



Supplementing your SU with bug bounty hunting

Crowdsourced security – The state of bug
bounty hunting in 2025

#whoami

- Studied the cybersecurity master 2020-2022
- Pentester at TDCNET
- Teacher of 'Fundamentals of Cybersecurity' @ AAU
- Various pentesting certs (OSCP, OSCE3)
- Freetime bug bounty hunter (why I give this talk)
- Web security researcher.



Why give this talk?

- In Denmark there very few bug bounty hunters
 - It's nicer to have someone to work with
 - More hunters means more programs launching in Denmark (my hypothesis)
- Relevant for application security jobs
- Because bug bounty hunting is fun!
- Because you can earn some money while studying.



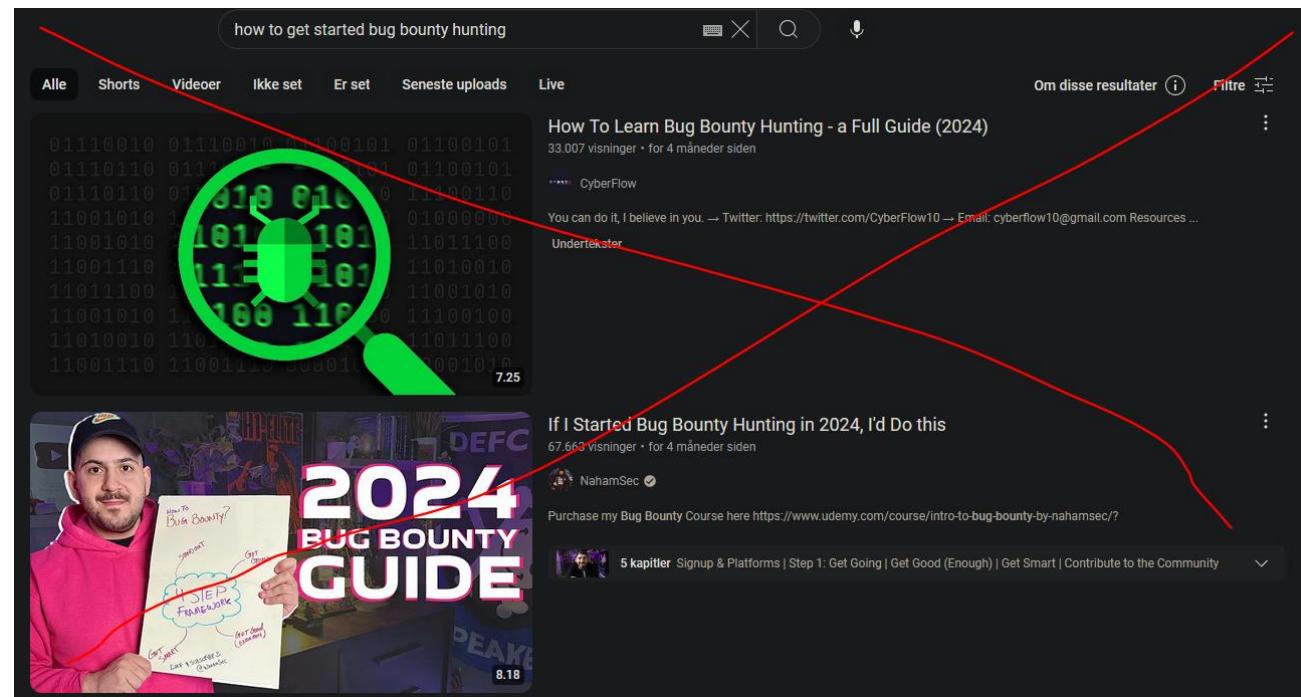
What this talk will be

- - Overview of bug bounty hunting:
 - Covering the basics of what bug bounty hunting is
 - The economics of crowdsourced security
 - Some thoughts about the state of bug bounty hunting in 2025
- How to get started and how to succeed
 - Finding a program and starting to hack
 - How to report bugs the right way
 - Common pitfalls to avoid
- ~1 hour long presentation



What this talk won't be

- Technical deep dives into different bug classes
 - There are better resources for this on the internet
 - Not suited for a talk
 - Would be too superficial to do in 30 minutes
- Me disclosing bugs I have found
- A surefire instructional guide for you to make money on bug bounty hunting
 - Hunting for bugs != actually finding valuable bugs
 - Finding a unique valuable bug can be hard
 - There isn't a direct guide, it requires creativity
- Non Web bug bounty programs
 - Blockchain, binary, scada/OT, whitebox bug bounty hunting won't be covered



Transparancy on my own journey

- Transparency is important for this talk
 - I don't want to give you the impression this is super lucrative
 - Nor do I feel I need to hide what I've 'earned'
- ~600 hours of hunting Feb 2024 – Feb 2025
- ~135.000 dkk of earnings in this timeframe
 - Most time spent on the same closed programs
 - 6 criticals 5 highs, 20 mediums, 4 lows
 - 28 eur (216 dkk) / hour for the work
- Payout can vary a lot
 - Some programs pay a lot for mediums, others little
 - Criticality of bugs found are different between companies

Yay!

Great news! Your submission **ATO leads to full compromise of [REDACTED]** has been awarded a **bounty**.

Please make sure your information is complete and your identity is checked on the Intigriti platform so we can transfer your funds. You can find all information regarding payouts and taxes in the following [knowledge base article](#).

Feel free to share the good news on X (former Twitter), but don't forget that you can't share any details on the vulnerability without explicit permission of **Schibsted**.

When sharing, we will automagically add a snapshot of your profile to your Tweet.

Oxlime

338pts	REP. ALL TIME
0pts	REP. LAST 90 DAYS
N/A	STREAK
DK	COUNTRY

Share on X

Transparancy on my own journey

- Transparency is important for this talk
 - I don't want to give you the impression this is super lucrative
 - Nor do I feel I need to hide what I've 'earned'
- ~600 hours of hunting Feb 2024 – Feb 2025
- ~135.000 dkk of earnings in this timeframe
 - Most time spent on the same closed programs
 - 6 criticals 5 highs, 20 mediums, 4 lows
 - 28 eur (216 dkk) / hour for the work
- Payout can vary a lot
 - Some programs pay a lot for mediums, others little
 - Criticality of bugs found are different between companies

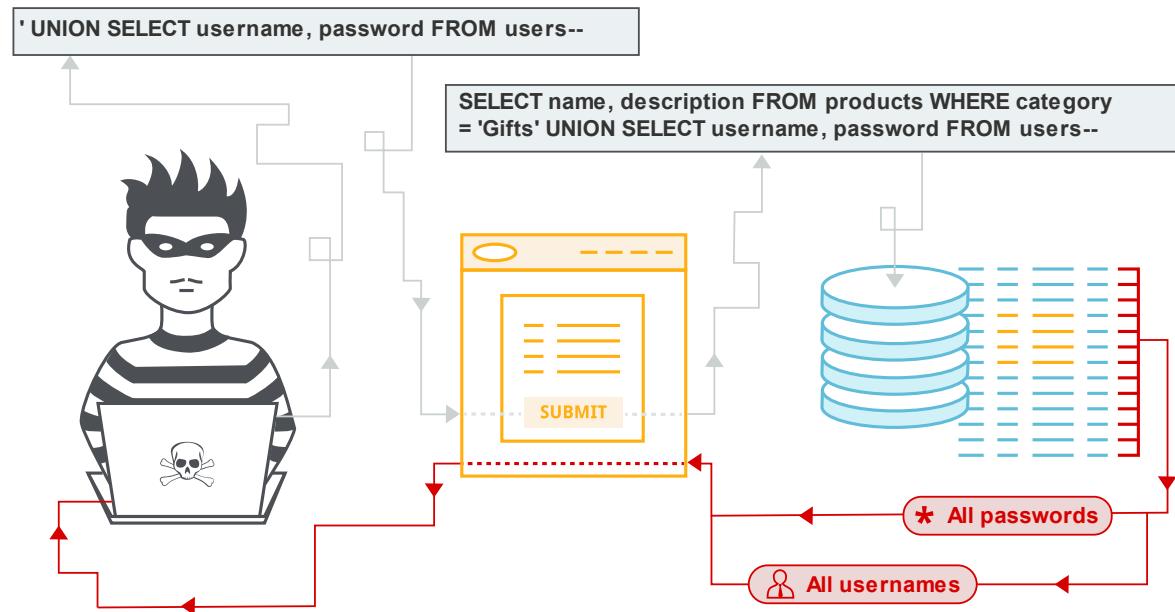
PAYMENT METHOD	TYPE	AMOUNT	STATUS
PayPal @gmail.com	Bounty	€332	Paid
PayPal @gmail.com	Bounty	€1,632	Paid
PayPal @gmail.com	Bounty	€50	Paid
PayPal @gmail.com	Bounty	€75	Paid

Lets get started

Bug bounty hunting involves finding and reporting vulnerabilities in software systems to earn rewards. It's a collaborative effort between researchers, middle-men and companies to improve security.



Scenario: You just found SQL injection in a large car companys login form



Value	Type
{ASP.sqlinjection_aspx}	SQLInjection {
{ASP.sqlinjection_aspx}	object {ASP.sc
{System.EventArgs}	System.Event/
null	System.Data.S
"Data Source=MICHAEL\\MIKE;Initial Catalog=NorthWind;In	string
"Select * From Products Where ProductID = 5;Drop Table Ac	string
"Select * From Products Where ProductID = 5;Drop Table Admin--"	

What now???

Different types of bug disclosures



Full disclosure

Finding a security vulnerability and publicly disclosing the details around it as early as possible



Responsible disclosure

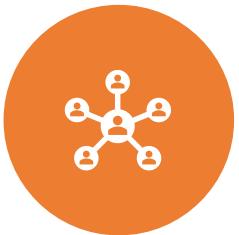
Finding a security vulnerability and giving the vendor a 'heads up' for them to create a fix. The vulnerability is disclosed after providing a fix to vendors/customers



Private disclosure

Finding a security vulnerability and disclosing it to the vendor, without ever going public with it

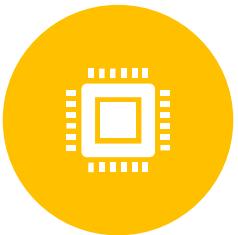
Bug bounty actors



PLATFORMS: THIRD-PARTY SERVICES (E.G., HACKERONE, BUGCROWD) THAT HOST BUG BOUNTY PROGRAMS, CONNECT RESEARCHERS WITH ORGANIZATIONS, AND MANAGE WORKFLOWS.



PROGRAMS: ORGANIZATIONS OR COMPANIES OFFERING REWARDS FOR VULNERABILITIES IN THEIR SYSTEMS, DEFINING SCOPE, RULES, AND PAYOUTS.



RESEARCHERS: ETHICAL HACKERS WHO PROACTIVELY IDENTIFY AND REPORT SECURITY FLAWS IN EXCHANGE FOR REWARDS AND RECOGNITION.



TRIAGERS: SPECIALISTS (OFTEN PLATFORM OR PROGRAM STAFF) WHO VALIDATE, PRIORITIZE, AND MEDIATE VULNERABILITY REPORTS FOR REMEDIATION.



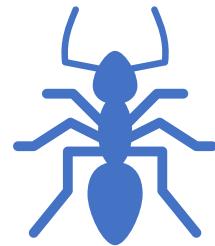
SECURITY TEAMS: INTERNAL TEAMS AT ORGANIZATIONS THAT PATCH REPORTED VULNERABILITIES AND IMPLEMENT LONG-TERM FIXES.

BBH VS VDP



Bug bounty hunting programs are where organizations reward individuals for finding and reporting security vulnerabilities in their systems.

- Private disclosure or responsible disclosure
- Monetary rewards or similar are given to the researcher for their report
- The amount paid correlates with the seriousness of the vulnerability
- RCE usually pays higher than finding cross site scripting for example



Vulnerability disclosure programs (VDP) are more open ended ‘guidelines’ for how a security researcher should disclose their found vulnerabilities

- No money is involved
- The researcher may be rewarded ‘points’ on the site associated with the vulnerability disclosure
- These sites may also host bug bounty programs.

VDP

 Public Open

9Altitudes / 9Altitudes - Vulnerability Disclosure Program / Detail

[Detail](#) [Leaderboard](#)

Description

The 9Altitudes Vulnerability Disclosure Program (VDP) program to review no-bounty assets.

9Altitudes is a European player with the main office in Belgium providing digital transformation for our customers focused on 3 main industry clusters – manufacturing, services, and wholesale & distribution. As a Microsoft Gold partner, we are mostly Microsoft-oriented with some own-IP and are an ever-expanding organization by way of merge & acquisition.

Bounties ⓘ

This is a responsible disclosure program without bounties.

Follow program 

Want to participate?
Feel free to join in, this is a public program

This program is publicly available to all researchers.
Good luck and happy hunting!

Create submission 

Bug bounty program

 Public Open

BMW/BMW Group Automotive / Detail

[Detail](#) [Leaderboard](#)

[Follow program](#) 

Description

The BMW Group looks forward to working with the security community to find vulnerabilities in order to keep its products and customers safe and secure. We are committed to working with you to verify, reproduce, and respond to legitimate reported vulnerabilities covered by this policy. Within this program bounties can be received by reporting vulnerabilities that are in the scope of program and marked as "Eligible". Please take note of the current scope outlined below.

Bounties ⓘ

	Low 0.1 - 3.9	Medium 4.0 - 6.9	High 7.0 - 8.9	Critical 9.0 - 9.4	Exceptional 9.5 - 10.0
Tier 1 €	500	2,000	5,000	10,000	15,000
Tier 2 €	100	500	1,000	2,000	5,000

Want to participate?
Feel free to join in, this is a public program

This program is publicly available to all researchers.
Good luck and happy hunting!

[Create submission](#)

[Ask scope question](#) >

[View my submissions](#) > 

Bug bounty platforms

- Several bug bounty **platforms** exists that facilitate programs.
 - These platforms provide all communication, payout, vulnerability triage, etc. Between security researchers and the bug bounty programs.
 - These platforms take a cut from the program owners for each vulnerability disclosed
 - Alternatively the programs pay a fixed fee to be on the platform.

hackerone



YesWeHack



bugcrowd

Bug bounty platforms – big players

- Huge companies like google, facebook, apple etc. Host their own bug bounty platform.

The screenshot shows the homepage of the Meta Bug Bounty platform. At the top, there is a dark header bar with the Meta logo, a "Meta Bug Bounty" dropdown menu, and links for "Tools", "Leaderboard", and "Learn". On the far right of the header is a blue button labeled "Submit a report". Below the header, the background is a dark gradient. In the center, the "Meta" logo is displayed above the large, white, sans-serif text "Meta Bug Bounty". Below this title, a message reads: "If you believe you have found a security vulnerability on Meta (or another member of the Meta family of companies), we encourage you to let us know right away." Underneath this message is another blue "Submit a report" button. At the bottom of the page, there is a black footer section containing the text "Bug Bounty rewards" on the left, and two pieces of information on the right: "Total rewards for 2024" followed by "\$1,113,126" and "Total rewards to date" followed by "\$15,606,146".

Meta

Meta Bug Bounty

If you believe you have found a security vulnerability on Meta (or another member of the Meta family of companies), we encourage you to let us know right away.

Submit a report

Bug Bounty rewards

Total rewards for 2024
\$1,113,126

Total rewards to date
\$15,606,146

How? – Researchers perspective



Starts with the **hacker** who wants
to earn some money from bug bounty hunting

How? – Researchers perspective



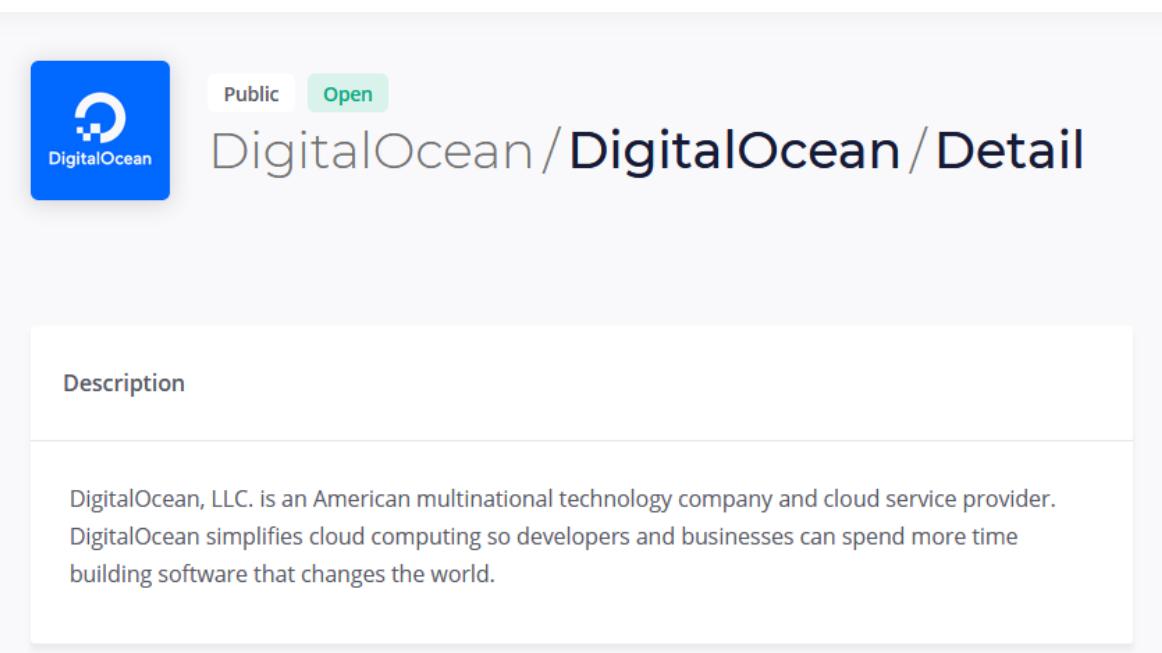
Starts with the **security researcher** who wants to earn some money from bug bounty hunting

How? – Researchers perspective



They sign up to one of the bug bounty platforms

How? – Researchers perspective



A screenshot of a DigitalOcean project detail page. The page has a blue header bar with the DigitalOcean logo and navigation buttons for 'Public' and 'Open'. The main title is 'DigitalOcean/DigitalOcean/Detail'. Below the title, there's a 'Description' section containing the following text: 'DigitalOcean, LLC. is an American multinational technology company and cloud service provider. DigitalOcean simplifies cloud computing so developers and businesses can spend more time building software that changes the world.'



They find a program they want to hunt for vulnerabilities on

How? – Researchers perspective

 Public Open

DigitalOcean / DigitalOcean / Detail

Description

DigitalOcean, LLC. is an American multinational technology company and cloud service provider. DigitalOcean simplifies cloud computing so developers and businesses can spend more time building software that changes the world.



Safe harbour for researchers is applied

DigitalOcean considers ethical hacking activities conducted consistent with the Researcher Guidelines,

the Program description and restrictions (the Terms) to constitute "authorized" conduct under criminal law.

DigitalOcean will not pursue civil action or initiate a complaint for accidental, good faith violations,

nor will they file a complaint for circumventing technological measures used by us to protect the scope as part of your ethical hacking activities.

If legal action is initiated by a third party against you and you have complied with the Terms, DigitalOcean will take steps to make it known that your actions were conducted in compliance and with our approval.

[Hide safe harbour ^](#)

The security researcher accepts the 'Safe Harbor policy'

A “safe harbor” is a provision that offers protection from liability in certain situations, usually when certain conditions are met. In the context of security research and vulnerability disclosure, it is a statement from an organization that hackers engaged in Good Faith Security Research and ethical disclosure are authorized to conduct such activity and will not be subject to legal action from that organization.

Hackerone Safe Harbor FAQ

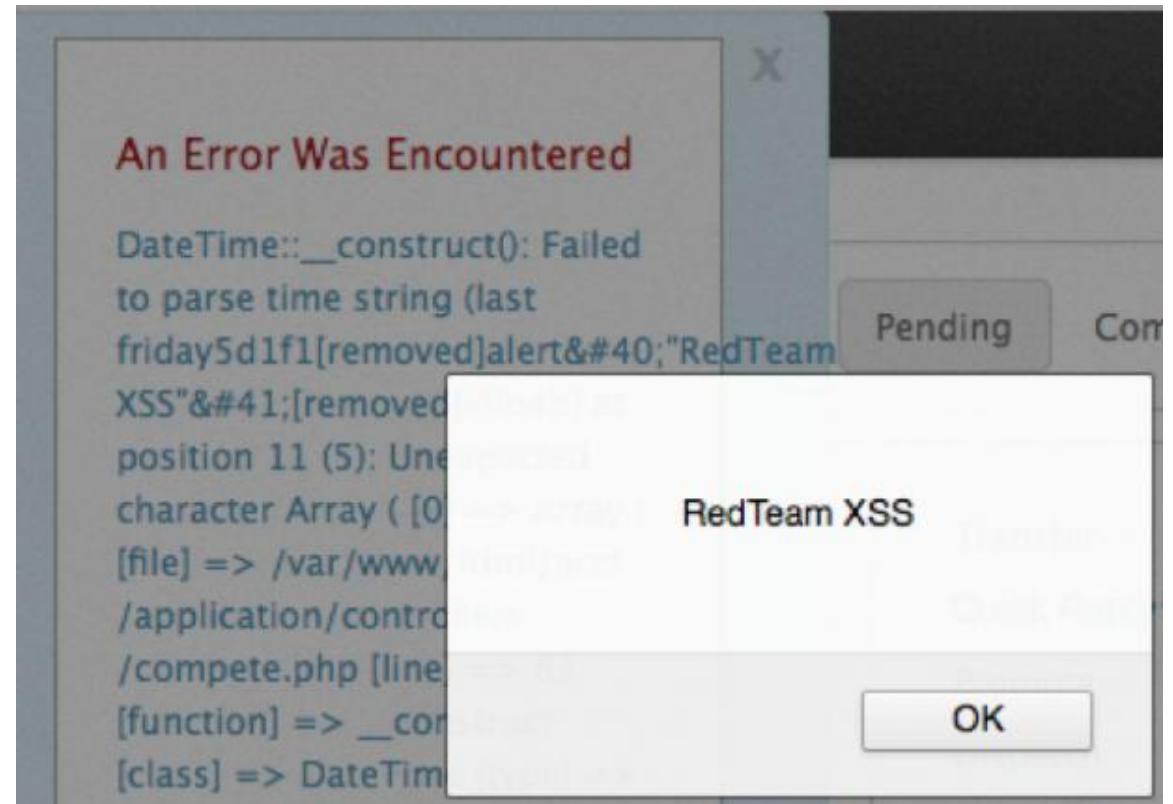
How? – Researchers perspective

- The researchers sees what is in scope and what is out of scope

Domains ⓘ			Give feedback >
TIER	TYPE	Filter text	
All	All	Show all descriptions ▾	
		*.digitalocean.com ⓘ	Tier 2 Wildcard
		169.254.169.254 ⓘ	Tier 2 IP Range
		api.digitalocean.com ⓘ	Tier 2 URL
		cloud.digitalocean.com ⓘ	Tier 2 URL
		*.db.ondigitalocean.com ⓘ	Out of scope Wildcard
		*.digitalceanspaces.com ⓘ	Out of scope Wildcard
		*.doserverless.co ⓘ	Out of scope Wildcard
		*.k8s.ondigitalocean.com ⓘ	Out of scope Wildcard
		*.ondigitalocean.app ⓘ	Out of scope Wildcard
		Assets created by other DigitalOcean customers ⓘ	Out of scope Other

How? – Researchers perspective

- The security researcher then finds a vulnerability
- This could be a stored xss vulnerability



How? – Researchers perspective

- The security researcher then writes a detailed report on the issue
 - This is a critical step; this should be done as professionally as possible.
- The report should include everything needed to replicate and understand impact
 - Show a detailed proof of concept (POC)
 - Provide all steps to reproduce
 - Explain the impact of this finding (what can an adversary do with this vulnerability)

A screenshot of a Twitter post from the account @giddsec. The post is titled "#485748 Stored XSS on reports." It has 218 likes and a timestamp of January 25, 2019, at 8:22am UTC. The post includes a detailed report about stored XSS, its description, steps to reproduce, a PoC, and its impact. The report states that stored XSS is more damaging than non-persistent XSS and occurs when a malicious script is injected directly into a vulnerable web application. It provides four steps to reproduce the vulnerability and notes that it has been tested on Firefox and Chrome. The impact is described as an attacker being able to steal data from anyone who checks the report.

#485748
Stored XSS on reports.

218

giddsec submitted a report to X (Formerly Twitter).

Summary:
Stored XSS can be submitted on reports, and anyone who will check the report the XSS will trigger.

Description:
Stored XSS, also known as persistent XSS, is the more damaging than non-persistent XSS. It occurs when a malicious script is injected directly into a vulnerable web application.

Steps To Reproduce:

1. Go to <https://app.mopub.com/reports/custom/>
2. Click New network report.
3. On the name, enter payload: ">
4. Click Run and save then XSS will trigger.

Demonstration of the vulnerability:
PoC: [REDACTED]

Tested on Firefox and chrome.

Impact
The attacker can steal data from whoever checks the report.

[Request support](#)

/ Stored XSS on information links

Code: M673R88D

LAST UPDATED 26/03/2024, 11:43:49

BOUNTY €445 Show details

CREATED 09/03/2024, 15:50:42

BONUS €0

SEVERITY Medium | 5.4 ⓘ

TYPE Stored Cross-Site Scripting

STATUS Accepted Show history

▲ Report

Domain

*.

Tier 1

Wildcard

Endpoint / vulnerable component

www contactinformation/<UUID> & picture/<UUID>

Proof of Concept / description

I am very happy to report what I believe is my first high on this program :-)

I have found a 1-click stored XSS vulnerability that a low privileged, verified user can create, through the draft feature of

When creating a listing the following requests are made when making a draft ("Gem kladde")

ID	Endpoint	Method	Request	Status	Count
10510	https://www	PUT	01e9580b-3383-45d8-a8a0-36611acf188a	200	537
10511	https://www	PUT	01e9580b-3383-45d8-a8a0-36611acf188a ✓	204	526
10512	https://www	PUT	1e9580b-3383-45d8-a8a0-36611acf188a ✓	204	526
10513	https://www	PUT	1e9580b-3383-45d8-a8a0-36611acf188a ✓	204	526
10514	https://www	PUT	1e9580b-3383-45d8-a8a0-36611acf188a ✓	204	526
10515	https://www	PUT	1e9580b-3383-45d8-a8a0-36611acf188a ✓	200	537
10516	https://www	PUT	1=01e9580b-3383-45d8-a8a0-36611acf188a ✓	200	537

All these requests except for the endpoint allow for setting freetext, by setting the text to a unicode version of an XSS payload, it is possible to bypass cloudflare WAF and inject javascript code.

When a subsequent get request is made to that endpoint, the backend misinterprets the right content-type to give back, and gives text/html, provoking an XSS.

example requests:

PUT '01e9580b-3383-45d8-a8a0-36611acf188a HTTP/2

Host:

Cookie omitted

```
{"contactName": "\u003c\u00069\u0006d\u00067\u0020\u0073\u00072\u00063\u003d\u0027\u0027\u0020\u0006f\u0006e\u00065\u0072\u00072\u003d\u00061\u0006c\u00065\u0072\u0028\u0031\u0029\u003e", "contactPhone": "15531553", "contactAddress": "", "contactPostalCode": "1553"}
```

And its subsequent GET request at: <https://www.01e9580b-3383-45d8-a8a0-36611acf188a>

Will fire the unicode encoded XSS payload.

ProjectDiscovery is the



Pinned

[nuclei](#) Public

Nuclei is a fast, customizable vulnerability scanner powered by the global security community and built on a simple YAML-based DSL, enabling collaboration to tackle trending vulnerabilities on the ...

● Go ⭐ 22.2k 🏷 2.6k

[nuclei-templates](#) Public

Community curated list of templates for the nuclei engine to find security vulnerabilities.

● JavaScript ⭐ 9.7k 🏷 2.7k

[subfinder](#) Public

Fast passive subdomain enumeration tool.

● Go ⭐ 11.2k 🏷 1.3k

[httpx](#) Public

httpx is a fast and multi-purpose HTTP toolkit that allows running multiple probes using the retryablehttp library.

● Go ⭐ 8.1k 🏷 873

[naabu](#) Public

A fast port scanner written in go with a focus on reliability and simplicity. Designed to be used in combination with other tools for attack surface discovery in bug bounties and pentests

● Go ⭐ 5k 🏷 574

[cvemap](#) Public

Navigate the CVE jungle with ease.

● Go ⭐ 1.9k 🏷 127

Different approaches – Deep dive into functionality

- Goal: Understand the target so well that conflicting functionality becomes apparent, the goal is to find logic bugs.
- Why? Logic bugs can be the most serious vulnerabilities, breaking confidentiality or integrity.
- How to navigate this approach?
 - Use the application normally for some time
 - Observe requests being made to understand the structure and architecture of the site
 - What technologies can be inferred to be in use?
 - How does the application respond to bad input?
 - What does the auth model look like?
- This approach is more **manual**

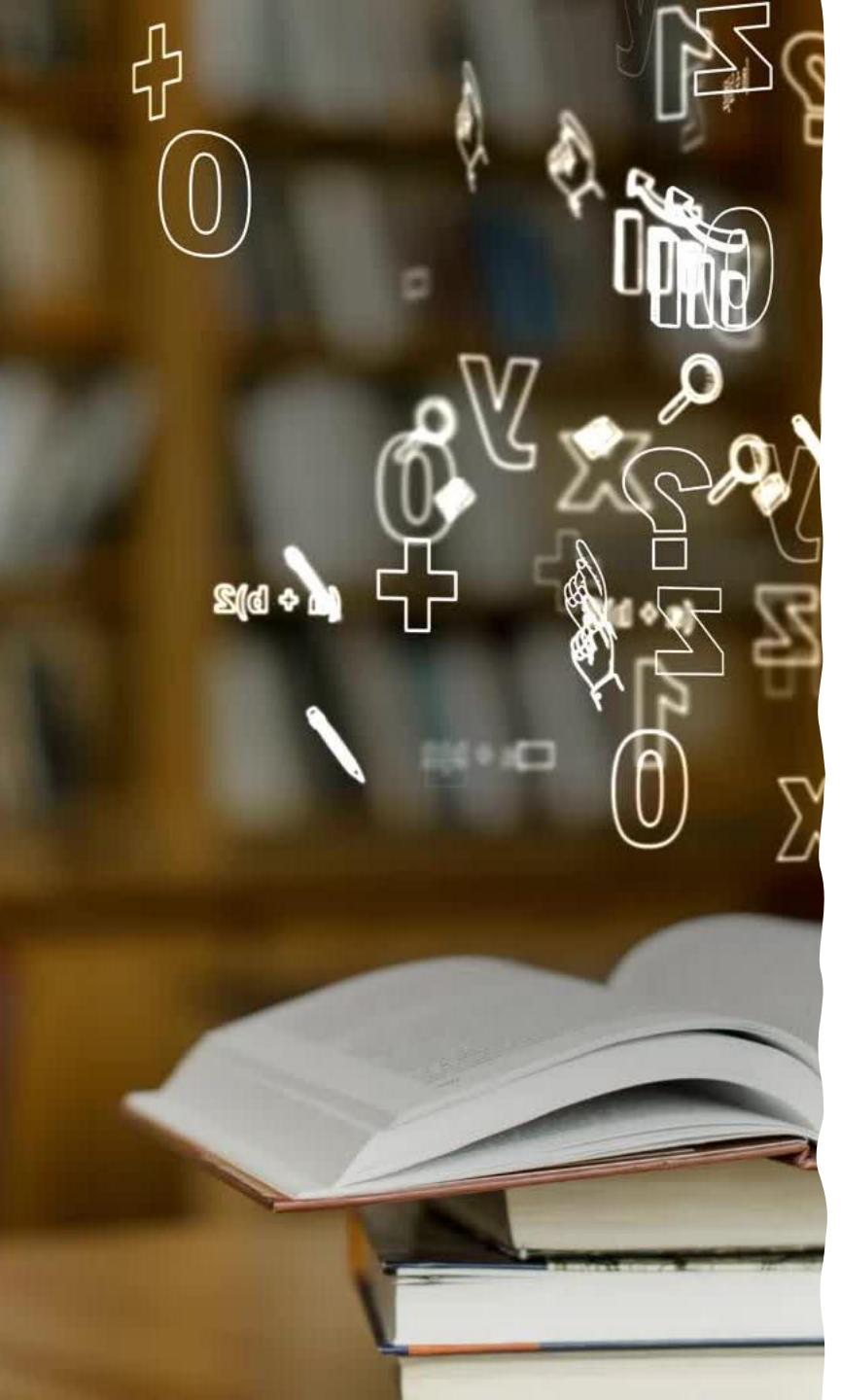


Deep dive vulnerabilities

I will just point to portswigger academy here

The screenshot shows a section of the PortSwigger Academy website dedicated to deep-dive vulnerabilities. The main title is "Deep dive vulnerabilities". Below it, a statement says "I will just point to portswigger academy here". The main content area is titled "Deep dive vulnerabilities" and lists nine topics with their descriptions and lab counts:

Topic	Description	Labs
SQL injection	SQL injection is an old-but-gold vulnerability responsible for many high-profile data breaches. Although relatively simple to learn, it can potentially be used for some high-severity exploits. This makes it an ideal first topic for beginners, and essential knowledge even for more experienced users.	18 Labs
Authentication	Go to topic →	14 Labs
Path traversal	Go to topic →	6 Labs
Command injection	Go to topic →	5 Labs
Business logic vulnerabilities	Go to topic →	11 Labs
Information disclosure	Go to topic →	5 Labs
Access control	Go to topic →	13 Labs
File upload vulnerabilities	Go to topic →	7 Labs
Race conditions	Go to topic →	6 Labs



Where to practice & gain knowledge

- Play CTF
 - Focus on web category
- Play Hackthebox
 - Web category or machines heavy on web
- Portswigger academy
 - This is where I would go and where I still go to learn
- Certifications
 - Can be expensive, but the notion of an exam may help motivate
- Published reports
 - Find published reports on h1 etc and learn from them

Additional resources



Critical Thinking Bug bounty podcast

Super technical podcast on hacking



Hacktricks

Go to site to get tips on hacking methodology



Web application hackers handbook

By Dafydd Stuttard and Marcus Pinto



The daily swig blog

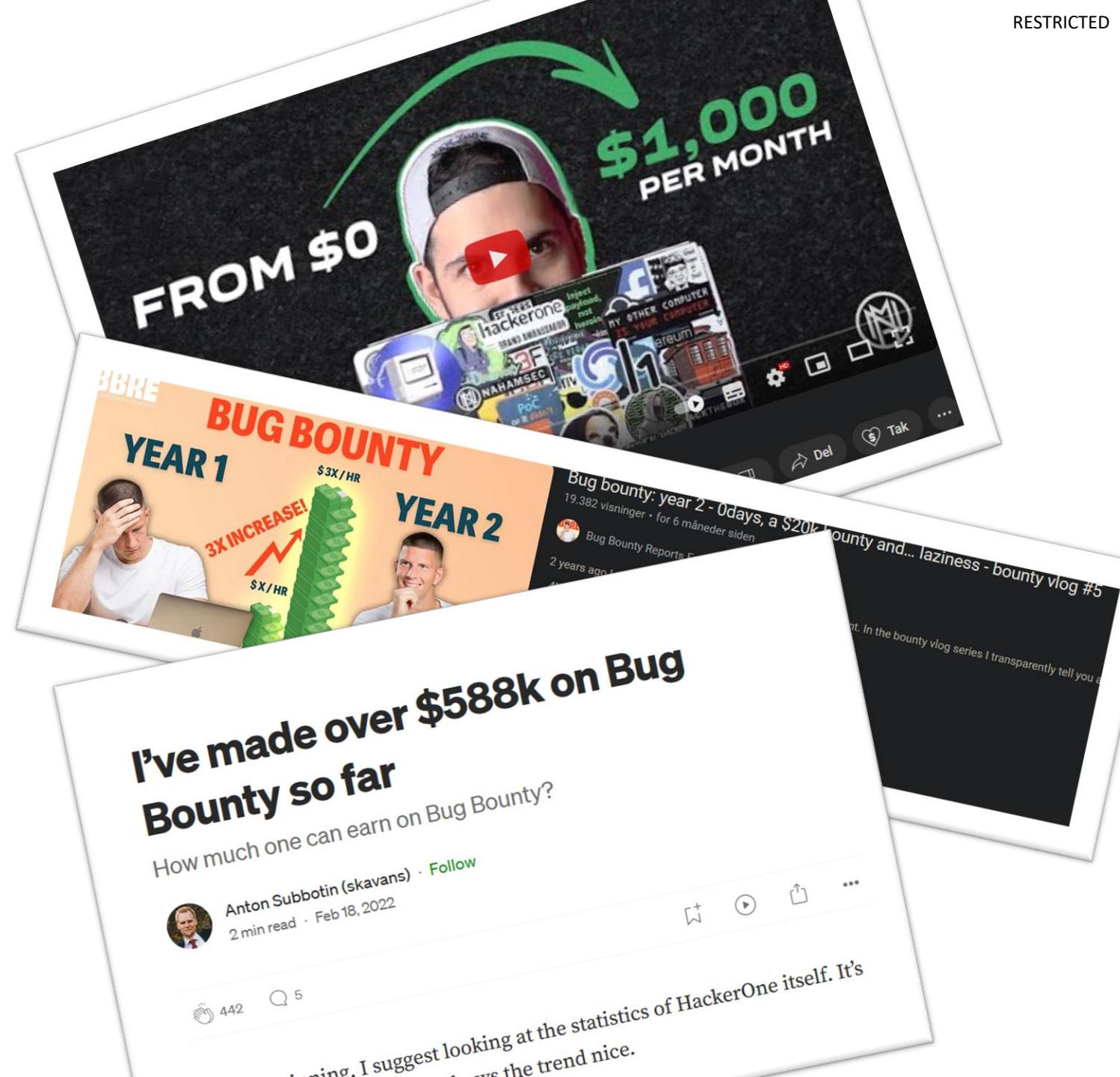
Portswiggers cybersecurity news blog

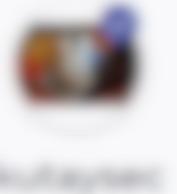
Lastly a few random thoughts

A word on good bug bounty hunting culture

And how not to behave

- Bug bounty is becoming more and more popular
- Many content creators shill it with promises of earning tons of money
 - Oftentimes they also sell courses to or bootcamps to "zero to hero"
- This creates a huge influx in hunters
 - Not in itself a problem
 - But problematic if everyone thinks their issues are important
- Creates a huge increase in 'low value reports'
- Fucks up the triagers day
 - And ruins it for other hunters





Profile picture

RANK

#2084

REP. ALL TIME

30 pts

REP. 90 DAYS

30 pts

STREAK

High

COUNTRY

Turkey (Türkiye)

LINKEDIN

X (Former Twitter)

Activity

	submission in Red Bull has been rejected [Out of scope] by Red Bull	1 day ago
	created submission in Red Bull from Red Bull	4 days ago
	submission in UZ Brussel has been rejected [Out of scope] by UZ Brussel	4 days ago
	reated submission in UZ Brussel from UZ Brussel	5 days ago
	submission in ! has been rejected [Informative] by Schibsted	5 days ago
	reated submission in !	6 days ago
	submission in ! has been rejected [Out of scope] by Vinterd	about 2 months ago
	created submission in !	about 2 months ago
	submission in ! private bug bounty has been rejected [Not applicable] by	about 2 months ago

Submission stats

ACCEPTED

2

VALID

37.5 %

TOTAL

8

Top contributions

The Coca-Cola Company Vulnerability ...

Libelle

Last contributions

Libelle

The Coca-Cola Company Vulnerability ...



#156098



XSS At "pages.et.uber.com"

Share: [f](#) [t](#) [in](#) [y](#) [o](#)

TIMELINE



raghav_bish submitted a report to Uber.

August 2, 2016, 2:23pm UTC

Vulnerable Domain :<https://pages.et.uber.com/>**Vulnerable Link :**https://pages.et.uber.com/icecream/?lang_id=5**Edited Link With Payload :**[https://pages.et.uber.com/icecream/?lang_id=5%22%20onmouseover%3dprompt\(document.domain\)%20bad%3d%22](https://pages.et.uber.com/icecream/?lang_id=5%22%20onmouseover%3dprompt(document.domain)%20bad%3d%22)[https://pages.et.uber.com/icecream/?lang_id=5%22%20onmouseover%3dprompt\(document.cookie\)%20bad%3d%22](https://pages.et.uber.com/icecream/?lang_id=5%22%20onmouseover%3dprompt(document.cookie)%20bad%3d%22)**Payload Used :**

" onmouseover%3dprompt(9020) bad%3d"

" onmouseover%3dprompt(document.domain) bad%3d"

" onmouseover%3dprompt(document.cookie) bad%3d"

5 attachments:

F109155: xss-1.JPG

F109156: source.JPG

F109157: xss-2.JPG

F109158: xss-3.JPG

F109159: xss-4.JPG

bugtriage-rob closed the report and changed the status to ● Informative.

August 3, 2016, 6:56pm UTC

Thanks for your report.

While we appreciate your efforts to help keep Uber secure, I'm afraid this doesn't qualify for this program as the domain *.et.uber.com is out of scope for this program. You can find the list of in-scope properties on our program page: <hackerone.com/uber>

Thanks and good luck in your future bug hunting.



raghav_bish posted a comment.

Respected...

Its a request Once you patched the vulnerability "Do disclose the report"

August 16, 2016, 10am UTC

 [lyoung-uber](#) reopened this report. August 16, 2016, 4:54pm UTC

 [lyoung-uber](#) closed the report and changed the status to ● Not Applicable. August 16, 2016, 4:54pm UTC
Closing as Not Applicable since this is out-of-scope.

 [raghav_bisht](#) posted a comment. August 16, 2016, 5:12pm UTC
@lyoung-uber you fucking asshole mother fucker I know this is "Out of scope" and your team member [@bugtriage-rob](#) marked it has Informative and closed the report, still I didn't argue about it and accepted it.....fucker.
I respectfully asked you to disclosure my report and you moron mother fucker deducted my Reputation Point

Bloody Mother Fucker TAXI DRIVER.....

 [lyoung-uber](#) posted a comment. August 16, 2016, 5:34pm UTC
Hi [@raghav_bisht](#),

First off I wanted to apologize for not writing a longer response when I updated the report state, that's my fault. However as you acknowledged yourself this is not in scope per our hackerone.com/uber:

Out-of-scope Properties

*.et.uber.com - The underlying software here is exacttarget which Uber does not have control over.

It's important that our reports are tracked correctly both for HackerOne's statistics and our own internal metrics. With that said, that absolutely does not excuse your behavior:

I respectfully asked you to disclosure my report and you moron mother fucker deducted my Reputation Point
Bloody Mother Fucker TAXI DRIVER.....

Consider this your **only warning** that any similar behavior or violation of the hackerone.com/uber (such as public disclosure of in-scope bugs before they are remediated) will result in a ban from our program.

 Ritik Raj 
@Cyber_Ritik

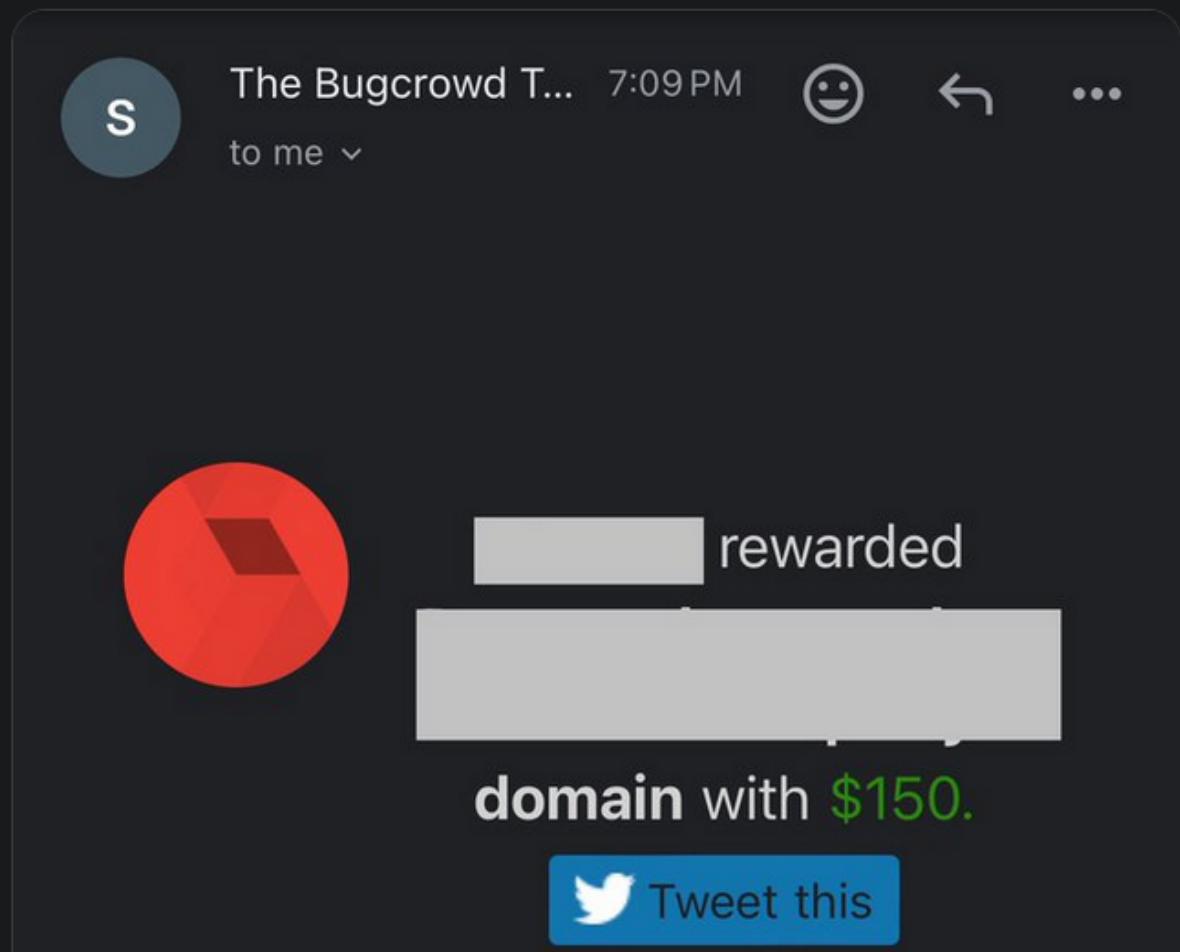
...

After 50 dupes, 70 N/A. I finally earned my first ever bounty from
@Bugcrowd !

Hard work always pays off 

Thanks @ADITYASHENDE17

#bugbounty #infosec #cybersecurity #ItTakesACrowd



Issue detail:-

The web server contains a robots.txt file.

Issue background:-

The file robots.txt is used to give instructions to web robots, such as search engine crawlers, about locations within the web site that robots are allowed, or not allowed, to crawl and index.

The presence of the robots.txt does not in itself present any kind of security vulnerability. However, it is often used to identify restricted or private areas of a site's contents. The information in the file may therefore help an attacker to map out the site's contents, especially if some of the locations identified are not linked from elsewhere in the site. If the application relies on robots.txt to protect access to these areas, and does not enforce proper access control over them, then this presents a serious vulnerability.

Issue remediation:-

The robots.txt file is not itself a security threat, and its correct use can represent good practice for non-security reasons. You should not assume that all web robots will honor the file's instructions. Rather, assume that attackers will pay close attention to any locations identified in the file. Do not rely on robots.txt to provide any kind of protection over unauthorized access.

We have found more bugs/vulnerability in your website. Kindly clarify if there is any payout if we disclose them to you?

We understand but my team worked very hard to find these bugs in your website. We have found more. If you can pay us small token of appreciation 100-150\$ we will submit all of our reports.

A word on healthy hunting mentality

- Finding a bug is exhilarating and an awesome feeling
- However try to compose yourself and not get too excited
 - It may be a duplicate
 - The company may push down the severity
 - You may have missed some 'out of scope' or 'known issues' section
- Incorporate the 'submit and forget' mentality
 - Send in the report, answer the question
- Celebrate when a payout is registered.
 - Its better to not keep thinking about your reports and the potential earnings.



Request

Pretty Raw Hex

Response

Pre

```
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
4 Content-Type: application/json
5 Date: Mon, 08 Jul 2024 11:52:43 GMT
6 Expires: 0
7 Pragma: no-cache
8 Server: Caddy
9 Server: Skipper
10 Strict-Transport-Security: max-age=31536000 ; includeSubDomains
11 Vary:
origin,access-control-request-method,access-control-request-headers,accept-encoding
12 X-Content-Type-Options: nosniff
13 X-Frame-Options: DENY
14 X-Xss-Protection: 1; mode=block
15 Content-Length: 862
16
17 {
    "id": 45400,
    "email": "a@example@gmail.com",
    "password_hash": "$2a$08$ju",
    "created_at": "2024-07-06T13:29:05.300696Z",
    "updated_at": "2024-07-06T13:29:05.300697Z",
    "password_reset_token": null,
    "password_reset_token_expiration_date": null,
    "login_count": 1,
    "first_name": "a",
    "last_name": "aad",
    "phone": "+45 33224421",
    "preferred_language": "en",
    "activation_token":
    "74bd2c14b04",
    "activation_token_expiration_date": "2024-08-05T13:29:05.300711Z",
    "admin_granted_by": null,
    "admin": false,
    "uuid": "3b6d5627-a692-4b20-916f-2f35a5462455",
    "position": null,
    "deleted_at": null,
```



Search



0 highlights



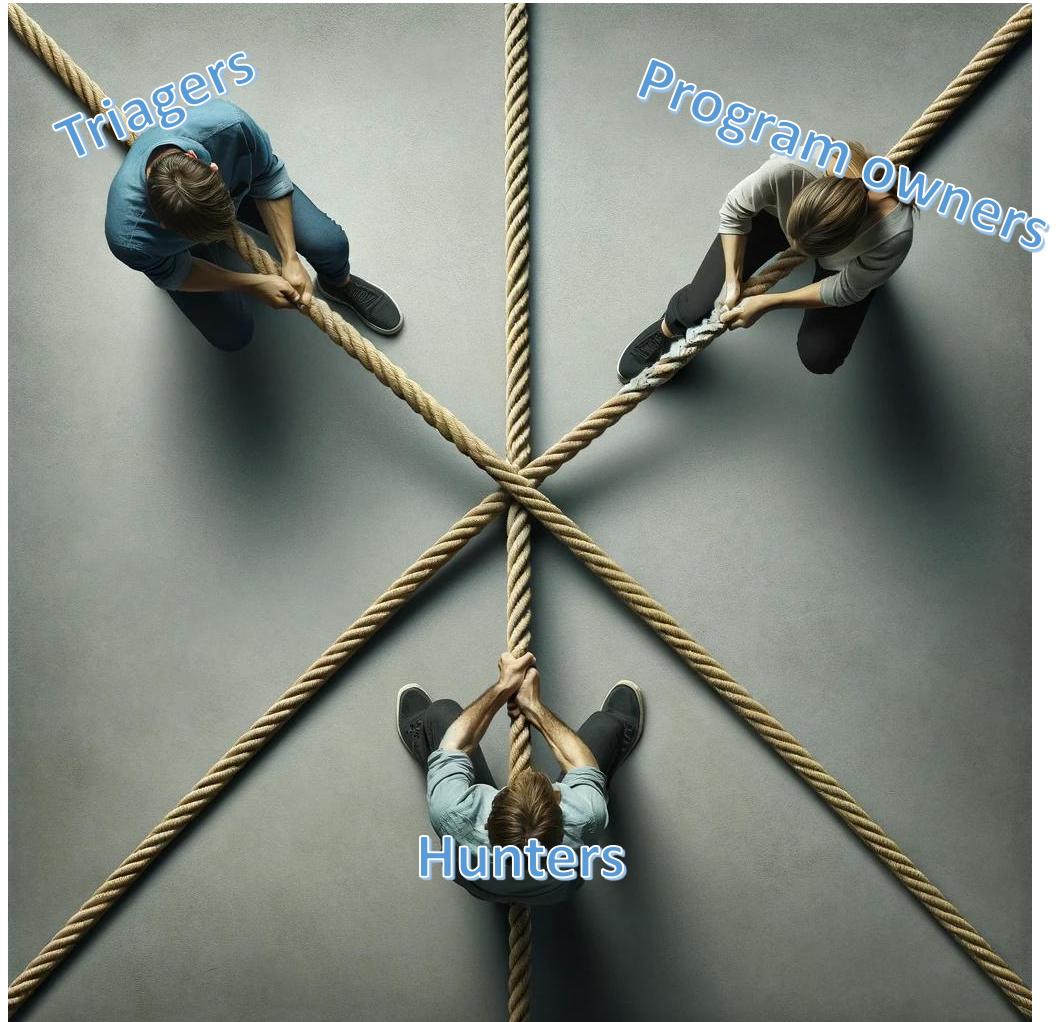
0 highlights

Done

 Zalando SE / Zalando Bug Bounty / IDOR leads to mass user info leakage			
Code: ZALANDO-5P9FB2LU			
LAST UPDATED	25/07/2024, 02.00.00	BOUNTY	€0
CREATED	08/07/2024, 13.53.43	BONUS	€0
SEVERITY	Critical	TYPE	Insecure Direct Object Reference
STATUS	Archived / Duplicate	DUPLICATE OF	ZALANDO-2BSHY1KA Show details
	Show history		

A word on the power dynamics

- Researcher wants severity to be high to get higher payout
- Program wants severity to be lower to provide lower payout
- Platform needs to balance this out, don't want to lose researchers, but also don't want program managers to feel that they are getting their moneys worth



A word on recent discussion around VDP's

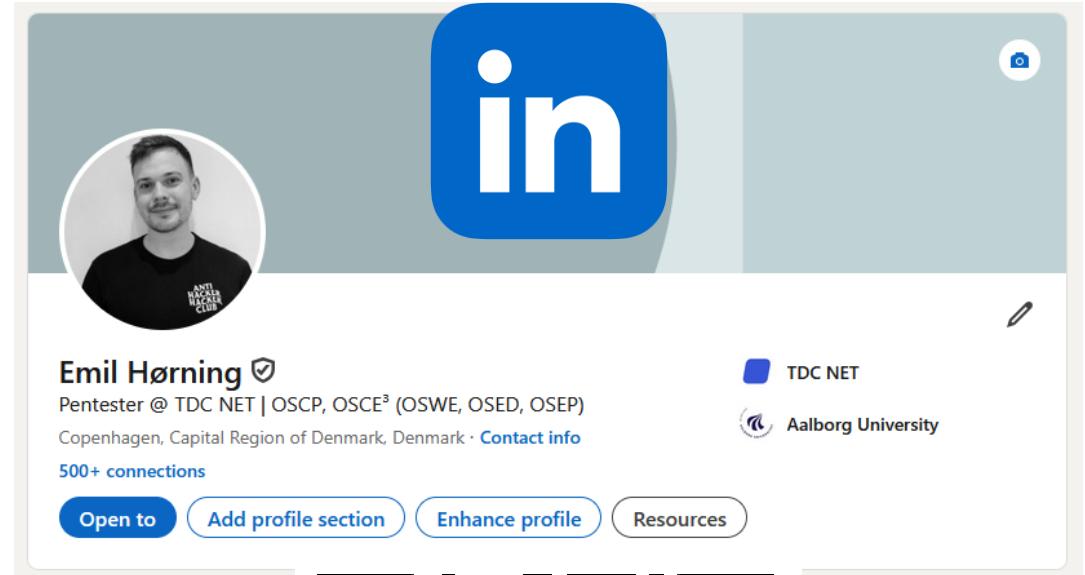
- **Resource Drain on Platforms**
 - VDPs consume significant resources from bug bounty platforms, overwhelming the system with high volumes of submissions and leading to triage team burnout.
- **Impact on Paid Bug Bounty Programs**
 - Paid programs face delays in handling critical reports due to the resource allocation towards VDPs, risking exploitation of vulnerabilities and hampering communication and feedback.
- **Substandard Report Quality**
 - Many VDP submissions are low-quality or duplicate issues, exhausting triage teams and reducing efficiency.
- **Unfair Labor Practices**
 - Researchers often work for free on VDPs, benefiting large companies without compensation, and many focus on VDPs to boost their reputation rather than finding valuable bugs in paid programs.
- **Negative Impact on Experienced Hunters**
 - Experienced hunters may miss critical vulnerabilities in paid programs due to the diversion of their efforts to VDPs, discouraging valuable contributions.

The image displays six separate screenshots of responsible disclosure programs from different companies, arranged in two columns of three. Each screenshot shows the company logo, the program name (e.g., 'Red Bull / Red Bull Public'), a green 'Open' button, a 'Responsible disclosure' link, and a timestamp indicating when it was last updated and submitted.

- Red Bull:** Last updated: 2 months ago * Last submission: 16 minutes ago
- Sixt:** Last updated: 8 days ago * Last submission: 4 days ago
- Nestlé:** Last updated: 8 days ago * Last submission: about 13 hours ago
- Ubisoft:** Last updated: 1 day ago * Last submission: about 18 hours ago
- The Coca-Cola Company:** Last updated: about 22 hours ago * Last submission: about 20 hours ago
- Revolut:** Last updated: 4 months ago * Last submission: 24 days ago

"Every bug is a story waiting to be told, and every bounty is a reward for the relentless pursuit of digital truth."

Questions?



A screenshot of a LinkedIn profile page for Emil Hørning. The profile picture shows a man with a beard in a black t-shirt. The LinkedIn logo is prominently displayed. Below the profile picture, the name "Emil Hørning" is followed by a verified badge (a blue checkmark). The bio states: "Pentester @ TDC NET | OSCP, OSCE³ (OSWE, OSED, OSEP) Copenhagen, Capital Region of Denmark, Denmark · Contact info". It also mentions "500+ connections". At the bottom, there are buttons for "Open to", "Add profile section", "Enhance profile", and "Resources". To the right of the profile, there are two company logos: "TDC NET" and "Aalborg University".

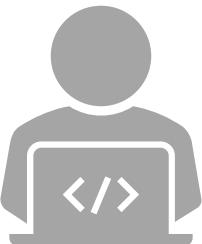


✨ Add me on LinkedIn ✨

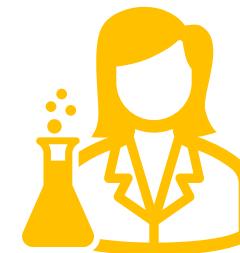
What now?



If you want to hunt – Go hunt on a platform of your choice



If you want to learn hacking - Go to portswigger academy and solve some labs



If you want to read research – Find and read papers on BBH*