpx
- https://github.com/ffuf/ffuf

THANK YOU

Any questions?