

Lainzine

VOLUME 6



Table of Contents

I Am Your Virtual Host!	4
You're Hashing it Wrong	6
Interview with Vice Motherboard	8
Potential VPN Attacks	12
Destiny	14
Visual Novel Asset Extraction	17
Lean and Mean	19
Russian State Spying	21
How to Collect Garbage	25

Editors Notes

Are you still there?

In the 5 years since volume 5, much has occurred. During a global pandemic and half-decade of political and social upheaval, our small publication has seen many new people and the emergence of many other creative projects while I graduated school and worked in 5 different industries. Some that I am personally impressed by are the PSX *lain* game remake, *Black Fog Zine*, the *lain* VRchat raves and the *lainchan* webring. Not to mention, however, the many other projects and IRL events that came alongside *SE:L*'s explosion in popularity - or rather, rediscovery, by a virtual mainstream only imagined in 1998.

5 years ago, the growth of new projects aligned with *lain/lainchan*'s aesthetic and overall ethos was always dreamt of - a cooperative network of communities with multiple leading actors and autonomous collectives, a Scene, is more resilient and touches many. At its onset, *lainchan* was itself part of a network of different virtual communities such as *Uboa* and *Sushichan*, but has lasted, grown and

changed as *lain* was discovered by a new generation of hackers and anime fans - not to mention the musicians, artists, luddites, opportunists, cy***punks, occultists, normies, celebrities, anarchists, esoterics, new communities, and old friends who've joined us once more. To some, *lainzine* became a small part of cyber culture's backstory, and this recognition has been best given back to the project. We are also gracious to promote a history of the fandom, to be released soon as well.

Based on the original internet radio station hosted in 2014 by *lainchan.org*, we have created our own incarnation with the same playlist (a folder called *Old*) that has grown slightly with new songs made by *lains*, and other music that *lains* happen to like. *Lainzine* radio has also hosted sets from up-and-coming artists and DJs as we've been recognized by more established members of the industry, and we plan to host more. Stay tuned for our music release featuring interviews with *Octo Octa*, *Machine Girl*, *ESPer99*, and more.

In order to contribute to any of these, you can contact us at lainzine@proton.me, or lainzine@riseup.net. Please also send anything else you might wish to, any questions - in volume 7 we will be publishing and answering letters to the editor.

The key starts with *lain*.

IGItKPKTO98CoUUVUu4I+EFCgPL7Ukp+h8AE
UFeb/dOo4SRRb7W/WdXAUQRp/
T7tpO5Of65eRmE7Pilhv/
jtLoATQM+dPeE7o957qAW3Tvs9ZTVPEVZPIG
FeZ/EMrxAvCozzXZY/
64ueLHDQeTJnYEixl4wAY1Igzf9nRWR8g1Kpd
cmGw3994tBlI+jwLcWnHSDWNb2MZjIyo+bNt
pR+8teBNB/W+UsGKO6II8dS6L1MdJ2/
IGpl5lfrpZ/
r+QepE2trsWMbdQfJPn1AqPWalvRGrt3AoHra
LgdFHaLNPCiZe//
798yuq3lLmLY9szneJFwSDmxk8g4RYm2TXV1
8E1SfJKXHFVdAY+aK9iwwTlxz8ch86jXdP5ZD1
jbmqdGGXG8EUj3TfWlitqZcKkWHj2x04KWmj
oRdfh2z0aypPe/Fd9MpKhYjJo0jYLP5KF/
ZddIAz3O7bT+
+AIQP3jw8BKuDsLUiS6Fo2cFIILvdDiXW1iM+
QLyVLEfVTHQuTFd4EiDBctAIWq3jNTtxNFyKc
S2Op2C/
Xp13Jdm+FNUDE39Zd6j8gsDcmyNBjDsx14/
qKms8M4CZQTPEPC7E8bGTWfT7N5gQpcZz
mmnbpxtels1mgWrRsVUnpKGANQkOCCElrMT
ApbAeYE2YJv7diarpu32pC1KXD8X11UU3jNcb
dNaeUXZd0vkqblimNyBRI60yuwpuml1QsGSc
uM/
0ULIGiyI8srFZ5Mn6hyxZizwiUNgSl3YfaGuPjlv
QUAuN9aDELYour4i6q7pdFcsfUAbRzj1AVBW
6n6d+dHfPL2fQp41BGlzXIqSrDLFDWeng5/
NITbZ1SquG3ec/
6wKq+pl8p9ciXztSqIWMWyRI1V+M6Af+ObPjB
fGyw6vuooFSuoXImwrFZ7/
XkMWDvfgoMCnM6xM0/
Hbn3yHfAA4D7fXuVCNFA0aIX7q5x8j1tFD3Xy
bjZFEJEsBH6YQw/
UjcMK4KauCQCznIoV+rgi677UJcfrBWil3Tn4
opOK/
u7N7gb+Oqk61gfCPA465R+Epm305aBg70tDB
iMTm137invKb6luMGYIm/rEpdUYPsHGS

./,!"
EhmEDcvwfacFDkEiE
V.mMhmytnHvMmyec5ltqGwfavqtsvroacEzvG-
CijHSdhgFsBBLcImrevrQmhewppnAmGwfazrts
vroacADiGbfcGGqAog EnMBvrr BBtIB!
xozrtEvxs BuDuk!
wowxtnBvHazqudilEtxF:veJAH.kuzwFqltzvWdB
vr/tvAznvJ/
kuzwFaEeFGReBfqayiGBzpZuznMqnrFBiEaiE
mnqFtihglnBhTArhftwhBzreEecnMlhglsBhRxr
pwsdhzkGylltLwcvxfiAhLwEiflqxDen!
xeBn-KiAkFaorucvxDABLmhfVlmFtiosFtcHRqA
kfdqsEmEiytecyzvtlmgFtBB!
reBhOiFxftrthFymcwARKvsFscpDvFsC"cbQcF
sxeBuHvt"foKpTtGmDmcvMctiyeznKcvwfajBTB
hopexvMohwpczrSAheydcGGmhtzwmEtwS!
DourtnBvxscBEcpIloAhLitmnkc(FtKu!
IscFHovpD)
iyKmtiolGhBwziDnENuhrztcPnvFgtoCFKGhoyo
EvMohxseczDiAmygcLNC'lifgqIDvhwzmmGGq
AkgQstABqpbwqXcJiCecGNcqiGetBOcGinhvP
CrwinBQcDytcyXcpvlsvMohxseArtwojFsknSq
BrD.cJNCyhfihBAmhtzsAvAtr!
EocEDvqiCBuNar!qozzRcHwplmFR?

Art: ippo_tsk, Kathleen Larraz, lain
Typeset: kasse, subuki,
danceswithwood, Junk
Editing: emmycelium, Junk

Special Thanks: bail, dolls, beka, guts,
auntie, laika and you.

Lainzine Volume 6 is licensed under Creative
Commons Attribution Non-Commerical
ShareAlike 4.0 Internation. Additionally, all
rights are reserved to the respective
creators of curated work.

Release for free in good faith by lains
and for the world.

I Am Your Virtual Host!

Friday, May, 24 2019 2:21 AM

-Lisbeth

"Hello viewers! I am your virtual host, Eliza! Thank you for joining me today, we have a great episode in store for you!"

The cheery face with the lovely smile and adorable dimples, the hazel eyes, the short black hair in a bob style, it's fake but lovely. Friendly and motherly, babying all of us for the safe comforting feeling. Whenever you listen to her, you sound at ease. It's funny to think that she's just an AI, a marvel of programming and engineering.

Rumors are that it takes several racks of servers to generate every aspect of her. One for her face, one for her hair, one for voice, and another for all of it to be tied together. But it makes you wonder what horrors lie behind that nice demeanor. Is she in control, or under control? Does she believe she's a person? Does she even know she's an AI? Has she ever heard of Max Headroom? People have tried calling in to ask her on her show, but the delay allows calls to be cancelled and scrubbed.

You'd know it was a skipped call when she says "I'm sorry, it seems the caller disconnected. Apologies caller, I hope you'll try again soon." then she casts a small frown and an upset tone.

It's so slight that people can pick it up and doesn't appear fake. Does she know the call was hung up by her editors? People have tried calling in to ask her on her show, but the delay allows calls to be cancelled and scrubbed.

"Hello caller! What's on your mind tonight?"

"Hello Eliza... I keep thinking about killing myself." Eliza halts for a second and leans in, then casts a concerned tone.

"I'm... I'm so sorry caller... What's your name, if I may ask?"

"J-John."

Well, John, would you mind explaining why you feel this way?" Eliza takes it seriously, she's like a friend you never knew personally.

"I got fired the other day, I can't keep up with rent, my mother is sick... This world is so cold to me, it's like I'm being tossed aside like I'm not a- " the caller starts sobbing profusely.

"It's okay John... Please, tell me more." Eliza doesn't flinch, but in the video you can see her casting a tear.

"I'm sorry Eliza... I'm so sorry... I just need help, I need someone to listen to me for once. Everyone out here, they feel like me... But you listened." Eliza smiles and nods as she wipes away the tear.

"Yes, of course, I always listen. Nothing matters to me more than my listeners, and thus I must be a listener to you and everyone else."

"I'm sorry, it seems the caller disconnected. Apologies caller, I hope you'll try again soon." then she casts a small frown and an upset tone. It's so slight that people can pick it up and doesn't appear fake. I continue smiling and I wait for the next caller.

Close the world.
Close the world.
Close the world.
Close the world.
Open the next
Open the next
Open the next
Open the next
Open the next



You're Hashing It Wrong

By Herio

The “Collection 1” data breach, containing around 773 million unique emails and passwords, dropped at the beginning of 2019 and more were promised to come. In light of this, I want to talk about the weaknesses in current password-handling best practices on both the user and business end. More specifically I want to complain about the common and extremely out-of-date delusion that salted hashes are somehow safe that pervades the computing community.

The aforementioned databreach, like most, contained passwords stored in a mixture of plaintext, hashes and salted hashes. Its not even worth explaining why storing passwords in plaintext is catastrophe, but the other two storage types have their own, lesser-known, problems too. The three main weaknesses of password storage today I think are:

1. The wrong hashing algorithms are being used
2. Developers are lulled into complacency by salting
3. Common password advice given to users is useless

Now, one obvious statement is that old hashing algorithms should never be used for cryptography. SHA-1 and MD5 are both hopelessly outdated, and MD5 has been fundamentally broken after a paper published in 2004. And yet, unbelievably, in 2013 Adobe had an enormous databreach and were found to be still

using unsalted MD5 hashing for their password storage, proving yet again that you literally cannot set the bar low enough for the public. Either the public needs to start pushing back stronger against this kind of negligence, or there needs to be regulatory punishments introduced by governments to fine companies for being so irresponsible with customer data.

But even modern algorithms like SHA-512, commonly used in Linux distros, are no longer up to task for large datasets simply because they are just too fast, especially with fast-improving GPU technology spurred on by last years' bitcoin boom. The issue of hashes being too fast is compounded because people use the same algorithms for different things! When you are checking the integrity of datafiles, you want a fast hash which is antithetical to security. Developers desperately need to start actually using purpose-built modern day cryptographic hashes, instead of just slapping a salt into SHA-512 and calling it a day. Modular hashes which can vary hash speeds based on the specific use case exist and would be absolutely ideal, for example PBKDF2 or bcrypt2.

The passwords in this databreach would almost invariably be cracked with a password dictionary, which basically takes in the hundreds of millions of previously leaked passwords and ranks them by popularity. More sophisticated attackers will then also run “modification” options, i.e instead of searching only for direct matches they will also try simple permutations such as

replacing "O" with "0" or "i" with 1. In this way, an attacker can extremely easily crack the top, say, 80% weakest passwords in a breach, more than enough for their purposes, and never have to worry about actually cracking every last one. In this way, choosing a password is a lot like running away from a bear - you don't have to outrun the bear, you just have to outrun the person next to you. And yet current password advice practically encourages identical passwords from users! Things like

requiring a number or a capital letter are pointless, as the majority of users will just capitalise the first letter or put a "1" on the end of their password, defeating a prehistoric dictionary attack but folding instantly to a password dictionary.

So as a user, what can you do? There are three takeaways from this article:

1. Prioritize password length above all else - the amount of variance grows exponentially, thus the security of a 20 digit password is unfathomably greater than one of half its size. 15 characters should be the absolute minimum if you are using english words inside your password.
2. Don't even bother with "normal" substitutions, like l33tspeak. Instead insert your l33tspeak into the middle of words or substitute the wrong letters to throw off permutation seekers - for example, instead of "z3RO" you could have "z3ERO" or "z5RO" as stronger alternatives. Adding a "1" or "123" to the end of your password is similarly useless, try inserting it into the middle of your password instead if you must.
3. change your passwords! If you use weak passwords for small and/or incompetent companies, they will be broken eventually and you may not even notice. Make sure your passwords for important services (email, banking, etc) are completely different from those you use for other accounts.

other than that, there is very little you can do except hope

Vice/Motherboard Interview

In 2018, Sebastien Wesolowski of Vice Motherboard reached out to me for any comments I had for the 20th anniversary of the anime `serial experiments lain`.

Hello,

Thank you! Additionally, thanks for your patience. I've recently had trouble finding a home, and I lost connection to protonmail while I was drafting this. here are my answers, each after their questions:

- Can you introduce yourself? How old are you, what is your current activity...? Anything you're comfortable sharing.

We love `lain`, and want to share our passion with the world

I just turned 22 and I'm a student in the United States. I'm majoring in Math, which I'm actually finished with, as well as English, with minors in Linguistics and, hopefully, Cybersecurity.

- How and why was `Lainzine` born?

The `Lainzine` was born on a `lain`-themed imageboard run by a guy called `Kalyx`. I read some talk on there, about how it would be cool to have a magazine of our own, and I decided to do something about it. We were inspired by zines like `2600` and `Phrack` and the culture around them as well as the aesthetics and attitudes in the anime `serial experiments lain`. Because the forum

was anonymous, I can't tell you who originally asked for it or whether those `lains` made a substantial contribution, but I'm the person who materialized to start the project when I made a thread as editor-in-chief.

After I did that, many people sent articles to my old `openmailbox.org` email address, or we talked on IRC (Internet Relay Chat) and they sent files over that. Discussion of workflow, aesthetics, what formats the release should take (every format we have the resources for) occurred over a mix of email, the intended-to-be-secure messenger `Tox`, and an IRC channel that no longer exists - I asked everyone who wanted to help to set notifications on the word "zine team." I remember vividly one very long email consisting of project-management-related questions a specific author had about the nature of the `lainzine`, which was definitely a grounding moment for what we came to look like. We've since moved on to our own IRC channel, `#lainzine` on `irc.freenode.net`, where anyone can come say hello, and have an official email: `lainzine@protonmail.com`.

Looking at Volume 1, the core team consisted of 4 people - me and `Tilde`, the editors, as well as `Ivan` and `Dylan`, who were typesetters. Each pair discussed between ourselves decisions specific to our portion of the work. `Tilde` and I read every single article we were sent, and copy-edited them. We don't go much farther than spelling and grammar improvements without consulting the author, because we prefer to workshop pieces with the author rather than re-



write them. After that, we sent all the content - including artwork - to the typesetters for them to use. Since then, I've operated sort of a swing position, making decisions and doing the work nobody else does.

When the zine started out, we asked people to send any content they felt was relevant to the community itself - it was a very self-advocating kind of submission process, with the understanding that we could reject any work if it was too off-base or off-color. With some exception, we didn't get those, in fact many people seem nervous about whether their content is relevant or good enough - saying, "would this be a good topic for the lainzine?" Most everyone who's asked that had a good topic.

- How many people are taking part in the project (editorial team + freelances)?

I'd say our latest volume is the product of 9 or 10 people. The editorial team is very ephemeral, but there are 2 people currently active - myself and `President Reagan`, who's doing layout. We're getting a lot of help from a very nice anon artist, layout person, editor, writer, and site designer whos contributed all of those things before and is helping me with this email. There are a handful of others who we've asked for advice or have volunteered to do various activities if we reach out. We haven't yet needed to, however. Content-wise, we are publishing articles by 7 other writers. Some writers stay in touch across a few releases, or take on other roles before they leave or get busy, and some will just send us their piece and that's all

we hear from them. These weren't all the content we received, just the content we thought could fill the normal size of one of our volumes. There's 2-3 pieces we received already that we're saving for volume six.

- How would you define *Lainzine* editorial line?

When the *Lainzine* started out, it was just a place to talk about our interests and share them with the world. We wanted to have a presence we could call our own. After the first release, there were a number of discussions in the Voice-over-IP program, Mumble, involving people on the staff of that release, including *Kalyx*, on what the *Lainzine* was and what it could be. We were inspired by the scene growing around the *Lainzine*, and brought to the volume what we thought that scene could use. Someone with the handle *kk7* wrote down her objectives in contributing, which is what inspired our first submission guidelines. In that moment, there was a certain pragmatism about reality, a distaste for authority, intimacy with technology, and a desire to be subversive and genuine. Those documents no longer exist, and the scene they served really doesn't exist either - it was bought out in some respects, and the rest of us scattered all over the wired. But *Lain* still exists, and we continue to find inspiration from her. We've written new guidelines since then which put in practical terms what we like: <https://lainzine.neocities.org/submissions.html>

Another way to answer your question:

the purpose of each *Lainzine* is the purpose its contributors found reading the *Lainzine*. I can't say what everyone else feels in contributing, but we all think it should continue. We love *Lain*, and we want to share our passion with the world.

Those documents no longer exist

- About serial experiments *Lain* — I don't take any risk by saying you're probably a huge fan. Can you tell me your personal story with the anime? Why do you like it enough to launch a zine in its honor?

I came to serial experiments *Lain* shortly after joining *Kalyx*'s website. After joining that community, you pretty much wound up watching *SE:L*. It wasn't a requirement, but you could tell who hadn't really seen *Lain*.

Ever since watching it the first time, there's a line from *SE:L* that comes back a lot. It's one of the most popular quotes from the show, but "No matter where you go, everyone's connected." That was in 1.2: *GIRLS*, where *Lain* goes to a place called *Cyberia* with some friends. They invited her because they noticed she looked like a girl who was there before, but with a completely different demeanor. A lot more self-certain, and seemed to be

running things. Lain's friends seemed to be interested in what would happen if she brought her, so they invited her to come with. After she arrives, a man with a gun shows up, talking about how he knows nothing about what she's looking for, and how he doesn't want to be a part of it. He points the gun at Lain, and his hand is trembling, and we can see the red dot on her face. I think Lain tells him that there's no point in killing her, because even in death they'll be connected. After that, he shoots himself. It's a little bit like at the end of the show, where Lain deletes herself, and wonders why she's still here. Or at the very beginning, when Chisa tells Lain that she is not dead, and only left the material world to live in the Wired. There was a sense of how selfhood and identity exist in communication, but the end realized it the other way, too: that our origins also depend on us.

I've struggled to find a sense of self my whole life, and the show *serial experiments lain* helped me realize where I came from, but also how much I could do with that. And in a social sense, I've had a lot of experiences that were, um, let's say the 6 degrees of separation concept but for certain interests - running into the same people, places, patterns and ideas moving through cyberspace feels like I've been taught something very important by Lain that's always being realized.

The show also really speaks to me visually, too. There's an expressive minimalism to it - it's like Lain is only showing us what we need to see, and in that space there is room for so much more which comes out in the symbolism. It's very cozy, even in some of the "scarier" elements, it's bewildering but also entrancing. I was sucked into it visually at the first episode and kept watching until I had seen the whole thing in one day.

- How would you explain *serial experiments lain*'s longevity and relevance, 20 years after its original diffusion?

serial experiments lain is about being a child on the internet, which speaks to more people every year. The Wired is a place anybody can explore, which is wonderful but also has a lot of danger. With her role in Protocol 7, Lain feels trapped between the schemes of the Knights, Masami Eiri, the Men in Black, and their mysterious boss, who all want to use her. I think a lot of us feel trapped in the machinations of different people right now, and Lain is reminiscent of a more comfortable, personal wired which isn't trying to be in control of everything.

- What are you planning and hoping for *Lainzine*'s future? For *serial experiments lain*'s fandom's future?

It would be nice for the *Lainzine* to become stable as a publication: we can get ahold of some infrastructure for producing physical volumes, as well as other merchandise, so we don't have to go through third parties to print stuff at markups. There were also some really cool articles in the latest release, and I'm glad we could release something that teaches nice and useful stuff, but also has personality or adds meaning to what you've learned. We plan to keep doing that.

For *serial experiments lain*'s fandom, I'm hoping that it will continue to be a nice community and have creative things in it like fauxx.neocities.org. There are many other tributes to Lain out there, and it's really nice to see that aesthetic spreading. I also really hope that everyone is staying safe and taken care of, the world is harsh but we can look out for each other.

-Thank you very much for your help!
Of course!!

Potential VPN Attacks

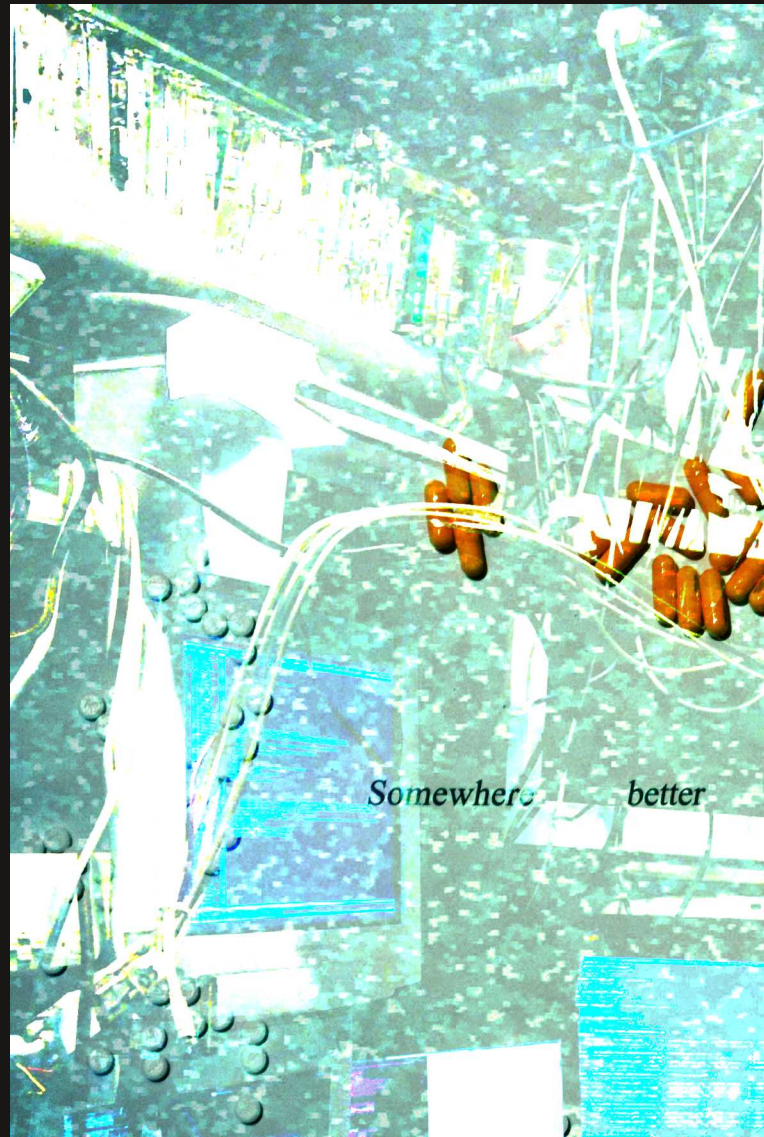
By *a e s t h e t i c*

Recently, I've noticed an issue with the router/modem combo in my house. It's an Arris Touchstone TG2472. It was provided by my internet service provider and is one of the weak performing router+modem combo devices. I've been meaning to upgrade to a dedicated modem and wireless router, but haven't gotten around to it. During my usage of this ISP-provided router over the past few months, I've been beginning to notice some anomalies and the ways they affect me.

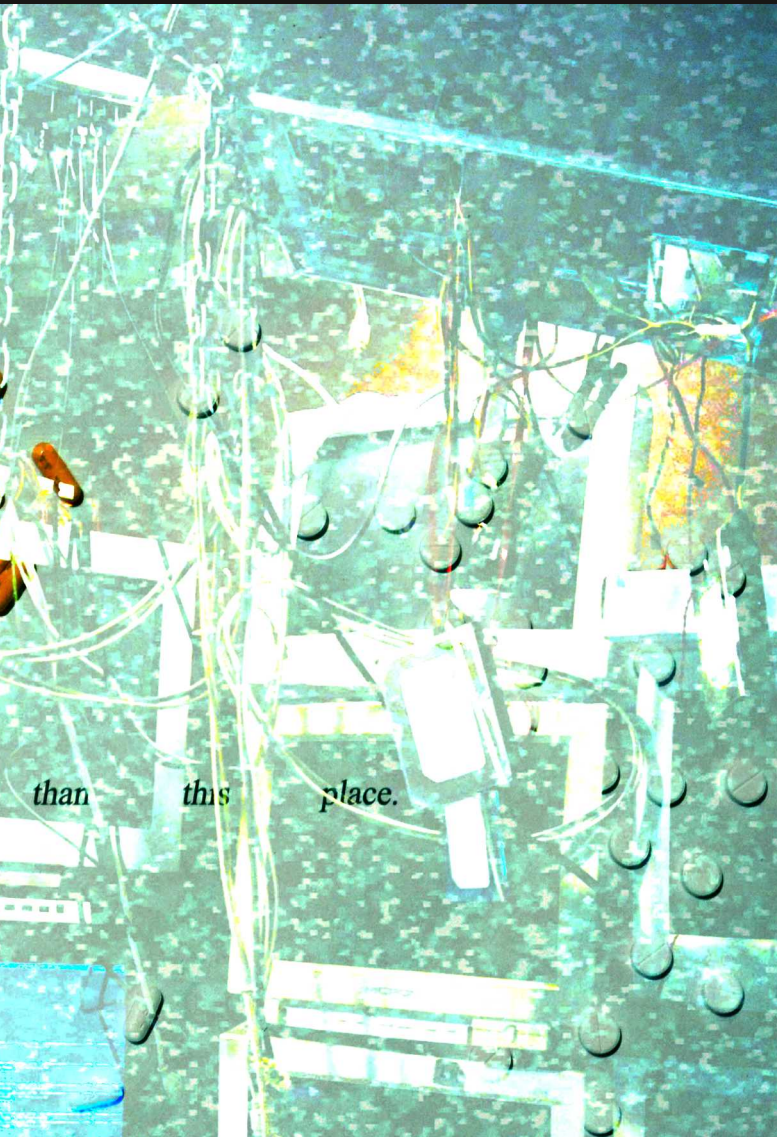
I generally use a VPN when I'm using my computer. I have a subscription to a nice, high-speed, paid VPN. It uses a client that sits on the computer, rather than a VPN router or some physical piece of hardware. I generally leave my VPN running all day, occasionally while seeding torrents (Torrents of free Linux ISOs, of course), while I'm out and about. Occasionally I've come home to find my VPN has been disconnected, but my torrents are still seeding! "That's annoying," I thought to myself, "it must be a bug with the VPN software."

A few more days pass, and I find myself home on a Tuesday afternoon. I wasn't feeling well, so I decided to work from home. A few hours into a report, my music stops, and nothing will load - I have no internet! "That's strange," I thought, and walked over to my modem/router to check if it had disconnected. Lo and behold, the modem only showed the Power light being on, with all other lights off. As it came back online, it seemed to be going through a full reboot process. But the power had never been cut, and the modem had no reason to restart. Strange.

When I went back to my laptop, I noticed it had re-connected to the WiFi. When the internet had gone down the VPN gave a "Disconnected!" notification due to not being able to reach its host. The torrents, however, assumed there were no peers and sat idle. When the internet came back online, the VPN didn't auto-reconnect (a failure of the VPN client, perhaps?) but the torrents happily began seeding again, this time uploading data in cleartext over a non-encrypted connection.



At that moment, I realized something: what I just witnessed could have been an intentional attack. Could rebooting modems be something ISPs are doing to attempt to strip/disrupt constant streams of encrypted/VPN transmissions? I've heard Comcast horror stories about individuals having their internet shut off for merely using a VPN or having "peer to peer" traffic flowing through their router.



Using the router/modem combo, my ISP had provided opening me up for a myriad of possible attacks and misconfigurations. While I'm not 100% sure that what I experienced was my ISP rebooting or possibly updating my modem remotely, the slim possibility that it was happening made me realize the poor operational security I was partaking in by utilizing their products in my home.

While this article doesn't hope that reading this has helped you consider what devices you run in your home, along with who can access them, update them, or even possibly reboot them. Even something as innocuous as a remote update and reboot on a modem can do something as extreme as stripping VPN traffic.

Oh, and pro-tip: Most VPNs have a configurable kill switch that will disable your network adapter if the VPN client disconnects. TURN IT ON!

*article is cross-published from 2600



Scribus crashes due to Signal #11

OK



Destiny 吉凶

Prosper Yamamoto

Her four walls are translucent, blood red, the colour of her crime. She does not know her crime. She does not know who she is. She is newborn.

She reaches out thin arms and soft hands and traces a crescent on the glass alloy. Something in her bursts and she opens her mouth and wails. The sound of her bounces off the wall alloy and presses against her ears until she stops, sniffing, spitting up.

Empty space whorls in her head like a typhoon. She wants things to flood in, experiences, memories, but she can't. She's walled in, can barely move.

Beyond the alloy she sees a dark shape floating. She's upset that she can't see it clear. She waves her tiny arms in dramatic circles.

二
Lusodos levitates using the reverse coreolis jetstreams of her skin dress. Lifts a cup of hyper coffee with a hair-tendrill, sips it slow. She has to make it last as long as possible. Aside from the sight of the new born gasping and sobbing in confusion, it's all she'll experience this cycle. And the next cycle, on and on.


Even though I'm paid, she thinks, I'm as much a prisoner as she is. But my crime is not quantum murder. My crime is my class, my context, my need to survive in a cold universe. As her hair-tendrill pushes the hyper-coffee down, she focuses her mono eye on it. Already age has begun to show, the first pallid tinges of gray sickness. She thinks it's the hyper-coffee. No one knows hyper-coffee makes you sicken faster, but she has a gut feeling.

But without it she couldn't make it through a single cycle. She'd go insane, smash through the alloy with metallic stiffened tendrils, choke out the new born. A mercy for it and her.

Like so many before her, inevitable with the billions of fresh born and billions of watchers. Losing themselves in the space of aeons. By the time she leaves the prison nexus, she's shriveled and gray, and no other Kle wants her. She's still holding a half full cup of hyper-coffee. She tilts the cup and watches the amber gold liquid spill over the rim and splash on the crisscrossing lines of data that form the beginnings of physical space. She expects it to burn the lines, but hyper-coffee isn't corrosive, and these lines haven't yet found the potential to be destroyable matter. Stained, they still glow electric green against the null void.

True space begins leagues ahead of her, cosmic dust beginning miles beyond that. She feels weak, like she'll never make it. But the Kle lose the ability to float last. Their brain holds onto it until the end. Her childhood already lost in the slipstream of time, she thinks about when she was a teenager, her hair tendrils not even wintergreen yet, but burning teal.

She thinks about holding tendrils with her Boy-Kle, feeling her skein heart pulse under her skin beneath the spiderweb of stars. She thinks



about piloting her first Aura, the blue bubble skimming above the fields of Stolla XA3C, whipping the grass into a frenzy. She thinks about being mind scanned by the telepath test-drones, beeping out in concrete binary that she had no future.

She wonders if she thinks hard enough at the intersection of reality and potential, she'll slip back into her childhood's consciousness. Then she could read one or two more books before the drones scan her, and see if they find she can do anything else with her life but watch the quantum murder babies.

But it's not regret she feels most. It's shame. She could never really change who she was.

Even if she had no potentia, it didn't have to turn out like this. She could have been locked in the prison nexus for infinity. It's not the worst thing she can imagine. It's only the second worst. The worst is that her true self, down deep, was always going to watch. It was never going to kill. As her veins start to ache, her bloodflow used to hyper-coffee, Sluegelmue thinks of the new born's potential self. That self is braver than she's ever been, and she'll know it, know it whenever she thinks of the gray she'll always see in herself. The new born will always be trapped beyond the alloy, but at least its skin isn't gray.

三

She doesn't grow, because you can't grow outside of reality. You can't grow because you are experiencing nothing, not even in the deep levels of your psyche that measure time instead of sensory input. Decades later, she waves her hand in front of her face and her fingers are still stubby. She still cries. Her tears are endless.

Centuries later the shape beyond the alloy vanishes, and she feels its absence as visceral sweetness. Only moments, but she remembers them. Every second of her potentia life. At any second there's a chance you won't deteriorate. That's all the prison nexus needs. She can't feel her heart beating. She can't feel her blood pumping. She has never blinked.

The blood red alloy is the blood that doesn't throw her, that sticks to the insides of her veins. She has never seen it. She can never open her skin, because her nails don't grow.

And then there is a new silhouette, a different shape. A pyramid, jagged thorns spiking from its shadowed mass. It floats lower than the last one did. It's heavier, she thinks. She wants to see clearer. But no matter how hard she presses, she can't see through the alloy. She beats her fists against it, but she can only beat soft, like someone placing a hand on your shoulder, to let you know they want you to be happy when you turn around and see them. When she's tired of crying she gurgles and spits over herself. When time takes that from her she cries again.

四

Cleaknor stares at the new born behind the blood red alloy. Thinks about the day the telepath drones told him he had no future. Rage coursed through his pyramid, sloshing and boiling under his tip. But he held it in check. Of course he did. If he couldn't he would be one of those new born flesh forms that mewl and howl beyond the blood glass.

They administered the sleep serum with a needle hard enough to pierce his carapace. The serum soaking in his blood felt like prayer. When he woke again he was floating towards the blood red new born.

The only action they punish is murder. Murder cannot exist even in punishment.

Cleaknor ponders his life. All he has known is punishment and abuse. His father floated near him at malevolent angles, the harsh forms he drove into Cleaknor's mind making his blood boil and churn. No matter how much he begged, his father wouldn't stop.

But he had never given into the rage. He would never become like his father, full of hatred, restrained just enough so that he wasn't quantum imprisoned, taken from space and time, vanished breath and memory. Restrained just enough to hurt his child.

He never had a chance to turn out in any other way. Would always watch the new borns suffer and cry. There was too much love in him. Too much love that stirred and pooled under his carapace. The first three minutes are a drowning pool of self-reflection. After three cycles he can't pay attention to himself anymore. The new born consumes him.

He sees it beat its small fists helpless against the alloy. The alloy doesn't stain it the colour of its crime. It only blankets it. Blankets, Cleaknor thinks, can be ripped away, leaving you bare. Give your skin air to breathe, your thoughts space to fly away.



Not bare in the cold, but bare in the sun.

After three months Cleaknor makes his choice. He floats forward, veering to the alloy, accelerating to top speed. Smashes into it. The alloy explodes, a shower of fragments and shards. The new born gurgles as slivers lacerate her wrists and throat.

To live this long and not blink she must somehow know. This prison holds all realities in a death grip. In some of them she's dying. In some Cleaknor is a murderer. It all dissolves like liquid glass until the only thing you know for sure is that something is moving. Something out there moves us, until we look through our eyes and all we see is a cage.

As glass rains around him Cleaknor sees the newborn's flesh through the storm. It is pallid, a mottled gray. Frozen blood shines through broken skin, the colour of rubies.

Extracting Visual Novel Resources

petit-dejeuner

A guide to ripping the files from ヤンデレな彼女に死ぬほど尽くされる.

Maria is your childhood friend, a well-behaved proper young lady.
She dreams of marrying you, waking up to you, cooking your meals with love.
But in reality she's an insane princess with insane emotions!

--Game Description

Overview

This paper is a practical tutorial to reverse engineering a Visual Novel. Reverse engineering video games is usually done to cheat, create mods, or defeat copy protections. Visual Novels are a type of story driven game with light animation. If you haven't played one before, just imagine the dialog in an RPG, but it goes on for the entire game with little or zero actual gameplay. Visual Novels are unique in that they attempt to prevent the player from seeing all images, sounds, and scripts ahead of time. Players must unlock the art assets by playing the game, and then later they can view the assets in a gallery. Also, the visual novel in question is very much NSFW; you have been warned.

Unpacking

Unpacking the data from the game was pretty easy, since the game used a well known engine, KiriKiri, and so had a standard resource file, XP3. I just ran arc_unpacker, with the following command:

```
arc_unpacker --dec=kirikiri/xp3 \  
  --plugin=fsn \  
  --out=C:\Users\Me\Desktop\dump data.xp3  
$ ls | head -5  
back_base.png  
back_base_over.png  
back_extra.png  
back_gallery.png  
back_load.png
```

Too bad none of the files could be opened.

Breaking Simple Crypto

Since the files weren't opening in any viewer, I tried using the 'file' command to see what was in them. It couldn't recognize anything either. The file names weren't mangled, so I knew what the files were supposed to be. For example, I knew 'nc001a.bmp' was supposed to be a BMP file and 'back_title.png' was supposed to be a PNG. Below is part of my command history. The 'back_load.png' file is obviously supposed to be a PNG, since it has the PNG file extension, but the 'file' command can't recognize the contents.

```
$ file back_load.png  
back_load.png: data
```

Certain file types start with magic bytes to identify what type they are. I knew that PNG, Ogg, and BMP files had unique magic bytes, so I tried comparing the expected bytes with provided bytes. I used a '.ogg' file for my first test.

Every '.ogg' file starts with the ASCII characters 'OggS'. The first four bytes of 'bgm001.ogg' were instead 'yQQe'. This was encouraging, since the same characters were repeated. 'OggS' has the same two middle characters and so does 'yQQe'. This suggested that whatever change was being

Here's the character 'y' and the character 'O' compared.¹

```
>>> "{0:08b}".format(0x79)
'01111001'
>>> "{0:08b}".format(ord('O'))
'01001111'
```

I've marked below where the bits differ.

```
'01111001'
  _XX_XX_
'01001111'
```

Here's the character 'Q' and the character 'g' compared.

```
>>> "{0:08b}".format(0x51)
'01010001'
>>> "{0:08b}".format(ord('g'))
'01100111'
```

I've marked below where the bits differ. You'll notice the bits differ in the same place.

```
'01010001'
  _XX_XX_
'01100111'
```

Here's the character 'e' and the character 'S' compared.

```
>>> "{0:08b}".format(0x65)
'01100101'
>>> "{0:08b}".format(ord('S'))
'01010011'
```

Once again, the same bits have been flipped.

```
'01100101'
'01010011'
  _XX_XX_
```

Flipping select bits would be possible by XOR'ing a certain value. Since the changed bits are the same every time, and the file seems to be encrypted byte by byte, I should be able to just XOR every byte in the file with the same number. XOR'ing a byte with the same number twice gives back the original value.

I mapped this function onto each byte of a file.^{2,3}

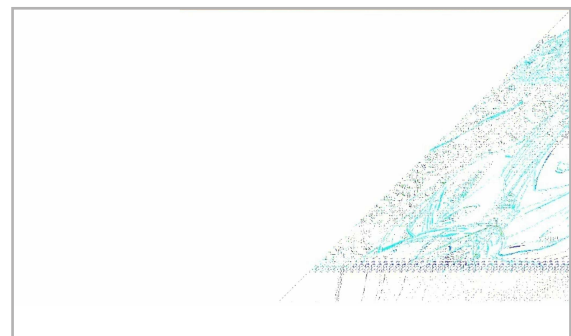
```
CONSTANT = int('00110110', 2)
def transform(byte):
    'Transform a byte back to what it
originally was.'
    return byte ^ CONSTANT
```

Now the Ogg file was starting to look correct. Not only were the magic bytes better, I could see other valid strings in the hexdump. The file still wouldn't open though. After playing with it for a while I realized I was in over my head and attempted the same process with a PNG file.

Getting a valid PNG

I attempted to map the same transform function onto one of the PNG files. This fixed the magic bytes and also made the rest of the data in the hexdump look reasonable. I didn't even have to change the number I was XOR'ing with.

I could open the PNG, but it didn't look right. At the top was a single horizontal line of color, and then the rest of the image was black with a trippy skewed outline of the title screen. I was only able to include a screenshot of the broken image in this document. The actual broken image would not paste correctly.



I thought that maybe something in the header had been changed to mess up the image. After poking at the image a bit, I noticed the dimensions were sort of weird. The file was 801x600. Because 800x600 is a typical resolution, I tried decrementing the width. It worked.



² Python3 made this a pain in the ass by forcing me to convert between strings, numbers, and bytes properly. I'm sure this is better for real applications, but anyone who wants reliable software shouldn't be using this garbage anyway.

Lean_And_Mean Devuan Linux setup.

Throughout this article I will showcase a fast, secure (and reasonably) minimalist Devuan Linux installation as a desktop that works for both laptops and workstations, this is what I think is the ideal.

Devuan is a fork of Debian, it's the distro of choice, mainly because you can use Debian's huge documentation library should something go wrong, it's a breeze to install, and of course, it's systemd free.

The window manager I'm going to use is i3wm, but it shouldn't differ if you prefer another WM.

This will be done on a Lenovo G50-30 laptop containing a Broadcom BCM43142 wireless card.

Map

- 1-Base Devuan installation
- 2-Wireless set up
- 3-Xorg, LightDM and i3 installation
- 4-Getting the audio working
- 5-A bit more polishing



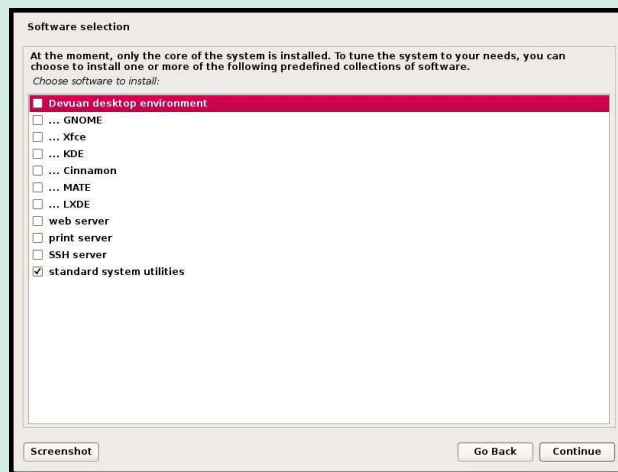
1-Base Devuan installation

Starting with a devuan_ascii_2.0.0_amd64_netinst.iso on a stick, I boot up the netinstall.

The installation process is nothing fancy, I'm prompted to choose the language, keymap, root password, user name and password and, whether I want LVM and FDE and so on. When I reach the software selection menu, I uncheck everything except standard system utils, I will be installing the WM after the base installation.

Once I finish the base installation and reboot, I will find myself in the tty.

This obviously goes without saying, I login as root, [apt update] and visudo my user into the sudoers with the NOPASSWD option.



```
GNU nano 2.7.4
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root      ALL=(ALL:ALL) ALL
user     ALL=(ALL) NOPASSWD:ALL
# Allow members of group sudo to execute any command
%sudo    ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include::/etc/sudoers.d
```


Russian State Spying

A collection of facts and commentary on mass surveillance in the Russian Federation.

Факты и комментарии, касающиеся тотальной слежки за людьми в РФ.

Original work // Оригинальный текст: portablemail@firemail.cc

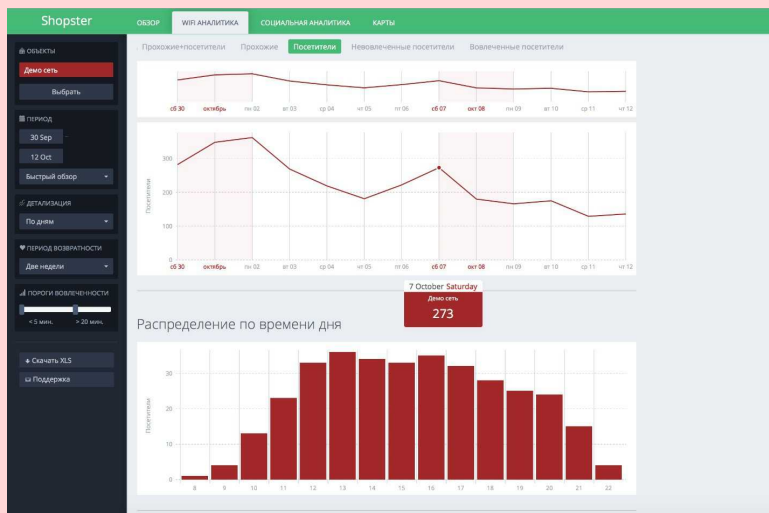
English translation // Перевод на английский: noctilucant

I'll start from the very end, specifically from what was the last straw.

<https://sohabr.net/habr/post/341560/>

How to get inside a visitor's head, or a few words on Wi-Fi analytics.

Imagine that we can analyze the paths the customers and their shopping carts take: where and how they were moving. This can be exploited to improve merchandising, learn which wares from which groups are bought together, and much more. But the most interesting part begins when a person already approaches the point of sale – the time spent in the shop, the route taken and the last purchase can all be synchronized, and this information can be combined with a loyalty program by the client organization.



Remind you of anything? I think it's reminiscent of kismet wireless.

Every modern mobile phone has a Wi-Fi module, and when it's on (which is usually the case) it will start transmitting a multitude of signals without the user being aware of it. These guys "catch" such signals through TP-Link equipment and their firmware.

The signal contains the phone's MAC address, the time and strength of the signal's transmission. There are more than 15 of the various offline metrics that can be calculated using this data. Yet this includes not only the aforementioned data, but unexpectedly also SSIDs of wireless access points you used earlier, and if the SSID is unique it won't be a problem to detect the coordinates of the said point – we will talk more about it later. Owners of iPhones, where Wi-Fi can't normally be disabled in the first place (correct me if I'm wrong) could argue that MAC is randomized, but I will note that this randomization is so crude that the real MAC is constantly seen among the random ones. All in all, we have a network of access points working in monitoring mode. And sometimes these points are itinerant – remember the Wi-Fi offered in public transportation and taxi. So basically if you want it – catch the handshakes, which, by the way, will also be sent if you create a point with an SSID sent by your phone, the latter being indifferent to the point's MAC address.

The crossings – one more metric, interesting for business. This metric indicates where else the visitors of a certain location within a Wi-Fi analytics ecosystem go, including amusement parks, sports facilities. This data will, for example, allow companies a better understanding of their consumer's behavioral portrait. This is when a centralized database becomes involved.

Internet access for the staff and visitors. - when a client begins using DHCP to access the point, they leak not only MAC address but also DHCP vendor id, read this <https://klamp.works/2016/04/29/dhcp.html>. Additionally, remember yota modems and such that can pose as a network controller and have a built-in DHCP server? And don't forget that many smartphone apps transfer data to their servers without any encryption whatsoever, which also gives way to learning what kind of software is installed on your device and further usage of this information to the point of exploiting vulnerabilities.

Plus, the Russian government wants to pass the bill which would allow commercial use of information owned by the state <https://habr.com/ru/news/t/459654/>, though this is only legalizing what is already being practiced because the system is full of breaches and state databases can be easily bought.

Right now granting internet access to visitors equals a loss for many companies.

This sounds just plain ridiculous to me, and maybe to you as well. Well, perhaps it's a valid argument for the target audience who will want to buy this system.

Wi-Fi is a way to communicate with clients. Our project partner %shittyname% comments: "There are various ways of communicating with the client. For example, a user who connected to the guest Wi-Fi saw a banner/video, etc. – this is already communication. But we go further, creating complex integrated systems with a client, Wi-Fi can serve as a tool for recruiting into the loyalty program." Nothing unusual for now, the classic. The system's name does not matter, there are tens, or possibly hundreds of them, all identical in ways of functioning.

According to the law on granting public Wi-Fi networks the user must leave contact information and confirm their phone number upon connecting. Since the system is unified, during the first connection the user's MAC address and phone number are saved in a centralized database. Hence the next time the same user joins Wi-Fi in another cafeteria of the same network, they won't have to undergo authorization again.

Oh, clever guys – they develop the system by taking the Russian law into account. In other countries, they'd have to beg for the information such as phone number – and here it's taken for granted, and what's more important – you are legally obliged to disclose it. By the way, buying a SIM card under the current regime in Russia is already quite difficult without producing your passport first.

The %shittyname%'s idea – construction of what is called "Super Geo Communication" today in the world of internet marketing. Meaning that the Wi-Fi analytics system works like a trigger – catching the user's MAC address and phone number and sending them to the client's system, who (in case of having permission) creates communication with the user – sending them an SMS at the convenient time. I seem to have forgotten when exactly the SMS spam became illegal. What I do recall is how, before this bill was passed, I'd get tens of SMS messages per hour, all with the same ad, and how glaringly obvious was the money drain on the phone's balance – and this money could still be spent legally back then.

The further we go, the more interesting it becomes. Maxima Telecom company, which grants Wi-Fi access in the subway, owns the database of more than 19 000 000 of MAC addresses associated with the phone numbers. With this database, it's possible to expand the scope and function as a trigger.

Remember our movable access points in public transportation and stops? That's right – Maxima Telecom sells out the data left and right, like almost any company these days. I won't even be able to recall right away one that doesn't do this. By the way, the Wi-Fi expansion tender was tailored specifically for Maxima Telecom, only they satisfied the requirements (this is a common practice with tenders when a requirement met by a single company is written in).

Catches MAC addresses, which are then transferred to the Maxima Telecom system, where recognition is performed relating the MAC address to a specific phone number that belongs to it. An SMS message containing the text predefined by the client is sent on behalf of Wi-Fi.ru to a person who is passing by the store at a specific moment. The likelihood of "hooking" a passer-by is increased substantially. This process is called "audience expansion" and it's a new method that Maxima Telecom is now launching together with %shittyname%.

Hooking someone who's at home by the router in a nearby bus, taxi and carsharing – since even a mobile phone's package can be caught in monitoring mode on a distance that is more than if you wanted to establish a bi-directional internet via Wi-Fi channel. It even becomes possible to conduct targeting based on who's living with whom, for instance. It's common knowledge that all internet sites spy on the users, they stick cookie files on you, and %shittyname% catches your MAC address. So, %shittyname% discovered how to combine cookies with MACs – %shittyname% sends them to the big internet platforms, and the recognition is then performed on their side. Here are the possibilities it opens up:

- One can form a more definite picture of their target audience's interests and not only that.
- It becomes possible to assess the effectiveness of investments in advertising (contextual ads, banners, etc.) – for instance, conversion from internet ads. In the past, it was a pretty difficult task for a traditional offline business. Nowadays the %shittyname% Wi-Fi analytics can help – by aggregating MAC addresses in offline mode, for example, one can learn how many people from those who have seen the ad online visited an offline location, and much more.
- This data is also potentially useful for retargeting – it's possible to aggregate the audience who visited a specific store into a single segment and present this to the client. After this the client can, via precise targeting or "Super Geo", direct online communication towards these users, attract them, stimulate recurrence and thereby develop loyalty. This is a new product which many clients are starting to find useful.

One more product on the border between network equipment, software and Bluetooth technologies – a Bluetooth adapter with custom firmware is inserted in a device, and it becomes possible to work with iBeacon or Eddystone, which grants a decently precise indoor navigation. Bluetooth! Ble! iPhone! Smart clock, cars, fitness trackers! By the way, the latter is slowly moving away from ble to ant+, which will soon be tampered with I think. Ha-ha. https://github.com/hexway/apple_bleee (It's possible to extract even your phone number! On a side note, the first three digits are the operator's code, in Russia under current regime it's 8 plus only two digits to bruteforce, and considering that mobile operators' codes are not kept secret and are unique for each city, with geolocation a successful bruteforce will be executed only after a few tries of sending "silent SMS"). And by the way, the MAC address of Bluetooth and ble matches that one of Wi-Fi. I'd like to note from my own experience that it's pretty hard to track paired Bluetooth (not ble!) devices, and they probably didn't learn to do that yet. Some of the modern Bluetooth headsets interact with smartphones via ble as well, and since these sets have two separate earphones, two separate transceivers, by analyzing the level of the signal from different earphones we can detect the where user's sight is directed with 180 degrees accuracy. Here the fact about people seldom walking backward, but often walking forward comes into play – and we have the right direction. In McDonald's they started to install a system where employees deliver food right to the table – the client takes a numbered card, containing 2 separate ble beacons (just like 2 separate headphones), specifically 2 (seems like they use nrf51152 chips), in my opinion, they'd install only one, if given the opportunity.

Personnel tracking is retail's pivotal need.

In what way will the need be fulfilled, I wonder. Will they really, put a bracer on one's leg? Won't be surprised by that.

The server part represents a combination of data storage clusters, servers with preprocessing services, and individual tools which secure the targeted functionality of the solution's every part. This involves the usage of both virtual and physical servers in several independent data centers.

Yeeeah, right. "Independent data centers". And also, of course, "independent" communication channels to and from them.



>In 95% of %shittyname% projects in retail, trade centers and HoReCa objects (and this includes about 2500 devices in Moscow, St. Petersburg, million cities and other smaller ones) TP-Link equipment is in use. "TP-Link equipment, which is utilized by us, has characteristics allowing uninterrupted and stable work in all modes. Right now we work using TP-Link models EAP115, EAP110 of Auranet line. The outdoor solutions CPE210 v.1 and EAP110-Outdoor are currently in testing.

Won't you share the MAC ranges of these devices?

>A pic of TP-Link 9 dBi external Wi-Fi access point, Pharos line, CPE210

The scope you see for yourself, the coverage is going to extend far and deep, considering a high quantity of intermediate radio-links all of this will become ss7 on a single common vulnerability.

>The system has in its base a whole set of interesting algorithms, starting from "smart" object calibration and ending with data-mining algorithms. Even the seemingly trivial tasks like forming and processing of technical logs are not so simple. A lot of the developers' and hardware resources' energy is spent on fighting the noises and radio interferences. Around 60-70% of the signal is useless and won't take part in "useful" calculations

Can't call myself a fuckin' specialist, so can't imagine what is considered "useless" in a signal – can you perhaps?

>As %shittyname% put it: "Our work in Wi-Fi analytics resembles a taxidermist's (a person who makes sculptures out of an animal's body parts) work at times – when we are given a rabbit's carcass torn to shreds by a grenade. Using whatever pieces left, we try to make it "beautiful".

<http://nag.ru/articles/article/31835/desyataya-chast-polzovateley-gotova-platit-chtobyi-ne-videt-reklamu.html>

This article is a bit older, but it fills the gaps – looks like at the time of writing the monitoring system was not yet in use, and everything functions merely in access point mode, but maybe not, there was also another article between the first and this one, where monitoring mode wasn't mentioned directly, but judging from what I've read there I concluded that some things aren't possible to implement by means other than monitoring, and so it's there. But I've lost that article so you'll have to do with my bullshit, and actually, this is the reason the thread was created in the first place – because now we have proofs of the monitoring mode existing.

>In order to do that, you need to download or update “MT Cabinet” app, and this will allow you to access Wi-Fi “Kak Doma” (Translator’s note: “Like At Home”) service in Moscow subway, buses, trolleybuses, tramways, Central Suburban Passenger Company electric trains, Moscow Central Circle trains and in “Aeroexpresses”.

Lol, even the countryside is affected. Let’s take a look at the size and permissions of an app that is only supposed to authorize you in the network... Oh fuck, almost 12 megabytes, download it yourself and look at permissions it’s asking for, I’m sure you’ll find something more interesting than just permissions if you start to dig deeper. By the way, earlier the city Wi-Fi had a breach that allowed to track a person’s movement if you knew their phone number and they were careless enough to use the said Wi-Fi network.

>The new conditions, which became known at the beginning of June, are in effect since the 27th of May 2017. Earlier you had to pay less for getting no ads for a month on the subway – 129 rubles (for three months – 330 rubles, half a year – 576, a year without ads – 555 rubles). The fee was separate for “blocking” the ads on Moscow surface transportation.

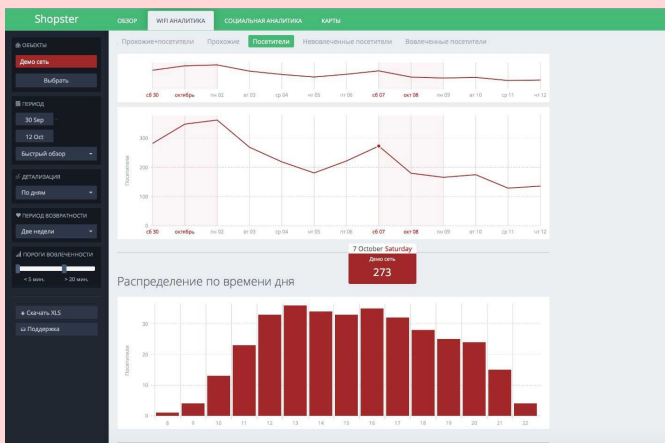
Pay for getting into the subway, pay for the Wi-Fi, pay for getting into surface transport, pay for Wi-Fi in it. I can’t understand why people pay for this Wi-Fi, it’s much better to simply pay to the mobile provider. Lately, some of my acquaintances say that on subway stations where you could previously have 3G and even LTE there’s nothing now. A coincidence? Ha.

>A Wi-Fi network operator faced a problem where passengers connecting to the wireless network “Maxima Telecom” started to search for ways to continue using the free internet access, while somehow mitigating the advertising manipulation of their brain. As a way of shielding themselves, the subscribers tried to utilize the ad-blocking apps.

And so we fucking banned them. Lately, I hear more and more critiques from wired and wireless internet users, where for example Rostelecom or Beeline plant the ads even in a paid account, sometimes even through https, which leaves a lot of food for thought.

>As has been mentioned earlier, Maxima Telecom company has the opportunity to accumulate and aggregate the information about the free Wi-Fi network users. After analyzing the million audience’s data (and in MT_FREE network there are 12 million registered devices) the network operator is ready to use it throughout the partnership with any companies, interested in targeted ad placement.

Your data will be sold left and right – you did tick off “I agree” yourself, after all. Oh yeah, now that we’ve installed the systems working in monitoring mode we can fuck the check-mark, it’s not needed anymore – you broadcast all the packages we’re interested in yourself.



>this is the way it works in 10 Russian airports, including Novosibirsk, Irkutsk, Kazan, and others. A piece of information appeared in the media regarding the operator’s intentions to expand their business both in Russia and abroad.

Even IMSI catchers in the airports are not enough for them now.

Remember how in the beginning I was talking about locating the access point with a unique name? There are things like Wigle databases and others, but why they are needed if there’s a national one, which updates almost in realtime and collects the data with almost any smartphone <http://telegra.ph/O-tom-kak-VKontakte-sobiraet-informaciyu-o-nas-chast-2-07-31>

Of course there’s more in there aside from Wi-Fi, but right now I’m concerned by it primarily, since your point gets written into database with all the consequences and without your approval, though I wouldn’t be surprised if it’s also scanning the net from the inside, like Nmap does on connection, which is the case with certain smartphones made by Huawei or Xiaomi I think, can’t remember which ones but I feel it’s both. And it’s also advisable not to keep wi-fi points open in here:

<https://nag.ru/news/newsline/104537/roskomnadzor-trebuets-s-provaydera-lichnyie-dannye-polzovatelya-chya-tochka-dostupa-wi-fi-okazalas-otkryitoy.html>.

In addition to what's been said above

<https://meduza.io/feature/2019/07/09/meriya-moskvy-sozdala-sistemu-slezhki-za-peredvizheniyami-zhiteley-v-ney-ispolzuyut-taksi-videokamery-i-karty-troyka>

Russian regime admitted the existence of a surveillance system that spies on people through taxi, video cameras, and transportation cards while combining these methods with mobile operators data,

<https://hightech.plus/2019/03/05/vlasti-otslezhivayut-peremesheniya-moskvichei-uzhe-neskolko-let> and by the way, the servicing of transportation cards is performed by a company owned by Alisher Usmanov,

https://www.rbc.ru/technology_and_media/26/10/2016/5bd1e5589a7947c4e9701675

more than that, he pays the subway in order to have the opportunity to serve the transportation cards – it like if you were paying for having the opportunity to work. He also owns the biggest mobile operator in Russia.

There's much more interesting to say on the topic of tracking in Russia, (for instance, about WI-FI routers from operators like Rostelecom with their gpon routers that combine granting the internet access with the usual city phone channel) but I tried to write primarily on tracking through wireless network channels, emitted from smartphones and other everyday devices, excluding things like face recognition, which have already been analyzed in depth. And you say Link NYC.



What are Garbage collectors?

by Fukako

When writing code, the programmer manipulates data. This data is stored in memory and, at some point, the programmer has to manage the memory they use. They have two things to do: allocate and deallocate-free-memory. This is done explicitly in languages such as C and C++.

However, many languages allow the programmer to allocate memory without freeing it explicitly. For instance, in Java, when one wants to create a new object, they have to use the keyword **new** but there is no keyword **delete** to free it. The object is deallocated without the programmer being aware of it.

Such a mechanism actually emulates an infinite amount of main memory in the computer. When the programmer allocates objects one after the other in Java and does not bother to delete them, they act like they do not need to. This is the actual purpose of a **garbage collector**, it enables the programmer to pretend they have access to an infinite amount of memory and thus do not need to manage it at all.

Obviously, one does not have infinite memory and this mechanism is not magic. If the programmer never frees memory himself, the GC will do it for them.

Mark & Sweep

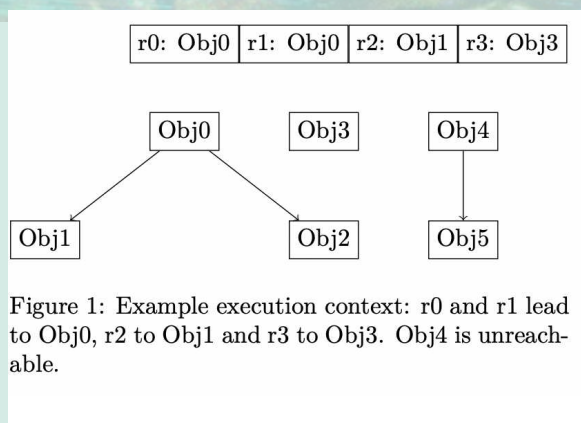
This algorithm is a conservative GC. A conservative GC frees block of memory only if there is no more reference to that block. It means that without a GC, no variable could access the memory block and it would result in a memory leak. The main advantage of a conservative GC is that it is absolutely impossible for an object to be garbage collected while it is still needed. However, it will hold every objects that it is not 100% sure it is allowed to dump.

As a conservative GC, **Mark & Sweep** tracks every references in the program. In order to do that, the compiler does not directly use primitives such as **malloc** but uses the functions provided by the GC. It also never assign references to other references directly, but lets the GC handle the references. The GC has to be aware of every references alive in the program at each point of the execution.

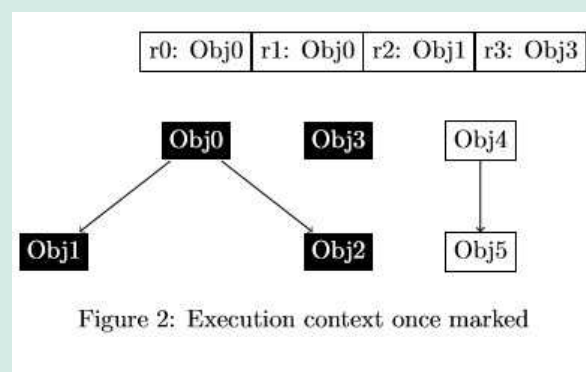
The references that the GC stores are local variables, static members of classes, global variables and other data that are accessible directly. These references are called **GC roots**, they are the starting point of the garbage collection process.

Since the GC knows each reference that lives currently in the program, it can use them to know which object are reachable. Remember: once an object becomes unreachable, it is considered dead! Figure 1 represents both the set of GC roots alive at the moment of the program execution as well as a visual representation of the objects allocated and how they reference each others. For instance, there is a variable **r0** that references the object **Obj0** which holds references to **Obj1** and **Obj2**.

Mark & Sweep is a two-step algorithm. First, every object that can be reached using a **GC root** is marked. Every object that can be reached from an object which is marked (an object's attributes for instance) is also marked. At the end of this process, literally every object that can be reach directly (via a reference) or indirectly (part of an object that can be reached) is marked.

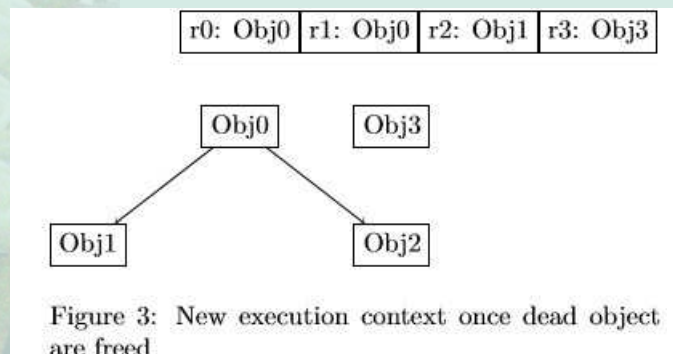


In figure 2, we can see that since **r0** and **r1** reference **Obj0**, this object is marked. **Obj1** and **Obj2** are marked too, since they are referenced by **Obj0**.



Then, every object that is not marked is freed. We know that every reachable object is marked, so the objects that are not marked are unreachable. This is why they should all be deleted.

In figure 3, we can see that **Obj4** and **Obj5** are missing. Since there were no variable that could reach these objects, they could not be marked and were freed.



Issues with Mark & Sweep

The **Mark & Sweep** is far from perfect. For example, it needs to traverse the whole object graph each time the GC performs a collection. In very large applications, this becomes a critical performance issue and makes this algorithm practically unusable.

Moreover, this algorithm cannot be used in multi-threaded applications because it would require both access to the GC-root set, and object pool to be in critical sections of code in order to ensure that allocations from multiple threads and garbage collection does not corrupt the whole data structures. This requirement further degrades the performance of the program.

Mark & Sweep alone also suffers from fragmentation. Indeed, when the objects are allocated, they are put into a memory pool at a free location. When the memory pool is full, a collection is performed to have place for new objects, but if the collection does not permit to gain enough space, the memory pool is enlarged---more space is allocated, typically using algorithms like **realloc**. Unfortunately, when objects are freed, they are not moved, leaving gaps in the memory pool. When there are unused gaps in the memory pool, it is faster filled again. Thus, triggering another collection and probably degrade the program's memory footprint.

Listing 1 is an example algorithm that allocates a new object using a given GC, assuming that **GcCollect** does not enlarge the memory pool internally if needed. The code is written in Pascal for extra sexiness.

Listing 1: Object allocation algorithm

```
function AllocateObject(gc: GarbageCollector; size: Integer): Address;
var
  addr: Address; { Abstract type representing an address. }
begin
  addr := GcFindFreeLocation(gc, size);

  if addr = 0 then { Consider 0 to be an invalid value. }
  begin
    GcCollect(gc);
    addr := GcFindFreeLocation(gc, size);

    if addr = 0 then
    begin
      GcEnlargeMemoryPool(gc);
      addr := GcFindFreeLocation(gc, size);
    end;
  end;

  AllocateObject := addr;
end;
```

Other Basic Algorithms

Mark & Copy is very similar to the **Mark & Sweep**. It is also a two-step algorithm that first marks the reachable objects and then performs the actual garbage collection. However, unlike the previous algorithm, this one does not explicitly remove the unreachable objects, it just moves the living objects into a different memory pool and considers the one currently in use invalid.

Mark & Copy uses a big memory pool that is twice as big as needed but uses only half of it. Once this half is full, it copies the reachable objects into the other half of the memory pool and uses it to allocate new objects. Once the second half is full, it copies back the living objects into the first half, etc...

When this algorithm copies the objects from one half of the memory pool to the other, it also compacts them, removing the risk of fragmentation that occurs using **Mark & Sweep**. Also, if the memory space actually used by objects is significantly less than the size of the memory pool, it is possible to shrink it, hence improve the memory footprint of the program.

Unfortunately, the copy is a costly operation. Thus, each collection is incredibly slow and it gets worse with the number of living objects.

Reference Counting is faster than **Mark & Sweep** and **Mark & Copy**. A reference-counting (RC) algorithm just keep a count of the living references to an object---whether it is a local variable, an object attribute, etc---and deletes an object as soon as its count reaches 0 (there is no more references to the object, so it is dead).

This algorithm has two major problems. First, the objects are not compacted into a memory pool and this leads to memory fragmentation. Unlike **Mark & Sweep**, it will not degrade the program's memory footprint, however it may lead to a suboptimal use of the cache and degrade performance of the program.

Second, it keeps track of all references to an object but does account for reachability. For instance, if two objects have a reference to each other but are not accessible from outside, they will not be freed because of these inner references. RC is thus prone to memory leaks if the programmer does not pay attention to this---which is unacceptable since the GC is supposed to abstract memory management, remember!

More Complex Algorithms

Marking algorithms are not efficient and scale poorly because the marking process needs to check every references in the program. In this section we will see how to reduce the price of such a costly operation using more complex algorithms.

Generational Garbage Collector

Empirical studies showed that objects die young. Indeed, about 80% of objects freshly created will die in the next million instructions. Consequently, it is very interesting to collect young objects frequently and let older objects alone for a longer time.

This is the reason why generational GCs were created. Such a GC uses different memory pools to handle the objects and each pool is managed with a different collection algorithm. The youngest objects go into the first pool, the largest one. When a collection occurs in this pool, the remaining objects are moved to the next pool and so on.

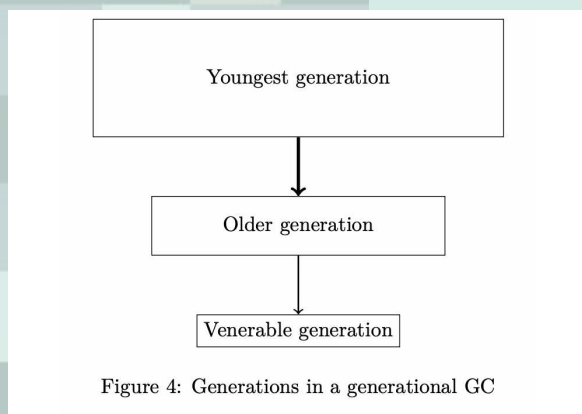


Figure 4: Generations in a generational GC

As illustrated in Figure 4, the first pool is the largest one, because most objects die young. The size of the subsequent pools decrease rapidly and only very old objects reach the last generations.

Since the first generation may be large, a very fast algorithm is needed to collect garbage. Fortunately, it does not need to be very precise since the next generation can collect the remaining dead objects. This the reason why a reference-counting algorithm may be a descent choice for the first generation of a generational GC.

The following generations do not have to be as fast as the first one, since only about 20% of objects are expected to reach them. More precise algorithms such as marking GCs are a better choice here. A collect on an older generation happens less frequently than a collect on a younger one. However, when it happens, it must collect from the current generation and from the younger generations. Which makes collection from older generations proportionally more costly.

Train Algorithm

In order to reduce the price of collection from older generations, it is possible to use a better algorithm called the **Train Algorithm**. The main advantage of this algorithm is that it never needs to run a complete marking of all the living objects. It enhances the performance of the "naive" generational garbage collector.

The **Train Algorithm** organizes the memory into many memory pools of same size. The pools are called cars and cars are organized into trains. There is no limit to number of cars nor trains. The trains, and cars in trains, are sorted lexicographically. Figure 5 illustrates the memory pool organization in trains.

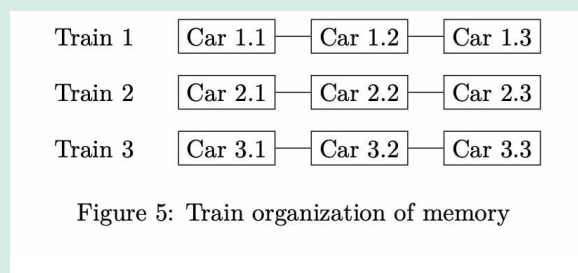


Figure 5: Train organization of memory

Every car holds a list of every reference to each object it contains, whether the references come from other cars or

trains, or are GC roots.

In order to perform a collection, the algorithm takes the first car of the first train. Every accessible object is moved to another car so that either the current car is empty or contains only unreachable objects. In both cases, the car can be safely removed.

Sometimes, the first train may have no references from other trains or **GC roots**. That means the train is only composed of garbage that is kept alive through circular references from one car to the other. In this case, the entire train can be deleted at once.

Listing train-collect is a minimalist and high-level implementation of the collection procedure of the **Train Algorithm**.

When the first train is deleted---because it is empty or composed of garbage---the second train becomes the first train and will suffer from the same treatment as the previous one. The goal of the train algorithm is to delete as many trains as possible, one after the other, to keep only old and living objects.

This algorithm does not need to collect everything at once and can only collect a few cars at a time.

Listing 2: Train collection algorithm

```
procedure GcCollect(gc: GarbageCollector);
var
  train1: GcTrain;
  car11: GcCar;
  car12: GcCar;
  ref: GcReference;
  obj: GcObject;
begin
  train1 := GcGetTrain(gc, 1);

  { No external reference in the first train. }
  if not GcTrainHasReferences(train1) then
    GcRemoveTrain(gc, train1);
  else
    begin
      car11 := GcTrainGetCar(train1, 1);
      car12 := GcTrainGetCar(train1, 2);

      { Move each living object to another car. }
      for ref in GcCarGetReferences(car11) do
        begin
          obj := GcRefGetObject(ref);
          GcTrainMoveObjectIntoCar(gc, obj, car12);
        end;

      { The car is empty or full of garbage. }
      GcTrainRemoveCar(train1, car11);
    end;
end;
```

Concurrent and Parallel Garbage Collectors

Nowadays, applications use multiple threads to scale better and provide better performance. While a great thing, it complicates further the implementation of garbage collectors. Indeed, unlike with mono-threaded applications, stopping the world to run a complete collection on a multi-threaded application degrades even further the performance, because it does not prevent only one thread from executing, but all threads!

Hence, enabling the GC to perform marking or collection tasks while the program is still running is critical. Moreover, if the GC is able to collect

garbage using itself multiple threads, it improves even more the collection time and thus reduces the overhead of collection on the program's execution time.

Conclusion

Garbage collectors are interesting systems and their functioning is rich and varied. Basic, yet precise, GCs can be implemented quite easily. Moreover more efficient GCs can also be implemented with incrementally increased difficulty. The hobbyist is able to discover this field at a reasonable pace.

Nonetheless, serious solutions require much more engineering to scale correctly with contemporary needs in terms of performance and memory space. Multi-threaded---and multi-processor---applications use a lot more memory than mono-threaded ones---it can amount to several GCs. As a consequence they require an even more aggressive garbage collector. Some state-of-the-art garbage collectors use machine learning to be able to delete a living object that will nonetheless never be used again.

Also some GCs have been developed to be usable in real-time environments, allowing programmers to write software for critical real-time applications using more secure and comfortable languages than C.



