Bug Business #7 – Get to know _jca_, Intigriti's Top Hacker in Q2

BY ANNA HAMMOND · AUGUST 3, 2020 · LAST UPDATED ON MARCH 6, 2025

Bug Business is a series of interviews in which experts from the bug bounty industry shine their light on bug types and trends. Today, we sat down with Belgium-based Johan (_jca_) and discussed his recent successes in Bug Bounty, the methodologies and techniques he uses, and how his kids useless pile of toys can be attributed to his bug bounty career.

Hi! Can you tell us a bit about yourself, who you are and how you got into bug bounty hunting?

Hi, my name is Johan, I'm 41 years old and I'm from Antwerp. Happily married and father of 2 boys. I'm working as a CyberSecurity Expert at the Belgian Cyber Emergency Response Team, or CERT.be. We do incident response, malware reverse engineering and pentesting, with the goal of making the Belgian vital infrastructure the least vulnerable in the world.

I have been playing with computers since I was 12. Hacking was still very illegal in those days, so I couldn't really nurture "the gift".
After a few jobs here and there I ended up in the DevOps team of a Belgian security company, where I was surrounded by incredibly talented people, from whom I learned a lot.

That company decided to run a Bug Bounty Program, and I was lucky enough to be involved in the triage process.
That really opened my eyes, I could do that too! A couple of months later I submitted my first real bug, a payment bypass.
The emotional rollercoaster is what really got me hooked.
The thrill of the chase and the frustration of not finding anything. The excitement when you discover some weird behaviour, the promise of something more.
And then finally the relief and satisfaction of getting that callback, or seeing the response take exactly the time you specified in the SQL command, even popping a stupid alert box.
It's really addictive.

So, what does your life look like now? Do you do bug bounty full-time or as a hobby, and how does it fit into your life?

Being a high ranked bug hunter helped me to secure my current job. I'm more of a weekend warrior. I need to balance my day job, familiy life and bug hunting and that's certainly not easy. Some months I feel the need to "go deep" on a target, other months I'm just surface scratching. I have never really considered going full-time, my day job is still the most important as it is a stable source of income. Also, I like my job.

Now some technical questions... How do you approach a target? Do you follow a pre-defined methodology? And would you recommend testing few functionalities for all possible bugs, few bug classes across all endpoints, or anything else?

Mapping the attack surface is really important. Become a "power user" of the application. The legacy and beta features will reveal themselves.
I always have a few iterations in mapping a target, some features only become apparant in some edge-cases.
I use some "probe strings" with special characters for injection bugs that I've been lucky with in the past. No polyglots, the goal is not to have a working exploit, just to see if it breaks something, or if the string gets transformed in a certain way.
I don't have a methodology, I'm basically "following my nose". A bit like poker, you're not just playing the cards, you should also play the people.
What were the devs thinking? Are they fond of custom headers? There might be some you haven't discovered yet.
Are they using angular in some places and other places not? There might be a transition going on in the dev team, or 2 teams that might not check each other's inputs and outputs.
If you find one encoding/filtering mistake, there's bound to be others.
Are they relying on UUID's for security? Are they sure the UUID is not revealed somewhere else?
I focus on bug classes that have to most chance of success on a given target.

Does recon play an important part in your bug hunting? And how does it look like for you?

I use Amass and aquatone for recon. I don't focus on recon much, but you have to do it.
You could always find that forgotten server. There can always be a DNS entry pointing to a private address space, that's something you can use later if you find an SSRF.
Maybe that staging server is not pointing to Cloudflare like the production servers.

Do you have any favorite bug classes or types of targets that you focus on the most, and why?

Not really. I try to have an understanding of most bug classes. I see bug hunting as a way to keep up with different technology stacks. Even if I don't find anything, I still learn from it. If you focus on what you know well, you'll be less frustrated for sure.
But if you focus on what you don't know, you'll learn more.

What was the most interesting bug you found (or your favorite)?

Caching bugs are always cool, because they are weird.
I had one where I could add a GET parameter called _escaped_fragment_ to the url of a single page application. The only difference it made was the capitalization of the headers, camelcase in one instance, lowercase in the other.
So the request ended up on a different server, but why? As it turned out, the _escaped_fragment_

parameter was a legacy way of letting the reverse proxy know the response needed to be rendered as plain HTML (not as a single page app) so it could be indexed by search engines. The reverse proxy send the request to a different server that would handle that. But the vulnerable page was already HTML, no client-side modifications, so the difference was not noticable, except for the headers.
Looking further, the other server would honor the X-Forwarded-Host header and replace the Host header with the value of X-Forwarded-Host.
So now I could put my own webserver in the X-Forwarded-Host header, and see that it was prerender.com, a third party service, that was rendering the page in a headless chrome. Prerender.com would send back the rendered response (of my website with XSS payload), and the other server would send that back to the revproxy, and the revproxy served my page to the user. Very weird, but not really a vulnerability as I could not serve this to other users. By this time I had already send so many requests that I noticed that there was caching going on, on the return path, for a couple of seconds based on the url of the request. But the url contained this weird GET parameter called _escaped_fragment_, so no user would hit the cache if it was based on the url.
The documentation provided the answer, the GET parameter could be replaced by another header, called X-bufferbot.

So now I could send a request to the application, added a header X-Bufferbot: true to instruct the revproxy to redirect to the other server, which in turn would forward to prerender.com. Added an X-Forwarded-Host header so that prerender.com would render my malicious website with XSS, instead of the actual app.
On the return path, the response would be cached, based on a “clean” url, for a couple of seconds. Every user hitting the homepage within that timeframe, would hit the cache. The cache would serve my malicious content instead of the actual target.
And since my payload was running under the domain of the target, I was able to access the session data stored in the local storage.
All an attacker needed to do was resend the poisoned request every 2-3 seconds.
I enjoyed this bug very much, it took a long time to figure out what was going on, since the only clue was the capitalization of the headers.

What does your arsenal look like? Which types of tools do you rely on, how do you choose them and which would be your favorites?

Amass, aquatone, dirsearch, @tomnomnom’s waybackurls.py and Burp.
Pretty basic. Better to have a few tools that you know well, than many tools you barely know how to use efficiently.

Let’s talk about automation. Many hackers leverage it for recon, mass-scale tests and even automated reporting of bugs like subdomain takeovers. But others prefer to focus on logic or advanced bugs that can only be found with manual testing. Where do you stand regarding this question of automation? Do you use it, and do you think it is worth spending time on?

It might have been easy money at one time, but I feel like too many people are fishing in the same pond right now for it to be profitable, except for the happy few.
Security-wise, from the point of view of the client, I think it’s actually a good thing that unpatched firewall appliances and default credentials get found and reported quickly.
After all, APT’s and cybercrime gangs always take the road of least resistance. If an rce poc comes out, you’re gonna get scanned in the hours that follow.
Might as well get scanned by the good guys. I would advise the clients & targets to not close these reports as informative because “90 day patching window” because those guys save you a ton of headache, by

avoiding an easy zero-click compromise.
It's not because you're not ransomwared in that 90 day window that you're not compromised. Some criminals wait for months inside the network and then strike.
So I recon there's a market for it, but it's not for me.

What advice would you give your past self about bug hunting?

Write better reports! When you're really deep into a target it's easy to forget that some things are not so self-explanatory as you think it is.
Take your time to write a good report, unless it's an xss in the search bar, then be fast!
Every cool bug is preceded by at least 2-3 moments where I was about to give up. Keep going! Other people are still finding bugs after you left, there is always one more bug to find. "Spray and pray" with payloads only gets you so far, understand the app like a power user would, then break it.

One huge hurdle hackers face is information overload. How do you keep up with the fast pace with which attacks and tools evolve? And what would you tell beginners who feel overwhelmed with the amount of information to learn?

It's difficult for sure. If this was a Hollywood movie, everything the protagonist needs to learn and train for would be condensed into a short series of shots called a montage.
In real life there is no montage. Processing information is the part where you need to put the time and effort in. You can learn new skills or attacks in one of three ways: imitation, reasoning and trial-and-error.
Imitation is the easiest, learn from the people who already did it.
Reasoning is cool and original, but takes time and effort.
Trial-and-error, aka "dumb luck". It sometimes works, but mostly not.
For beginners I would advice to start by looking for easy bugs in easy places. You'll drown in duplicates, but it shows you are on the right track.
Then go for easy bugs in hard to find places. This shows you mapped the application well, might even score a bounty here.

Move on to hard bugs in easy places, programs with a low max bounty or responsible disclosure even are a good starting point. You scored some bounties, now it's time to practice and improve your skillset, build some confidence too by finding cool stuff. The real reward is confidence and experience, not money.

Last step are the hard bugs in hard to find places, now it's time to target that 5 digit max bounty program. I don't know what it's like on this level, I'll tell you when I get there. Ask @intidc or @arneswinnen or some other natural talent who make this shit look easy.

What is the coolest thing you did with your bounty money?

Ok, this is confronting. I haven't done anything cool with the bounty money. Just an overall increased standard of living. Bought some furniture, did some city trips. My kids now have a pile of useless toys. Took a cab instead of the subway. Mundane stuff really.

Which hacker(s) would you give a shoutout to, whether they are a mentor or a community member?

Basically all the hackers that take the time and effort to share knowledge, write blog posts and create videos.
That is hard work. Think about it, those people are good at what they do, they could be making money instead. But no, they spend time and effort creating free content to make other people better.

Have you already collaborated with other bug hunters? Can you share with us your experience, and if there is anyone you would like to collaborate with in the future?

I have collaborated with @honoki on a project outside the bugbounty context. Really cool guy and amazing hacker, be sure to follow him if you aren't already.
I'm always willing to collaborate, but with the work/life/family balance it's not always easy to find the time.

What are your expectations of bug bounty platforms, and why did you choose Intigriti?

The most important thing for me is that I need to feel that somebody has my back. That the game is played fair. I think this is where Intigriti excels.
Worst thing that can happen is that you report a bug and you see it getting fixed without even getting a reply on your bug report.
That has happened to me on all platforms except Intigriti.
My experience with the Intigriti staff is that they are both friendly and capable, a very valuable combination.

Thank you so much for this interview! Any last words?

At least for me, the two most important questions in bug hunting are "How does this work?" and "What happens if I do this?"