Meet the hacker: Tom Hudson

BY ANNA HAMMOND · JULY 1, 2021 · LAST UPDATED ON MARCH 6, 2025

In our 'meet the hacker' series, we're taking the time to talk with Intigriti community members that have an impressive track record, an unusual methodology or have made valuable contributions to the community. This time, we were talking to Tom Hudson (aka. TomNomNom), who is well-known for his plethora of hacking tools that he currently hosts on Github.

Hi Tom! Please tell us a bit about yourself, who is Tom Hudson?

Hi, I'm Tom. I currently work for Detectify as a security researcher.

Before that, I was a technical trainer for a big gambling company in the UK. I was a solutions architect, a lead software engineer. I was a DevOps sort of a person, and before that I was kind of a software engineer writing PHP for a company that builds educational software. And a whole bunch of other stuff before that as well.

In terms how we're going to the security side of things, that was through bug bounty. I was never professionally a security person up until now. I just kind of found my way here.

When was it that you heard about the term bug bounty for the first time?

I think it was sometime around 2016, 2017. The company I was working for at the time launched a bug bounty program. I had never heard of it. This sounded too good to be true.

I am very much of the age where I grew up with the Internet, and with the knowledge that if somebody is going to give you money for something on the internet, it is probably a scam. You know, I would have been really pretty wary of it, if it wasn't for the fact that it was my employer doing it.

Much to my delight, they decided that employees could take part in the bug bounty program, as long as they weren't submitting bugs against code that they wrote. It was the number one rule. And I got involved that way. It was sort of a, I think, had a wedding booked and not paid for. And there was this thing saying, hey, you can make some money doing something that I happened to be okay at. And the rest is history, as they say.

We all know you for creating a ton of tools. How did you start coming up with the idea that you want to create tools for the community?

As I mentioned, I spent quite a long time as a software engineer, but also in sort of DevOps and sysadmin sort of roles. I was pretty used to just automating and scripting the things that I was doing. Even if it was a relatively small task, because I had a decent amount of programming experience, it would be usually faster to script something.

After a while, I sort of realised that there were things I wanted to do, that there didn't seem to be tools for. So I said, I'm going to build my own. I saw no reason to keep them private. If that makes sense, I often find myself talking to people about things I've done. And it's way, way easier if I can say, here's a link to some source code. You can see exactly what I mean, instead of just trying to describe what something is.

> **"Even if it was a relatively small task, [...] it would be usually faster to script something."**

What's going on in your head from the first idea to the tool's release?

I would say as just about everything, it starts with the problem. Often it will be a case of, I'm trying to do this thing as part of doing some bug bounty hunting or something like that.

It's probably easier if we use an example. I have a tool called "anew", which is not actually a security tool. It's just sort of a general utility Unix tool. It's for appending lines to a file, but only if those lines don't already exist in the file.

In doing my bug bounty hunting, I have dozens and dozens of text files. A text file full of domains, and a text file full of URLs and a text file full of parameters, and so on and so forth. I don't want duplication in those files.

Back then, I was kind of using a few different tools to solve this problem. I thought it'd be really neat if I could just solve this by piping it into one thing. So, that was my problem. It was really cumbersome to perform this task.

The first version is usually just going to be something that works, doesn't have to be elegant. There's not gonna be a README or anything. It's going into the hacks repo. It's going to be effectively a proof of concept.

That's sort of the first or second stage. Have a problem, find some way to solve it and just come up with a proof of concept. And after that come the refinements. You find bugs in things, or you find out that your original assumptions were invalid, or that it's clunky to use, and you need to figure out what the interface is.

Let's talk a little bit about the place where you are coming from. You are obviously from the UK. I'm curious, can you tell us a little bit about the bug bounty hacking community in the UK? How big is it?

I would say that the bug bounty community in the UK is fairly small. I think that the hacking and infosec community is pretty big though. You know, there's a handful of bug hunters that are pretty well known from the UK. Katie Paxton-Fear for example, who is fantastic. She puts out a lot of great content. There's Zseano as well. We've got Alex and lots of other people. It feels like quite a minority compared to the US or India. Especially as the population is much smaller here.

Back to actual hacking. Do you have any tips for our audience what you do when you approach a new target?

I think my main advice is figure out how things work. There's a real temptation to jump into trying to exploit stuff. But if you haven't taken the time to understand how the system actually works, in which way it's supposed to be working, it can be much harder to spot when things go wrong.

If you don't understand the different types of data that are being passed around, which token is used for which thing as part of the authentication flows, for example. What happens if this particular piece of data is exposed? Is this piece of data sensitive?

Some of the best advice I've ever received actually was probably from Inti, which is read the documentation. Some of the most critical bugs are laid out for you in the documentation, should you care to read it. Just read it with that mindset of looking for inconsistencies or things where systems disagree with each other about how some pieces of data should be treated.

One humorous question at the end. Would you rather want to build a hacking tool in C or in Rust?

That's a tough call actually. I think if I had to publish it and have people actually use it then Rust (I would have to learn Rust first for the record). If I actually had to just do it for personal use, I'd probably pick C.

Do you have anything left you want to share with our readers?

I want to tell people to stay curious and learn how things work. Learn how things work and why they work and that will help you more than memorising different payloads or anything like that ever will.

Did you like the short form of the interview? Do you want to hear more from Tom? Watch the full conversation between Pascal and Tom right now on Youtube: