



PHRACK PROPHILE - Halvar Flake

Name: Thomas Dullien
Handle: Halvar Flake / HalVar on IRC
Handle origin: Children's cartoon about a smart viking child
AKA: elwood_blues@viking.ruhr.com in the 90s
Affiliations: TESO
Active Since: I think 1996 or 1997?
Country: US, Germany, Switzerland, France
Website: thomasdullien.github.io
GitHub: github.com/thomasdullien
Projects/Code: Uh. Yeah.

Heavily abridged print version. The digital version is twice as long.

Background

I'll steal Gera's approach and have a short version and a lengthy historians version:

When I was 8 or 9 I asked my brother "how do you crack copy protections", and he said "it's easy, you just invert a jump". I didn't know what a jump was, but remembered the advice. I learnt assembly at 16, then started cracking and virus writing, then moved into reverse engineering and hacking/vuln-dev, started training government folks on doing that to finance my math degree, accidentally started a company, got acquired by Google when my health was bad, recovered for a few years, got tired of security, started another company in performance optimization, sold that, then spent a few years caring for family. Now interested in LLM and AI stuff, because everything is in motion and new and interesting.

Version for historians and really bored readers:

Dad taught me BBx BASIC when I was 8, but didn't do much with it. I typed in magazine listings to get games, but couldn't debug them to get them to run. I asked my brother how to crack copy protections, and he said: "It is easy, you just invert a jump. You need to know assembly." - and that's all I remembered. I became the artsy kid

instead, drawing, and as a teen trying my hand at editing Quake 3D models, using a software called "QuakeMe" that had some limitation in the shareware (non-paid) version.

I went to the US for a high school exchange year. At Barnes&Noble I saw "Master Class Assembly Language" by Wrox Press and bought it. It taught me what a JMP is, how to put the CPU into protected mode, how a DPMM extender and paging works, and all the things I needed. I installed an IRC client and found the cracking scene, joining #cracking4newbies (you'd get booted from #cracking immediately if you weren't obviously good). I started learning and cracked QuakeME with a *terrible* butcher job, but it worked. I was hooked. I read ed!son's cracking tutorial, then stumbled into what would become Fravia's reverse engineering website.

I had found a practically infinite fountain of puzzles and peers to puzzle with. The EFNet cracking scene was a very vibrant place. An entire generation of curious kids, spanning 10-15 years age difference, congregated in shared spaces. There was a lot of 1990s weirdness (from today's perspective toxic or even sociopathic), but many weird kids found their communities. I met many people that are now important in the security world there - too many to name. Rolf Rolles, scut, acp, random_, Neel Mehta, Stephan Esser, Anders Vogh, Solar Designer, vizions, the_grugq, the_owl, G-R0M, iceman.de, Nishad Herath, etc. etc. etc.

I hacked my first computer in 1996 by abusing the PHF vulnerability, a machine at a Mexican university. I didn't know what to do from here, as I didn't know Unix at all. I read VLAD and 29A magazine. The Win32.Cabanas source code and article in 29A were an absolute eye-opener for me, and I meditated on Win32.Cabanas for a long while. Solar Designer's Crackme that included custom virtual machines and everything involved with executable packing (PECrypt, Procdump32). I read Phrack, too, but until Smashing the Stack, a lot of hacking articles seemed alien to me. The cracking scene was full of assembly, interrupts, some crypto even, and many hacking articles were "hey we social engineered our way in" and "here is a manual for some obscure telco system". Fascinating, but far away from what I knew.

"Smashing the stack" was an eye opener to me. And writing shellcode was easy if you were used to writing keygens in assembly. But as a cracking scene kid I knew nothing about UNIX, or using a shell other than the terrible CMD.EXE.

I had two years of German high school to go. At the time, there was a conference/party called "UC3D" - "underground conference 3D" - which was one of the last places where the cracking scene, the demo scene, and the hacking scene still had touchpoints (the scenes had largely split and diverged). I took a train to Bingen in Germany and met random and lorian, who I had known online for years. The party had live-on-site-demo-coding competitions, but also a "hardware compression" competition (where you had to crush a piece of hardware to the smallest possible volume).

Shortly thereafter a "mass exodus" from the cracking scene into the hacking scene happened. Scut, an early TESO members also had home in cracking, invited a bunch of crackers into TESO. It worked out great. The cracking scene folks merged with the existing highly technical hacking folks, and for a brief moment TESO was among the most respected "hacking groups" for sheer technical output. ADM was quite possibly deemed cooler, but TESO was up there, and for a brief moment we had an extremely collaborative culture. We wrote up everything we learned in "TESO informationals", text files that described what we were up to - many of which would've justified a conference talk. We hung out on IRC, we exchanged literature recommendations, and helped each other solve problems.

Near high school graduation I saw a post on Bugtraq about a stack overflow in Netscape Enterprise Server, an important commercial webserver at the time (it ran CNN.com, which was an important news website, culturally about as relevant as Fox today). The people that had posted the overflow seemed to struggle with exploitation, and I emailed them telling them I could do it. They emailed back that if I succeeded they'd send me \$1k. I sent the exploit the next day. They offered me a job in Bangkok, and given that I was about to graduate from high school and had a few months of downtime before starting my replacement service for the mandatory military service, I agreed. I was happy, because in the past year my home life had gotten very difficult due to things outside of my control, so a path offering me a quick escape and some money was great.

I graduated from German high school and took a flight to Bangkok the next day. It was there that I met a few old-school hackers that had started a security consultancy. I also met Fyodor Yarochkin. We worked together, and I consider him a close friend and a great human being to this day. It was fun times -- software was so full of bugs,

and people were not used to buying a pentest and getting hacked by zero days in core infra services. It was also fun to be 18 and in a foreign land with good food and interesting work. I remember Fyodor and me playing around with restricted x86 shellcode at the time, and managing to do alphanumeric shellcode with '_' and '-' added (don't ask me why I needed those characters now), and scripting IDA trying to do static analysis using IDC. It was fun. Fyodor also got me interested in all the strange other architectures at the time, from PA-RISC (where the stack grows upwards) to SPARC (to this day the most beautiful assembly).

I returned to Germany to start the military replacement. It was in a home for troubled youth in a very rough part of town. The job gave me about 2 hours each morning to hack on my own stuff before the kids arrived.

While in Thailand I had agreed with a fellow windows hacker, Joey__, that we should meet. He was one of the few people at the time that wrote Windows exploits (I can really just remember a handful of nicknames: darkspyrit (Barnaby Jack), Joey__, a Japanese person that called himself UNYON / dark penguin (?) that later worked at EEye ... and a bit later Greg Hoglund (?).

Everybody kinda knew (of) each other.

I had always admired Joey__ and we agreed submit to the same Blackhat conference so we could meet. I submitted, and got accepted, and somehow Joey__ didn't submit -- I'm hazy on the details -- but through this I ended up presenting at Blackhat Amsterdam 2000 on reverse engineering. Afterwards I was asked if I could talk more about this. I was 19, fresh out of high-school and needed a way to finance myself (I was not going to allow myself to be dependent on my parents, given the difficult situation at home).

I started doing reverse engineering and exploit development trainings at Blackhat in order to finance my studies.

I got into contact with all the usual three-letter suspects. In my first class, an experienced "DoD" reverse engineer (with like 13 or 15 years experience doing reverse engineering, an unimaginable timespan for a 19 year old) sat in the class with some junior folks and I was confused why they'd send someone with that pedigree to a class like mine, taught by a kid that taught himself to reverse just a few years ago. The person was extremely nice, and

explained that he's mainly there to check out how I'm doing things, but that he doesn't have time to train all the people that need to learn this stuff.

Contact with the US intelligence sector was a bit unsettling at first -- I had grown up in a *deeply* pacifist core family (both my parents saw WW2 as children, and both had deep emotional scars from it - which led them to embrace pacifism to an extreme extent, in spite of a very prussian and quite martial family history). So everything to do with the military, with espionage, and with state-to-state conflict was deeply suspicious to me, and more so to my father. But the IC intelligent, resourceful, and often quite introspective about the moral and ethical dimensions of their work. This soothed my instinctive antipathy.

TESO suffered from leaks. So TESO's previously free-wheeling internal research culture nosedived, and everybody tried locking stuff down more. The leaks did not stop (perhaps some of the infra was compromised), and this got so bad that TESO members would often only learn of the research of another TESO researcher *after* that research had gotten leaked and published.

I had to decide on a subject to study at university. I went to the only subject that didn't have people queuing for it: Mathematics, with the intent to do two semesters to teach myself how to think clearly. I had also read Schneier's "Applied Cryptography" and thought some math would be useful.

For the first time in my life a subject was genuinely hard for me, and I was middling, if not in the lower half of students. The first lecture opened with the Prof saying "look at your neighbor to your left and to your right. If you're here by Christmans, they won't be.". I was too proud to quit, so I slogged on, and eventually had the realization that if I ever wanted to be just an epsilon above useless, I'd have to decide to love it. I did, and I still love it.

9/11 happened a month or two into my first year at uni. Everything went crazy and complicated. The GWoT started, and my view of US actions had to become more nuanced very quickly, in particular in the run-up to the second Iraq war. I still earned my money doing trainings.

I was studying mathematics and struggling with it, only to fly to Las Vegas or other places at regular intervals to teach a class full of people much older than me (and often

in government roles) on reverse engineering and vulnerability development. It was very schizophrenic, and a tiny bit isolating.

In 2003 Microsoft published a bunch of patches for the Windows DCOM system after the hacking group LSD had reported a bunch of Windows remotes. The In order to mitigate the risks MS wanted to have the patches available but provided no information about the vulnerabilities.

I got annoyed by having the patched executables but no way to understand what the bugs were that we patched. Microsofts release executables were shipped with something that is called PGO nowadays (basic block reordering based on profiling data -- Microsoft even had a patent on this stuff), and their patches weren't -- so any byte-level comparison was out because everything had gotten shuffled.

The fact that all the bytes were shuffled but the graph structure had remained stable for almost all functions led me to build the first prototype of BinDiff.

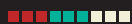
Near the end of my Vordiplom (roughly equiv to undergrad, but not a stand-alone degree) I took a cryptography seminar. I presented my part of the seminar, and it was the only time in my undergrad that I did well. The Prof took me aside after the seminar, and told me: "Listen, I see your grades and how you're doing here. And I see how you did on this seminar. My recommendation to you is: Go do something that *really* interests you for a change."

I had heard that Dobbertin had just become a Professor in Bochum, and I decided to switch universities with the goal of writing my master thesis on MD5. I thought at the time that the construct looked shaky enough to have a go at it. I'd have a semester off beforehand. I decided to go to Argentina, where many friends were working for Gera's company, CORE.

Those 2-3 months were some of the best in my life.

BinDiff really became useful, and I hacked up a website for the company around it. I flew to LA from Argentina to teach a training class on patch analysis.

There was demand.



PHRACK PROPHILE - Halvar Flake

This Microsoft PCT SSL security patch hit while I was in Argentina, and it was BinDiff's first "big moment" -- the tool pinpointed the fix, and within 24h of patch release there was a working and reliable exploit in CORE Impact, allowing remote access to any unpatched IIS installation with SSL enabled.

Good times. It was here that I first learned about Gera's heap visualization work that he had done for the dtlogin exploit, and it was an epiphany for me.

I returned to Germany and moved to Bochum, into a new flat, in early summer 2004. In August 2004, Wang et. al. published their MD5 collision, and then visited Bochum. I was invited into the discussions with Dobbertin and his grad students and Mrs Wang. This was extremely interesting, validating but also disappointing: My hunch about MD5 fragility had been correct, but now I'd need to work on something else.

Unfortunately, Prof Dobbertin fell ill shortly after the semester started, mostly disappeared from University life and eventually died of cancer. I was stranded in Bochum, the the reason why I was there gone, and I had to wait for the Cryptography department char being re-appointed. At the same time people wanted to buy BinDiff.

(Modest) revenue started coming in, but enough to reduce my travel schedule for trainings. And a short while after, Rolf Rolles contacted me - he had manually decompiled the BinDiff plugin binary to re-compilable source code, and wanted to see if he could work with me on the tool. We found an arrangement.

I also was introduced to Ero Carrera, who was working at F-Secure at the time and had also looked at using the graph structure of malware to perform malware clustering. He also joined the "company", and we ended up renting an office.

The roadmap was: Use BinDiff revenue to build a tool that allows input crafting to reach a given point in a program (BinNavi), and to use the money from that to build BinAudit, a hypothetical bug detection program that would identify security issues in binaries. Over the 7 year history of the company, we never started actual real development on BinAudit, but we did build BinDiff, and BinNavi, including the REIL IR that was supposed to be used to feed into the solvers.

We also had to deal with many suspicious customers that wanted to drop by our office and pay in cash, and misc. other madness. We ended up having the company secretary do basic background checking of our customers. The stories.

ynamics was fun, we grew to 10-12 people, but it was also deeply unhealthy for me. I ran myself ragged: Every time our money in the bank would dip below my safety margin I'd go and sell extra trainings to make up for the shortfall, then sales would recover and I'd still have to do the trainings, while trying to finish my degree(s) (first my MSc, then I embarked on a PhD). So developed stress-induced immune failure -- I got shingles, whooping cough, and a bursitis in short sequence, with the bursitis spiking inflammation markers beyond all reason. My doctor eventually told me: "Young man, we are seeing symptoms in you that we normally see in 80 year old men with immune failure. Whatever you are doing to yourself, if you want to see the age of 35, you need to stop doing that."

I took a vacation and started surfing, and decided I need a way out of the treadmill. We started scheming about potentially finding a buyer, initially in the antivirus industry; we got very lucky that the Chinese Gov got caught hacking Google at just the right time.

We joined Google - it was an exciting place in 2011 - and Google made me a much better computer scientist but a much worse reverse engineer (just because the nature of my work shifted a lot). My involvement in vuln-dev shifted more theoretical -- I had had the important "insights" into the proper way to define what Sergey Bratus called a weird machine in the late 2000s.

I was very happy to be able to collaborate with Mark Seaborn when Flippy the Laptop with terrible RAM was found and sparked Mark's successful run to his Rowhammer exploit. I took a sabbatical, finished the weird machine paper (a slog, but important) and returned into an IC role at Project Zero. P0 was full of awesome human beings, and being in a research group like that felt somewhat liberating.

I grew frustrated with security. So after sitting next to Jann Horn and learning about eBPF by watching him, I decided it is time to do something in performance land, and quit Google to start a performance optimization company.



I got together with a vuln-dev friend of mine, Sean Heelan, with whom I had discussed a lot during his PhD work. The company coincided with COVID, and was bought by Elastic 2.5 years later, just as my mother went into coma and died.

And ... well, it's 5 years later now. LLMs are now a thing and are reshaping everything, including security, and I find myself back looking at some security stuff, because things are changing rapidly enough for everything to be *interesting* again. As Hunter S. Thompson said: When the going gets weird, the weird turn pro, and I'm here for the ride, at least momentarily.

Inspiration

For some unknown reason, I told other kids when I was 8 or 9 that I'd become a hacker and steal \$1bn from the defense budget. I don't know where that came from.

Early Influences

My brother. Then probably Fravia, and the rest of the cracking scene. I never knew ed!son, but his cracking tutorial was huge for me.

Memorable Experiences

Writing shellcode on my AMD 486 knock-off, and being too clever by half: To avoid NUL bytes, I'd push the bytes for "pop ebp, call ebp" onto the stack, and then issue "call esp". It calculated the delta offset just fine on my machine, but I didn't know Intel CPUs had an erratum in the call esp instruction, so most machines on the internet would crash.

Standing on a train platform in summer 2000, eating from a bag of cherries and slowly realizing the full impact of pre-malloc integer overflows.

Reading OpenSSH again and again, and mistakenly assuming fatal() just "prints and exits", not realizing it does a bunch of heap cleanup, hence missing my best shot at having my own OpenSSH remote.

Meeting FX in pre-Kathrina New Orleans and ending up on a stage singing Karaoke to a song with him that he didn't know.

How did Phrack influence you and helped shape who you are?

Without Barnaby Jack's Phrack article, I don't think I would've continued with Windows hacking. I was a teen while route ran Phrack, and I admired everybody that was involved. And to some extent the values touted by some old textfiles are values that seeped into all of us?

What is your favourite bug/exploit?

Each one has their own little story. They end up like pets.

One of my favorites was a RealServer bug where RealServer would take a request URL and break it up into parts, splitting the string at '/', and putting a sequence of char* into an array of char* on the stack - so you'd overflow the stack buffer with char* to your own data, and you'd overwrite EIP with that pointer, so it was really "send the right string and watch it get executed".

I remember Dave Aitel being confused about accidentally executing code while fuzzing *without doing* anything. Gorgeous bug.

Will mitigations eventually make exploitation impossible?

I think properly implemented fine-grained memory tagging can make the vast majority of memory corruption bugs impossible or at least fiendishly hard to exploit. I don't think we have a much better mitigation on the cost/benefit tradeoff currently. Time will tell.

There will be other bugs, less powerful perhaps, that will still be worth exploiting.

Would you recommend newcomers to contribute to open source projects?

I recommend everybody contributes to open source. Spreading source code makes us all richer. I also understand that economics can dictate not sharing the source code -- so there is no moral judgement if you don't -- but I generally think that sharing code when you can is a fundamentally altruistic move that makes everybody better off.

Your opinion in the infosec scene now vs then?

We have all gotten much older :) and we can now talk about health issues as much as bugs. Middle-aged mutant ninja turtles indeed.



PHRACK PROPHILE - Halvar Flake

We need to recall that the 90s were a weird time, geopolitically, as some serious scholars seriously considered "the end of history" and that we'd all just trade and be happy, a form of capitalism-meets-hippie-ideals. 9/11 put an end to that, and the aftermath gave everybody stark reminders what power and steel and torture means. I share some of Gera's concerns about "mercenaries everywhere", I don't really like the militarization of everything. But it's nuanced (read the digital version for more verbosity).

To finish: I used to joke that there are less than a dozen reasons why working in security is fun, and I know most of them personally. Then over time I had to increase that number, and nowadays I could probably name 3 dozen or more people whom I have gotten to know through security that I hold in high esteem. So while perhaps the percentage of people I like has gone down, the overall number has gone up, and since I am good at avoiding people I don't like, my life is a bit richer with the scene today than it was with the scene back then, by sheer numbers.

Recommendations

Technical Books:

"A New Era in Computing" - 1993, MIT Press. Best book. I steal ideas from it all the time.

Beyond that: Too many to list or too few to list. I have shelves full of books I will never touch again, but can't bring myself to throw away.

Other Books:

Heinlein's "The Moon is a harsh mistress"

Lem's "Memoirs found in a bathtub"

Mathematical biographies or autobiographies: Ulam,

Rota, Thorpe, Erdos, Friedman

Anything by Giancarlo Rota

Essays by Vladimir Arnold

Anything by Serhij Zhadan

Descartes Error

A History of Computing in the 20th century

Reflections

Hacker Spirit:

Boundless curiosity, a desire to understand even if people say "no need to look at that", and a true desire for experimentation and learning by absorbing a topic and not just reading about.

Exploit Industry:

You don't argue with the existence of rain.

Career and Burnout:

There's a helpful analogy: Your life is a long car ride, and each of the aspects of "you" is a passenger. You need to find a way to keep them all happy, and that begins by listening to them, and not locking one of them into the trunk for years.

Insights

Important principle:

As a child I understood the "canary" principle: Enforcement of rules is often by making an example of the most egregious offender. If you're breaking the rules, don't be in the first line. And if the first line gets hit, hit the ground and stay there for longer than reasonable.

Hacking Milestones:

I recommend "not getting arrested, sued, or seriously harmed" is pretty high on the list to achieve before you're 25. "Make decisions to make your 8 year old self and your 80 year old self proud" is the milestone for 40. After that, be aware that you're sitting in failing machinery that needs serious maintenance.

The "Art" of Hacking:

Finding unexpected uses of something, either for beauty, for fun, or actual utility / gain.

Favorites

Programming Languages: I actually like C++ these days. Stockholm syndrome.

Pwnie Award: Yes. I got one, Lifetime Achievement, and it feels it was a lifetime ago.

Software: I loved SoftICE. I loved IDA. Nowadays, I just want a compiler and vim and the freedom to play.

Gatherings: The dutch hacker camps. HAL, HIP, WTF etc. -- each time I go, they refresh my spirit and make me want to be around a bit longer.

Locations: Bay of Biscay. Northern Spain. East Taiwan. Rocha in Uruguay. Portugal.

Tanegashima.

Personal

Other Interests:

I enthusiastically suck at a lot of things, but I love the process of learning so much.

Quotes

"Theo, there are no more bugs in UNIX" -- Duke in conversation with Theo de Raadt, around 2001, when Theo tried to quiz him for bugs.

