

ARKAKAPI

www.arkakapimag.com

BIMONTHLY CYBER SECURITY MAGAZINE

Chain of Indepence:
Blockchain

Introduction to
Cryptology

Thoughts on
Meltdown and
Spectre

Vulnerabilities
of Bluetooth

01
02
03
04

The Art of Data Hiding: Steganography

Anonymizing Internet from the Router with
OpenWrt and Tor

Set up Your VPN with "Kendi Bağlantım"
(My Connection)

Acting on the Sly: Overcome Obstacles with
DNS Tunneling



www.arkakapimag.com

Executive:

YEAR: 1 – ISSUE: 1 – ISSN : 2645-906X

Editor in Chief : Ziyahan Albeniz - ziyahan@arkakapimag.com
Editorial Operations Manager : Umran Yildirimkaya - umran@arkakapimag.com
Chief Business Officer : Oguz Aydinylmaz - oguz@arkakapimag.com
Publishing Coordinator : Sahin Solmaz - sahin@arkakapimag.com
Legal Adviser : Mehmet Pehlivan - mehmet@arkakapimag.com

Translators:

- Serdar Savaş
- Emre İyidođan
- Enes Özen
- Zekvan Arslan
- d1scharg3d

Web:

- www.arkakapimag.com
- twitter.com/arkakapimag
- instagram.com/arkakapimag
- facebook.com/arkakapimag

Hello World!

Arka Kapı is a hackers' magazine that began its journey in Turkey. 'Arka Kapı' stands for back door. We're publishing our fourth issue in Turkey as we introduce the first English issue to you.

The internet today hosts the core components of personal freedom: freedom of speech, freedom of trade, and freedom of self-advocacy. That is, as long as you are aware of it and protect it.

An average internet user can make use of this freedom through various free tools on the internet. There are multiple running headlines to protect this issue.

A united stand against mass surveillance and censorship, guarding the global net neutrality.

In Turkey, we became a hacker magazine which passed on tools and techniques to the readers but we also stood for the benefits of the information revolution that strengthened human rights.

We're hoping to do the same and more in our global issues. As an independent magazine, we need your support to reach our goals!

...

The spotlight for our first issue is "Anonymity." The mass surveillance we're going through seems very much like a chapter from the Orwellian dystopia except here everything we do is watched through our smartphones and wearable technology.

Most of you might think that you've got nothing to hide. However, protecting our privacy from Big Brother has a larger scope than the individual scale, as we saw in the very recent event that struck millions through Cambridge Analytica.

The only antidote against this mass surveillance is the correct and conscious implication of anonymity.

As Snowden says, 'Let's take back the internet!'

Special thanks to Netsparker Ltd. for bringing our first issue alive.

netsparker

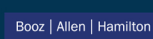
Web Application Security Scanner

Use Netsparker to Identify Exploitable Vulnerabilities and Other Security Flaws in Your Websites, Web Applications & Web Services Before Hackers Do.

Netsparker scanners employ the unique, dead accurate & fast **Proof-Based Vulnerability Scanning Technology** that automatically verifies the identified vulnerabilities with a proof of exploit, so you do not have to manually verify them.



Trusted by



WHAT'S IN THIS ISSUE

Possession is Nine-Tenths the Law: User Agreements

How to be a Shareholder of Google: Double your Income with Google AdSense

Introduction to Cryptology

Spotlight: Anonymity

The art of data hiding, ways to hide your IP and surf anonymously, and the configuration of TOR - the basic tool that we all need nowadays.

You'll learn about the technical details on setting up TOR through your router. In case TOR is blocked in your country, you'll learn how to set up a VPN to bypass that.

You'll also find out about DNS tunnelling to pass through obstacles.

The Art of Data Hiding: Steganography

Anonymizing Internet from the Router with OpenWrt and Tor

Set up your VPN with "Kendi Bağlantım" (My Connection)

Acting on the Sly: Overcome Obstacles with DNS Tunneling

Insider: Amanda Rousseau

Chain of Independence: Blockchain

Why you shouldn't store sensitive data in javascript files

Thoughts on Meltdown and Spectre Vulnerabilities

Vulnerabilities of Bluetooth: Past and Future

POSSESSION IS NINE-TENTHS THE LAW: USER AGREEMENTS

As individual users, we trust the free services provided to us on a daily basis (such as the search engines, email applications, and various social media) so much that we allow them to cover our lives almost entirely. Do you think on the basis of this trust lies the thought, “How nice of them to give us such services for free! Hurray world peace!” I don’t think so. Although it seems like we don’t pay anything in cash to use these services, we pay with our privacy and personal data. Yes, it’s correct, these companies store our data as a payment. What makes things more interesting is that we allow them to do this in the first place. How? The answer lies in the User Agreements.

The user agreement usually has an average of 47 pages and 3,294 words and promises us that we’ll be using a free service but doesn’t entirely tell that the company will take our privacy and personal data in return.

Next, next, next, next, I agree...

We open the doors to our privacy and personal data at the very start of the app we want to use when we’re signing up. We simply do that by putting a tick on the “I’ve read and agreed to the terms of services.” Our journey to the dark side begins here.

These terms we agree within a matter of seconds might become a massive threat for us. Here are two examples of the threats posed by the bulk of the iceberg:

A company in London experimented signing up to a public WiFi hotspot. Within the terms and conditions was that the WiFi was free only if “the recipient agreed to assign their first born child to us for the duration of eternity. If the recipient does not have a child, their favorite pet will be taken away for the duration of eternity.” When the experiment ended, reports showed that everyone signed away, proving that no one read the terms.

Another company further proved that terms and conditions aren't read by adding this term to the terms to a program they advertised: "Those who read the mentioned term and contacts us will be given a monetary prize." After the program was downloaded 3 thousand times and 4 months have passed, one person contacted the company through the mentioned email. The company awarded this person \$3000. It's important to note that reading these terms will not always award you monetary prizes.

In this article, we're going to evaluate the terms that no one reads, or gives up at some point while reading, and summarize what exactly the terms force us to agree on as much as possible.

The services we're going to examine are those provided by WhatsApp, Google, YouTube, Twitter, Facebook, Microsoft, and Apple. A legal notice we should be making is that none of the information given below is considered to be legal advice and unfortunately they might not be the most recent versions. Reading the information given here does not substitute reading the original terms and conditions.

WhatsApp

WhatsApp doesn't store your messages but stores the metadata regarding your message. As far as we're concerned, metadata is more of a wordplay, defined as data about data. Instead of what exactly the data is, it stores where the contents come from and where it goes and the timestamps. It might seem harmless, but as I mentioned above, it's just wordplay. For example, the famous hacker Kevin Mitnick found out about the planned raid to his house through the metadata of the FBI agents' phones. In his article, Kurt Opsahl sampled out what companies and governments can acquire from the metadata and proved how vital metadata could be: "They know you rang a phone sex service at 2:24 am and spoke for 18 minutes. But they don't know what you talked about. They know you called the suicide prevention hotline from the Golden Gate Bridge. But the topic of the call remains a secret. They know you spoke with an HIV testing service, then your doctor, then your health insurance company in the same hour. But they don't know what was discussed."



The files (photos, sound records, documents, location) sent over WhatsApp are stored in the servers for a while and deleted afterward. Unlike the end-to-end encrypted messages that aren't stored and sent directly to the recipient, files are stored and how long they're stored is not specified. This isn't a user-friendly term.

Your shared status is visible by everyone who knows your phone number. "Shared status" consists of your profile picture, status, last seen information, and is visible by whoever has your phone number (if you haven't blocked them) unless you opted out. Considering how privacy should be the default option and the exception should be publicity, this term is also not user-friendly. Users own rights to the aforementioned shared status, but WhatsApp has a broad license over these.

WhatsApp shares the data stored about the users with Facebook. WhatsApp's Privacy Policy indicates that Facebook bought WhatsApp in 2014 and that it holds the rights to share your data with Facebook and other companies it owns. European Commission had fined WhatsApp \$122 million for syncing WhatsApp numbers with Facebook accounts.

WhatsApp can share your data with third parties, stripping your data of its personal features. WhatsApp can also collect and share your data if the law requires to do so. However, that would be a somewhat optimistic view on this use, and we can define this ambiguity as the bulk of the iceberg.

Changes in the privacy policy will not be based on the users' opinions, and the user will not be directly informed of the changes. WhatsApp states that it'll revise the Privacy Policy but does not in any way hold the commitment to take your opinion or inform you of the changes. Continuing to use the services after these changes would be a form of agreement to the new terms. This article states how a seemingly user-friendly term can end up being harmful.

It's not Big Brother; It's Google!

Google applies to almost every civilized part of humans' daily lives by the services it provides like AdSense, Analytics, Search, Blogger, Translate, Drive, Gmail, Google+, Hangouts, Maps, and Youtube. Naturally, it collects various data from us. Google provides us with free services in various ways by building their million-dollar worth of servers. This sounds odd because their servers aren't for sale in stores. So how does Google form these servers if



they're not for sale? Google needs to have large servers to store the data and privacy we agree to share, making it world's 4th largest server provider.

Google collects most of your data including your email and analyzes them. Since the Google services are rather numerous, the data it collects is equally large: Search prompts, location services, map searches, videos we watch, ads we interact with, websites we visit that uses Google services, and other data using various Google services. Considering all this plus Gmail, the data would be highly significant.

With all this, Google has the automation system that analyzes the entire data including our email. When we get an email about a holiday, Google gives us ads about plane tickets around the email. This is considered a "skill" of Google!

Google doesn't share our data with ad companies. Although Google doesn't share data with ad companies, Google is an ad company with all the services it provides.

Google watches our activity in other websites and collects data on our access to these websites. According to Google Privacy Policy, when we visit a website that uses ad products like AdSense, social products like +1 button, or analyze products like Google Analytics our browser automatically sends specific data to Google. Among this data is the web address of the website and your IP address. By doing this, Google analyzes your browsing history and intends to send you personalized ads! It's important to point out that there are almost no websites that don't use Google ads.

Google makes profiles about us by compiling our data collected to "use with ads." If we sign in to Google, the data collected will be assigned to our Google account. If we don't sign in, Google can "connect personal data in certain services." In both instances, a profile is formed with the data collected. Google states that the reason it does this is to "provide personalized services by filtering out unwanted email, spotting malware, and bring personalized ads and search results." Should we trust Google with this?

Google does not share our data with third-party...Yes, it does. Google shares our data with third-party to process our data externally and for legal reasons.

Even if we delete our account and its contents, Google may continue to store our data. You're like a troublesome ex-partner Google! Google allows us to delete our browsing history, our blog, YouTube channel, Google+ profile and the Google account that's related to all these. However, even if we delete our data and close down our accounts, Google "may not remove the copies of the data you've deleted and may keep them in backup servers." Under this term

is another article which states that the license we give to Google over our data continues to be applied even if we delete them. This way it's an irreversible license.

Google doesn't inform us of the changes done in the Terms of Service, but the changes are applied 14 days after the announcement. Google can revise the Terms of Service at certain intervals. Changes that'll be made will be presented on the mentioned page, but we don't see an article about letting users know about the changes. Unless the changes "go against the new uses or legal reasons," they will be active 14 days after being published.

Working under Google, YouTube does not have its privacy policy. Just like in other Google products, the Google Privacy Policy applies to YouTube as well. However, YouTube differs from Google regarding User Agreements on user contents' copyright. As Google states, general and unique situations might affect the User Agreements. In this case, YouTube is considered a unique situation.

Even though we own the rights on the videos we share on YouTube, we still give away too much license to the service. Google does not own any rights on the contents we share on YouTube. However, the licensing we sign off has the right to sub-license, causing a chain-reaction of licenses. This causes the most substantial copyright issue of the past years. The licenses of the text comments we make are irreversible and consistent. This raises the question of who owns the lyrics written in the comments section...

YouTube can delete your content whenever it wants without giving reasoning and without informing you. Under YouTube's initiative, if the content goes against User Agreement or is a copyright infringement, it can be deleted without notification and reasoning. Recently the music videos that had millions of views were deleted for "groundless" reports, leaving the singer aggrieved.

Twitter

Twitter follows us on other websites too and collects data on our interactions on these websites. According to Twitter Privacy Policy, under the widget data title, visiting any website that has Twitter components, Twitter collects data through cookies. This way Twitter aims to present personalized ads to us.

Our interactions with Twitter services are collected and saved as daily data and are deleted or depersonalized in



18 months or less. Any interactions we make with Twitter services cause data like the source website, visited pages, location, searched words and cookie data, IP address, username and email to be synchronized and collectively saved in servers. These daily data is deleted or depersonalized in 18 months or less.

Twitter allows third parties to use cookies. Twitter doesn't share our data with third parties entirely, but this statement has exceptions. As a rule, Twitter doesn't share our data with third-party but through service providers or if the law requires, shares our data with third-party. Furthermore, Twitter can share our data with tied companies

Twitter can share depersonalized or public data with third-party. Twitter can share our data that's not personalized or data considered not private, visible to the public with ad companies. For example, if your profile isn't locked, your tweets, people you follow and your followers, and the links you click are shared with ad companies. Your name or communication information is not shared while doing so.

You own the rights on the things you share however the license you give Twitter over these is extensive. Twitter doesn't hold any rights on the content you share, but the license you give to Twitter has sub-licensing which is rather comprehensive. This license remains even if you delete your account.

Facebook

Facebook collects all sorts of data that we and others input about us. This includes signing up, forming or sharing content, messaging with others, the location of the photos we upload, the date of a folder we form, and the content we view. Facebook also collects the input given by others about us.

Facebook also saves the operation system, hardware version, settings, battery and signal, location (through GPS and WiFi signals), carrier, ISP, browser, and mobile number of the device we're using to access Facebook.



Facebook can decide to share our data with third-party with the reasons it spots. Fortunately, Facebook is a sensible company that hasn't been fined millions of dollars for personal data processing!

Facebook shares our data with applications that use its services. For instance, if we're playing a game with our friends on Facebook or if we're using the Facebook comment or share buttons on a website, the game developer or the webpage can collect information about the activity we have on the game or collect information on our activity on the webpage through comments or shares. When I told this article to my dad who plays games on Facebook, he stated that his game style was informed to our neighbor by Facebook. I'm quite intrigued about the relationship my neighbor has with Facebook!

When you download or use the third party services, your username or user profile, age range, country/language, your friends list, and your profile visible by everyone. The data collected by these applications, websites, or integrated services have their own terms and conditions.

Facebook keeps its discretion when it comes to the true deletion of the contents we remove. Yet we know that it doesn't delete any! When Facebook switched to the 'timeline' system, previously deleted content was visible on the profiles, meaning they were never truly deleted from the database.

Microsoft

Microsoft collects and stores many of your data. Your name and contact info, identity, demographics, likes and preferences, payment info, contacts and relationships, location data, content you input, chat you have with customer support, and the surveillance recordings you have the moment you enter a Microsoft store, are all examples of the data stored.



Microsoft collect your data from third-party as well. It purchases demographic data from other companies and adds to its collection. Using services of other companies that help discover your location based on your IP, Microsoft aims to personalize your geographical location and thus shares your information with such companies.

Microsoft doesn't state whether it forms a profile out of all the data it collects from each individual or not. Microsoft's Privacy Policy doesn't indicate whether the data collected while using the services are collectively stored and used as an advertising profile. The lack of such article can mean that Microsoft uses data from other services to bring together a profile. How cruel of you Microsoft!

Microsoft states that it uses the personal data collected to serve ads to users. Microsoft uses the collected data over its own services or third-party services to help choose ads presented

by Microsoft. The chosen ads may depend on current location, search prompt, and content viewed. The “ads based on interest” mentioned in this article are based on interests, demographic data collected over time, search prompts, preferences, usage data, and location data.

Microsoft can share personal data with government institutions without the ruling of the court. Microsoft Privacy Policy states that it can share all personal data including email contents, private folders, private contact information and files to abide the law or as a response to the legal process requested from any government institution.

Microsoft shares personal data with third-party ad companies. In some cases, Microsoft shares reports on the data it collects of users on the websites or ads with the ad-giving companies. Plus, it can directly share this data with service providers and allow them to work with Microsoft on choosing ads and services. The credibility of the partners of a company which shares its users' data with ad companies is highly concerning.

Microsoft allows ad companies that provide services to users, to collect data about you using internet pointers. Ad companies can insert internet pointers in its ads to generate and read self-identifying data.

Microsoft states that instead of selling software, it gives away a license, and this has negative aspects for the user. For instance, you cannot transfer software, software licenses, service access or the rights to use a service.

Apple

Apple collects specific personal data. When you make an Apple ID, purchase an item, download a software update, sign up to a course in an Apple Store, contact Apple or join an online survey, Apple may collect your name, contact info, phone number, email address, preferred communication, and credit card information. Apple states that it collects these personal data to use for advertising purposes.



Apple can give away the collected personal data to or acquire personal data from international establishments. However, Apple’s privacy policy doesn’t clear out who these “international establishments” are, and a new topic follows right after the statement.

Apple can expose personal data without the necessity of a ruling from the court or a formal request from a government institution. Apple privacy policy states that it can expose personal data if it finds fit according to the execution of the law or problems regarding the public or to protect its users and processes and abide by its terms and conditions.

Apple collects and shares your location information by default. Apple, its partners and license owners can collect, use and share real-time geographic location and other location information of your computer or device to provide location services to Apple devices. If the user is uncomfortable about this, the location services are stored anonymously, and Apple, its partners, and license owners will use the data to provide and improve location services.

Conclusion

The inspections and consumer protection laws that stop articles that are against customer benefits to be active in the non-negotiable lengthy user agreements of banks, sadly do not exist for user agreements of services. However public initiatives and governments are looking for solutions for users privacy. There are attempts within this goal to pass laws and begin inspections on services. In the recent years, Turkey has proposed ways to battle for the privacy of users. One of these is Article 6698 Protection of Personal Data and the regulations passed under this law, making it a huge step to protect personal data. We're hoping that regulatory institutions will end the 'governing' power of the services on its privacy-seeking users.

How to be a Shareholder of Google: DOUBLE YOUR INCOME WITH GOOGLE ADSENSE

In this article, I'm going to be talking about cookies which are the main structural elements of websites, and the significant vulnerability they pose, as well as the horrific results these vulnerabilities may bring. As we all know, cookies are small files that use browsers to collect our data such as the amount we spend on the website, remembering our passwords or settings, how we got to the website, how often we visit the website and many others.

I can almost hear you ask what the results of these vulnerabilities may be. Without further ado, I'd like to share one of these vulnerabilities which I've tested and shocked by the results. Assuming you've heard of Google AdSense and Adwords - the biggest in web advertising. Using this vulnerability the same AdSense account that made me only \$168 over a 3-year span gave me a whopping \$1200 a month. Within the 4 months of testing, I made approximately \$8800, and Uncle Google paid me nearly \$5000 of that. The website is the same, ads are the same, only 30-40 visitors a day, just using a computer and a simple bot written in Visual Basic Script generating virtual visits. How?

Once I consulted with a few of my expert friends and came to the same conclusion that this is serious, I decided to reach out to Uncle Google. Since this exploitation is against my morals, I shut down the bot as well.

Presenting all the details, I talked to the Google Security Team for days. They ended the conversation by saying "We took a look at your report and we can confirm that this is not a security breach, yet we'd be pleased to receive more details on your discoveries."

Honestly, I wasn't surprised at all because I didn't report this as a software breach and knew very well that they wouldn't have a short-term solution to this situation.

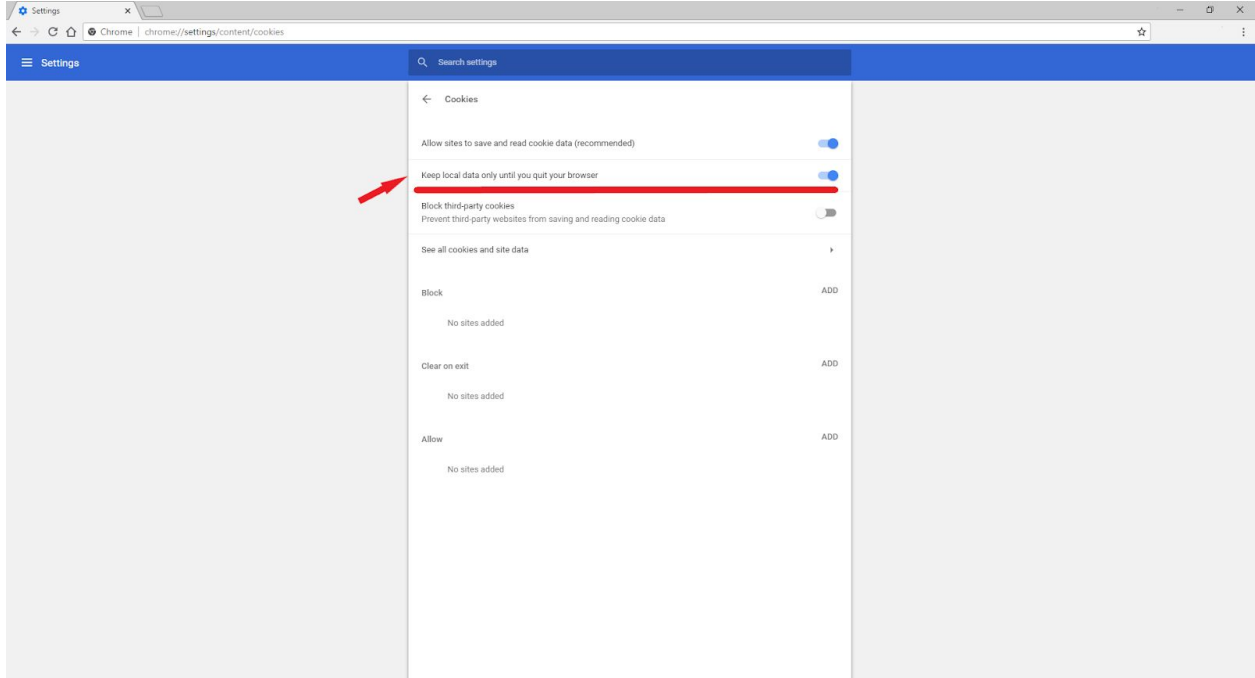
Actually, there are some easy radical solutions, like unclearable cookies and assigning non-changeable IDs to each browser, but none of these can be applied without breaking the User Agreement.

These wouldn't work because they'd be abusing rights and no one would want to be monitored without consent. I restarted the bot after 3 years, and the same situation was ongoing. I had tried to explain that this problem wasn't unique to Google but instead concerned YouTube, Yandex, Bing, Facebook and all other websites including small ones. Why should Google care? I make \$1000 using AdSense ads whereas Google makes 10,000-20,000 out of this. The losers are the ad givers.

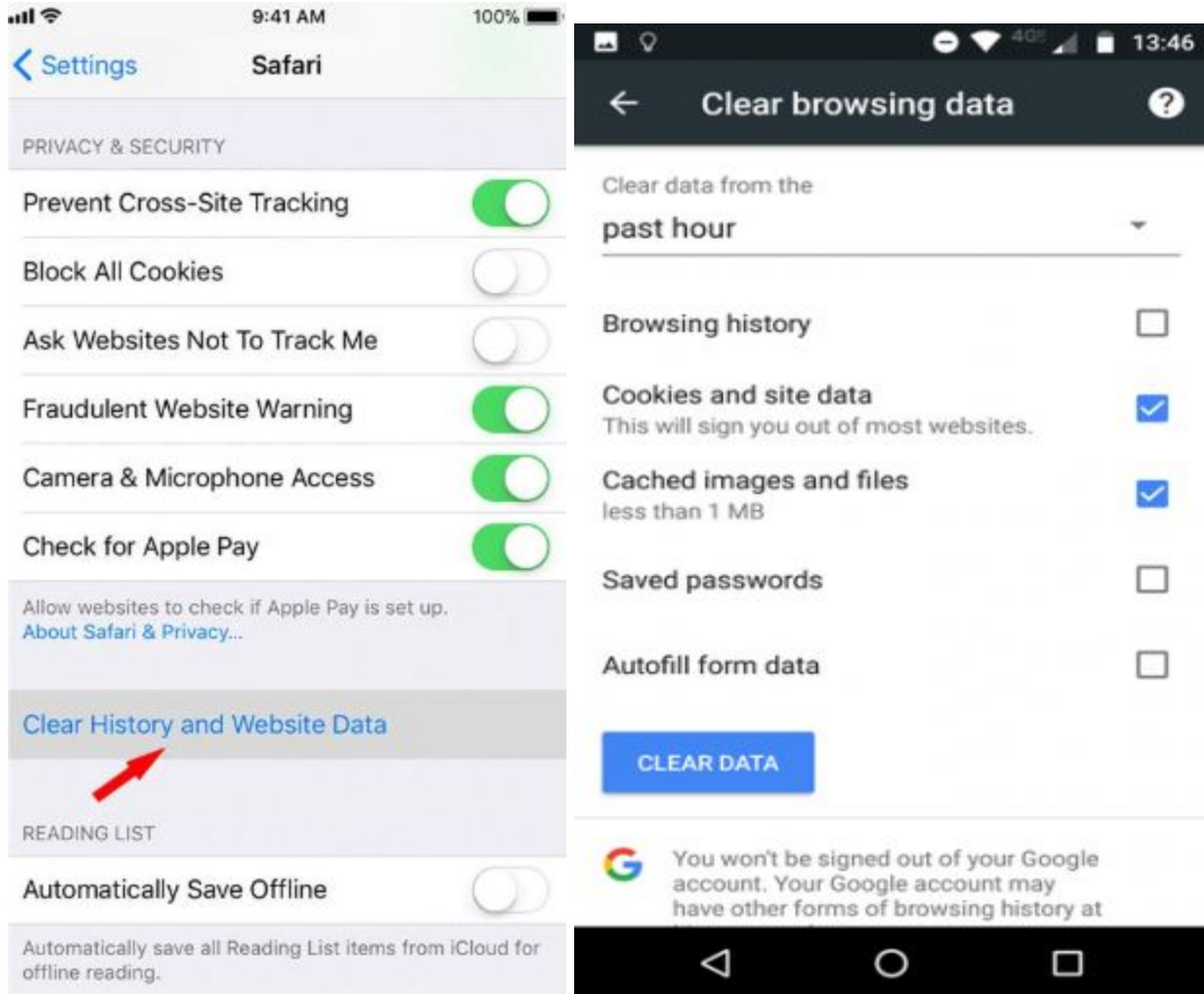
Let's get to the main story - how to generate these fake visits. First I want to make things more apparent by describing the manual bot-less way of doing this. As we all know, all browsers have privacy options. Although they portray small differences basically, they're all the same. I'll be giving the settings for the most preferred one, Google Chrome.

Browser Settings

Go to settings and scroll down and click on "Advanced." Then under the "Privacy and Security" list, go to "Content Settings." We're going to make configurations on how the cookies are used in the browser. All we have to do here is to activate the "Keep local data only until you quit your browser." Doing this will delete the cookies the moment we exit the browser. If we haven't changed the default settings, "Allow sites to save and read cookie data (recommended)" option will be on and "Block third-party cookies" will be off. To make sure, we go to Location settings in the previous screen and make sure "Ask before accessing (recommended)" option is on. If the location services are on, we must ensure to keep this option on as well. These are the only settings we'll be changing. Make sure to configure these on the other browsers on your PC.



As you know, Google won't bite it if the traffic only comes through desktop computers so it'll ignore most visits. Therefore we have to generate visits on mobile and tablet to enlarge the network. The settings I mentioned above are available on the mobile browsers as well but with one key difference. Instead of the auto-deletion of cookies once we quit the browser, we'll go to the settings of our mobile device and delete the cookies manually. For tablets, we're going to access the browser settings and find the Privacy list and click "Clear browser data" manually.



That's all for the browser settings. Next, we need to change our IP. It's as easy as deleting cookies. Once you unplug and plug the router that gets a dynamic IP, you'll acquire a new IP. On mobile, you switch to flight mode, wait a bit, and exit flight mode. That's it.

So now you can visit your website which has AdSense ads and make money! I know you're thinking it can't be THAT easy but try it for yourself and see. Visit your website directly, through a search engine, or a backlink and check out a page or two, then click on the ad and go around the website you're directed to, and exit the browser. Turn off your router and then switch it back on. If your device is mobile, switch to flight mode and clear the cookies. Then exit the flight mode and start the cycle all over again.

Important to Note

1. First and foremost, don't ever sign in to your Google account with the IP you use to generate fake visits. You'll get banned instantly.
2. Make sure to keep the ad showing number and the clicking number at a reasonable rate. When you're increasing the clicking, make sure to increase the visitor number at the same rate. For example, I increased the 30-40 hit number to 1000-1500 using natural hit programs. This way I inflated the ad showing rate. So the 100 show number became 2000-3000 suddenly. This way the CTR (Click Through Rate) didn't alert Uncle Google and staying at the 1-2% increased my website's clicking by 20-30 times.
3. Work with the site reports Google Analytics provides you and keep in accord with them. For example, if the traffic is 60% mobile, generate %60 of your fake visits through mobile devices. If 40% of the traffic is coming through Chrome, use Chrome with your fake visits. If there's region restriction, make a website from that region and generate traffic from that region. So if you're generating fake visits only from your network in Istanbul, then publish your ads on a website related to Istanbul. This way Uncle Google won't be alerted.
4. If your website doesn't have enough traffic through search engines, you can generate high traffic through backlinking another website with the help of JavaScript or directly. Uncle Google won't ask you why 90% of your traffic is coming from another website because you might have published an ad or someone may have mentioned you in the news.
5. Make sure to act normal when you visit your website. Take a look around and then click the ad and go around on that website. Don't click the ad in the footer more than the one in the header. Keep an eye on the rates.
6. Instead of clearing out the cookies, you might want to collect them somewhere and choose a random cookie to revisit the website. A user from a month ago should be able to revisit and click the ads for Uncle Google to stay quiet.
7. Make sure to choose the content appropriately and adjust the traffic and clicking accordingly. A personal blog won't have the same traffic as a news website.

8. Don't forget that you can use the same method to deceive other private ads on your website, increase the view numbers on your YouTube video, access and login to websites that banned you, and fulfill most of the criteria that SEO requires. You can also use this method to click the AdWords of rival websites to exploit their credits.

Now instead of doing this manually, generate a bot doing all this and you're officially a 10% shareholder of Google.

INTRODUCTION TO CRYPTOLOGY

Cryptocurrencies, specifically Bitcoin, that everyone's involved in, from regular people to high-officials, are constructed on this mysterious cryptology. Writing about this topic sounded appealing to me. Thus I decided to write a series about cryptology from past to present with the hint of popular science.

Introduction

Cryptology or cryptography (from Ancient Greek: κρυπτός, *kryptós*, "hidden, secret"; and γραφειν (*graphein*), "to write", or -λογία (*-logia*, "study")) is the practice of encrypting and hiding private data, securely transporting private data to the right recipient or in sum analyzing this whole process and identifying open spots. In many definitions, it's said that cryptology works with mathematical theories. The definition is understandable when we consider that it emerged after the Second World War with the development of public key mathematical algorithms, but it's somewhat incomplete. It ignores the methods that have used markings, shapes, or displacement to stay hidden for thousands of years.

1978 can be regarded as a milestone for cryptology. Until 1977-78 everything, including the key and the encryption, was kept secret because the key used to encrypt text with was also used in the decryption of the text. Until then, secrecy was the most important and the only vital element for cryptology, and that was about to change. Three researchers named Ron Rivest, Adi Shamir, and Leonard Adleman have published the first public key cryptography algorithm, later to be referred to as "RSA" by their initials. The secret key and secret cryptography, which the states have developed by devoting large budgets and carefully hid as their most intimate secrets, were no longer meaningful and were immediately abandoned. Things have changed drastically. The key and the encryption method were given out openly, down to the last detail.

RSA's mathematics had elementary school level simplicity. However, those who wanted to break it faced the frightening side. It took hundreds of years even for supercomputers to divide each encrypted letter or number into the multipliers of the correct number. In the future issues, we will delve into the details of this algorithm.

Cryptographic Methods

Cryptographic methods live until their algorithm is proved insecure or a better algorithm is found. They're then abandoned and never used again. Cryptographic methods are perfect during design. In practice, the weakest points of the methods are careless users who do not have sufficient knowledge about system operation and do not implement the directives completely. In other words, the human factor is the weakest link in the chain of cryptology.

Methods should be thoroughly tested before release, and user-induced faults should be tested as much as possible. Compulsory measures should be developed for the user to behave in specific patterns.

People and Cryptography

It is an unquestionable fact that we, as individuals or institutions, are a community that does not show the necessary importance and value of knowledge and data. Throughout the years I have worked in the IT sector it was a rare situation to encounter a user who conceptualized the importance of passwords.

When I tried to tell how simple choices like 123456, date of birth, the name of one's child are, I faced the surprising answer that users chose it for its simplicity.

It is also helpful to the development of information security culture to define the origin of the word "cipher" here.

The word "şifre" (Cipher in Turkish) is derived from the French verb 'chiffre,' meaning "to digitize, to code a text in the motive of obscuring." This verb is also derived from the French word 'chiffre,' meaning "number." The French word 'chiffre' is derived from the Italian word 'ciffra,' which means "zero or Arabic numbers." The Italian word 'ciffra' is derived from the word şifr, "zero" in Arabic. Until 1000 AD, Europeans used the Roman and Ancient Greek counting systems which were error-prone and consuming too much time. These people were stunned to see that the Andalusian Muslims could easily and quickly make calculations with a numerical system based on the Indian Base 10 system with decimals and zeros. There were rumors among the people that zero (şifr in Arabic) is a magical, mysterious number. Later they had to derive new words such as "null" and "zero," to clear out this confusion.

Cryptology Studies in Turkey

Before preparing the article, I searched for Turkish resources on cryptology on the Internet. I only found a handful of books of local authors, translations, bits of theses and lecture notes. Also, access to www.wikipedia.org, where the richest Turkish sources are found, is still blocked by BTK (Information Technologies and Communications Agency) due to unknown reason(s).

The first cryptology studies in Turkey began with a small workgroup at the Middle East Technical University in 1972. Along with the increased security needs of the public and the army, cryptology studies progressed. In 1995, National Electronics And Cryptology Research Institute (UEKAE in short) was established within the structure of TÜBİTAK (Scientific and Technological Research Council of Turkey) to bring a corporate status to this workgroup. In addition to various military projects, UEKAE also develops public security products such as AKIS electronic certificates (e-signature). However, there are still no crypto-analyst training departments in universities. While TUBITAK is organizing summer schools for college graduates; some universities, such as ITU (Istanbul Technical University), METU and Ankara University, are organizing certificate programs.



In recent years, the public has undergone a significant transformation under the e-government project. Unfortunately, I witness the citizens sharing their e-government passwords with other, unauthorized, people.

There is no secrecy of a state where the citizens do not care about their privacy. Because when those citizens become executive authorities of the state, they will carry out the state affairs with the same carelessness. We witnessed a group of high officials leaking crypto-phone conversations belonging to the state. However, we do not know whether these recordings are obtained through drop-ins or from decrypting the crypto. Even though the leaders claimed that the content of these speeches was fake, they didn't hold back from initiating a cryptographic phone development project.

In this issue, we will examine the ancient methods used to provide secrecy from state secrets to military strategies, from personal information to commercials. We will examine some of the simplest and some of the most complex systems and methods used, since the invention of writing, including the Enigma and the RSA.

Language, Writing, and Alphabet

The development of writing around 4000 BC, is the most important discovery in human history. Simple drawings on cave walls or rocks depicting people, animals, or scenes, evolved into symbols over time by acquiring abstract meanings. From walls and rocks, these symbols evolved into sentences and numbers on clay tablets, leather, and papyrus.



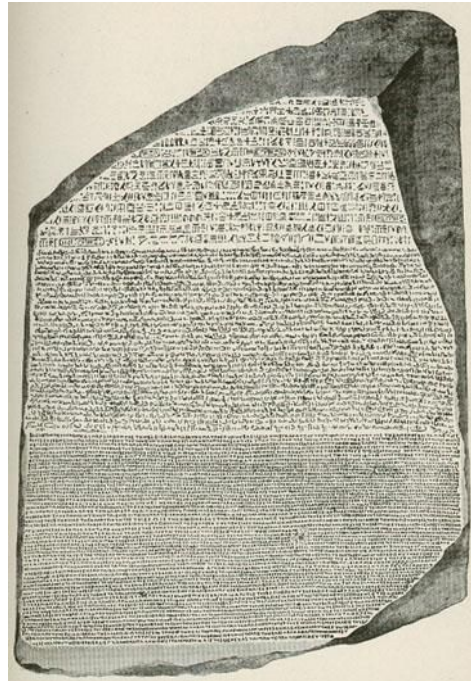
*Archaeological Museum of Sulaymaniyah,
Iraq - Lines of Epic of Gilgamesh, cuneiform tablet*

When writing materials were expensive, rarely found and educational institutions were not common being literate was a real superiority and a special privilege. The few writers, accountants, and tax officers who knew how to write, read, and calculate protected their privilege enviously. They would carefully choose a small number of apprentices, who would take their role. Thus, the complicated shapes on tablets and bone fragments were nothing more than a mystery for the public. Information circulated in the monopoly of scribes. So there is no inconvenience or contradiction in labeling the first literates as first cryptographers along with, even if they're not used that way, the first writings as the first cryptographic method.

It is essential to know the language, the writing, and the alphabet of the text that is being decrypted during the crypto-analysis.

Language, writing, and alphabet are in constant interaction with other languages, writings, and alphabets. Depending on the people who use them, they exhibit continuity. Their use diminishes against the dominance of more advanced kinds. Let's talk about two good stories as an example of alphabet-based hidden text analysis that we will work on in the future issues.

Hieroglyphic writings, primarily found on architectural structures and on papyrus sprinkled throughout the entire Egyptian landscape, kept the scientists busy as a mystery to be solved since the last literate was far gone. Everyone wanted to be able to read these writings and bring Egyptian history to light.



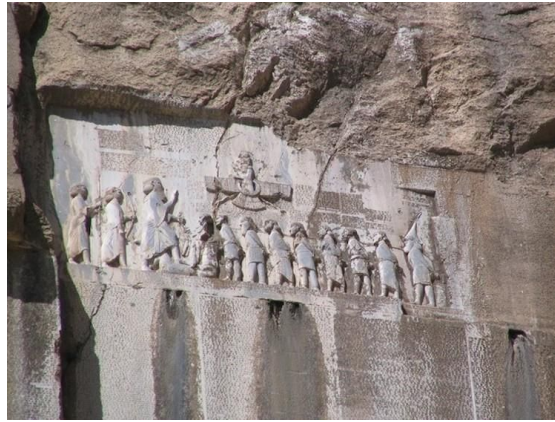
The Rosetta Stone

Under a castle wall, that was destroyed during Napoleon's Egyptian campaign in 1798, there was a large granite stone found which was bearing three texts in Demotik, Hieroglyph, and Ancient Greek. Jean-François Champollion, a French linguist and future professor of Egyptology, realized that the names of Ptolemy and Cleopatra in the Ancient Greek language were written in hieroglyphics inside a frame. He solved the text by matching and revealed that Hieroglyphs are not pictorial representations but an alphabet that produces sound. With the help of Coptic language, which was still-spoken, the ancient

Egyptian writings became legible, and the Egyptian historical researches gained a significant boost.

Henry Creswicke Rawlinson was an amateur archaeologist who served as an officer in The East India Company. He was assigned to Iran in 1838. Apart from his work, he started examining archaeological pieces in Iran. One day the local guides took him to the slope of a steep rock 50 meters high from the ground, The Behistun Inscription, built by the Persian Emperor Darius the Great in 5th century BC.

Nobody knew what was told in this inscription visible to everyone for thousands of years. No one was able to read or decipher it. He took permissions from the Iranian authorities with a heavy heart, copied the article by hanging on the rock with the help of a pulley system. For a long time, he tried to understand these symbols which made no sense. Rawlinson finally broke the cipher for this three-lingual inscription written in Old Persian, Elamite, and Babylonian. All the insurmountable obstacles in front of the inscription have been removed to understand the Mesopotamian inscriptions.



The Behistun Inscription

Lastly, I want to tell you about the Windtalkers movie. During the Second World War, seeking to retaliate for the difficulties they have suffered to decrypt the Japanese Purple code, the Americans enlisted a group of Navajo locals as radio operators. They told the Japanese to decrypt the Navajo language. I recommend the movie for those interested in cryptography.

See you in the next issue...

SPOTLIGHT: ANONYMITY

The Art of Data Hiding: Steganography

Anonymizing Internet from the Router with OpenWrt and Tor

Set up your VPN with "Kendi Bağlantım" (My Connection)

Acting on the Sly: Overcome Obstacles with DNS Tunneling

THE ART OF DATA HIDING: STEGANOGRAPHY

The first concepts that come to mind regarding security are undoubtedly data secrecy and privacy. These two concepts, quite similar, do differ in some points.

We can encrypt data with various algorithms or tools so that we can reformat it in a way that third parties cannot understand. However, in this case, the third parties are aware of the existence of this data and try to steal it by various methods. In the science of 'cryptography,' which aims to make the data private, the security of our data depends on its ability to be effectively encrypted with powerful algorithms.

Another method is to hide the data in such ways that its existence is obscured, and even to prevent third parties from accessing it in the motives of stealing. Steganography, which I want to emphasize on this article, comes to play at this point.

The word 'Steganography' in English is derived from the Greek words 'Steganos' (hidden) and 'Graphein' (writing). When we examine the history, we observe that we've always had the need to hide data for centuries, and various methods have been developed, mostly to be used in wars, diplomatic correspondences, and intelligence purposes. The first example of steganography is found in the Histories book of the ancient Greek historian Herodotus. During the Greco-Persian wars, Histiaeus, a Persian ruler, ordered his slave's hair to be shaved, only to scribble down the message to start an uprising on his scalp and sent this slave to Aristagoras when his hair grew back.

Again, some other interesting examples from the past are, writing down on wax tablets and pouring wax over them in the ancient times, using invisible ink in the Second World War, using micro-punctuation and null cipher techniques, using ordinary sentences to hide letters and sending encrypted messages, a tortured prisoner blinking in Morse code, and using pens that write down letters visible only in the ultraviolet light.

How advanced are the techniques of data hiding considering today's advanced technology? New techniques in the digital world are as follows:

- Hide messages in the smallest bits of image or audio files
- Change the echo of a sound file

- Placing data in the invisible or unused areas of a file

Some of the terminology used in the science of steganography are as follows:

Carrier or cover file: The original file that the confidential information will be hidden into.

Stego-medium: The environment in which information will be hidden.

Embedded (embedded/payload): Hidden data in the cover file.

Stego: The state of the file after the message is hidden.

Steganalysis: The process of detecting hidden data in a file.

Steganography Methods

Text Steganography

Although this technique seems rather simple, it is quite tricky to find the hidden data in the text. First, sentence structures are created in the text, and the letters are added according to the specified rules, and spaces are filled. There are no errors regarding expression in the text, but morphological errors can be found in some words. As an example, let's examine the cryptic text that a German spy used in World War II:

'Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects Pretext for embargo on products, ejecting suets and vegetable oils.'

He combined the second letter of each word and conveyed this message:

'Pershing sails from NY June.'

Text steganography has many other methods that can be used like line/word shifting, open spaces, character encoding, semantic methods, special character usage in words, and acrostics.

Image Steganography

With image steganography, one of the most commonly used methods, we can hide our messages inside the pixels of an image. In the “Least Significant Bit’ (LSB) in BMP technique,” we can hide data in the 24-bit image file BMP (Bitmap), which holds the image properties without any compression, making it ideal for storing data. Considering each pixel is 24 bits, changing 2 bits will not make a noticeable change. The color values of each pixel are kept in the 3-byte area containing red, green, and blue colors.

If we assume that the 3 pixels of the 24-bit picture are as follows:

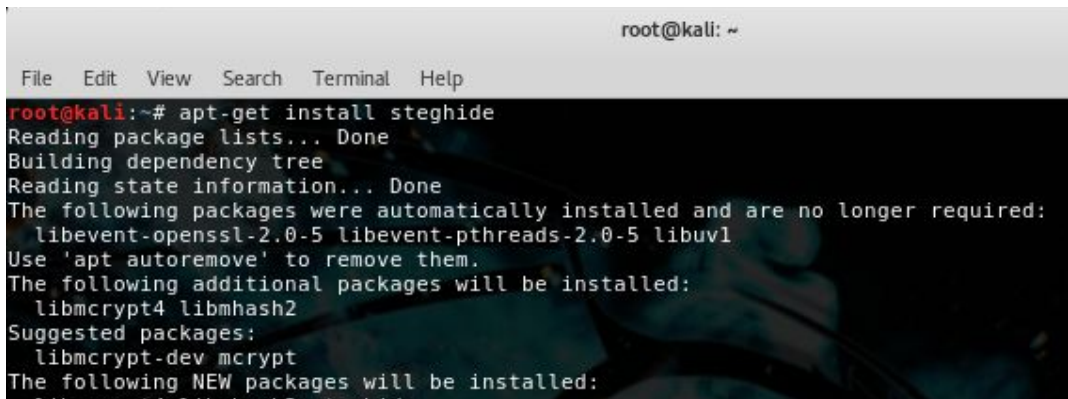
(00101101 00011101 11011100)
(10100110 11000101 00001100)
(11010010 10101100 01100011)

We get 11001000 or the binary number of 200 when we mark the LSB of every 8 bits.

(0010110**1** 0001110**1** 1101110**0**)
(101001**1**0 1100010**1** 0000110**0**)
(110100**1**0 1010110**0** 0110001**1**)

By using LSBs like this, the binary bits of the letters can easily be hidden in bits of the data. Let's see an applied version of hiding our data by installing the popular Steghide tool on our Linux operating system.

1. First, we use the apt-get install steghide command to install our tool.



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# apt-get install steghide  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  libevent-openssl-2.0-5 libevent-pthreads-2.0-5 libuv1  
Use 'apt autoremove' to remove them.  
The following additional packages will be installed:  
  libmcrypt4 libmhash2  
Suggested packages:  
  libmcrypt-dev mcrypt  
The following NEW packages will be installed:
```

2. After setting it up, we can see the available parameters by typing '*Steghide.*'

```

root@kali: ~
File Edit View Search Terminal Help
root@kali:~# steghide
steghide version 0.5.1

the first argument must be one of the following:
embed, --embed          embed data
extract, --extract      extract data
info, --info            display information about a cover- or stego-file
info <filename>        display information about <filename>
encinfo, --encinfo      display a list of supported encryption algorithms
version, --version      display version information
license, --license      display steghide's license
help, --help            display this usage information

embedding options:
-ef, --embedfile        select file to be embedded
-ef <filename>          embed the file <filename>
-cf, --coverfile        select cover-file
-cf <filename>          embed into the file <filename>
-p, --passphrase        specify passphrase
-p <passphrase>        use <passphrase> to embed data
-sf, --stegofile        select stego file
-sf <filename>          write result to <filename> instead of cover-file
-e, --encryption        select encryption parameters
-e <a>[<m>][<m>[<a>]    specify an encryption algorithm and/or mode
-e none                 do not encrypt data before embedding
-z, --compress          compress data before embedding (default)
-z <l>                  using level <l> (1 best speed...9 best compression)
-Z, --dontcompress     do not compress data before embedding
-K, --nochecksum        do not embed crc32 checksum of embedded data
-N, --dontembedname    do not embed the name of the original file
-f, --force             overwrite existing files
-q, --quiet             suppress information messages
-v, --verbose           display detailed information

```

- Using the command `Steghide embed -cf ogemi.jpg -ef parolalarım.txt`, I used `-cf` to specify the image to hide the data (cover file), and `-ef` parameter to specify the file to hide (embedded). You can review the *'Embedding Options'* section and add any parameters you want to include. We are also asked to create a *'passphrase'* as a security measure against the third-party while the data is being embedded.

```

root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~# cd Desktop/
root@kali:~/Desktop# steghide embed -cf ogemi.jpg -ef parolalarım.txt
Enter passphrase:
Re-Enter passphrase:
embedding "parolalarım.txt" in "ogemi.jpg"... done
root@kali:~/Desktop#

```

- We can examine the size of our ship after we load our text file on it. So let's get information about the content using the command `'steghide info ogemi.jpg.'`

```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# steghide info ogemi.jpg
"ogemi.jpg":
  format: jpeg
  capacity: 3.7 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "parolalarım.txt":
    size: 35.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes
root@kali:~/Desktop#
```

5. Finally, we use the command 'steghide extract -sf ogemi.jpg' to alleviate the burden of our ship and extract our passwords. In order to extract the file, we enter the passphrase we created in the beginning.

```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# steghide extract -sf ogemi.jpg
Enter passphrase:
the file "parolalarım.txt" does already exist. overwrite ? (y/n) y
wrote extracted data to "parolalarım.txt".
root@kali:~/Desktop#
```

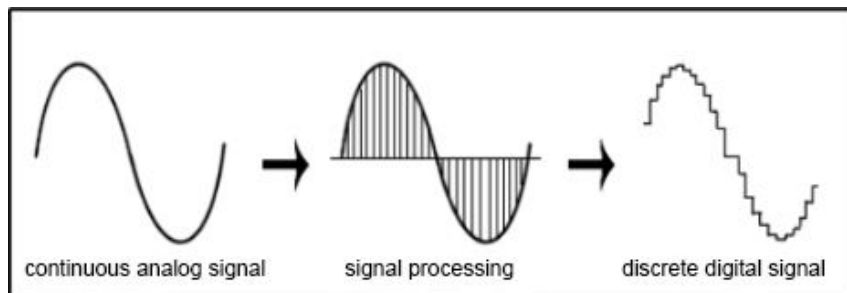
6. Finally, when we take a look at our image, we don't see a noticeable change before and after embedding the text file. So our ship is still waiting in the depths of the heart.



Sound Steganography

Sound steganography is used to hide data in a sound file or to mark it secure and firm. Secret information is hidden by being buried in sound signals. This method has severe and vital importance in some uses, such as battlefield communication and banking operations. Just like the image steganography, this embedding is done by changing the binary values. However, unlike the image steganography, the signal processing methods used for the audio file are far more complicated.

If we separate the audio signals as digital and analog, we can identify the digital sounds as discrete, and the analog sounds as continuous. Discrete signals are produced by processing continuous analog signals at a specific rate. For example, the digital audio processing rate for CD is 44 kHz. The image below shows a continuous wave of an analog audio signal processed to create a digital audio signal wave.



Sound Steganography Methods

Along with the advances in mathematics and signal processing, many methods have been developed to embed data in audio files. So I want to focus on the most common ones.

LSB (Least Significant Bit) Coding Method

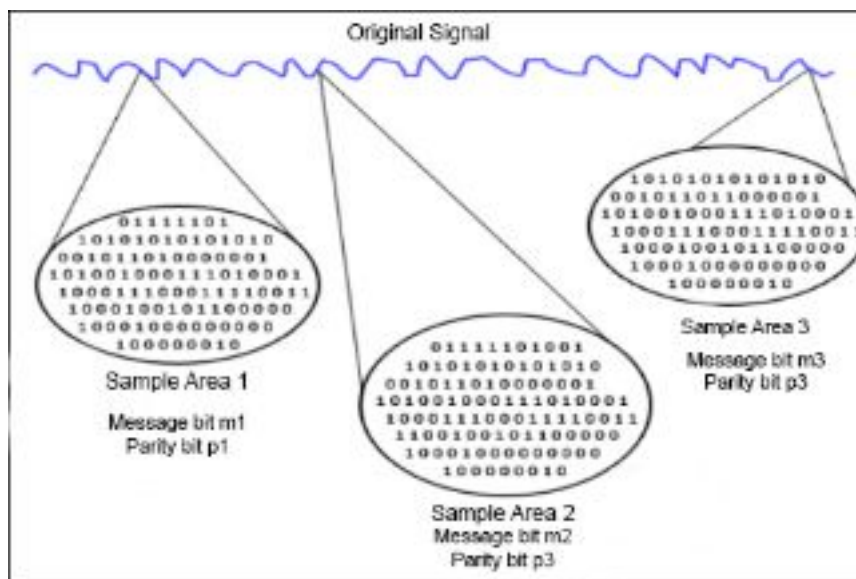
As I mentioned in the image steganography part, one of the most commonly used and easiest methods is undoubtedly the LSB encoding for audio files, too. If we depict an example for audio files, we can see in the following table that the binary counter of 'Hi' is encoded on the LSB.

Audio stream sample (16-bits)	"Hi" in binary	Stego audio Stream (w embedded message)
1 1 0 1 1 1 0 1 1 1 0 0 0 1 0 0	1	0
0 0 0 1 1 0 0 0 0 0 1 1 0 0 1 1	0	1
1 1 1 0 0 1 0 1 1 1 0 1 1 0 1 0	0	0
0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0	0	0
1 1 1 0 0 0 0 1 1 1 0 1 0 1 1 0	1	1
0 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0	0	0
1 1 1 1 1 0 0 0 1 1 0 0 0 0 1 1	0	1
0 1 0 0 1 1 1 1 0 1 0 1 0 1 1 0	0	0
0 1 0 0 0 0 0 0 0 1 1 0 0 0 1 1	0	0
0 0 1 1 1 0 1 1 0 1 0 0 1 1 1 0	1	1
0 1 1 0 0 0 0 0 0 0 1 1 0 0 1 0	1	0
1 0 0 0 1 1 0 1 0 1 0 1 1 1 0 0	0	1
0 1 1 0 0 0 1 0 1 0 1 0 0 0 1 0	1	0
1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0	0	1
0 0 0 0 0 0 1 0 1 1 1 1 1 0 1 1	0	0
1 1 0 1 1 1 0 0 1 1 0 0 0 1 0 1	1	1

With this method, we can not hear the changes in the sound. Messages are hidden by setting the frequencies above 20,000 Hz, inaudible for the human ear.

Parity Coding Method

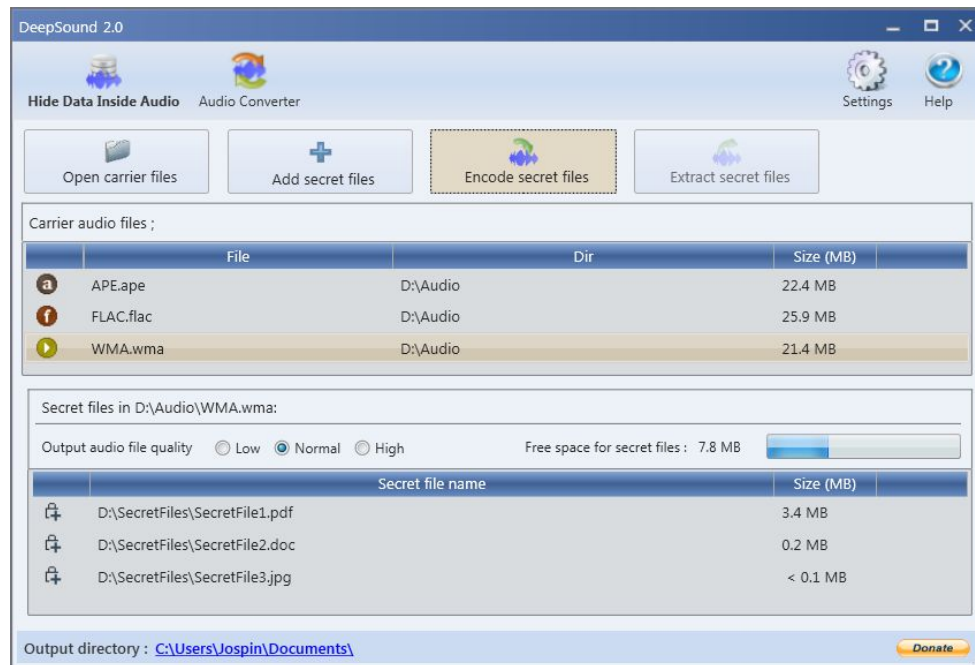
In the parity coding, the audio signal is divided into separate sample areas, and each message is hidden in the area within a parity bit. For this reason, this method offers a broader selection of options to hide the bits into and makes the change in the signal more unobservable.



Aside from these, there are other techniques like the phase coding technique which uses phase shifts to hide data, spread spectrum technique which attempts to distribute the covert data of the audio signal over the frequency spectrum as much as possible at the maximum level, and the echo hiding technique which hides the data using an echo added

to discrete signal in a sound file. However, I won't go further into details and show you how to embed data into sound files.

Let's switch from technical knowledge to practical knowledge. If you are watching Mr. Robot TV show, you probably saw how Elliot used DeepSound to hide data in audio files in season 1 episode 8.



In DeepSound, we select the audio file with the 'Open carrier files' option and select the file we want to hide with the 'Add secret files' option. We set the quality of the sound, and when we click 'Encode secret files,' the embedding process is completed once we select the desired format and set the password. We can access the hidden data when we use the application to open the audio file and click 'Extract secret files.'

Of course, aside from Steghide and DeepSound, dozens of tools have been developed. QuickStego, StegFS, StegoShare, Outguess, Stegbreak, Zsteg, OpenStego, Matroschka, AudioStegano, BitCrypt, MP3Stego, Xiao, Crypture, SteganographX Plus, rSteg, SSuite, Pícel, Camouflage, Hide'N'Send are just some of them. These include tools developed for both video and audio files.

Steganography is also used in watermarking to provide copyright information to text, audio, image or video files. Digital stamping is branched into visible and invisible. The visible stamping can be the logo found at the corner of any picture. In the unseen stamp, the person's private data is buried in the file to declare ownership.

Finally, "steganalysis" is the analysis and research on steganographic systems aimed at finding out if there is hidden information in a carrier file, and if so, obtaining this

information. It is divided into two segments, passive (scanning) and active (destruction/destruction). The steg-analysts examine various steganographic attacks. These types of attacks and their purposes are as follows.

File Only:	The attacker has access to the file and tries to determine whether or not there's a hidden message inside.
File an Original Copy:	The attacker could have copies of both the encrypted message and the original file.
Reformat Attack:	The format of the file is changed.
Compression Attack:	Compression algorithms are used to remove unnecessary information from a file.
Visual Attack:	It is a stego-only attack that infects some of the object to allow a person to look for visual anomalies.
Structural Attack:	The attacker can detect the presence of a message that analyzes the statistical profile of the bits.
Statistical Attack:	The frequency distribution of a potential cover file is compared to the theoretically expected distribution of the cover file.

From past to present, we can see how far the art of steganography has come when we study the historical process, and we can also predict that in the future, we will be able to progress faster and come up with new techniques. If we believe that privacy and secrecy are significant for people, steganography and cryptography sciences complement each other and occupy a vital and indispensable place in our lives.

References:

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.401.5678&rep=rep1&type=pdf>
<http://resources.infosecinstitute.com/steganography-and-tools-to-perform-steganography/#gref>
<https://www.slideshare.net/UttamJain/steganography-14902856>
<http://steganography-info.blogspot.com.tr/2008/04/steganography-and-attacks.html>

Anonymizing Internet from the Router with OpenWrt and Tor

Anonabox, the plug-in Internet router that claims to make your online activity anonymous, has raised nearly \$600,000 after Kickstarter suspended the controversial project.

We can accomplish the same task or even more by using some hardware with OpenWrt.

If you use a wireless repeater to install OpenWrt, you will have a Tor hardware with similar features of Anonabox. It will look just like Anonabox.

I will use Nexx WT3020H for this task, but you can prefer any other hardware. It surprisingly looks precisely like Anonabox.

WT3020H is actually just a wireless repeater, but we can modify it for our needs. If you adjusted this device as below, you could use it as a wireless repeater again.

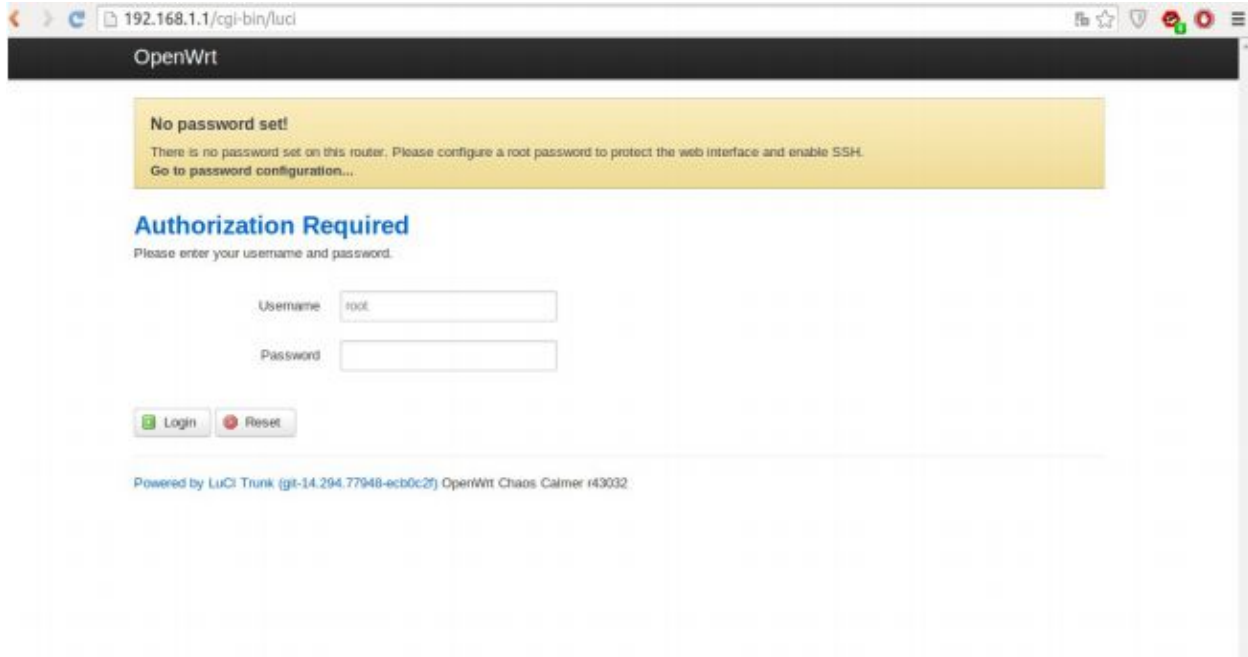
Installing OpenWrt

You must change the default firmware. Plug the RJ45 cable to the LAN port of the box. The default web interface will be at the address 192.168.1.1. Launch it on your browser.

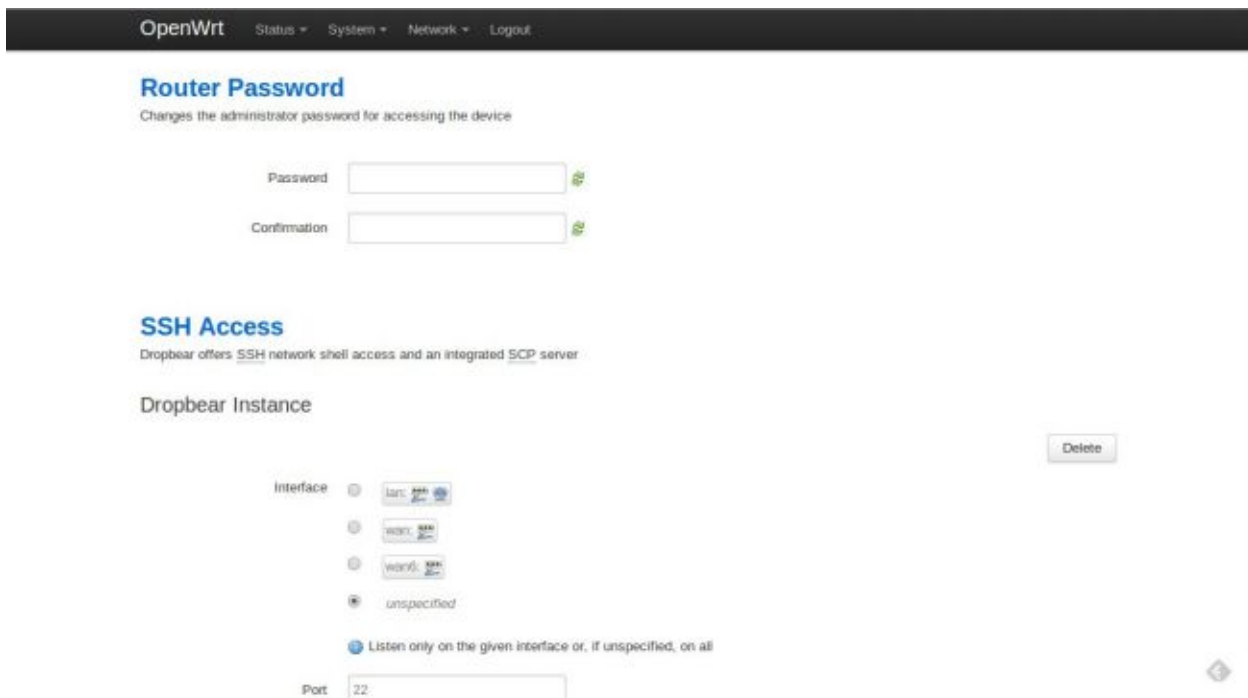
For the WT3020H version, the firmware is OpenWRT-ramips-mt7620n-wt3020-squashfs-8M-factory.bin and can be download from <http://onionwrt.link/download/>

Then use the upgrade image to upgrade your firmware.

After the completed installation, you will see this screen when you enter 192.168.1.1 with your browser.



You should set the root user's password. After logging in, you can activate SSH service from administration under the system tab.




After activating SSH, it is possible to adjust other settings, but I did extra work at LAN interface. Because 192.168.1.1 usually is router's address, I've changed IPV4 to 192.168.8.1.

With this adjustment, I can connect to the device with 192.168.8.1 as soon as the device is ready.

After these adjustments, we should be able to connect 192.168.8.1 IP address with SSH using the username and password we've set.

Interfaces

Interface Overview

Network	Status	Actions
LAN  br-lan	Uptime: 3h 27m 25s MAC-Address: 20:28:18:A0:A6:36 RX: 3.16 MB (26056 Pkts.) TX: 20.88 MB (29935 Pkts.) IPv4: 192.168.8.1/24 IPv6: FD3E:605A:1AB9:0:0:0:1/60	Connect Stop Edit Delete
WAN  br-wan	Uptime: 3h 27m 23s MAC-Address: 20:28:18:A0:A6:37 RX: 23.52 MB (24956 Pkts.) TX: 4.67 MB (12665 Pkts.) IPv4: 192.168.1.42/24	Connect Stop Edit Delete
WAN6  eth0.2	Uptime: 0h 0m 0s MAC-Address: 20:28:18:A0:A6:36 RX: 23.84 MB (27974 Pkts.) TX: 4.67 MB (12666 Pkts.)	Connect Stop Edit Delete

[Add new interface...](#)

Installing Tor

Before installing Tor, we should check if the device's wireless settings are working fine.

After clicking “WiFi” on the top panel, you can adjust settings like wireless network name, visibility, and password.

You should connect the device from its WAN input to router's LAN input using an ethernet cable. After joining this new wireless network, you should connect to the device with SSH using the IP address you set.

I've used this script to install Tor and complete other configurations.

After connecting with SSH, you can use this simple command to perform the installation.

```
wget -qo - http://onionwrt.us.to/install | sh -
```

If everything goes smoothly, you should see something like this on the screen below.

For the final step, you should restart the device. After connecting via WiFi, you should be able to browse the internet using Tor.

```

BusyBox v1.22.1 (2014-10-22 09:58:52 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

-----
|           |           |           |           |           |           |
| W I R E L E S S | F R E E D O M |
|           |           |           |           |           |           |
-----

CHAOS CALMER (Bleeding Edge, r43032)
-----
* 1 1/2 oz Gin           Shake with a glassful
* 1/4 oz Triple Sec     of broken ice and pour
* 3/4 oz Lime Juice     unstrained into a goblet
* 1 1/2 oz Orange Juice
* 1 tsp. Grenadine Syrup

-----
root@OpenWrt:~# wget -qO - http://onionwrt.us.to/install | sh -
Installing tor (0.2.4.24-1) to root...
Downloading http://downloads.openwrt.org/snapshots/trunk/ramips/packages/packages/tor_0.2.4.24-1_ramips_24kec.ipk.
Installing libevent2 (2.0.21-1) to root...
Downloading http://downloads.openwrt.org/snapshots/trunk/ramips/packages/base/libevent2_2.0.21-1_ramips_24kec.ipk.
Installing libopenssl (1.0.1j-3) to root...
Downloading http://downloads.openwrt.org/snapshots/trunk/ramips/packages/base/libopenssl_1.0.1j-3_ramips_24kec.ipk.
Installing zlib (1.2.8-1) to root...
Downloading http://downloads.openwrt.org/snapshots/trunk/ramips/packages/base/zlib_1.2.8-1_ramips_24kec.ipk.
Installing libpthread (0.9.33.2-1) to root...
Downloading http://downloads.openwrt.org/snapshots/trunk/ramips/packages/base/libpthread_0.9.33.2-1_ramips_24kec.ipk.
Installing librt (0.9.33.2-1) to root...
Downloading http://downloads.openwrt.org/snapshots/trunk/ramips/packages/base/librt_0.9.33.2-1_ramips_24kec.ipk.
Configuring libpthread.
Configuring libevent2.
Configuring librt.
Configuring zlib.
Configuring libopenssl.
Configuring tor.

```

Possible Problems

- If you get error messages while trying to run the script you can try to run the script's commands manually.
- If you can connect to the device but cannot connect to the internet, you may have to start Tor service manually or restart the firewall. In this case, you can run these commands from the terminal

```
/etc/init.d/tor enable
```

```
/etc/init.d/tor start
```

```
/etc/init.d/firewall stop
```

```
/etc/init.d/firewall start
```

- You may check iptables rules to see if there is a problem. If there is a problem, you can run the commands in script manually.

- For other problems you can check logs in the device interface.

Conclusion

After configuring the device correctly, you can use it anywhere with an ethernet cable.

You can browse the internet via Tor on your tablet, phone, computer, and all the other WiFi compatible devices without installing any additional software on them.

SETUP YOUR VPN WITH "KENDİ BAĞLANTIM" (MY CONNECTION)

Many useful websites like Wikipedia have blocked access in Turkey but there are multiple ways to access such blocked sites in our country. One of these is using VPN .

However, VPNs in our country aren't implemented correctly or what they are and how they work isn't known properly. Therefore, when we're trying to access banned sites on the internet, we're delivering all our traffic to people or organizations that aren't trustworthy.

In this article, I will not only describe what VPN is and how it works but I will also show you how to setup your VPN and be safe on the internet.

What is VPN?

VPN stands for Virtual Private Network. VPNs are internal networks established by individuals or organizations. In some cases, these internal networks are allowed to be accessed from a connection outside.

Here's a scenario to make it clearer. Let's say you signed up for an Internet Service Provider (ISP) and brought internet connection home. You get the IP address $X.X.X.X$ from the ISP. This IP address is assigned specifically to your connection. Therefore, when you access the internet from your home through any device, your IP address will appear as $X.X.X.X$.

However, at the same time, you have an internal network where your modem is broadcasted to devices like smartphones, computers, smart TVs, and more. Every device connected to this network gets an IP address from your modem only valid in the local network. These IP addresses must be in the same subnet mask as your modem. Generally, if the IP address of the router is 192.168.1.1, it gives out IP addresses like 192.168.1.2, 1.3, 1.4 and so on to the devices in the network.

The modem serves many roles such as acting as a router or a DHCP server. The broadcast is a communication method where all devices are equal. So it is a question/answer protocol. The modem is the rule maker here due to its functionalities like router and DHCP server.

The IP addresses that your modem assigns to the devices in the internal network are used by your device in the local network only. The TCP/IP packets your device sends out when it connects to the internet initially visit your modem and then readjust the IP address as the one given by the ISP (public IP) before it's delivered to its recipient. So the IP addresses of your smartphone and computer are the same on the internet.

So far, I tried to summarize how the network at home works. Note that we haven't reached the VPN part yet.

Now, suppose you installed a web server such as Apache on a computer at home and made it responsive to HTTP requests. Your computer is now both a 'client' and a 'server.' It is a client because before you set up a web server on it, it didn't have the mechanism to respond to requests. It became a server because it can now respond to requests.

In time, you added many files and contents to your web server. In fact, you weren't satisfied with one, so you installed 2 or 3 more servers in your local area network. To make it clearer, let's assign sample IPs to your servers in the local network:

- 1st server: 192.168.1.21
- 2nd server: 192.168.1.22
- 3rd server: 192.168.1.23

Now you can access your servers and the data found in them by typing in the IP addresses of your devices in the address bar of your browser.

The 3 servers you had grew in number and in time became 5, 10, and 15. Now you're beginning to forget which server kept which information. But you get the brilliant idea to assign domain names to these servers.

You can now access your servers by typing domain names such as 'school,' 'family,' 'game' instead of typing the IP address in your browser's address bar. What a beautiful system, right? Just like the internet! The Internet is a network where servers are connected and where each server and client has an IP address. You have created a network at home just like a miniature internet. Since this network belongs to you, it is a 'private network' – also called VPN.

Let's say a similar event happened at your workplace. The IT guys told you, "Well, we built a server for you with work files. Here is its IP address. But remember, it will only respond to

requests from your home IP address.” So your workplace has set up a server that you can only access from the IP address that the ISP assigned you.

But what if you have to go abroad for an extended period. But your school, family, game files were all on the servers on your local network. Since these servers are on the local network, you do not have remote access to them. What are you going to do? Are you going to open servers to the internet with port forwarding (NAT) from the modem? No, this won't work. If you do so, you will declare all those files accessible to the internet, meaning everyone can reach them!

Are you going to set up additional software for remote access to each server? No, not that either. You can access your school, family and game servers but you can not access the remote server at your workplace since it requires access from home.

Here's what you need to do: access your local network remotely. You can use an open source software to do this. For example, the most popular free VPN software is OpenVPN. You can remotely connect to your local network through OpenVPN. Every request you make to connect to the local network of your servers will reach this local network before it goes out to the internet. This way you will be able to access your server at work via VPN.

This is what VPN simply is. You set up a local network and connect to the network which has the contents you need. Thus, when you join that network, every request that needs to go out on the internet comes out of the connected local area network (VPN).

We use this architecture to access blocked sites. First, we rent a server located at a place where Wikipedia is not prohibited, then we install the necessary software and connect to the network. If we want to access Wikipedia later on, our requests will go through the VPN we are connected to and thus access Wikipedia without a problem.

What is “Güvenli Bağlantım” (My Secure Connection)?



“Güvenli Bağlantım” is a website that teaches you how to setup your VPN securely without speed loss to prevent interferences to your internet access. It helps you to setup a VPN entirely using open software and helps connect to the VPN you setup to surf the web safely and freely.

Now I'm going to describe how to setup your VPN with "Kendi Bağlantım" (My Connection). This section will consist of three subtitles as follows:

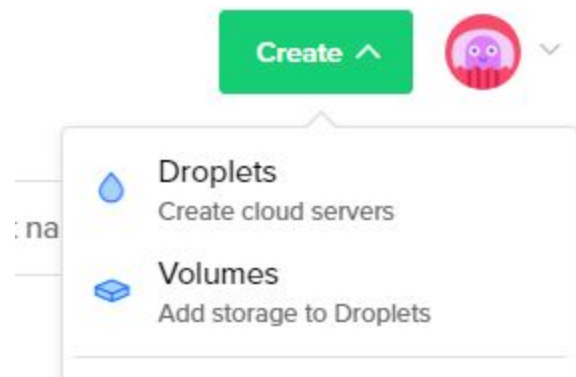
1. Server Rental
2. Server and OpenVPN Installation
3. Connection and program settings for PCs

By the end of these steps, you will be free and safe on the internet. We live in a geographical area where even a free encyclopedia like Wikipedia is blocked. It is very likely that we'll wake up to a morning where a new source of information is blocked. Use and teach VPN to tackle these obstacles and censorship on the internet.

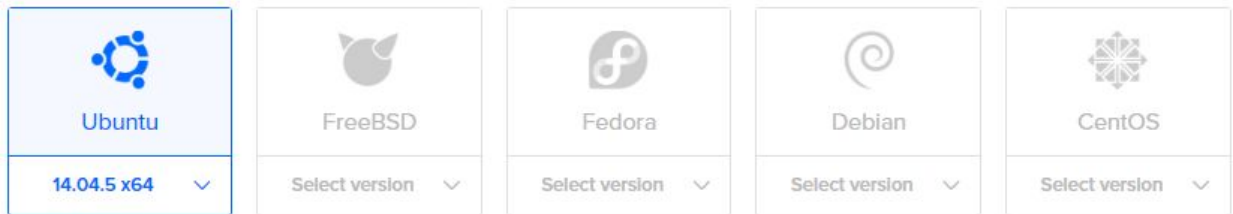
Server Rental

In this section, I will simply describe the steps to rent a server. For the sake of simplicity and reliability, we will rent the server from DigitalOcean.

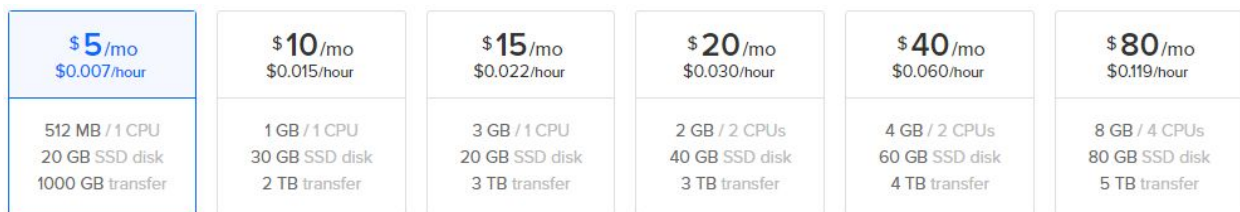
As the first step, you must sign up to digitalocean.com and enter your credit card information. My recommendation is to use a virtual credit card that has the limit \$0 and which can only be increased when shopping.



Next, click on the 'Create' button with the green background on the top right of the site as pictured above. Select 'Droplets' from the drop-down menu. 'Droplets' in DigitalOcean literature corresponds to 'server.' When the name of the hosting provider is the Ocean, we have to be a drop in it.



In creating a new droplet page, select the Linux version to be installed on the server. Select the 14.04.5 x64 version of Ubuntu on the far left.



Then, choose the package for the server's capacity. At \$5 a month, the package on the far left will be enough for us.



In this next step, choose the location of the server. Among the options, Frankfurt is the smartest location for us. Because compared to others, Frankfurt is the closest location to Turkey so we'll be able to navigate faster on the Internet.

How many Droplets?

Deploy multiple Droplets with the same [configuration](#).

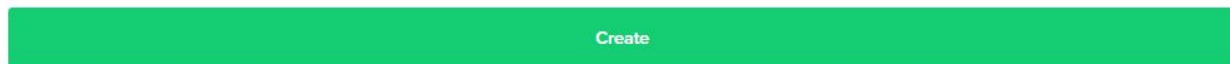
— 1 Droplet +

Choose a hostname

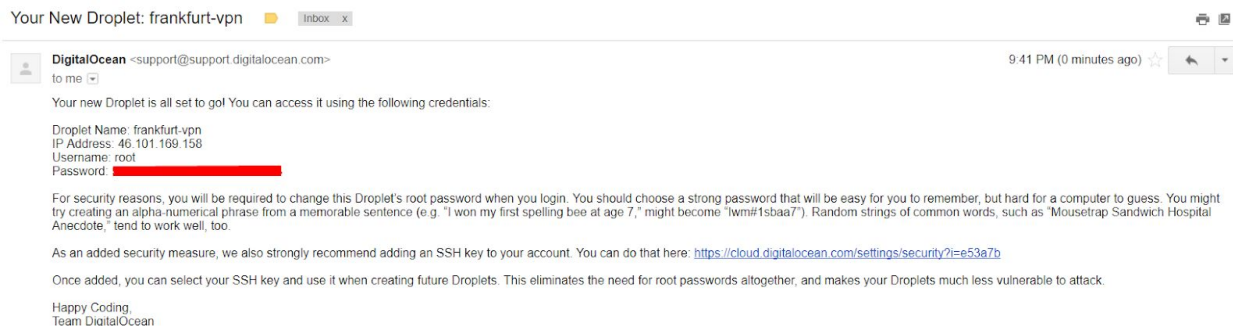
Give your Droplets an identifying name you will remember them by. Your Droplet name can only contain alphanumeric characters, dashes, and periods.

frankfurt-vpn

[Add Tags](#)



Finally, in the very bottom section, leave the left side as '1 Droplet' indicating to open 1 server and write the name of your server on the right side. I chose the name 'frankfurt-vpn.'



So the server is created in a matter of a few seconds, and the access information was sent by email!

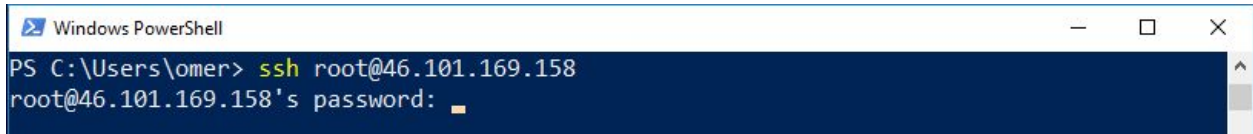
Server and OpenVPN Installation

First, we need to establish a remote SSH connection to our server using the information in the email from DigitalOcean. To do this, we need a terminal and an SSH software on our

computer. Linux and MacOS users can use the in-built terminals, but Windows users must use “Powershell.”

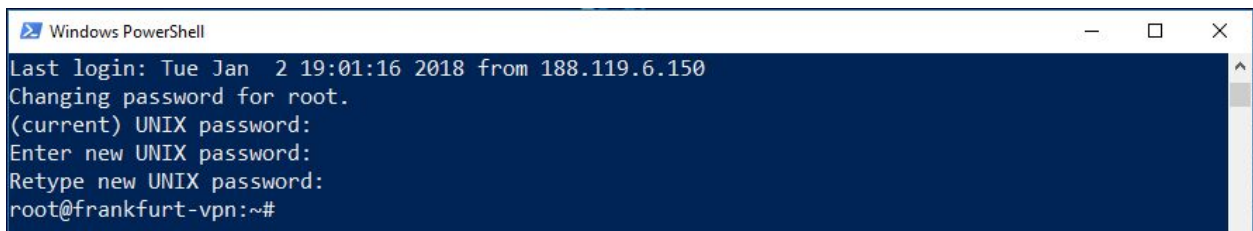
If you type “Powershell” in the Windows search bar, it’ll come up. After you launch Powershell, type the following command and press Enter.

```
ssh root@{IP_sent_in_the_email}
```



```
Windows PowerShell
PS C:\Users\omer> ssh root@46.101.169.158
root@46.101.169.158's password: █
```

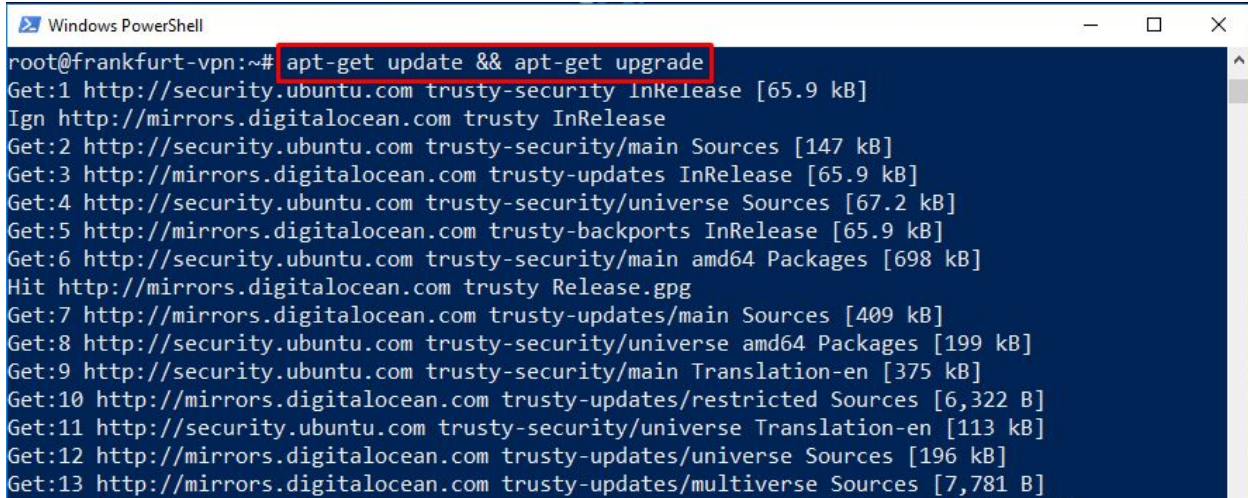
With the command above we will make an SSH connection to our server. After writing and executing the command, it will request a password from us. Your password is available, again, in the mail sent by DigitalOcean. If you copy the password from the mail and right click on your mouse on Powershell, your password will be pasted. But beware in this and the next password prompts, the password you type is invisible. You have to do the right-click just once. Don’t paste it multiple times assuming it’s not there. :)



```
Windows PowerShell
Last login: Tue Jan  2 19:01:16 2018 from 188.119.6.150
Changing password for root.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
root@frankfurt-vpn:~#
```

In the first step, it requested the server password. In the following two steps, it requests to set up a new password since it’s a first time login. Put a difficult password that no one can guess.

And we’re finally connected to our server.



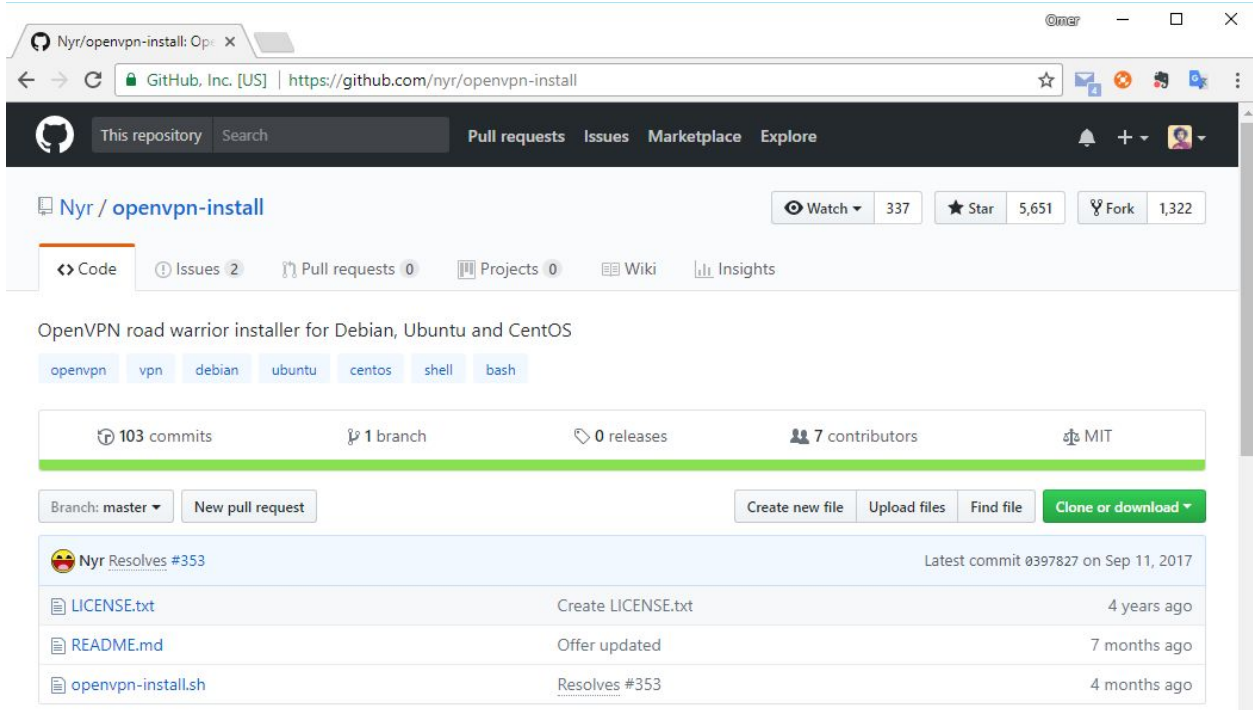
```
Windows PowerShell
root@frankfurt-vpn:~# apt-get update && apt-get upgrade
Get:1 http://security.ubuntu.com trusty-security InRelease [65.9 kB]
Ign http://mirrors.digitalocean.com trusty InRelease
Get:2 http://security.ubuntu.com trusty-security/main Sources [147 kB]
Get:3 http://mirrors.digitalocean.com trusty-updates InRelease [65.9 kB]
Get:4 http://security.ubuntu.com trusty-security/universe Sources [67.2 kB]
Get:5 http://mirrors.digitalocean.com trusty-backports InRelease [65.9 kB]
Get:6 http://security.ubuntu.com trusty-security/main amd64 Packages [698 kB]
Hit http://mirrors.digitalocean.com trusty Release.gpg
Get:7 http://mirrors.digitalocean.com trusty-updates/main Sources [409 kB]
Get:8 http://security.ubuntu.com trusty-security/universe amd64 Packages [199 kB]
Get:9 http://security.ubuntu.com trusty-security/main Translation-en [375 kB]
Get:10 http://mirrors.digitalocean.com trusty-updates/restricted Sources [6,322 B]
Get:11 http://security.ubuntu.com trusty-security/universe Translation-en [113 kB]
Get:12 http://mirrors.digitalocean.com trusty-updates/universe Sources [196 kB]
Get:13 http://mirrors.digitalocean.com trusty-updates/multiverse Sources [7,781 B]
```

First and foremost, we will update the repository and software on the server.

apt-get update && apt-get upgrade

You will have to wait a while after executing the command above. At the end of this wait, all applications and repositories will be updated.

Time to install OpenVPN.



Typically, the installation of OpenVPN is a bit complicated, but a Free Software volunteer wrote the code to ease these steps and shared it on GitHub. We'll take advantage of the tool this friend wrote.

GitHub link: <https://github.com/nyr/openvpn-install>

There is a command we need to run to use this tool also found on the GitHub page.

```
wget https://git.io/vpn -O openvpn-install.sh && bash  
openvpn-install.sh
```

When we write this command to the terminal and apply, we will encounter an output like the one below.

```
Windows PowerShell
First I need to know the IPv4 address of the network interface you want OpenVPN
listening to.
IP address: 46.101.169.158

Which protocol do you want for OpenVPN connections?
  1) UDP (recommended)
  2) TCP
Protocol [1-2]: 1

What port do you want OpenVPN listening to?
Port: 443

Which DNS do you want to use with the VPN?
  1) Current system resolvers
  2) Google
  3) OpenDNS
  4) NTT
  5) Hurricane Electric
  6) Verisign
DNS [1-6]: 2

Finally, tell me your name for the client certificate
Please, use one word only, no special characters
Client name: ev-bilgisayarim

Okay, that was all I needed. We are ready to setup your OpenVPN server now
Press any key to continue...
```

1. IP Address: The server is asking for the IP address. Press Enter as it is automatically filled in.
2. Select Protocol: UDP.
3. Port: Type 443 in case the VPN port on the network you are trying to connect may be blocked. No one blocks 443.
4. DNS: Select Google DNS.
5. Client Name: Enter the name of the device you're connecting with. For example, 'ev-bilgisayarim' (home-pc).

As a result, you will see an output similar to the following.

```
Finished!

Your client configuration is available at /root/ev-bilgisayarim.ovpn
If you want to add more clients, you simply need to run this script again!
root@frankfurt-vpn:~#
```

As you can see from the image above, you have created the file 'ev-bilgisayarim.ovpn' under the '/root' directory to connect to your VPN from your home computer.

If you want to connect to your VPN from more than one device, you can run the command above again and create a connection file with '.ovpn' extension for as many devices as you want.

Now we need to transfer the '.ovpn' connection file to your computer. We will use 'PuTTY SCP Client' software for this.

You can download the software from the 'Download' page on putty.org. Make sure to download the correct version of 'pscp.exe' as shown below.

pscp.exe (an SCP client, i.e. command-line secure file copy)

32-bit:	pscp.exe	(or by FTP)	(signature)
64-bit:	pscp.exe	(or by FTP)	(signature)

After downloading, we will go to the 'Downloads' directory of Windows and write the command we need to copy the file from the server on Powershell.

```

Windows PowerShell
PS C:\Users\omer> cd .\Downloads\
PS C:\Users\omer\Downloads> .\pscp.exe root@46.101.169.158:ev-bilgisayarim.ovpn C:\Users\omer\Desktop\ev-bilgisayarim.ovpn
The server's host key is not cached in the registry. You
have no guarantee that the server is the computer you
think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 256 49:ef:42:83:6b:f4:c3:7c:13:bc:e4:86:79:58:26:e7
If you trust this host, enter "y" to add the key to
PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without
adding the key to the cache, enter "n".
If you do not trust this host, press Return to abandon the
connection.
Store key in cache? (y/n) y
root@46.101.169.158's password:
ev-bilgisayarim.ovpn | 8 kB | 8.0 kB/s | ETA: 00:00:00 | 100%
PS C:\Users\omer\Downloads>
    
```

`cd .\Downloads\`

Go to the 'Downloads' directory with the command above.


```
.\pscp.exe root@46.101.169.158:ev-bilgisayarim.ovpn  
C:\Users\omer\Desktop\ev-bilgisayarim.ovpn
```

Download the 'ev-bilgisayarim.ovpn' file from the server to the Desktop directory of the computer using the command above.

Once you write this command, press 'y' to confirm to keep the key in the cache.

Immediately after that, you will be asked for your server password, and when you enter the password, the 'ev-bilgisayarim.ovpn' file will be downloaded to your computer.

ATTENTION: I wrote the command above according to my computer. If the language of your computer is different, the directory names may differ. Also I wrote the IP address of my server. You should write the IP address of your own server.

Connection and Program Settings for Computers

We need the OpenVPN client to connect to the VPN we have installed on our Windows device. To download the OpenVPN client we need to go to the 'Downloads' page of [openvpn.net](https://openvpn.net/index.php/open-source/downloads.html) (<https://openvpn.net/index.php/open-source/downloads.html>).

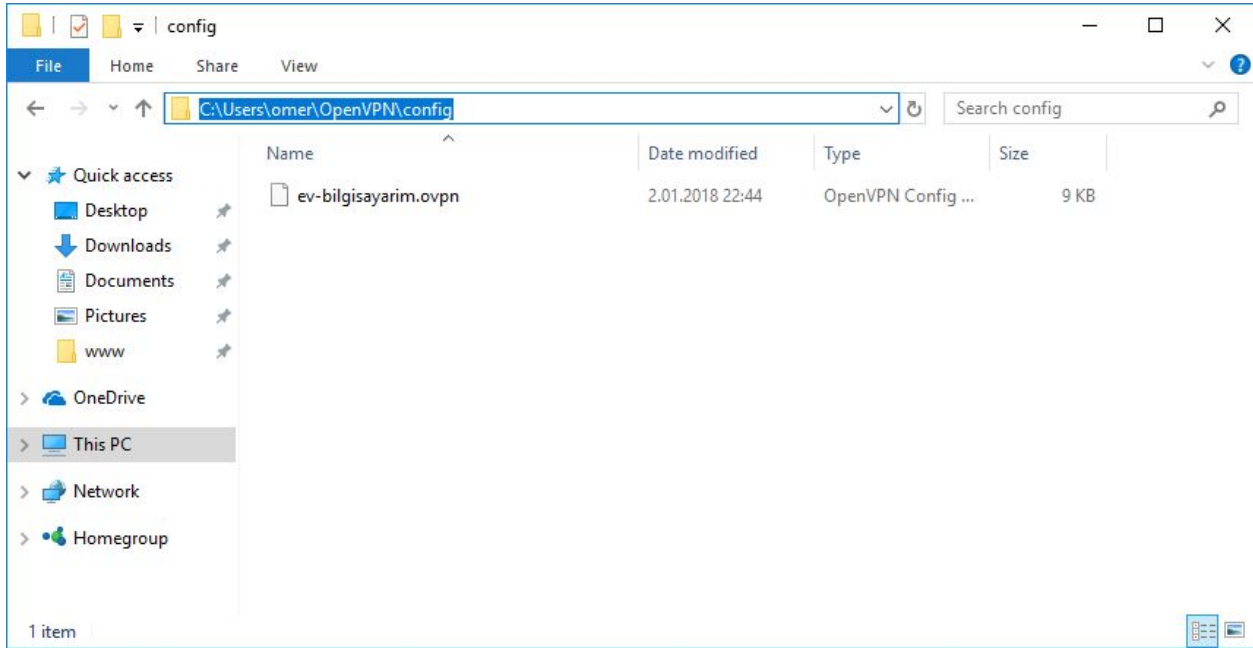
Source Tarball (gzip)	openvpn-2.4.4.tar.gz	GnuPG Signature
Source Tarball (xz)	openvpn-2.4.4.tar.xz	GnuPG Signature
Source Zip	openvpn-2.4.4.zip	GnuPG Signature
Installer, Windows Vista and later	openvpn-install-2.4.4-i601.exe	GnuPG Signature

When you go to the Downloads page, you will be welcomed with a similar table above. You have to download the most up-to-date version of the client, 2.4.4. the 'Windows Installer,' i.e, the 4th option above.

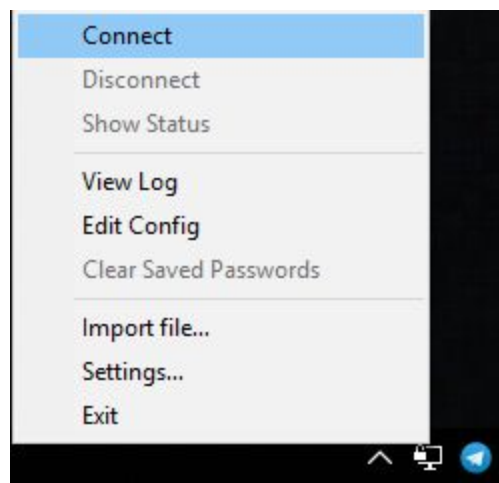
Then run the Installer you downloaded and click on the 'Next,' 'I Agree,' 'Next,' 'Install,' and 'Finish' buttons, without making any configurations.

That's all for the client setup. Now, to connect to the VPN you have created, you need to mount the 'ev-bilgisayarim.ovpn' file to the OpenVPN client.

Copy the 'ev-bilgisayarim.ovpn' file on your computer to "C:\Users\{username}\OpenVPN\config" directory. For example, on my computer, the username is 'omer' and the directory "C:\Users\ omer\OpenVPN\ config" looks like this:



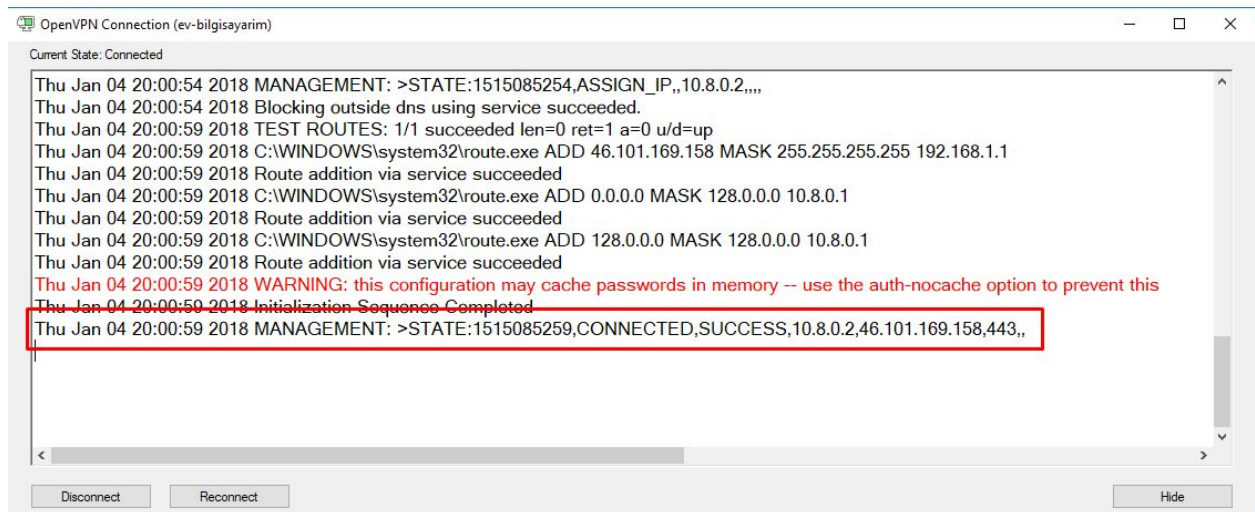
Once you've done this step, you can launch the 'OpenVPN GUI' shortcut on your desktop. After you run it, you will have the OpenVPN icon on the bottom right of our computer. All you need to do is to right-click on the icon and click 'Connect' as shown below.



And then you get a notification from Windows that it is connected to the VPN.



If you want to check for one last time, right-click on the OpenVPN icon on the right side of your taskbar. When you click 'Show Status' you must see the message 'CONNECTED, SUCCESS' as shown in the figure below.



It's that simple to setup a VPN! "Kendi Bağlantım" and Arka Kapı magazine wish you plenty of free and safe roaming on the internet in the new year!

ACTING ON THE SLY:

OVERCOME **OBSTACLES** WITH DNS TUNNELING

Let's imagine a scenario where you're at an airport or in a town square all alone, and you wanted to browse the internet.

Lucky for you, your WiFi receiver found many online modems around the square.

After connecting to a modem and browsing a website such as www.arkakapimag.com, guess what happens? A page wants your account credentials. I know you don't have an account but there is a link to create a "New User" at the bottom of the page. When you click this link, a login form that requests some very personal information like your cell phone number shows up.

Once again, you are all alone in the crowd!

Don't worry. This article aims to teach you how to find a way around obstacles like this and continue to surf the internet by DNS Tunneling.

This article is for educational purposes only, and we bear no responsibility for any damages caused by any software that is installed with our instructions.

The only way to understand this method is to understand DNS. DNS is a protocol also known as the phonebook of the internet. I know this analogy has been overused! Besides, I am sure that the 90s generation who is reading this article has never even seen a yellow phone book, but this is the best I can do until a better analogy found.

When we want to visit a page on the internet, we write the target website to the address bar. You get the website that you want to visit in a heartbeat. However, there is probably data transfer occurring very quickly in the background. How does this data transfer take place? Your web browser tries to detect the IP address related to this domain address when you enter www.arkakapimag.com. Once the IP address is found, an HTTP packet is sent to that address.

To see more details about this, you can read *Setup Your VPN with "Kendi Bağlantım"* written by Ömer Cıtaç on this issue of Arka Kapı Magazine.

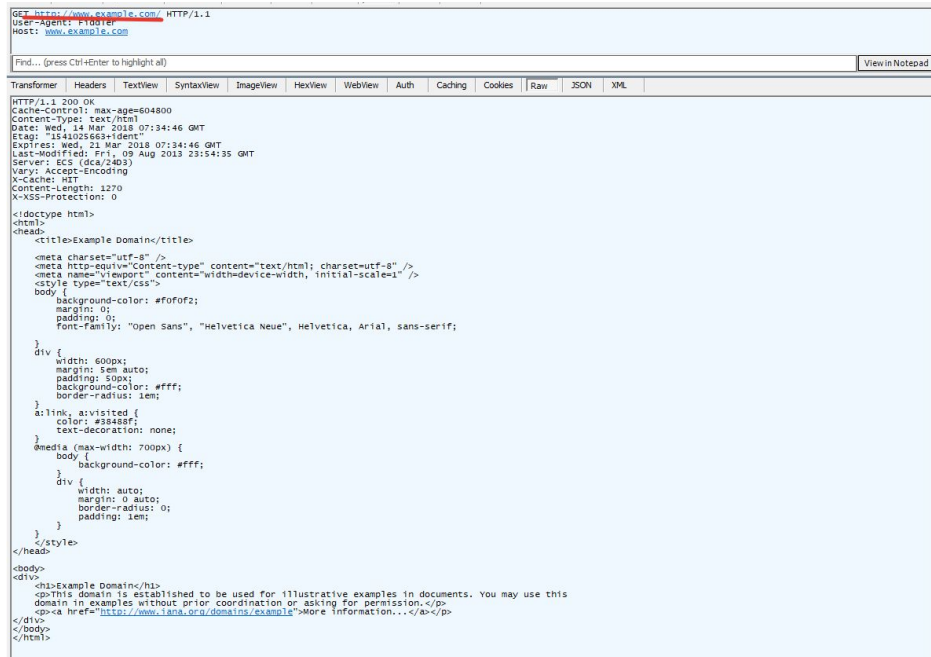
This method called DNS Resolution is seen as an innocuous process by almost all security mechanisms. Because the only process that is performed in this method is to demand and to get IP address information. The hotspots and the WiFi modems that request your personal details or a fee (like the public ones above) repeat the same action. However, many of the features in the DNS protocol may allow huge internet traffic to pass through this innocuous door. If I may say so, you can do things under the rose.

Let's assume a scenario where we create a DNS server under our control. Let's suppose, for example, the NS records of *freenet.arkakapimag.com* points to that DNS server. We start to listen port 53 (the default port of DNS protocol).

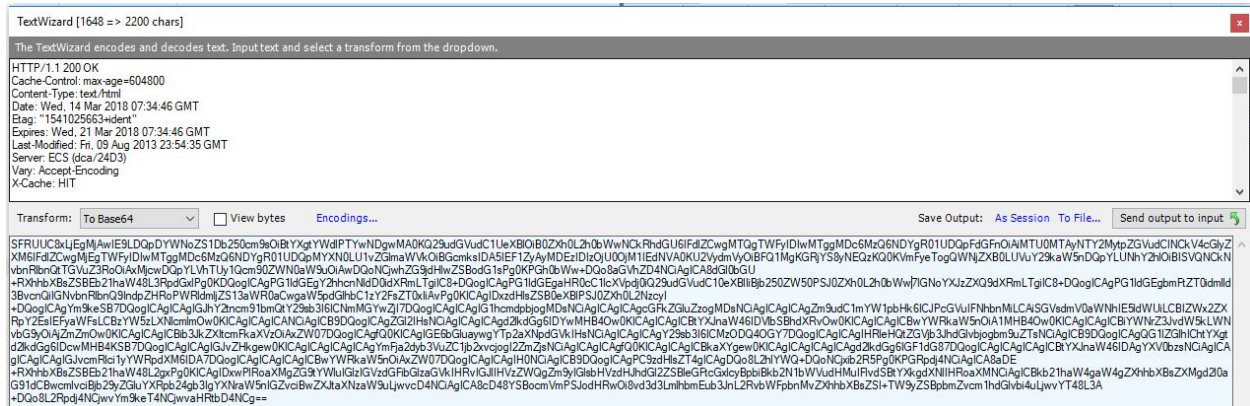
In case someone wants to visit a website through all the annoying circumstances mentioned above, let's ensure that this website is encoded with base64 and let's prepend the encoded address to the domain as a subdomain request. For instance, when we visit *example.com*, the base64 encoding of that address would be *ZXhhbXBsZS5jb20=*.

So, the address we visit will be *ZXhhbXBsZS5jb20=.freenet.arkakapimag.com*

The DNS address of *freenet.arkakapimag.com* recognizes the purpose of that DNS Resolution request. The content of *example.com* is requested, not the IP address equaled to *ZXhhbXBsZS5jb20=.freenet.arkakapimag.com*. Therefore, the server sends the base64 encoding content of *example.com* within the DNS records while responding to the DNS request.



The server that encodes this HTTP response with base64 gets the following value:



The server sends this base64 encoded data as a result of the DNS query. The client (or in this example - we), get the results that are responded to *example.com*, by decoding this response, and access it in our browser.

Don't worry. We will perform DNS Tunneling using *iodine* that is a software which takes all these steps for us. I wanted to mention these steps to make the underlying process of the tunneling clear.

We still need a server and a domain to do all this. The domain is *arkakapimag.com*. But, we will make the tunneling over the subdomain *freenet.arkakapimag.com* to avoid breaking the website *arkakapimag.com*.

We begin with the DNS settings first since the whole system is running over the DNS protocol. We add two DNS records to *arkakapimag.com*. One of them is a name server (NS) record for *freenet.arkakapimag.com*. This indicates the place that the requests will be captured while trying to access *freenet.arkakapimag.com* via different subdomains and that the operation will continue as described above. The requests will be captured over the server in which we install iodine. Let's say that the IP address of the server that we install iodine in is X.X.X.X. Also, let's name this DNS server: *freenetns.arkakapimag.com*.

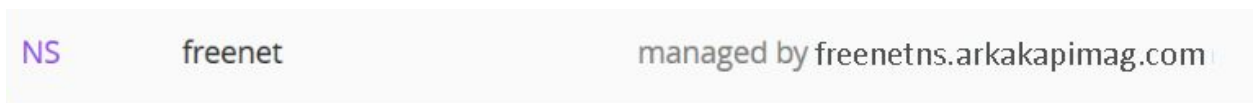
Now, we have to add two DNS records. The first record is the server that points to *freenetns.arkakapimag.com* that has *Iodine*. For this, we add a DNS record type A, and the domain address must be *freenetns*.



I am adding the DNS records over Cloudflare.

Here is the point: How will the subdomains be processed? Of course, via Name Server, also known as the DNS server. We cannot know which subdomains will be processed. This is because the websites that a user wants to visit are encoded with base64, and those make up the subdomains. So we will add an NS record for *freenet.arkakapimag.com* only instead of defining static subdomains. Then, all subdomain requests to *freenet.arkakapimag.com* will be sent to this DNS server, and the responses will be obtained from it.

So we add an NS record for *freenet.arkakapimag.com*. Let's ensure that this record points to *freenetns.arkakapimag.com* which we defined above.



Here is the second phase of the operation. We install iodine on the machine that is under our control and has X.X.X.X IP address.

We will run all terminal commands on the Ubuntu server. The package management (for example apt-get, brew) might be different than yours, but the basic functionality will be entirely the same.

It would be better if you can update your package management before starting. When I make the installation on the droplet that I bought from DigitalOcean (Thanks, Ömer), I realize that *iodine* doesn't exist.

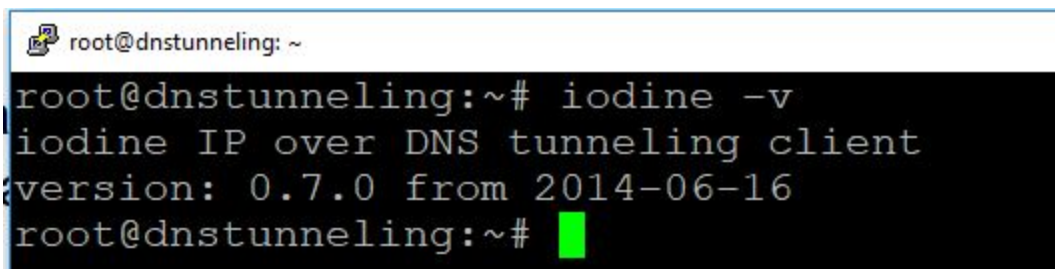
So, I am updating the package management first:

```
apt-get update
```

We can install *iodine* after update:

```
apt-get install iodine
```

To verify the installation, you can see the version info by running `iodine -v` in the command line.



```
root@dnstunneling: ~  
root@dnstunneling:~# iodine -v  
iodine IP over DNS tunneling client  
version: 0.7.0 from 2014-06-16  
root@dnstunneling:~#
```

Now, it is time to start *iodine*. With the following command, we start *iodine* on our server:

```
iodined -f -P acilsusamacil 10.0.0.1 freenet.arkakapimag.com
```

The descriptions of the parameters:

-f: *iodine* continues to run in the foreground on the terminal. You can easily stop the process with the keyboard combination of CTRL+C. Otherwise, it runs in background mode. *Iodine* can be seen in the terminal immediately after running the command.

-P: We set a password: acilsusamacil

10.0.0.1 is the IP address of the interface that *iodine* uses on our server. There will be a network setup between *iodine* and our client machine over a different network interface. Our client machine will get 10.0.0.2 or 10.0.0.3, probably.

freenet.arkakapimag.com: the domain that all requests are transferred as a DNS request.

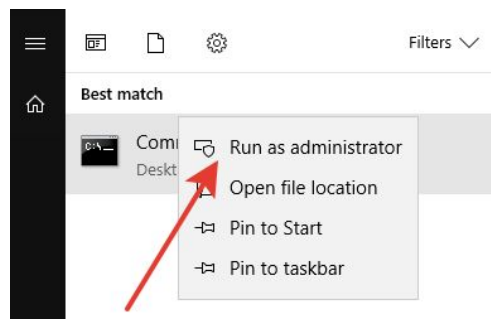
```
root@dnstunneling: ~  
root@dnstunneling:~# iodined -f -P acilsusamacil 10.0.0.1 freenet.arkakapidergi  
.com  
Opened dns0  
Setting IP of dns0 to 10.0.0.1  
Setting MTU of dns0 to 1130  
Opened IPv4 UDP socket  
Listening to dns for domain freenet.arkakapidergi.com
```

Now it is time to check the client, our machine. I assumed the client machine is a Windows 10 in this example.

First, we install the *iodine* client on our Windows 10 machine.

Go to <http://code.kryo.se/iodine/> and choose win32/64. The installation begins immediately. The file is just a small file of 243 KB.

Extract the ZIP file downloaded. Launch the command line with Administrator rights and go to the installation directory of *iodine*.



Change the directory to the installation folder of *iodine* and run the command below:

```
iodine.exe -f -P acilsusamacil freenet.arkakapimag.com
```

-f and -P parameters have been described above.

Note: I've hidden the IP address.

```
Administrator: Command Prompt - iodine.exe -f -P acilsusamacil freenet.arkakapidergi.com
D:\Ziya\Downloads\iodine-0.7.0-windows\64bit>iodine.exe -f -P acilsusamacil freenet.arkakapidergi.com
Opening device Ethernet 4
Opened IPv4 UDP socket
Opened IPv4 UDP socket
Opened IPv4 UDP socket
Sending DNS queries for freenet.arkakapidergi.com to 192.168.1.1
Autodetecting DNS query type (use -T to override).
Using DNS type NULL queries
Version ok, both using protocol v 0x00000502. You are user #1
Enabling interface 'Ethernet 4'
Setting IP of interface 'Ethernet 4' to 10.0.0.3 (can take a few seconds)...

Server tunnel IP is 10.0.0.1
Testing raw UDP data to the server (skip with -r)
Server is at XXXXX trying raw login: OK
Sending raw traffic directly to XXXXX
Connection setup complete, transmitting data.
```

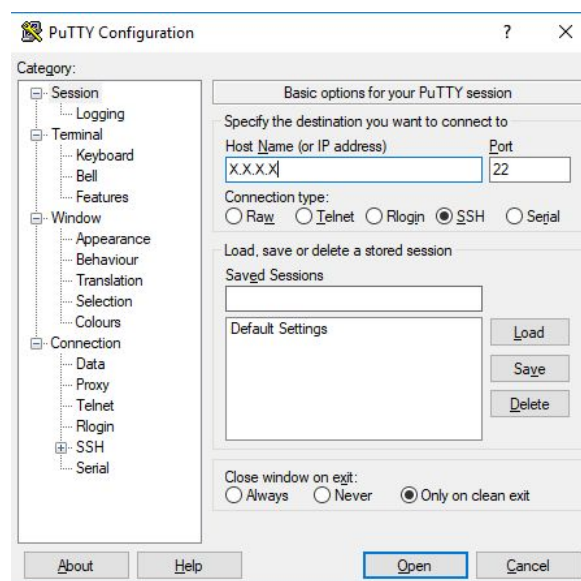
Now we've got two more steps remaining.

There will be a tunnel between the client and the server from now on. For example, we need to transfer the web surfing on the browser to this tunnel. There are many ways to do that. We can set up an SSH connection via Putty, connect this to a local port, and set up a socket connection over the browser.

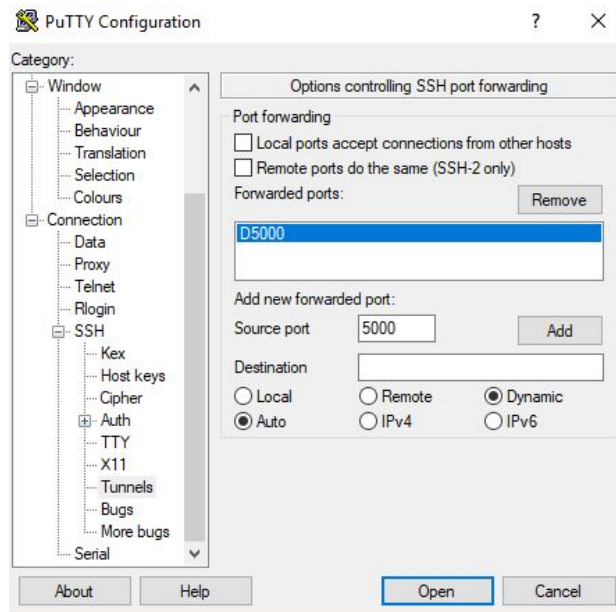
Is Putty installed on your machine? If not, please download the latest version of it clicking the link below:

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Open Putty. Specify that you will make an SSH connection to X.X.X.X (please write what your server IP address is) via port 22.



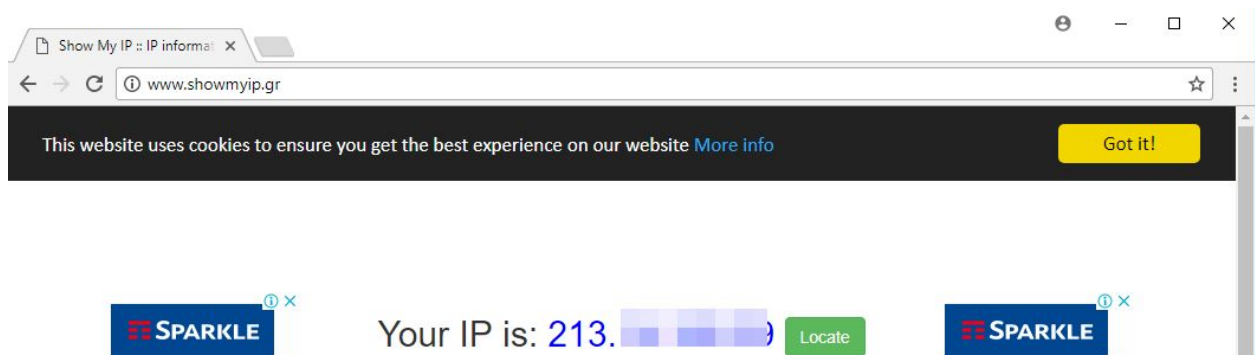
You still need to make several configurations under *Connection->SSH->Tunnels*.



In the window opened, enter 5000 Source Port, choose Dynamic from Destination, and click Add. You should see D5000 in the Forwarded Ports field.

Click Open. Enter the relevant information for your server connection in the terminal. If everything is OK, it means that the Socket connection that we will transfer the browser connection to is OK, too.

Let's double check before setting the browser. To confirm that our connection is done in an ordinary way instead of being done through the tunnel, we go to www.whatismyip.com or www.showmyip.gr and check our IP address.

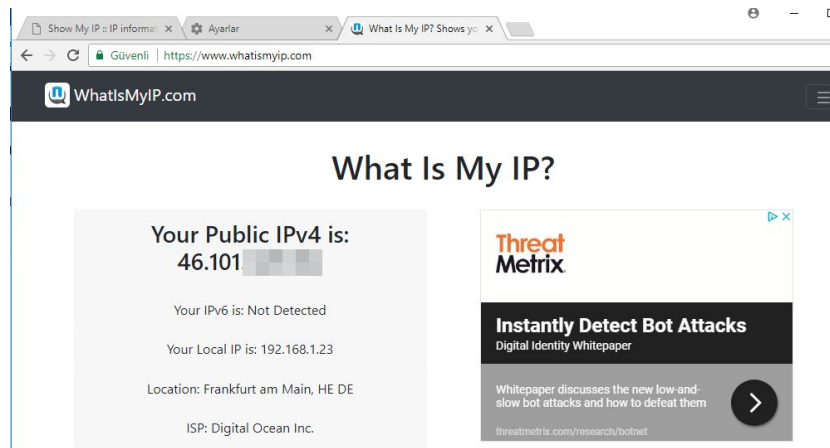


As you see in the screenshot, our IP address starts with 213. Now we configure the SOCKS settings in the browser.

You can reach out to Chrome's settings by entering *chrome://settings/* in the address bar. Go to Advanced settings, and then Open Proxy Settings.

In SOCKS configurations, enter 127.0.0.1 and 5000 for the address and the port field, respectively. Everything is ready now!

Let's recheck our IP address:



Congratulations! Your traffic in Chrome browser will be transferred over the DNS Tunnel. You can connect your mobile phone to your DNS Tunnel server using a tool named Andiodine, and enjoy it!

INSIDER

WOMEN IN SECURITY: AMANDA ROUSSEAU

Utku Şen - utku@utkusen.com.tr | Arka Kapı Magazine

“Those days you want to cry because you think you aren’t good enough should drive you to solve what areas you are weak in.”

Leaving behind DEF CON 26, we had the greatest pleasure to talk to Amanda Rousseau, who’s not only a review board member of DEF CON but also a white hat hacker, a reverse engineer, a senior malware researcher, and the voice of women in tech. A superwoman in her field, Rousseau gives us some insight on her job, the situation of the increasing presence of women in cybersecurity, and great advice for beginners.

Utku Şen: How old are you, where do you live, what are you doing in your current job?

Amanda Rousseau: Age, 30ish. San Francisco, California. In my current job, I look at a lot of malware. My job is to create proof of concepts for detection modules based on malicious behaviors that we research.

UŞ: When did you start to study computer security and why did you choose malware instead of something else? Could you please explain your journey into security and malware? What was your motivation?

AR: I knew of computer security in my teens, my father worked for the U.S. Air Force doing computer security. So basically, I was only exposed to it but at the time not too interested. I actually studied in university to be a graphic designer. So my first coding experience was Flash ActionScript at the time. My father convinced me to take one computer science class and I was hooked. Soon after I was a computer science department lab instructor and tutor. My first internship was in network security. After college, I was able to land a job at the U.S. Department of Defense Cyber Crime Center (DC3) where I learned on the job. I didn't start out as a reverse engineer at first. I had no idea what a reverse engineer was, I only knew that they were paid well. I had to work my way towards it. I started out as a forensic technician where I would break my fingernails prying open hardware to extract hard drives. Once I proved my worth, I was able to get training to move into dead-box forensics for criminal cases. A few months of working hard I was able to finally receive a mentor for reverse engineering. Coming from a creative background was tough. Switching between right and left brain all the time was draining. It was fun and also challenging. I was at the time okay at RE. I knew I needed to learn more so I decided to pursue a masters degree while working full time. This is where I honed my research skills. I still feel I don't know everything, constantly learning and grinding my way through problems is what drives me. There are no shortcuts.

UŞ: Did you encounter any difficulties just because of being a woman?

AR: Of course I had difficulty. No one takes you seriously until you have a chance to prove your worth. It was more of an obstacle you have to overcome, socially and physically. Every person is different: different cultures, personalities, strengths and weaknesses. It comes with experience to navigate around those situations. I didn't have a female role model in my field, I had to just deal with the challenges. All I can say is stay professional. Keep work and personal life separate until you can trust in combining the two.

UŞ: What do you think about current state of women in security? How can it be better?

AR: It's getting a lot better. There are more women reverse engineers than I ever knew 6 years ago. There are so many conferences and communities reaching out to the younger people. Getting exposure is what's important. I would not be in this field if I wasn't exposed to it in my teens. I still don't like identifying as a "hacker" but it's the only well known concept that people outside our industry understand. Getting paid to solve puzzles and break open

malicious software sounds more attractive.

UŞ: Do you think that the malware problem will be totally solved in the future?

AR: Never. Operating systems and software are developed by many teams and people. Someone is going to slip up. Some random person is going to find the holes and write malware for it. We are all still human.

UŞ: We are seeing that you are combining security and comedy together, and your drawing skills are great. Do you draw as a hobby?

AR: Well, I also have a degree in graphic design, so I can pretty much paint, draw, sculpt. But I only keep it as a hobby at this point. Comedy is what makes the negative aspects of computer security more bearable. Misery loves company.

UŞ: How does it feel to be a DEF CON review board member?

AR: It's a lot of work. I work full time but I also work in the evenings to review all submissions. That means cross-checking sources and reviewing code. I sit there and build the code and test it. I also spend time researching the subject to make sure the content is nuanced and valuable to the community. If they send in POC malware, I will sit there and reverse it. Try doing that 500+ times, but I do it because I love giving back to community. It's all volunteer effort.

UŞ: What's with unicorns?

AR: It's more of an inside joke. In silicon valley, a unicorn is basically a startup company that actually does very well after going public. I kept sending unicorn jokes to my boss, until it eventually became my totem I guess. Zombie unicorns seems like a halfway point of being not too feminine.

UŞ: What are your advices & messages to young people - especially women - who wants to be a malware researcher?

AR: If you like puzzle solving and have good attention to detail, you might like the field. Get exposure first, and see if you like it. The entry to the field is pretty steep. So find a mentor, take online classes, do CTF challenges, etc. Be proactive and seek out the information because it's not going to fall into your lap. There are no shortcuts, so you need to have your computer science foundations. You are probably going to fail at first, but trial and error is the best way to help you grow. Those days you want to cry because you think you aren't good enough should drive you to solve what areas your are weak in. If you don't know a programming language or subject, spend a day and prioritize time learning it.

Special thanks to Amanda Rousseau for being our first guest at our Insider series, Arka Kapı Magazine. Follow her on Twitter @malwareunicorn

CHAIN OF INDEPENDENCE: BLOCKCHAIN

In recent years the cryptocurrency world, where material value has increased and where people jump on the boat with the hopes of being rich, Blockchain made our agendas quite busy.

It wouldn't be wrong to tell you that Bitcoin has been the most popular cornerstone in this cryptocurrency world.

I'd like to touch on the technology and philosophy behind Bitcoin to be able to explain why it is so worthwhile nowadays.

What is Bitcoin?

After the global financial crisis of 2008, people began to have a distrust on financial institutions and institutions that regulate and supervise these mechanisms. A little later, Satoshi Nakamoto published an article titled "Bitcoin: A Peer-to-Peer Electronic Cash Payment System." If you search this title on Google, you can read the article.

Bitcoin was born as a digital currency without any central system, without anybody's intervention and manipulation.

Is Bitcoin just a currency? Of course not! I also see it as a rebellious child who riots against the financial world and the central institutions and allows people to transfer its digital presence peer-to-peer without being attached to the central institutions.

Now is the time to talk about the technology that makes this rebel child's hand so powerful: The Blockchain Technology.

What is Blockchain?

We can see the Blockchain network as a book where all transactions or digital assets are stored.

The most important feature of this book is that the data is not stored in a single point, but is distributed in the whole network. Even if one of the points where the data is held is lost, the data and system operation continue over the other points.

In blockchain, each chain is associated with a previous chain through a private encryption. Giving the book example again, each page is associated with the previous page through private encryption.

The change of a page in the book also makes the previous pages incompatible therefore the changes in the book will be instantly visible.

Keeping the book this way and its working mechanism helps us to give Blockchain full grades on security and transparency.

We all heard Blockchain with Bitcoin, and perhaps some of us might have thought that Blockchain and Bitcoin are the same things.

I like to define Blockchain technology as a protocol or a programming language. If we can develop solutions for different problems in a programming language, we can also work out different problems with Blockchain. For these reasons, I have defined Blockchain as a book where not only Bitcoin transfers are held, but digital assets and data are also kept.

Blockchain has brought solutions to our problems and let's talk a bit about them.

By the way, there is no obligation to be a miner on the Blockchain networks, as you will see in the examples below.

Blockchain Applications

Bitcoin

Bitcoin is the first cryptocurrency that we are all familiar with, which has been occupying the agenda in recent years with increasing its value, digital gold, so to speak. In addition to being the first cryptocurrency, we can also define it as an application that uses Blockchain technology for the first time.

Peer-to-peer money transfers are written in blocks obtained by solving the specific problem by the persons or groups we call miners.

Ethereum

Ethereum introduced by Vitalik Buterin in 2015 is the first cryptocurrency that is open source and allows the design of Blockchain-based smart contracts. The smart contracts that are created take place instantly when the contractor's conditions are fulfilled, without the need of intervention by anyone. If we need to give an example; Suppose two people, A and B, are doing a smart contract with an Ethereum base. According to the agreement, person A and person B have made an estimate based on the dollar rate on 31.12.2018, and as a result of this estimation, a contract has been made that some money will be sent if the estimate is reached. When the 31.12.2018 date comes, and if the relevant conditions occurred, whoever wins the contract will receive the prize automatically.

You can also write your own contracts with the open source code and the script language Solidity.

IOTA

An open source distributed book focused on solving problems related to the Internet of things. An application based on the verification model of the two previous processes determined by the system instead of mining and blocks, and can be verified in a very short time compared to Bitcoin.

Bithealth

An enterprise aimed at making healthcare services faster and safer. Here are some of the benefits of this initiative, which keeps people's health data on Blockchain:

- Patients' health data protected by a private key.
- It can be sent instantaneously to any part of the world at any time.
- Encryption of personal data instead of traditional data storage models for safer storage.
- If any change in the patient's health condition occurs, the doctor will approve this change with a timestamp. Notified to the person concerned.

Copyrobo

An application that records visual and written documents based on Blockchain with a timestamp. With this application, you can prove your visual, written documents as yours and show them as evidence.

In the near future, Blockchain technology will provide solutions to all of our problems.

Smart Contract

The protocol which program the conditions of a contractor and ensure that these conditions are followed and realized without the intervention of anyone is called "smart contracts."

Every programmable flow can be turned into a smart contract.

Smart contracts show up with Ethereum, is the first in the crypto money.

I want to give a few more examples for good measure.

Example: If you have rented a vehicle by signing a contract for a month, you have agreed that you will not be able to ride your vehicle again if you delay the monthly instalments two consecutive times.

Something happened and you delayed your two consecutive installments. You will not be able to ride your vehicle again, and when we think that your car is programmable, this contract automatically turns down the car's ignition without any human intervention. You can duplicate these examples and as I mentioned above, you can make all your transactions programmable as smart contracts

Example 2: When we want to rent a traditional house or stay in a hotel, we usually need a human factor. When we are accommodated by an officer, the entrance key of the house or hotel room must be delivered to us. If we do not have any contact, can we have a chance to enter the hotel? Programmable door locks with object internet (IOT), programmable accessories: coffee makers, lightings, etc. Let's think about renting a house for a day's stay through a smart contract, and make a contract. As a result of this contract, consider that the house door allocated to us during our stated date is opened with our private key. When time is up, imagine that it is not opening again. This scenario is currently running on the sloc.it Blockchain base, you will see if you search on Youtube or Google.

We can duplicate samples. Imagination and technology have no limits.

How do smart contracts work?

1. The subject of the contract is created.
2. The terms of the contract are set.
3. The contract is generated in the blockchain environment.
4. The contract is signed with the private keys of the contracting parties.
5. The blockchain controls whether the contract's conditions are met.

6. The provisions to be applied as a result of the terms of the contract are applied by the blockchain.

What are the advantages of smart contracts?

- Security
- Speed
- Cost
- Backup
- Standardization

I'm concluding my article here even though there's a lot to talk about. I think we've cracked the door open about Blockchain and smart contracts. I didn't go further into the technical details but I hope we will also touch on the work ethics in the upcoming issues.

See you then!

Why You Shouldn't Store Sensitive Data in JavaScript Files

It's been known for many years that storing sensitive data in JavaScript files is not only a bad practice, but also a rather dangerous one. The reason for that is relatively simple. Let's assume that you dynamically generate a JavaScript file containing the API key of your user.

```
apiCall = function(type, api_key, data) { ... }  
var api_key = '1391f6bd2f6fe8dcafb847e0615e5b29'  
var profileInfo = apiCall('getProfile', api_key, 'all')
```

Whenever you create a variable in the global scope like in the example above, you make it available to any website that includes your script file as well.

Why Would You Do Something That's Clearly So Dangerous?

The reasons why developers would embed sensitive information in JavaScript files are wide ranging. For inexperienced developers this may be the only obvious way to pass information that was stored or generated on the server side to their client side code. It may also save some additional requests to the server. However, an often overlooked aspect of this is browser extensions. Sometimes it's necessary to directly inject script tags into the DOM for it to use exactly the same window object. This wouldn't be possible with content scripts alone.

Is There a Way to Protect Variables?

We have talked about the global scope above. For JavaScript in browsers, a global variable is effectively a property of the window object. However, back in ECMA Script 5 there was only one additional scope, the function scope. That means if we declare a variable within a function using the var keyword, it is not globally available. With ECMA Script 6 an additional scope was introduced, the block scope and together with it the keywords const and let.

Both keywords are being used to declare a variable in a block scope, but you can not reassign variables that were created using const. If we omit the declaration with any of these keywords or if we use var outside of a function, we create a global variable and it is actually quite rare that we'd want to do that.


```
"use strict";
```

An effective way to prevent yourself from accidentally creating global variables is to activate strict mode. This can be done by adding the string "use strict" at the beginning of either a file or a function. It will then prevent you from using variables that weren't declared before.

```
"use strict";  
var test1 = 'arka' // works  
test2 = 'kapı' // Reference Error
```

You can use this in conjunction with so called Immediately Invoked Function Expressions (short IIFE, pronounced *iffy*). IIFEs can be used to create a function scope, but they immediately execute the function body. Let's see how this looks like.

```
(function() {  
    "use strict";  
    //variable declared within function scope  
    var privateVar = 'Secret value';  
})();  
console.log(privateVar) // Reference Error
```

On first glance this looks like an effective way to create variables whose content can't be read outside of their scope. But don't be fooled. While IIFEs are a good way to avoid polluting the global namespace, they aren't completely suitable to protect their content.

Reading Sensitive Data From Private Variables

It is (almost) impossible to keep the content of a private variable private. There are different reasons, some of which we will examine now. This is by far not an exhaustive list, but rather a reminder to show *why* you should never save sensitive data in your JavaScript files.

Overwriting Native Functions

The most obvious reason to decide against this dangerous practice is that you actually want to use the value of a variable to carry out a certain task. In our first example we need this key to make a request to a server. And therefore we need to send it over the network in clear text. Now there aren't an awful lot of ways to do this in JavaScript. Let's say our code uses the `fetch()` function.

```
window.fetch = (url, options) => {
```

```

        console.log(`URL: ${url}, data: ${options.body}`);
    };

    // EXTERNAL SCRIPT START
    (function(){
        "use strict";
        var api_key = "1391f6bd2f6fe8dcafb847e0615e5b29"
        fetch('/api/v1/getusers', {
            method: "POST",
            body: "api_key=" + api_key
        });
    })()
    // EXTERNAL SCRIPT END

```

As you see, we can simply override the fetch function and then steal the API key that way. The only prerequisite is that we can include the external script after our own script block. In this example we just log it out, but of course we could send it to our own server as well.

Defining Setters and Getters

Private variables may not only contain strings, but also objects or arrays. Objects can have different properties and in most of the cases you can simply set them and read their values. But JavaScript supports a pretty interesting functionality. You can actually execute a function if a property is set on an object or when it's accessed. This works with the `__defineSetter__` and `__defineGetter__` functions. If we apply the `__defineSetter__` function to the prototype of the Object constructor, we can effectively log every value that's assigned to a property with a certain name.

```

Object.prototype.__defineSetter__('api_key', function(value){
    console.log(value);
    return this._api_key = value;
});
Object.prototype.__defineGetter__('api_key', function(){
    return this._api_key;
});

// EXTERNAL SCRIPT START
(function(){
    "use strict"
    let options = {}
    options.api_key = "1391f6bd2f6fe8dcafb847e0615e5b29"
    options.name = "Alice"

```

```

        options.endpoint = "get_user_data"
        anotherAPICall(options);
    })()
    // EXTERNAL SCRIPT END

```

If the code assigns a property to an object containing the API key, we can easily access it with our setter. The getter on the other hand will make sure that the rest of the code is working correctly. This is not absolutely necessary, but can sometimes be helpful.

Custom Iterators

After we took a look at strings that are passed to native functions and objects with setters / getters, it's time to take a look at simple arrays. If the code iterates over an array with a for ... of loop, we can define a custom iterator on the prototype of the Array constructor. This will allow us to access the content of the array and still maintain functioning code.

```

Array.prototype[Symbol.iterator] = function() {
    let arr = this;
    let index = 0;
    console.log(arr)
    return {
        next: function() {
            return {
                value: arr[index++],
                done: index > arr.length
            }
        }
    }
};

// EXTERNAL SCRIPT START
(function() {
    let secretArray = ["this", "contains", "an", "API", "key"];
    for (let element of secretArray) {
        doSomething(element);
    }
})();
// EXTERNAL SCRIPT END

```

I'm not going to talk about the concept of iterators, since that's a little bit out of scope here. What's actually important is that we can access the whole array from within the custom `Symbol.iterator` method and therefore steal the secret value.

The Aftermath

As mentioned before, this is not an exhaustive list of possibilities that would allow an attacker to steal secret values from your script files. Even IIFEs, strict mode and declaring variables in a function / block scope will not always help you. My recommendation is that you dynamically fetch sensitive data from the server instead of writing it into your JavaScript files. In most, if not all cases, this is a sane alternative and may even be more maintainable.

Thoughts on Meltdown and Spectre Vulnerabilities

It's a great timing to publish the first issue of your magazine following the months that had frightening security vulnerabilities like Meltdown and Spectre.

I'd like to greet everyone who's interested in this magazine and these topics. Through this magazine I've had the opportunity to write down a few thoughts on Meltdown and Spectre vulnerabilities.

Why Security Problems?

There's a lot of interest on security problems. It has a huge marketing significance, too. What are the underlying causes for this? Why aren't our systems secure?

Almost all our systems are open to the entire world due to the existence of the internet. If you have a server that simply has port 22 open for SSH¹, it's highly likely to see multiple login attempts every second on the system logs. We're constantly under attack. Meaning there are constant intrusion attempts performed on our programs. Why do these attacks succeed?

One of the main reasons is the C language. C isn't entirely faulty here. In fact, the C++ language inherits the attributions of C. However, the problematic features of C can be defined with these two: (a) the boundaries of array access aren't controlled automatically, (b) arrays are obscure pointers in memory. Besides these two, an input that triggers stack overflow may write codes on stack, executing the attack code (buffer overrun vulnerability).²

The security problem I claim here is mainly a structural problem of the language. If the stack boundaries were controlled in each access and if the types of indexes were apparent, like in Pascal, the existence of such vulnerability would be no more. If controls took place in time in the iOS, MS Windows, GNU-Linux operating systems to prevent this, our systems would naturally slow down. But aren't we going to slow down the operating systems to prevent

¹ Of course you tied SSH to another port, right? If you haven't turned off port 22 and tie it to another port, go do it right away, don't wait for the article's end!

² Obviously the developers of C are innocent. The C language represented a major, an advanced step at its time. Eventually, C is a language invented to write a portable operating system (Unix) on the PDP-11 machine. This machine had a 64 kbyte memory. It didn't have virtual memory or a separate kernel state. Therefore small and fast was important, and a security implementation that we understand today was impossible back then. C language was designed for the conditions in 1973. The main problem is that we're still using this language 45 years later.

Meltdown Anyway?

Surely not every security breach is caused by such vulnerabilities. For example, the SQL injection depends on another vulnerability in programming languages, the type systems. Problem here is that the user input and SQL input are the same type; string. These two should technically be different types. Here lies a vulnerability of the type systems of programming languages. Text and SQL prompts are two different data types. The interpretation of these two as same entities causes a security vulnerability.

Unlike previous weaknesses, the Meltdown and Spectre aren't directly caused by a language problem. Spectre vulnerability can even be exploited over a browser using a code written in Javascript.

Surprisingly, the problems in Meltdown and Spectre are tried to be resolved on programming languages, namely, compilers. These compilers attempt to avoid producing machine code function arrays that can lead to Meltdown and Spectre. Since we can't change millions of processors right away, there's no other solution. LLVM and GCC C language compiler infrastructures are trying to cover up the Spectre/Meltdown weaknesses using the "retpoline"³ technique on their machine codes.⁴

"Legacy" Issue

"The tradition of all dead generations weighs like a nightmare on the brains of the living." — Karl Marx, The Eighteenth Brumaire of Louis Bonaparte

We're trying to administer our computer systems using 30-years-old operating systems⁵ written with a 45-years-old programming language. The number of the computers today are thousands more than back then, and their speed and memory have increased by thousands. When the systems we mentioned were developed, only a few devices were connected to the internet, whereas now almost all of them are connected. Situations changed, hardware changed but the software technology didn't change as much. Updating software isn't as easy as updating hardware.

³ Details can be found on the Google Project Zero Blog (<https://googleprojectzero.blogspot.com.tr/2018/01/reading-privileged-memory-with-side.html>), llvm.org and lkml.org websites.

⁴ This technique by Google uses the "ret," return command, of the processor instead of indirect jump. The "trampoline" concept stands for tail call elimination in both machine level (interrupt, exception, and case programming) and functional languages (continuation passing style). The word "retpoline" comes from the combination of the words "ret" and "trampoline."

⁵ Unix 1974, Microsoft Windows 1986, GNU Linux 1991. iOS is Unix-based, too. So the youngest is 27 years old.

Solutions for The Future

If you're using Raspberry Pi for a computer, you shouldn't worry about the Meltdown/Spectre vulnerabilities. The ARM processor in Pi doesn't allow these weaknesses. The ARM in Raspberry Pi is a simple RISC (Reduced Instruction Set Computer) processor, so it escapes these attacks by not doing speculative execution.

We have a nightmare in our hardware, too: CISC. The x86 processor architecture is a nightmare haunting us from the past. It makes the manual writing of the machine code easier with its complicated but tough commands. The RISC architecture, developed in the 1980s in Stanford and Berkeley, does the same calculation with fewer transistor and consuming less energy because it uses simpler and solo-circuit commands with a large register on the processor. The simple commands makes the processor's pipeline to work easier in the RISC architecture. The processor commands go through a pipeline in the processor step by step and multiple commands are evaluated at once. This causes some commands to be evaluated as "out of order."

For example a "jump" command is registered along with the following command. This is known as Branch Delay Slot. The programming of software that are coded in machine language is therefore a hassle. But who programs in machine language anyway? The registries translate the programs written in a high level language into machine language. So we can leave this hassle for the registries to deal with. The one who writes the registry solves the problem once, and we benefit out of that.

Intel and its followers resisted for CISC for a long while, but RISC won. The majority of the processors (95%?) in the world is RISC due to its low electricity consumption. The ARM chips (RISC architecture) in iPhone and Android devices cause this majority. Intel Pentium chips include a RISC core since the P6 model. CISC commands are translated to RISC before being registered.

The fact that RISC commands are directly used by programs brings another advantage. We can leave the "out of order" problems to the registry to deal with. The x86 processors have to work out these problems on their own. According to AMD, the speculative execution is performed by an AI unit found directly on the AMD chips.

However, the processor shouldn't have to make guesses. The producer of the machine codes has the knowledge about registry jumping and cycles. The successful logic of RISC was just this.

In the past generations, there has been a bit of a convergence between the RISC and CISC

processors. Meaning they started to resemble one another. This is why the latest ARM processors (RISC architecture) may be vulnerable to Spectre. In theory, ARM (and AMD which has a different x86 architecture than Intel) could've been vulnerable to Meltdown but researchers didn't succeed in practice.

Thoughts for The Future

Problems like Spectre/Meltdown would be easier to solve through the path chosen for RISC. The RISC approach is to maximize the performance of a processor kept simple by using the programming languages and their registries. Maybe in the long term this is smarter.

The Spectre/Meltdown problem is caused by the processor's speculative execution function. If the speculative execution was under the control of a code built by the registry like the Branch Delay Slot, the problem would be solved with changing the registry.

It seems like we're going into a general crisis. Processors and their dependency increase. We nearly reached a trillion processors globally. This means roughly 130 processors per person. How many processors do you have at home? Don't forget to count the credit cards and the chips in mobile phones' SIM cards. Right now, a Turkish reader owns around 100 processors.

However, the increasing power of the computers isn't secure at all. Neither the servers, nor the Internet of Things devices. The programs buried in Blockchain are problematic too. We got to do something!

When we're all doomed with the speculative execution, I'd like to do some speculation:

Big changes always seems difficult. Backward compatibility is an issue foreseen by Karl Marx in 1848. We can't get rid of the weirdness of x86 and the C language. Regardless, sometimes change comes from our side, unexpectedly. For years, the question "When will GNU Linux win in desktops?" has been asked. I asked the least. It didn't happen. This fight has been ended without us realizing. Desktop's significance decreased. On the other hand GNU Linux and similar were victorious on the servers side. On personal use, mobile phones and tablets passed the desktop PCs in popularity. Not just MS Windows but Intel was given up on, too. The winners were the Unix/GNU Linux based iOS and Android. They did it with the ARM processors. What's more interesting is that, both of these formed environments with nearly single programming languages. Android unfortunately chose Java. Apple chose Pascal initially, then Objective C, and then Go. The important thing here is that the user was given limited programming languages to ensure a sustainable experience.

Unfortunately, as companies Google and Apple aren't brave enough to take radical steps in the programming languages issue. Sadly, Microsoft is far more advanced in the research of

programming languages. In Europe, Microsoft Research hosts a good amount of researchers. C# is a more secure language than Java in terms of type security. Languages like F# and F* don't come from Google and Apple. Go, Dart, and Swift are all missed opportunities.

Is it possible to go through a change in paradigm to overcome these security problems? We had something similar with Android. A new processor type (relative), new operating system (GNU Linux), and a single-language environment (sadly, Java).

The Blockchain world⁶ is in need of secure, approvingly working programs. Languages like Simplicity, Obsidian, F* are suggested to end this need in the Blockchain world. Generally, functional, statically typed in a sophisticated way and sometimes Turing incomplete. Turing incomplete can resolve the halting issue in finitistic languages. The accuracy of programs can be proved, too.

In the Blockchain world, these new languages transform into basic computer science theory. Lambda calculus, simple typed lambda calculus, and SKI combinators all underlie these languages.

If we're going to solve these problems once and for all, we have to review the programming issue. And the hardware-software relationship will play a huge role here. It was managed in a small scale by RISC jump. A huge change took place in the hardware-software relationship.

Such developments are possible in the Blockchain world because it's relatively new. Can a newer and more secure hardware-software relationship appear in an unseen sector that backward compatibility isn't necessary? A sector where programming language and hardware is compatible and secure. Of course we'd have to leave behind our dear friend GNU Linux.

Can there be a hardware that's compatible with a static type system, like Haskell, possibly finitistic like Turing incomplete, with a functional language? I don't know. RISC's victory took 25 years after its invention. The computer world is so lazy and conservative when it comes to new technologies, that it's difficult to expect fast developments.⁷ Otherwise we'll continue with patches and our world will be far more dangerous with broken systems.

Computer Science Education

Another thought on this subject is about how many computer science/engineering programs

⁶ Crypto-currencies consist of trust-chains. (Check out my link in T24 <http://t24.com.tr/yazarlar/chris-stephenson/bilgisayar-bilimcisi-gozuyle-kripto-para-ve-yatirimciya-tavsiyeler,18768>). However, applications like Blockchain technology and automatic agreements can be a significant area.

⁷ Think of physical science. In 1932 Neutron was discovered. It wasn't known until then. In 10 years, nuclear reactors and in 13 years the atomic bomb was made. Both of these were dependent on the existence of Neutron. It took longer for Microsoft Windows legacy to remove the 16 bit code than the development of the atomic bomb.

can give technical information to understand spectre/meltdown issue. 25 years ago they would. Now? This isn't a problem restricted to Turkey. I'm checking out the curricula that haven't been updated for a long while. But to my understanding, the education has been deprived of its contents. Are undergraduates taught about virtual memory, page tables, their cache, memory cache and pipeline functions? Can they write a program using machine code? Do they understand about the production of compiler code? Do they comprehend the role of high-level languages? How about lambda calculus, Turing completeness and incompleteness, type systems? Personally, I used to give lectures about most of these in my classes 20 years ago. I think now many colleges do not teach these.

If I'm wrong, do notify. If colleges are going to be useful in computer sector in the future, these subjects should be of high importance. Or else this job will be left to other institutions. I don't have much hope for universities (besides a few elite institutions).

Verdict

The meltdown/spectre problem agitated me. Hopefully the things I wrote about stimulated the readers for new ideas/thoughts. Good luck for all of us.

Vulnerabilities of Bluetooth: Past and Future

Bluetooth, which falls into many of our routines with cell phones, is now a technology widely used in many aspects of our lives. With IoT and Industry 4.0, almost all devices will be connected to the internet and communicate with each other. This technology is benefited not only by home devices but also covers a wide range including door locks, unmanned air vehicles, cardiac pacemakers, wireless speakers, and earphones. Through the mesh network technology emerged with the ever-evolving Bluetooth 5.0, coverage area can be expended up to 30,000 kilometers. Well, have you ever wondered how Bluetooth technology has come to the world? In this article, I will talk about Bluetooth, and help you understand its internal structure and operation logic, and give you the chance to examine the risk incorporated in our lives by Bluetooth, as well as the attacks against it.¹

Rise of Bluetooth

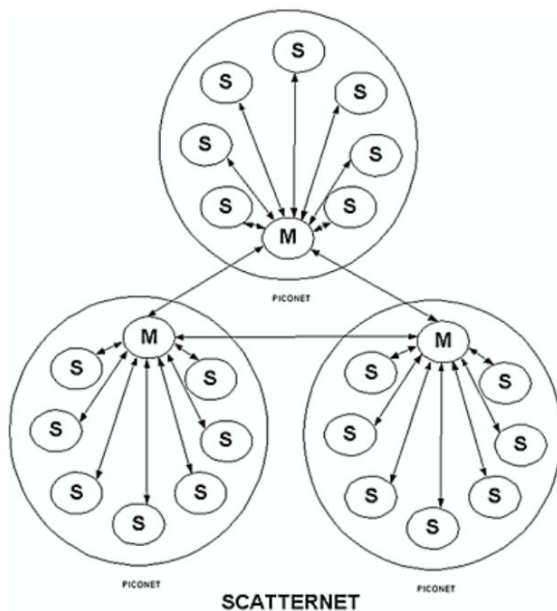
In 1989, the Ericsson company started to work on finding the alternatives to serial communication (RS-232). In 1989 and 1992, Dr. Jaap Haartsen, an employee of Ericsson, has obtained two patents which provided the basis for the technology. In 1996, other companies like Nokia had begun to search for a short-ranged protocol like Bluetooth. These companies have gathered under a group called Bluetooth Special Interest Group (SIG). Today, the trademark and the license of the technology belong to this group. Bluetooth technology is named after the king of Denmark and Norway, Harald Blatand, who lived in the 10th century. Harald had engaged with numerous wars and succeeded to unite Scandinavian tribes. The name Bluetooth was inspired by the interest of the Scandinavian King to a blue fruit, blueberry. Bluetooth was designed to be a universal standard to provide short-range communication between portable or stationary electronic devices. It can also be defined as the technology which made the serial communication wireless because, at the time, the devices had corresponded with each other using serial communication.

¹ Bluetooth Special Interest Group consisted of 5 base companies. These companies are Ericsson, Nokia, Intel, Toshiba, and IBM. Apple and Microsoft have joined them later. SIG is composed of 32302 members, 594 partners, and 7 supportive companies. It is free of charge to join SIG and you may join on www.bluetooth.com. If you have a desire to be a part of the working groups or to set the specifications, you can upgrade your membership to associate membership with a small charge of fee. You may visit the website for further information about associate membership or you may be informed of the Bluetooth protocol, blogs and news by obtaining normal membership.

Bluetooth, which works over the 2.4 GHz frequency band, is designed to communicate wirelessly on three power classes; short distance (10 – 100 cm), standard distance (10m), and long distance (100m) (Sridhar, 2008).

Although Bluetooth started to play a crucial role in our daily lives, there are still specific areas that are unknown to the users. One of these is the security risk consisting of vulnerabilities in the technology (Laurie, Holtmann, & Herfurt, 2006).

Bluetooth technology has the structure of a parent device and seven child devices connected to it. This structure is called PicoNet. Child devices can be located from the parent device maximum 10 meters away. PicoNet structures can unite and form Scatternet. Parent device can communicate with child devices; however, it is impossible to talk directly to child devices.



Bluetooth Security

Bluetooth protocol defines three security models. First security model does not come with any security implementation. None of the required steps for self-protection of devices are applied. Second security model provides service level security. In other words, even though an application that uses the secure service while communicating is somewhat protected, there is no extra precaution about the device security. With the third security model, since connection level security is applied, and it is safe against some unauthorized access methods. Every Bluetooth service has a security mode in its roots, and there are three services to provide security. While some services use authorization and authentication, others only use

authentication, or they may be open to all devices. There are two different security models on devices using this protocol. These are trusted devices and untrusted devices.

In short, Bluetooth provides security on three different layers; security modes by protocol, authorization/authentication by applications and services, and trust model by devices.

Attack Methods

There's a significant attack vector to Bluetooth protocol and devices using it. There are many attack vectors, originating from the aforementioned vector, with colorful names like Bluesnarf, BlueBump, BlueJacking, BlueSmack, Bluestabbing, and so on. All these attacks attempt to have unauthorized access to the victim's cell phone, using a vulnerability in Bluetooth. That special access makes many attacks possible like data exchange, device control, expanding authority, attacking other devices, and removing evidence. We usually think that the range of Bluetooth devices is short. However, it is possible to increase the range up to 1500 meters with the use of Bluetooth technology and high-gain antennas, and even more with the development of Bluetooth 5.

Attackers' general practice is generating unexpected results by transferring erroneous files. When a system receives an unanticipated incorrect file, and if the determined security level is low, the system either goes into an unstable state or collapses. Attackers benefit from this and may perform various attacks on the vulnerable devices.

After such attacks, the list of actions that can be taken is pretty broad. To illustrate; texting and calling on behalf of you; viewing, changing, and updating your files; interfering in private life by creating sensitive files such as pictures, videos, and sounds; data theft, and stealing financial values related to it; converting your device into an assailant device. In short, an attacker may perform any action a core authorized (root) account can access on that particular device.

This is usually because the Bluetooth and similar chips are directly connected to the main chips and the lack of authorization restrictions on the main chip itself. Namely, whenever Bluetooth without any restrictions has a vulnerability, it becomes the target of the attackers.

Blueborne

In September 2017, the Blueborne vulnerability proved once again how frustrating the Bluetooth technology could be. With these attacks, remote code execution was possible. In other words, the interaction between the owner and the device is exposed to the attacker.

Then the question is, what is the difference between Blueborne and other attacks? Attacks mentioned above are applied with user permission like a file transfer request, any connection request, trust to device request, etc. and it means the end-users always know about it. If they accept such requests, they become vulnerable to attacks. Blueborne has shown us the impacts of remotely exploited vulnerabilities and other similar vulnerabilities.

Remotely Exploited Vulnerabilities

Remotely exploited vulnerabilities are those that can be used without any user interaction. There are three cornerstone rules of this method:

1. Does not require any human activity for exploitation.
2. Does not make any complicated assumptions on the active system state.
3. Shall leave the system stable after exploitation.

All in all, it seems possible to perform all the mentioned attacks when you are completely unaware.

If we go back to the Blueborne matter; an attacker can sneakily access a person's or the masses' devices from kilometers away with a Bluetooth attack and manipulate them in line with his plans.

Bluetooth Vulnerabilities

I will speak about the attack vectors I mentioned above, starting with explanations of the concepts.

Bluesnarf

Perhaps the most famous attack method. OBEX (object exchange) protocol is designed to gather job cards and other objects at the application layer of Bluetooth. Bluesnarf attack can gain access to public files with the OBEX GET request. If the victim's Bluetooth driver software is misconfigured, the attacker has access privilege to all files in the device using GET request. In most cases, this service does not require authentication, and that is why anyone can use it.

Bluesnarf++

This attack is similar to Bluesnarf, but the main difference is the way of accessing the file system. If an FTP (File Transfer Protocol) server is run on OBEX, a connection can establish with this service without any pairing due to OBEX Push service. They obtain read and write authorization without the need to authenticate and pair.

Bluebug

This vulnerability causes the attacker to execute unauthorized processes on a Bluetooth device. Some devices can accept the AT (Ascii Terminal) commands used by Bluetooth technology, over Bluetooth. That is why the attacker controls the device using AT commands. The attacker, who succeeds to capture Bluetooth protocol, is free to use any configuration on the device and has full control of the device.

BluePrinting

This method allows capturing information like the brand and model of the device. First three digits of Bluetooth MAC address tells some information about the device and its manufacturer. Apart from that, applications, open ports, the device's brand, model or even Bluetooth software version can be accessed. With such detailed information, the attack vector can be narrowed down.

HelloMoto

This attack exploits the vulnerability found on some devices of Motorola's mismanagement of "trusted devices." Using the OBEX Push service, the attacker starts to send vCards (the virtual structure that holds personal information). The attacker then cancels the dispatch and creates an unsuccessful transmission. This unsuccessful transmission does not remove the attacker from the trusted list. Thus, the attacker can connect to the headset profile without authentication. After the connection is established, the device can be controlled with the AT commands.

BlueBump

This attack requires a little bit of social engineering. The idea is to maintain a secure connection with the victim. It is possible via a file transfer session or sharing a virtual business card. If the victim adds you to the trusted devices list after the transmission, the attacker requests from the victim to delete the connection key without disconnecting. The victim removes the connection key, and is unaware of the attacker's present connection and continues to do ordinary work. Meanwhile, the attacker requests to regenerate the key using

his ongoing connection. As a result, the attacker's device rejoins the victim's trusted device list without authentication and can access the victim's cell phone until the key is disabled.

BlueDump

In this attack, the attacker must know the addresses paired with the Bluetooth device (BDADDR). The attacker connects to the victim by replacing his address with an address the victim is connected to. of the device connected by the victim and connects to it. Since the attacker does not have any connection key, whenever the attacker wants to connect the victim's device, it'll give out the 'no connection key' error (HCI_Link_Key_Request_Native_Reply). In some cases, it makes the victim's device to delete the connection key and enters into pairing mode again. The attacker can read the key change event of a device in pairing mode. In this case, the attacker can remove the trusted device from the list and have the right to make a connection. He also has a chance to carry out a man in the middle attack by getting involved with key change procedure.

BlueChop

The purpose of this attack is to break the Piconet network for the devices connected to Scatternet. The attack benefits from the device's ability to connect multiple devices and constitute an extensive network (Scatternet). The attacker changes his address with a device connected to Piconet and makes a connection with the parent device, destroying the Piconet connection.

Authentication Abuse

Authentication is necessary for all devices to use the service. In other words, a device cannot use a service without authentication and connection to it. Devices, connected to the service provider device, can use all the unauthorized services. In this attack, the attacker tries to connect to unauthorized services to abuse them.

BlueSmack

BlueSmack is a DoS attack, and it can be created using Linux BlueZ layer. It is possible to request an Echo request from another Bluetooth device at L2CAP (application/service) layer. Similar to ICMP ping logic, the purpose of L2CAP ping is to control the connection and calculate the duration of the send-receive period. Due to l2ping coming with BlueZ, the attacker can carry an attack by changing the size of the packets (600-byte size is ideal with the -s parameter).

BlueBorne

Using the vulnerabilities found on the Bluetooth stack, BlueBorne provides connection to other devices without notifying them and runs commands in the device with maximum authority. As a result, all operations can be made by running commands on the device (listening, data changing, reading, tracing). This is caused by the Bluetooth chip's ability to connect to the main chip without being subjected to security check and having all the authorities at the same time.

Car Whisperer

In this attack, the attackers use the default PIN codes in car radios. Devices are connected to the cars by mimicking a phone. After hooking up, they can play sound from the music system found in the car and listen to the microphone. With an attacking antenna, a vehicle going at 120 km/h can be exposed to an attack for 15 seconds from a distance of 500 meters.

Bluestabbing

Just like Bluesmack, this attack is also a type of DoS attack. The intent is to generate an error on the Bluetooth device or to crash it. The Bluetooth device name is encoded with UTF-8. However, not all devices that establish a Bluetooth connection support UTF-8. For this reason, while attempting to list the Bluetooth devices which include UTF-8 characters, those that do not support the encoding crash and restart the device. The attacker can prompt an error on the devices and may put devices into an unstable state.

Bluespoofing

The attacker copies a device found in the trusted device list in the victim's phone. Device address, service records, and similar data are copied. It simulates and copies protocols and profiles and imitates the copied device. It turns encryption off and tries to reconnect without any password. On this account, it is probable to connect to the victim, who accepts not encrypted connections, as another device.

Bloover

Bloover is a Java/Bada application where attack vectors above can be tested. It supports bad object creation attacks as well as Bluebug, Hellomoto, Bluesnarf. It is used to test the Bluetooth protocol.

BlueStalker

It is a commercial follow-up service. It identifies the phone number through Bluebug SMS and intercepts the confirmation message. It performs GSM location tracking.

Bloonix

It is a Linux distribution for testing Bluetooth devices. It is a live version, has 2.6 kernel, contains the latest BlueZ protocol, generates a report, and stores tools for testing Bluetooth connection and device configuration (a kind of Bluetooth attack test kit).

Blusniping and BlueToone are the names given to ways of increasing communication distance of Bluetooth devices by attaching directional antennas to Bluetooth transmitters and receivers. With this directional antenna, a Bluetooth connection can be made from up to 4 kilometers away.

Other Practices

There are a few other methods that are not mentioned here but can be examined online. Some of these are listed below.

Bluehacking, Marphing, Bluejacking, Bluesnafting, Bluetoothing...

Future Of Attacks

Bluetooth technology has an extensive structure. Like most standards used today, it starts with a structure similar to the IEEE OSI Reference model. Mainly (physically), the protocol stack begins with radio and baseband and ends with the application layer at the top. The only difference of the protocol stack, which is similar to the TCP/IP architecture, is that unlike similar protocols (using TCP/IP stack in application layer) Bluetooth uses the broad protocol and application range that SIG has defined for high-level applications. There are some protocols that have not been elaborated in detail in this defined application. There are a lot of apps that have to be tested, a lot of protocols, and lots of internal/intermediate communication that has to be studied because of their lack of being adhered to the standards. That is why it is vulnerable to attacks. The specification for the definitions of the protocols is 2822 pages long. For this reason, researchers did not conduct detailed research on the Bluetooth protocol while investigating other high-level protocols in detail. The other protocols (according to Bluetooth) are easier to understand, and the more complex nature of Bluetooth has kept it away from these studies. For these reasons, it is possible to find a lot of unexplored

attack methods on Bluetooth.

Attack Protection

To protect yourself from the attacks I mentioned before, do not leave the Bluetooth connection on and do not accept unreliable Bluetooth connections. Keep your Bluetooth software up-to-date and remove unused Bluetooth devices from your list of trusted devices. These methods will protect you from most of the attacks coming via Bluetooth but don't forget that there is no perfect security, and that attack vectors will continue to evolve as technology improves.

“



First we thought the PC was a calculator.
Then we found out how to turn numbers into letters with ASCII and we thought it was a typewriter.
Then we discovered graphics, and we thought it was a television.
With the World Wide Web, we've realized it's a brochure.

Douglas Adams
1952 - 2001

”