The Bug Bounty Effect From DEVSECOops! To Security Success

about:me

Emil Vaagland
Head of Product Security
Vend

FINN Mulighetenes marked

Varslinger | Ny annonse | Meldinger | Min FINN

Søk etter direktør eller deltidsjobb

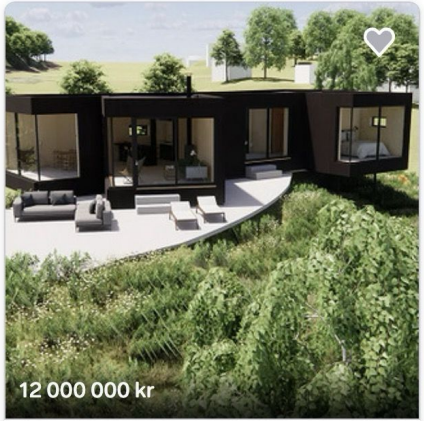
Kart

Torget | Bil og campingvogn | Reise | Båt | MC | Nettbil | Pakkereise

Jobb | Eiendom | Nybrukt elektronikk | Feriehjem og hytter til leie | Nyttekjøretøy og maskiner | Mittanbud

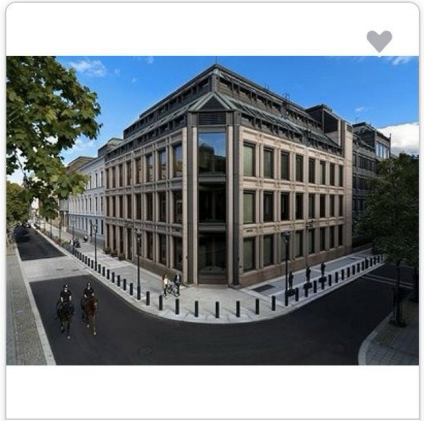
Anbefalinger til deg

Hvorfor anbefaler vi disse annonsene?



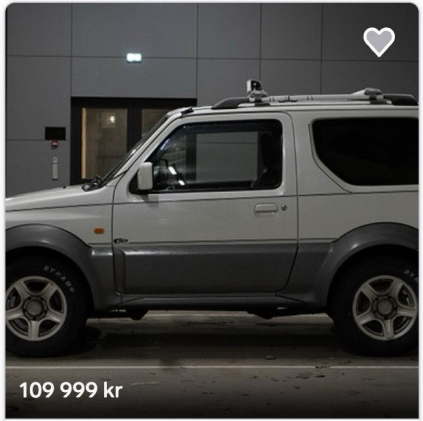
Bryggja

Eksklusiv arkitekttegnet hytte ved sjøen under oppføring - Opplev



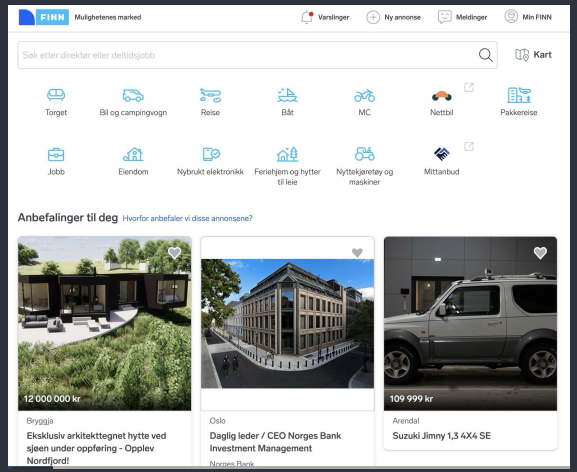
Oslo

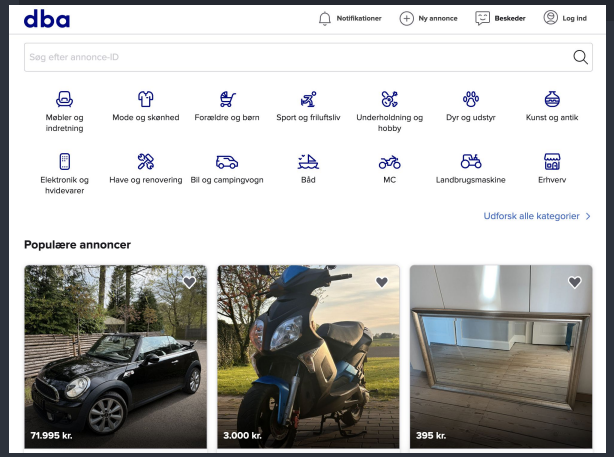
Daglig leder / CEO Norges Bank Investment Management

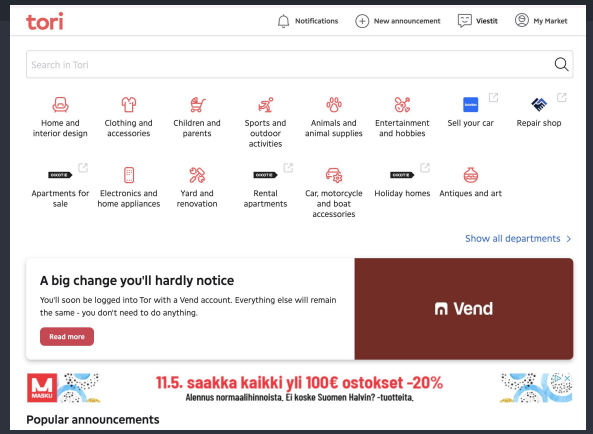


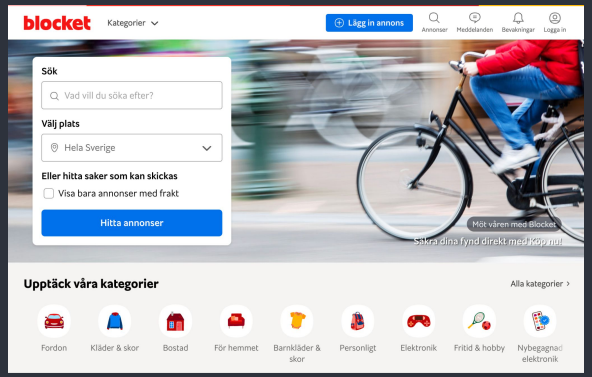
Arendal

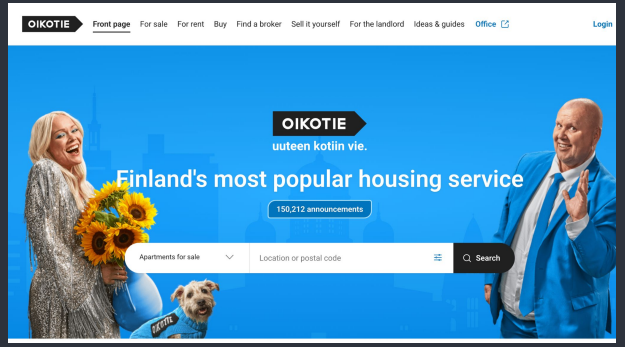
Suzuki Jimny 1,3 4X4 SE


FINN
Mulighetenes marked
Varslinger
Ny annonse
Meldinger
Min FINN
Søk etter direktør eller deltidsjobb
Kart
Torget
Bil og campingvogn
Reise
Båt
MC
Nettbil
Pakkereise
Jobb
Eiendom
Nybrukt elektronikk
Feriehjem og hytter til leie
Nyttekjøretøy og maskiner
Mittanbud
Anbefalinger til deg
Hvorfor anbefaler vi disse annonsene?
12 000 000 kr
Bryggja
Eksklusiv arkitekttegnet hytte ved sjøen under oppføring - Opplev Nordfjord!
Oslo
Daglig leder / CEO Norges Bank Investment Management
Norges Bank
109 999 kr
Arendal
Suzuki Jimny 1,3 4X4 SE


dba
Notifikationer
Ny annonce
Beskeder
Log ind
Søg efter annonce-ID
Møbler og indretning
Mode og skønhed
Forældre og børn
Sport og friluftsliv
Underholdning og hobby
Dyr og udstyr
Kunst og antik
Elektronik og hvidevarer
Have og renovering
Bil og campingvogn
Båd
MC
Landbrugsmaskine
Erhverv
Udforsk alle kategorier
Populære annoncer
71.995 kr.
3.000 kr.
395 kr.


tori
Notifications
New announcement
Viestit
My Market
Search in Tori
Home and interior design
Clothing and accessories
Children and parents
Sports and outdoor activities
Animals and animal supplies
Entertainment and hobbies
Sell your car
Repair shop
Apartments for sale
Electronics and home appliances
Yard and renovation
Rental apartments
Car, motorcycle and boat accessories
Holiday homes
Antiques and art
Show all departments
A big change you'll hardly notice
You'll soon be logged into Tor with a Vend account. Everything else will remain the same - you don't need to do anything.
Read more
Vend
11.5. saakka kaikki yli 100€ ostokset -20%
Alennus normaalihinnoista. Ei koske Suomen Halvin? -tuotteita.
Popular announcements


blocket
Kategorier
Lägg in annons
Annonser
Meddelanden
Bevakningar
Logga in
Sök
Vad vill du söka efter?
Välj plats
Hela Sverige
Eller hitta saker som kan skickas
Visa bara annonser med frakt
Hitta annonser
Möt våren med Blocket
Söka dina fynd direkt med köp nu!
Upptäck våra kategorier
Alla kategorier
Fordon
Kläder & skor
Bostad
För hemmet
Barnkläder & skor
Personligt
Elektronik
Fritid & hobby
Nybegagnad elektronik


OIKOTIE
Front page
For sale
For rent
Buy
Find a broker
Sell it yourself
For the landlord
Ideas & guides
Office
Login
OIKOTIE
uuteen kotiin vie.
Finland's most popular housing service
150,212 announcements
Apartments for sale
Location or postal code
Search


Bilbasen
Sælg din bil
Log ind
Favoritter
Kundeservice
Gjensidige
Tilliden er gjensidig
Vælg en bilforsikring, du kan have tillid til
Danmarks største markedsplads for biler
54.862 annoncer i dag
Personbil
Mærke
Model
Årgang
Kørte km
Drivmiddel
Pris
Vis 44.381 biler
Nulstil
Udvidet søgning
Populære søgninger

Some Vend Numbers for Context

- Going towards the same platform
- 1000+ applications per market
- 5000+ deployments per week
- 450+ developers / 70 teams
- All the languages and frameworks

Nice attack surface

My Security Journey (in current job)

2014-2018

Developer in product team
Small scope

- ➔ Secrets Management

2018

Security Engineer in group function
Huge scope

- ➔ Scanner orchestration, vulnerable dependencies, dynamic web app scanning

2019 - 2022

First FTE in FINN, Security Manager
Scope: one brand, 200+ developers
One-man show

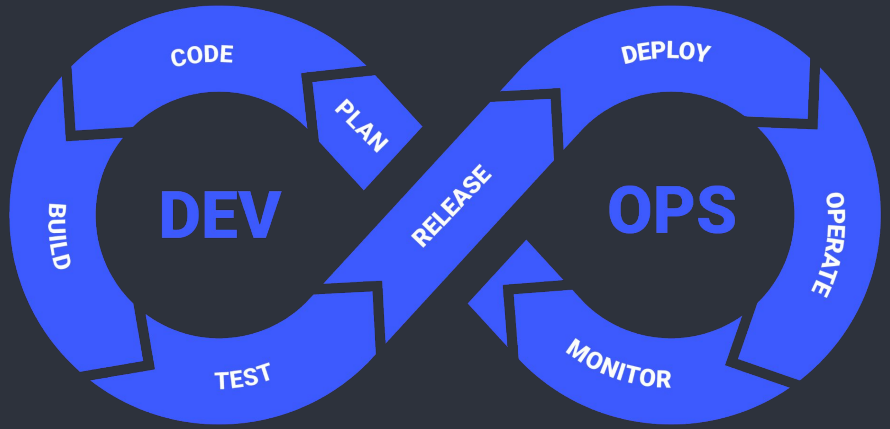
- ➔ Pentesting, bug bounty, code scanning
- ➔ Security champions

2022-2025

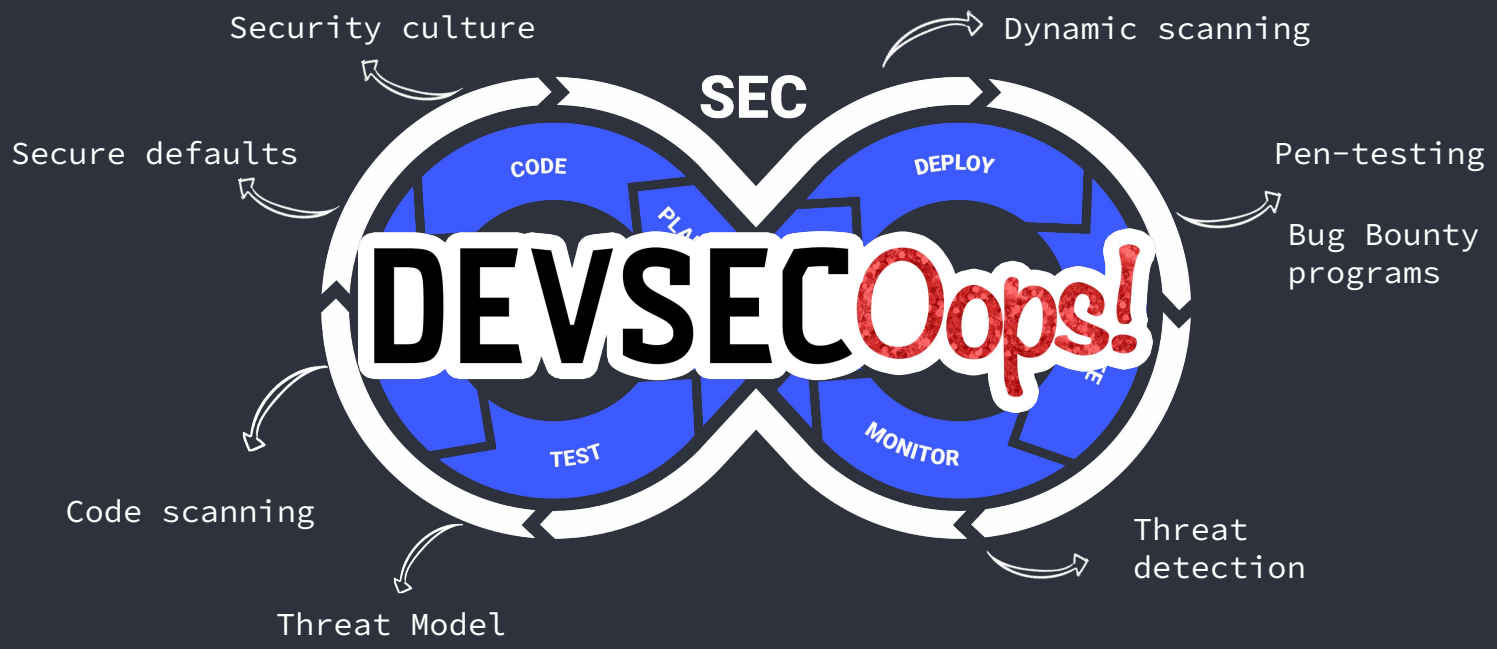
Team of 5 Security Engineers
Scope: Many brands, 450+ developers

- ➔ Cloud Security & more bug bounty programs

The DevOps Lifecycle



Sprinkle Security Across Everything



Vulnerability Sources

Vuln Sources

Different tools

Code

- CodeQL
- Dependencies
- Secrets

Dynamic

- Web app
- Attack surface
- Cloud scanning

Testing

- Pen-testing
- Bug Bounty

Internal

- Developers
- Security Team

Which source do you trust the most?

Imagine you have one critical finding from each source, which one would you prioritize first?

- Code Scanning
- Dependency (software library)
- Web App Scanning
- Attack Surface / Cloud
- Pen-test / Bug Bounty

Focus on What Matters

Verified Vulnerabilities

- Exploitable in production
- Real, provable impact
- Must be fixed
- Usually found by humans (👋AI)

Busywork Vulnerabilities

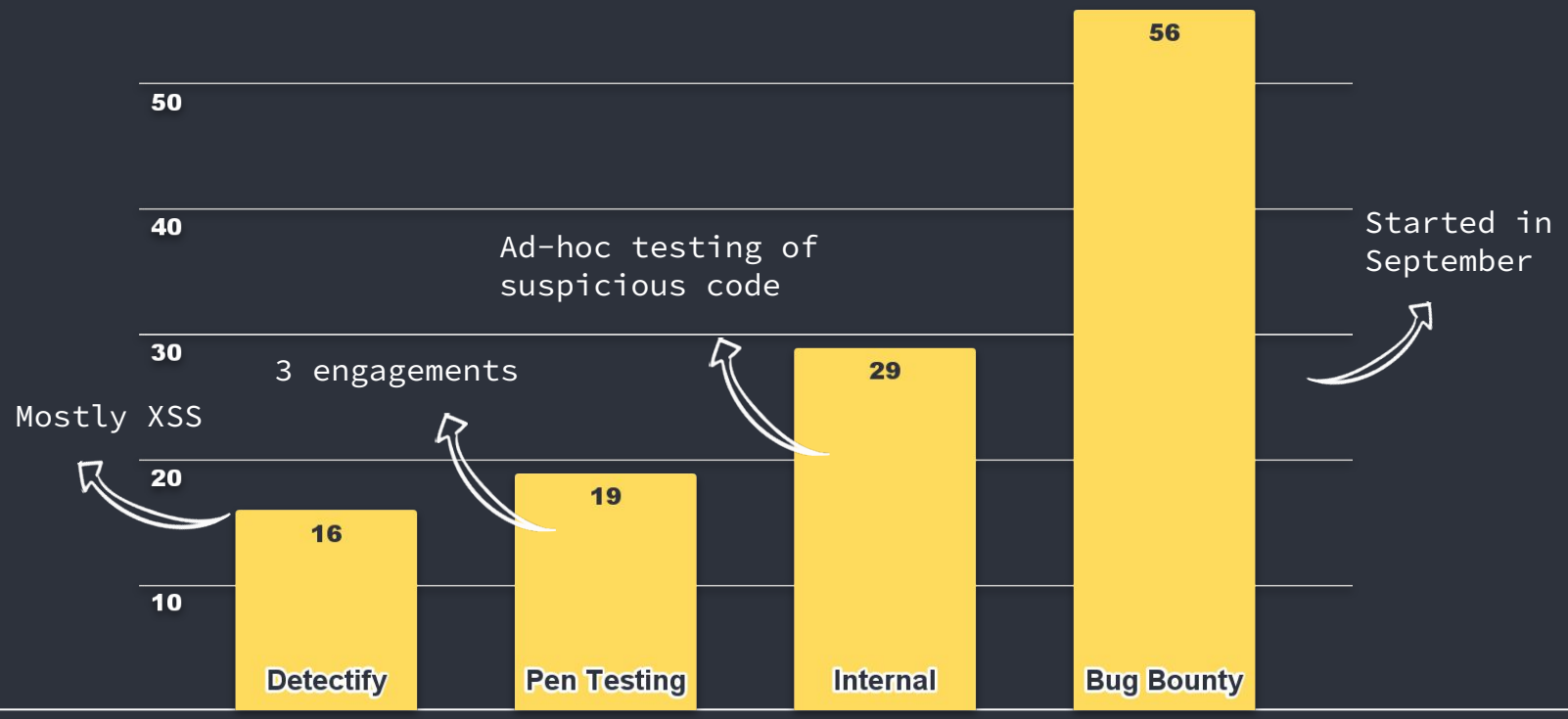
- False positives
- Theoretical correct
- Lacks context
- Mostly from automated tools

No risk reduction

Lost dev time

“It is time to ignore most dependency alerts” - https://blog.doyensec.com/2024/03/14/supplychain.html

2019: Verified Vulnerabilities



Pentesting Journey at FINN.no

- **Been doing it for 15+ years in many different forms**
  - Release test
  - Monthly on-site testing by external partner
  - Two larger tests per year
- **Works well with few releases**
  - But not with over 1000+ deployments per week
- **Vulnerability forecast:**
  - Foggy with a high chance of undiscovered bugs in production

Vulnerability Fun Facts From 2019 Era

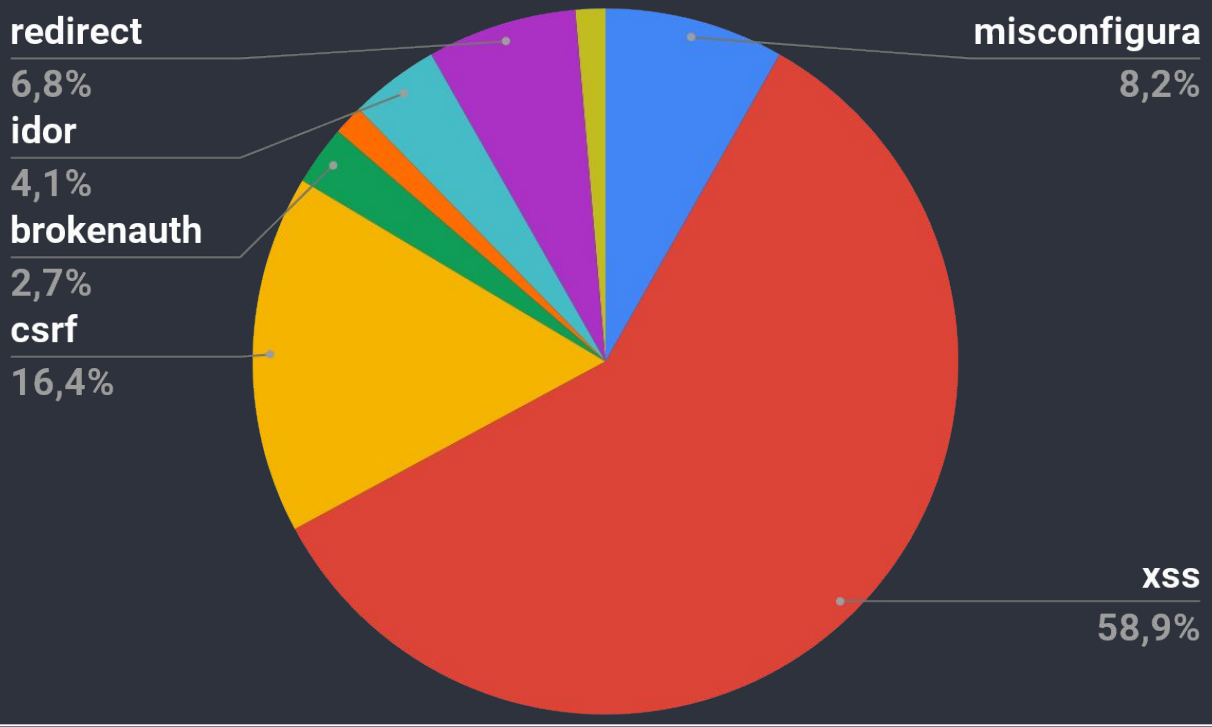
Lifespan

- **Avg exposure time**
  - over 800 days
- **Oldest finding**
  - 11 years in prod

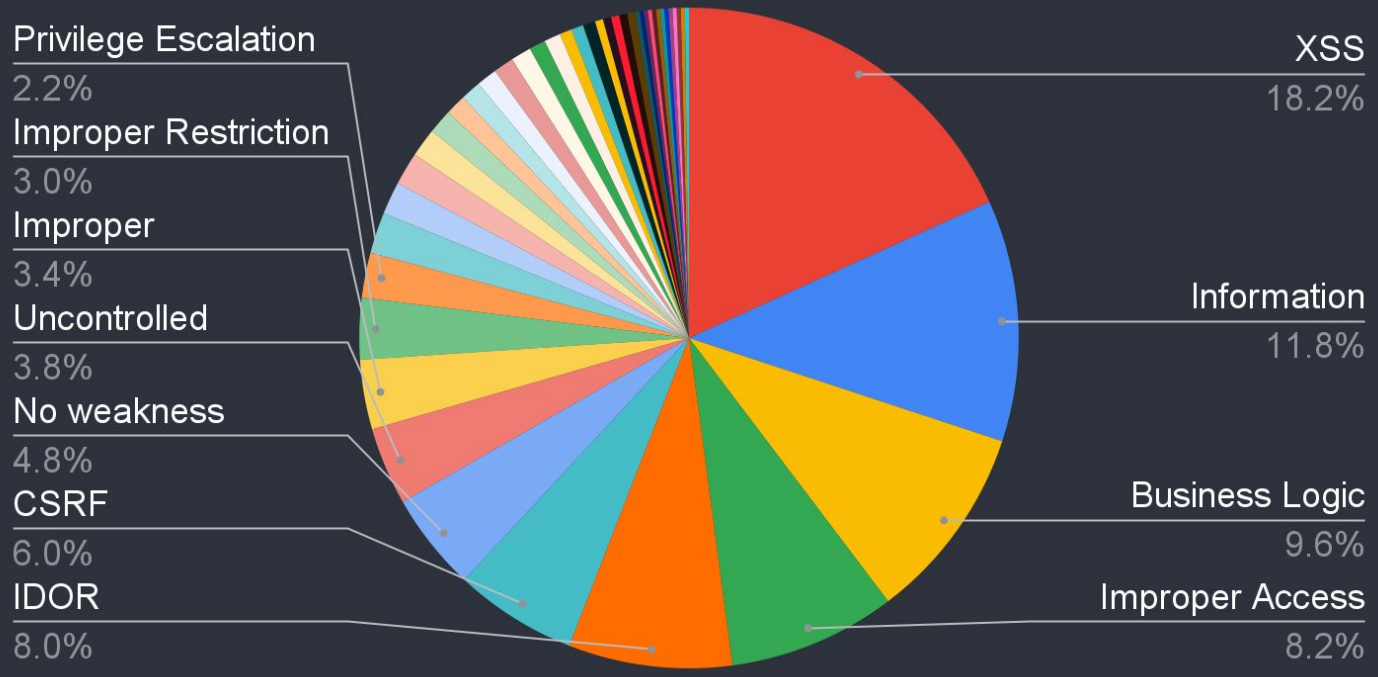
Discovery

- **Findings per year**
  - 15 on average
- **Cost per finding**
  - > $2000

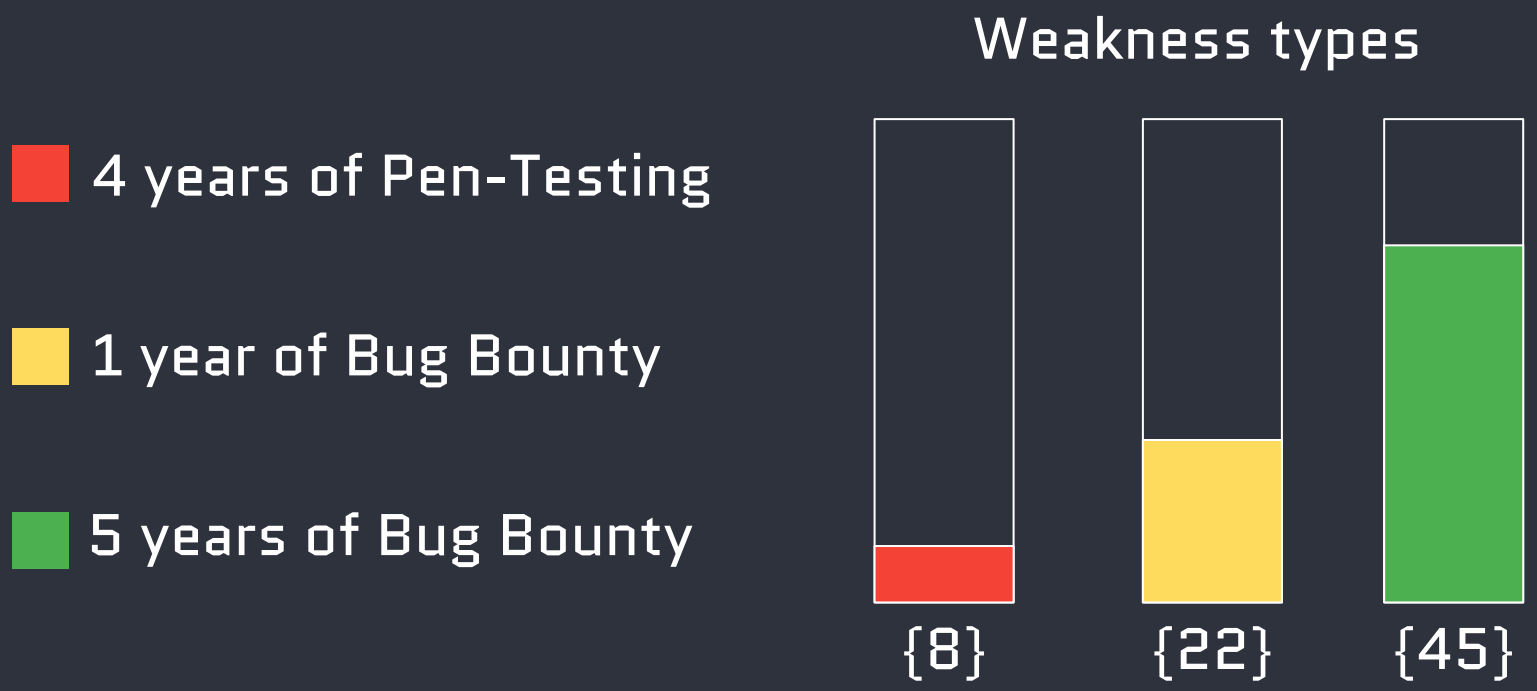
Low Weakness Diversity



The Bug Bounty Effect



Bug Bounty Gives Better Diversity



What is a Bug Bounty Program?

- **Crowdsourced security testing:** pay per valid finding
- **Usually hosted on a platform:** HackerOne, Intigriti, BugCrowd
- **Program types:** Public or Private
- **The platform deals with:**
  - Sourcing and paying hackers
  - Receiving reports and sometimes triage
- **You decide scope and policy for the researcher's behavior:**
  - What domains are in scope or out of scope?
  - Rate-limits, required headers
  - "Use bb platform email, do not spam end-users"

The Life of a Bug Bounty Report

h4x0r

Triage

40%

Invalid

60%

Accepted

No Action Needed 🔁
➔ Duplicate, Informative, Out of Scope

Rejected ❌
➔ Spam, N/A, Low Quality

Edge Cases 🟡
➔ Accepted Risk

Will be fixed ✅
➔ Time to fix dependent on severity

Bug Bounty Benefits

- **Many eyes on the target**
- **Diverse backgrounds and skill sets**
- **Better coverage and continuous like testing**
- **Effective:** Scope once & pay per finding
- **Being a part of the bug bounty community and building a good reputation**

Bug Bounty Challenges

- **Hard to test admin interfaces / back office in prod**
  - Requires you to setup dedicated test environments
- **Duplicate submissions:**
  - If your teams are slow at fixing issues
- **Scanner noise, not following policies**
  - Happens, but can easily be handled
- **Too many critical findings at launch?**
  - Pause program and fix before relaunching

How much do we pay?

- **We pay the hackers based on severity and business impact**
  - We started at $100-$3000, now we are at max $6000
- **Since 2019:**
  - ~$200.000 paid in bounties (FINN)
  - ~$400.000 across 6 programs
  - Median bounty around $200-$400 the last years
- **Cannot compete with “Big Tech” payouts:**
  - Shopify: max $200,000
  - Google: max $151,515


It's
NICE
to be
IMPORTANT
but it's more
IMPORTANT
to be
NICE
- Scooter
INSPIRING
SLIDES

Competing for bug bounty talent

- **Just be nice**
  - Friendly and respectful tone in messages
  - Lax on scope definition: accept all reports with impact
  - Pay fairly, increase severity/bounty instead of limiting
  - Working with hackers to determine the maximum impact
- **Fast response times**
  - Better than the top 20 programs on HackerOne
  - Top 20 measures in hours/days
  - We measured in minutes/hours

Competing for bug bounty talent

bogdantcaciuc posted a comment.

@emilva

Damn, that was quick.

It really motivates me to look more and focus on your program.

Awesome job.

Thank you!

Being Lax on Scope Saved us!

Hi,

I am purely inquiring to see if this program would accept submissions for sensitive data exposure inside docker images from their org https://hub.docker.com/u/fiaas is owned by Schibsted.

This asset is not in scope, and would not be accepted by platform triage ❌
But we were intrigued, and accepted it! Critical issue found ✅

Vulnerability Explanation

- **A docker image published on Docker Hub in 2019 contained a GitHub PAT with OWNER privileges in the organization**
- **The public *fiaas* GitHub org:**
  - Our Kubernetes deployment platform
  - Core infrastructure component with high privileges
- **Potential scenario:**
  - Sophisticated attackers could easily misuse this to supply chain attack & ransomware us hard

2019 Docker bug

2025 Report received

6 years exposure time

Why Did it Happen?

- Our developers were innocent, this was caused by a bug in Docker multi stage builds

18.03.0-ce

2018-03-21

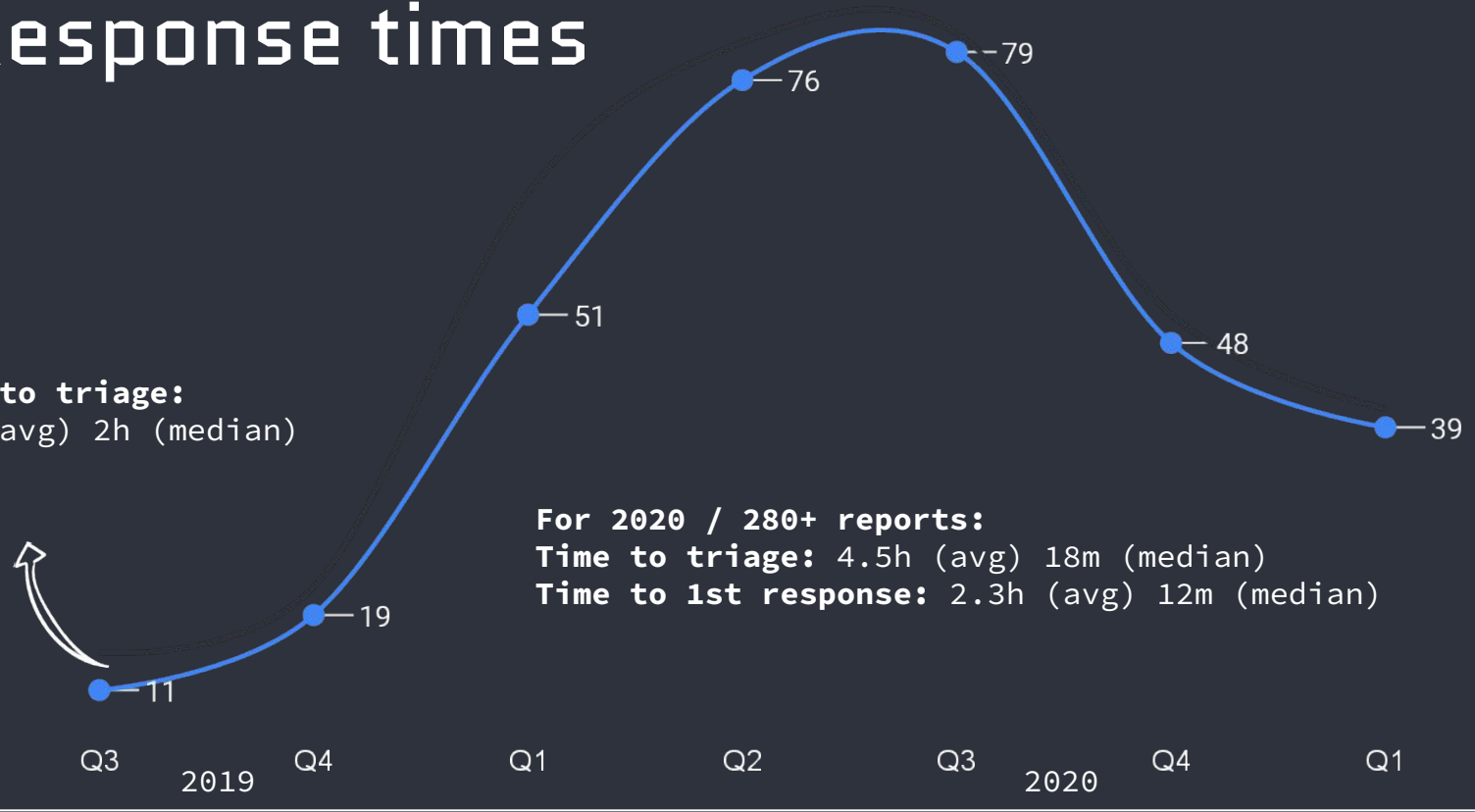
Builder

- Switch to -buildmode=pie moby/moby#34369
- Allow Dockerfile to be outside of build-context docker/cli#886
- Builder: fix wrong cache hits building from tars moby/moby#36329
- Fixes files leaking to other images in a multi-stage build moby/moby#36338

Lessons Learned

- **Our container secret scanning would have found it today**
  - But this container was not in use, too old
- **Scan everything, even old stuff!**
- **Ban PATs and use fine-grained access tokens instead**
- **Use short-lived OIDC tokens wherever possible**
- **Being lax on scope definition saved us**
- **Add all your open source projects to scope!**

Response times

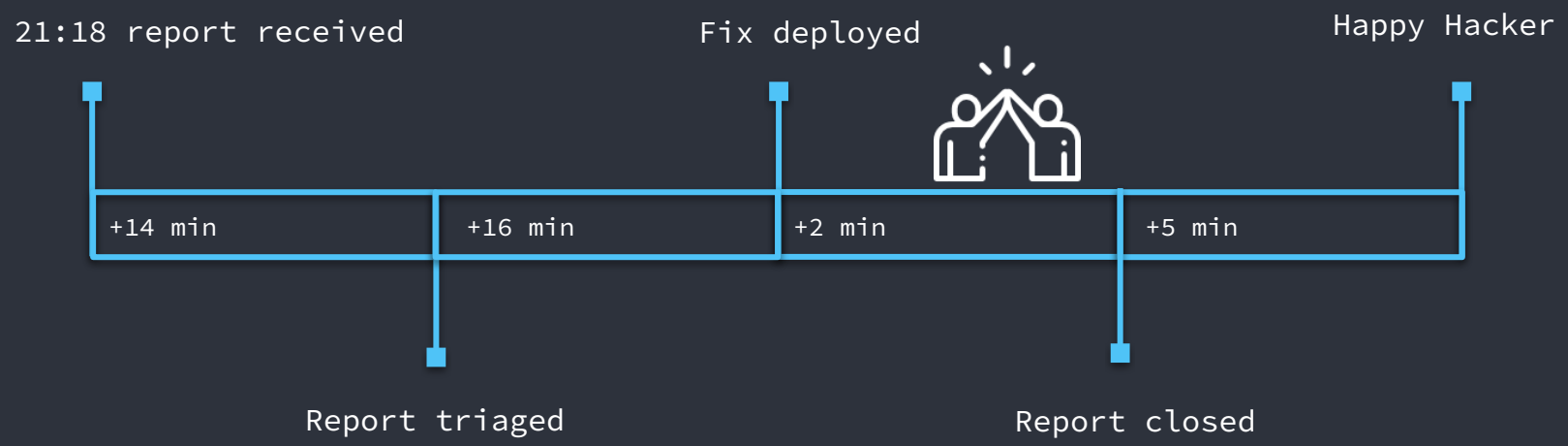


Time to triage:
21h (avg) 2h (median)

For 2020 / 280+ reports:
Time to triage: 4.5h (avg) 18m (median)
Time to 1st response: 2.3h (avg) 12m (median)

Critical fixed on a Saturday Night

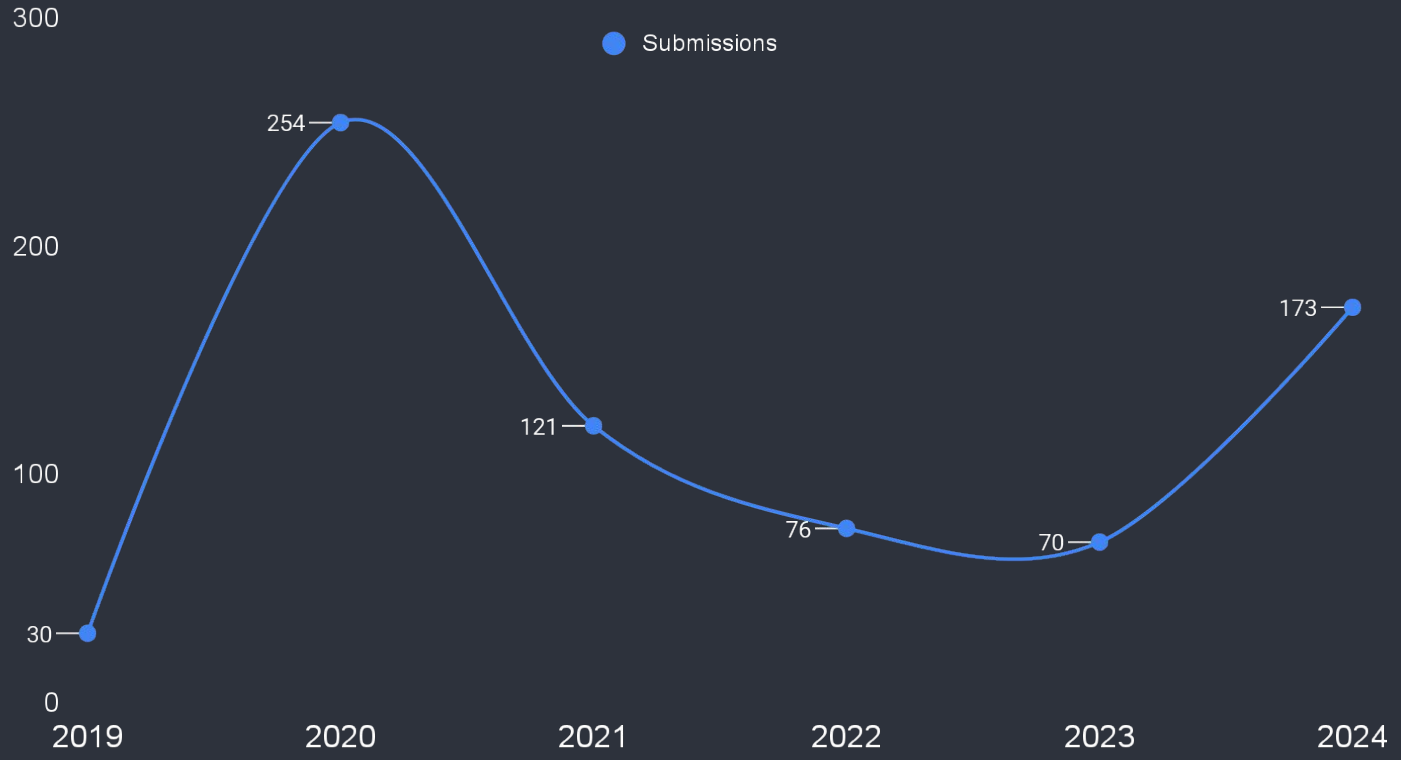
“/actuator/httptrace exposed”



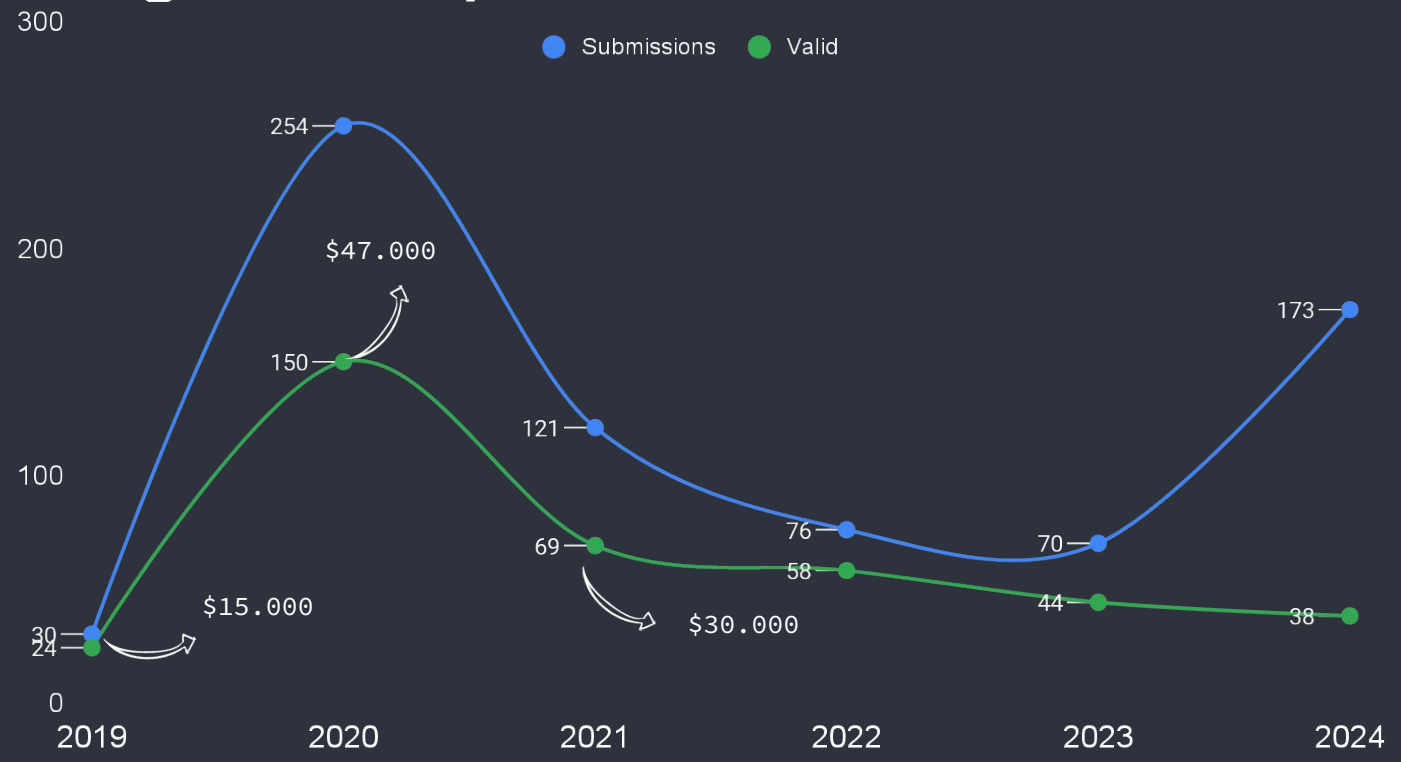
Vulnerability Transparency

- **Give platform access to all developers gives you:**
  - Awareness about all reports coming in
  - Faster response on critical findings
- **“Critical fix on a Saturday” not possible without it**
  - We could always ping somebody
  - Easier when they just show up and fix it

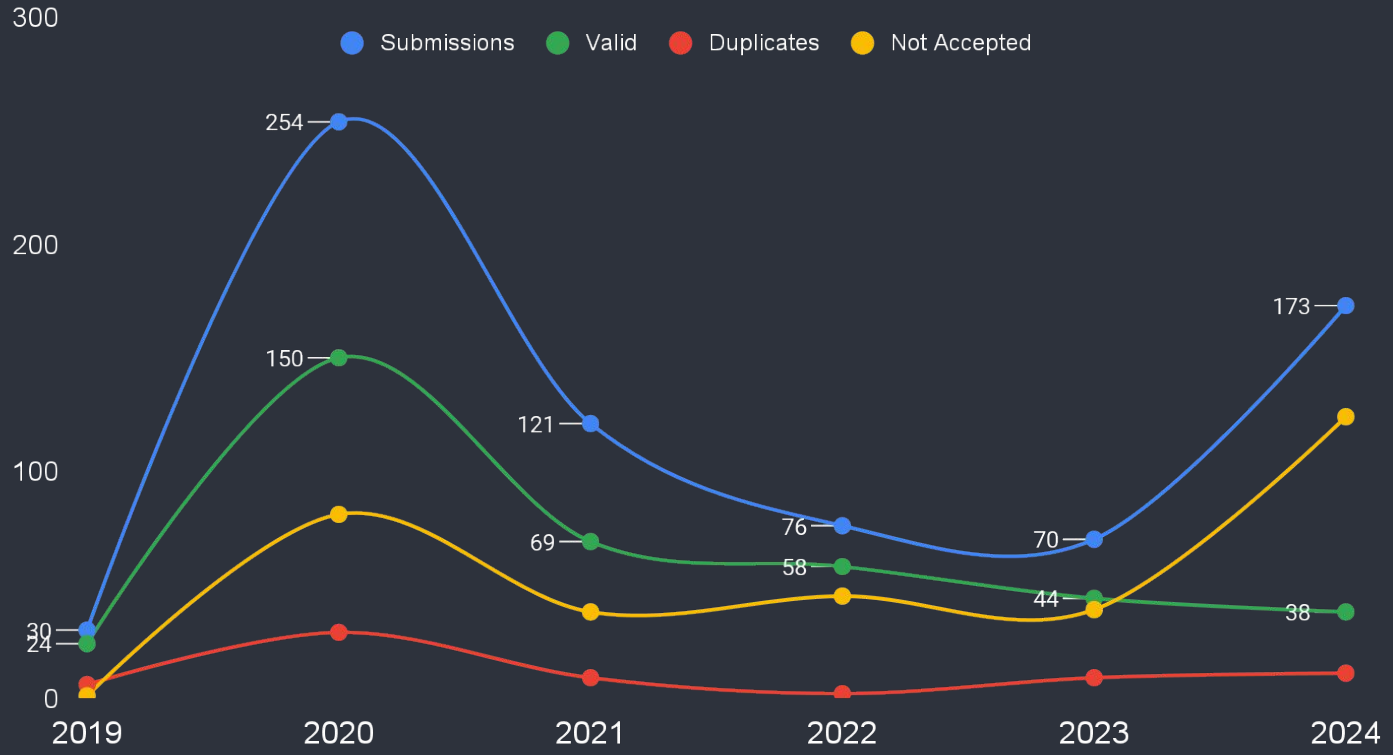
Bug Bounty Rollercoaster



Bug Bounty Rollercoaster



Bug Bounty Rollercoaster



Top 10 Vulnerabilities

By Count	By Spend
XSS	XSS
Information Disclosure	Subdomain Takeover
Access Control	Information Disclosure
Misconfiguration	Access Control
IDOR	IDOR
Authentication	Authentication
Open Redirect	Denial of Service
CSRF	Business Logic Errors
Web Cache Poisoning	SSRF
Business Logic Errors	HTTP Request Smuggling

Critical bugs: more than just AppSec

Cloud & Infrastructure

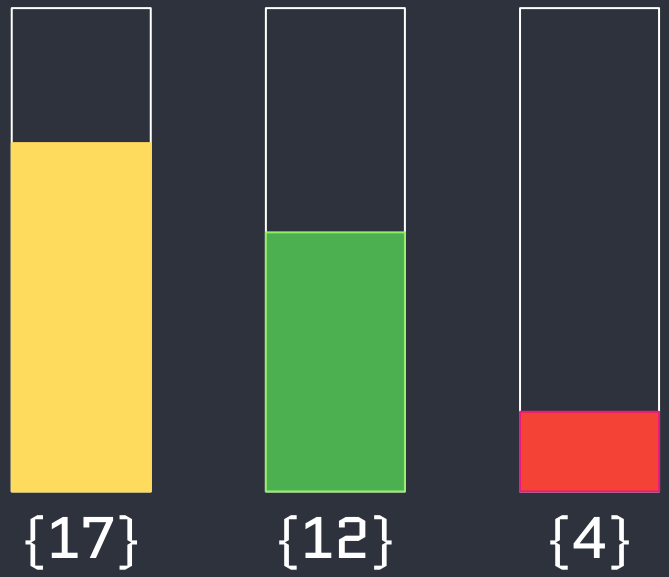
Misconfigurations: subdomains, load balancer bypass

Application Bugs

IDOR, XXE, SSRF, Auth

Secrets in The Wild

High privileged API tokens



Report Quality

- **Running a private program helps**
  - Less random automated bounty beggars
- **Closing bad reports is not a big time sink**
  - “XSS” via Console:

3. Inject Payload:

   Paste the following payload into the console and press Enter:

```
var payload = '<img src=x onerror="alert(document.cookie)">';
var div = document.createElement("div");
div.innerHTML = payload;
document.body.appendChild(div);
```

3 Degrees of Low Quality Reports

No Bounty

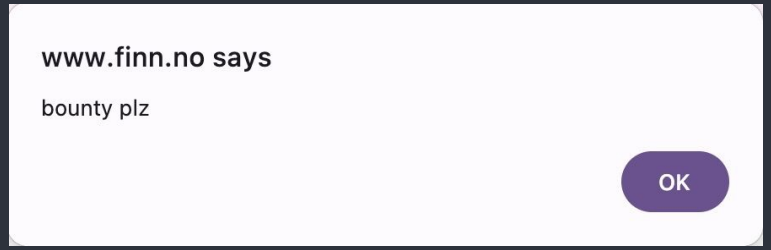
No bounty accepted

Beg Bounty

Begs for bounty or positive close status

Threat Bounty

Threatens to publicly disclose if no bounty or positive close status



Denial of Funds Attack [DoF]

- We usually have about 10k EUR in our bounty pool
- Program **auto-suspends** if bounty pool is depleted
- The platform reserves the bounty amount ***before* triage**
- What happens if one researcher spam us with “Criticals”?
  - **Denial of Funds attack!**

DoF Attack + Threat Bounty

- **We got three high severity reports**
- **Program was auto-suspended**
- **Two of those:**
  - Almost same title
  - Same content, different order
  - … but different severity
  - No impact whatsoever - Likely AI generated
- **Closed as *Not Applicable***

Threat Bounty ⚠️

Hello,

Why did you give me negative? Does this mean that these addresses that I reported are not important for you? If it is not important, there is no need to give a negative because this information is related to your company.

I will definitely follow up on this case because we report the same information on Hackerone and at least they count it low, but I don't understand at all why you gave me a negative.

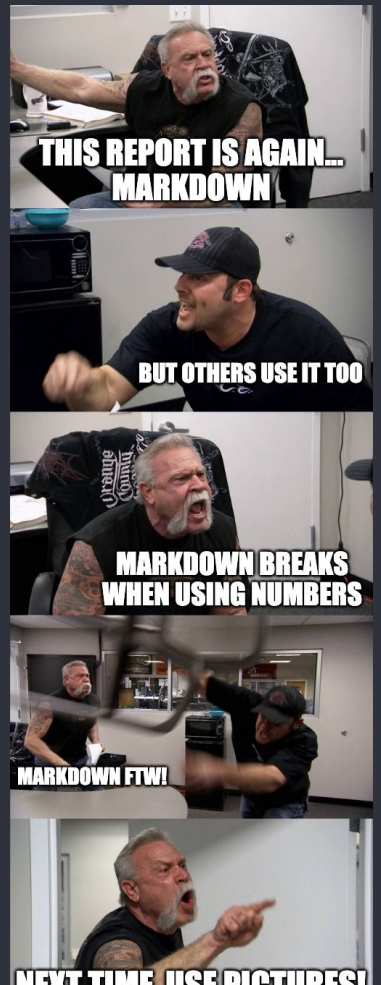
If you agree, I will post this information publicly on Twitter so that you will know the opinions of other people that this decision of yours was not correct. Do you agree to hear other people's opinions?

Because this topic is very important to me and you should not give me negative feedback and it should be changed to informative. If you don't intend to do this, please let me know as soon as possible if you want to change or would you like to give a negative score for this report?

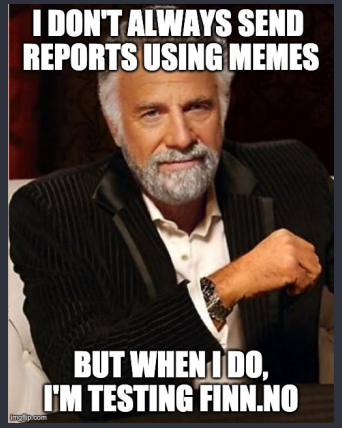
Meme Bounty


LEFT
EXIT 12
Other programs
FINN.no
tk0
imgflip.com

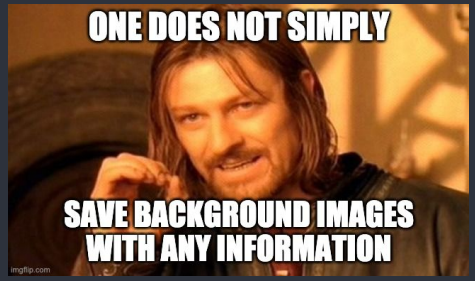

CHALLENGES
CHALLENGES, EVERYWHERE
imgflip.com


THIS REPORT IS AGAIN... MARKDOWN
BUT OTHERS USE IT TOO
MARKDOWN BREAKS WHEN USING NUMBERS
MARKDOWN FTW!


ANDROID/IPHONE SAVES SCREEN WHEN SENT TO BACKGROUND
SHOW ONLY LOGO
imgflip.com


I DON'T ALWAYS SEND REPORTS USING MEMES
BUT WHEN I DO, I'M TESTING FINN.NO
imgflip.com


IF YOU THINK ABOUT IT
THIS IS THE FIRST EVER MEME REPORT IN HACKERONE
imgflip.com


ONE DOES NOT SIMPLY
SAVE BACKGROUND IMAGES WITH ANY INFORMATION
imgflip.com

Bug Bounty Program Impact

- **We found and fixed a lot of old vulnerabilities**
  - Over 700+ vulnerabilities fixed
- **We are discovering vulnerabilities faster than before**
- **The number of findings per year are decreasing**
- **It is effective**
  - Cost per finding $400 vs $3000 before

Only 2 out of 700 reports were caused by a vulnerable dependency

Key Ingredient in AppSec Program

- **Builds security awareness among developers**
  - All reports are open for anybody to read
  - Devs like to talk about new interesting findings
- **Over 700 verified vulnerability reports**
  - Valuable vulnerability data & metrics
  - Helps us focus our AppSec efforts

No SQL Injection found on FINN.no since 2014

No SQLi Bugs Happened Organically

Secure Defaults FTW

Most libraries encode input *automatically*

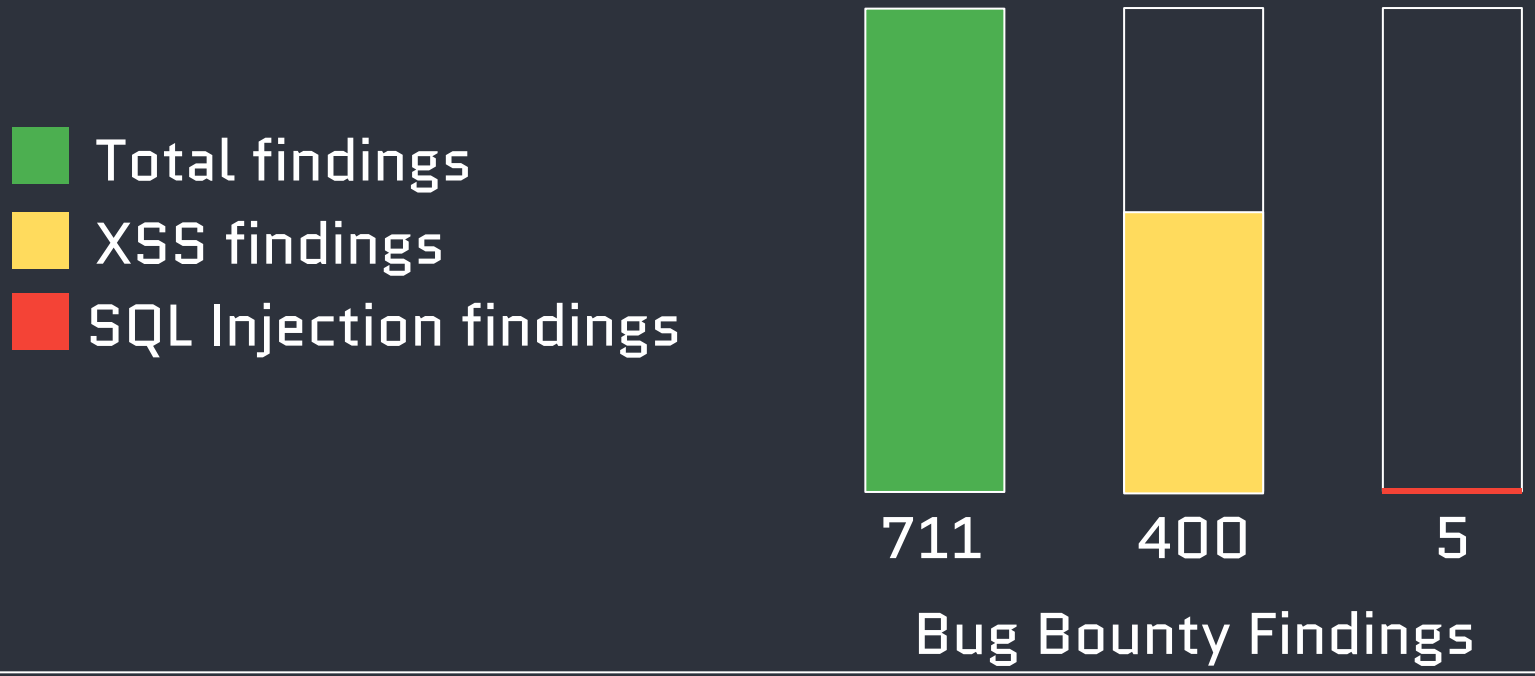
Dev's do not need to *manually* encode

Did we look close enough?

10 years of pen-testing
5 years of Bug Bounty
A lot of Code scanning
24/7 Dynamic scanning
2x Bounty promotion campaigns

Secure defaults:
Squash bug classes, not individual bugs

Secure Defaults in Practice



“Weaponizing” the Vulnerability Data

- **Created CTF challenges based on bug bounty reports**
- **35 challenges in the categories:**
  - XSS/CORS, JWT, Cryptography, SSRF, XXE, /internal-backstage/, secrets in github/CI/CD
- **Released a few challenges per week until main event**
  - Kept teams on their toes
- **Developers loved it**

Simplified XSS example

```
app.get("/", (req, res) => {
    const { query = "" } = req.query;
    res.send(`
      ${heading}
      <body>
      ${req.query ? `<h2>Results</h2><p>No results for ${req.query}</p>` : ""}
      <script type="application/json">${JSON.stringify({ req.query })}</script>
      </body>
      </html>
`);});
```

Sanitized by DOMPurify

```
app.get("/", (req, res) => {
    const { query = "" } = req.query;
    const sanitized = DOMPurify.sanitize(query);
    res.send(`
      ${heading}
      <body>
      ${sanitized ? `<h2>Results</h2><p>No results for ${sanitized}</p>` : ""}
      <script type="application/json">${JSON.stringify({ sanitized })}</script>
      </body>
      </html>
`);});
```

DOMPurify Removes Dangerous bits

```
DOMPurify.sanitize('<img src=x onerror=alert(1)//>'); // becomes <img src="x">
DOMPurify.sanitize('<svg><g/onload=alert(2)//<p>'); // becomes <svg><g></g></svg>
DOMPurify.sanitize('<p>abc<iframe//src=jAva&Tab;script:alert(3)>def</p>'); // becomes <p>abc</p>
```

DOMPurify 

Is there any foot-gun potential?

Well, please note, if you *first* sanitize HTML and then modify it *afterwards*, you might easily **void the effects of sanitization**. If you feed the sanitized markup to another library *after* sanitization, please be certain that the library doesn't mess around with the HTML on its own.

Foot-gun Potential

```
app.get("/part4", (req, res) => {
    const { query = "" } = req.query;
    const sanitized = DOMPurify.sanitize(query);
    res.send(`
      ${heading}
      <body>
      ${sanitized ? `<h2>Results</h2><p>No results for ${sanitized}</p>` : ""}
      <script type="application/json">${JSON.stringify({ sanitized })}</script>
      </body>
      </html>
`);});
```

Foot-gun example

```
// Removes dangerous <script> tag
DOMPurify.sanitize('<img src=""><script>alert(1)</script>');
// outputs <img src="">

// <script> tag is not dangerous inside attribute quote context
DOMPurify.sanitize('<img src="<script>alert(1)</script>">');
// outputs <img src="<script>alert(1)</script>">
```

Payload example

We send payload: <img src="</script><script>alert(1)</script>">

<h2>Results</h2> No results for <img src="</script><script>alert(1)</script>">
<script type="application/json">{"sanitized":"<img src=\"</script><script>alert(1)</script>\">"}</script>

```
    </form>
    <h2>Results</h2><p>No results for <img src="</script><script>alert(1)</script>"></p>
    <script type="application/json">{"sanitized":"<img src=\"</script><script>alert(1)</script>\">"}</script>
```

The Long Tail of Vulnerabilities

- **Bug bounty by far outclasses other activities**
  - Effective in terms of $ and vulnerabilities
- **Still we spend time and resources on other activities**
  - Code/Dynamic scanning, Cloud tools, Pentesting
- **While others yields less verified vulnerabilities, they often yield *different* types of vulnerabilities**

- **All in all this gives us better assurance**

Only 5 out of 20 critical bugs could be found with code scanning.

What if I could only do one thing?

Code Scanning

- Hard to roll out
- Findings lacks deployment context
- Can annoy developers if done badly
- High cost per year, no guarantees

Bug Bounty Program

- Scope once, hackz everywhere!
- Mostly real exploitable bugs
- Developers only see real bugs
- Pay-as-you-go for bounties

Traditional Advice: Roll out a SDL with a bunch of tools and practices before bug bounty.

Spicy Advice: Launch a private bug bounty program and do some real risk reduction.

5 Years of security.txt vs Bug Bounty

security.txt

- 1-2 valid findings
- 99% spam

5 findings

Bug Bounty Program

- > 700 valid findings
- < 1% spam

All companies **should** do bug bounty

How to Launch a Program

Get money & a platform

- Test different platforms
- Managed triage service = slow response times

Triage process

- Set bounties based on business impact
- Get inspired by other programs

Scoping

- Start small and expand as you mature the program
- Do pen-tests before launch?
- 24/7 scanning to catch low-hanging fruits

Communication

- Inform the org about it!
- Onboard people in your process
- There will be scanner noise

Platform must-haves: SSO

Why?

Give everybody easy access
Manually managing access sucks
Importing to JIRA sucks

Impact

No time spent on manual processes
Developers talk about reports!
Security vulnerability awareness
No JIRA-headache

Platform must-haves: Good API

What is good enough?

Should not be painful to use
Be able to import reports
Export data/metrics
Automate missing things

Examples

Silly rate limits / auth methods
Avoid paying for known findings
Crunch custom metrics
Use data in other platforms
Auto assign reports based on URLs

Automation Example

- **All our deployments are tagged with an *owner***
- **We can find a deployment from an URL**
- **Report with url finn.no/user/api**
  - Lookup owner from ingress /user/api
  - Find team: ‘account’
  - Assign report to team ‘account’
  - Notify their alert slack channel

Platform must-haves: Report disclosure

Why is it important

Knowledge sharing
Increase activity
Community building
Shows fair treatment

Impact

Bypasses!
More reports!

Regex is pronounced /rɪˈgrɛts/

- **Any URL with** /internal-backstage/ **or** /_/ **is blocked**
- **“Hiding” endpoints like** /metrics, /health, /actuator
- **The blocking is done by a RegEx in HAProxy configuration**
- **Vulnerability forecast:**
  - Cloudy with a high chance of RegEx Bypass

Regex Bypass Galore

Hacker #1	Hacker #2	Hacker#2	Hacker#3
Bypass: /ib;/env	Bypass: /ib;abc/env	Bypass: /_;/env	Bypass: /_;abc/env
September 2019	January 2020	January 2020	August 2020

Lessons Learned

- **RegEx is hard & report disclosure is effective**
- **Bug had been in production for 3 years:**
  - No pentest discovered it (or other tooling)
  - Bug bounty discovered it after 7 days
  - … And kept re-discovering it!

Dependency Confusion

Recipe:

1. Find name of an internal package
2. Publish a public version with higher version number
3. Profit! (Literally)

Impact:

Attacker code can then run on:

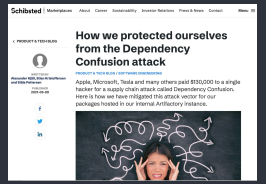
- Developer’s machines
- Build Systems
- In production

[1] https://medium.com/@alex.birsan/dependency-confusion-4a5d60fec610

How we protected ourselves...

@alex.birsan Research released Feb 2021

Blog post released



RCE via NPM reported

+8 days | +14 days | +2 months | +1.5 years

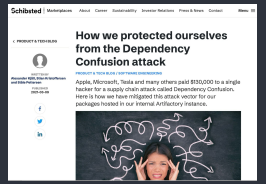
✅Mitigated

Artishock open-sourced

Blogged it, Built it, Forgot to use it!

@alex.birsan Research released Feb 2021

Blog post released



RCE via NPM reported

+8 days

+14 days

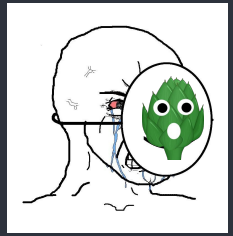
+2 months

+1.5 years

✅Mitigated

Artishock open-sourced

From Blog to Bug Bounty



- **The researcher referenced our own blog post**

> I do often reference your blog post here for mitigation when submitting similar reports
> https://schibsted.com/blog/dependency-confusion-how-we-protected-ourselves/

- **The blog post was more impactful than the tool?**
  - Sometimes words are mightier than the sword
- **After this the team made the tool run 24/7**

Expect the Unexpected

- **Wow-factor reports**
  - Critical bugs that lived in production too long
- **Reality check**
  - What if the bad guys found this first?
- **Continuous learning**
  - Each surprise finding drives improvements

The Bug Bounty Effect

- **Bug bounty has been priceless for us**
  - Key ingredient in our AppSec program
- **We still do pen-tests and all things** DEVSECOops!
  - Secure defaults are effective
- **Launching a program is easy & impactful**
  - Every large company should have one
- **The key to bug bounty success**
  - Be nice and be fast on response

Thanks!

Do you have any questions?

Emil Vaagland @ LinkedIn
twitter.com/emil_no
emilpls.bsky.social
emil@vend.com

DEVSECOops!