Topics in Network Security: CLI Email Client with Antivirus

Shahar Bashari, 203428131

In this project, I chose to implement a CLI email client, with antivirus capabilities.

The client can send and receive emails through Gmail / Outlook inboxes.

In case a message is to be sent / received with an attachment, prior to the action's invocation (sending or saving to the disk), the client will run it through the extensive analyzation of VirusTotal.

Should the attachment be found clean, well, good – it will be downloaded or sent according to plan.

However, if it is found infected, the action will cease, and the user will be prompted to view the full report that denied the request.

CLI Email Client Demo:

Now, it is time to show the client's capabilities.

First, I will send a regular old email to my personal account, with a clean attachment (clean.txt which contains the string "123").

CLI Email Client Demo:

```python
def cmd_send(cmd):
    args = parse_send(cmd)
    msg = MIMEMultipart()
    msg['Subject'] = args.subject
    msg['From'] = _user
    msg['To'] = args.recipient
    msg['Date'] = formatdate(localtime=True)
    msg.attach(MIMEText(args.body))
    if args.attachment_path is not None:
        with open(args.attachment_path, 'rb') as f:
            f = f.read()
            if vt_scan(f) == -1:
                return
            attachment = MIMEApplication(f, Name=basename(args.attachment_path))
        attachment['Content-Disposition'] = 'attachment; filename="%s"' % basename(args.attachment_path)
        msg.attach(attachment)

    server = smtplib.SMTP_SSL(_smtp_ssl_host, _smtp_ssl_port)
    if login(server) == -1:
        return
    if send(server, args, msg) == -1:
        return
    server.quit()
```

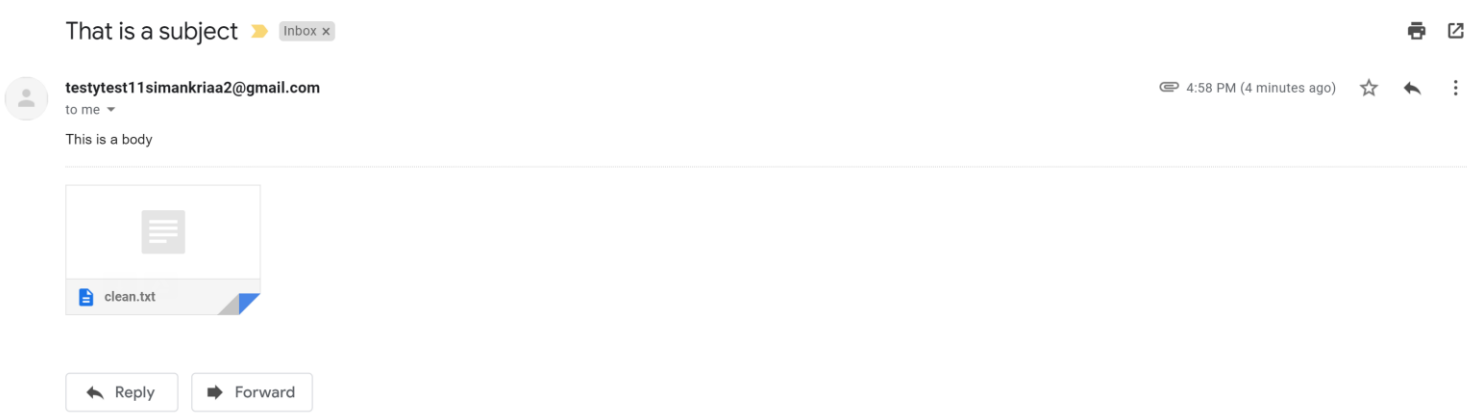
```python
def login(server):
    print("Attempting to login...", end=' ')
    try:
        server.login(_user, _password)
        cprint("Login successful!", Fore.GREEN)
    except Exception as e:
        cprint("Login failed :(", Fore.RED)
        cprint(str(e), Fore.RED)
        return -1
    return 0


def send(server, args, msg):
    print('Sending "{}" to "{}" with attachment "{}"...'
          .format(args.subject, args.recipient, args.attachment_path), end=' ')
    try:
        server.sendmail(_user, args.recipient, msg.as_string())
        cprint("Send successful!", Fore.GREEN)
    except Exception as e:
        cprint("Send failed :(", Fore.RED)
        cprint(str(e), Fore.RED)
        return -1
    return 0
```

cmd_send

CLI Email Client Demo:

```
C:\Users\shaha\PycharmProjects\MiniNetSe2\venv\Scripts\python.exe C:/Users/shaha/PycharmProjects/MiniNetSe2/email_client.py -usr [redacted] -pwd [redacted] -api [redacted]
<NetworkSecurityMini192> send -r shahar.bashari@gmail.com -s "That is a subject" -b "This is a body" -a clean.txt
Scanning file via VirusTotal.com... Clean!
Attempting to login... Login successful!
Sending "That is a subject" to "shahar.bashari@gmail.com" with attachment "clean.txt"... Send successful!
```

Send message with clean file



Message was received

CLI Email Client Demo:

Next, I will try to send an infected file to my personal email and will be stopped by VT's scan.

CLI Email Client Demo:

```python
def vt_scan(f):
    f_md5 = hashlib.md5(f).hexdigest()
    print("Scanning file via VirusTotal.com...", end=' ')
    try:
        file_report = _virus_total.get_file_report(f_md5)
        if file_report['results']['positives'] > 0:
            cprint("Infected :(", Fore.RED)
            print("Dump results (y/n)?", end=' ')
            dump = input('')
            if dump == 'y' or dump == 'yes':
                print(json.dumps(file_report, sort_keys=False, indent=4))
            return -1
    except Exception as e:
        cprint("Scan failed :(", Fore.RED)
        cprint(str(e), Fore.RED)
        return -1
    cprint("Clean!", Fore.GREEN)
    return 0
```

vt_scan

CLI Email Client Demo:

```
<NetworkSecurityMini192> send -r shahar.bashari -s "That is also a subject" -b "Doesnt matter cause it wont be sent" -a infected.txt
Scanning file via VirusTotal.com... Infected :(
Dump results (y/n)?
```


```
Dump results (y/n)? y
{
    "results": {
        "scans": {
            "Bkav": {
                "detected": true,
                "version": "1.3.0.10239",
                "result": "DOS.EiracA.Trojan",
                "update": "20190614"
            },
            "MicroWorld-eScan": {
                "detected": true,
                "version": "14.0.297.0",
                "result": "EICAR-Test-File",
                "update": "20190616"
            },
            "CMC": {
                "detected": true,
                "version": "1.1.0.977",
                "result": "Eicar.test.file",
```

```
    "scan_id": "275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f-1560693858",
    "sha1": "3395856ce81f2b7382dee72602f798b642f14140",
    "resource": "44d88612fea8a8f36de82e1278abb02f",
    "response_code": 1,
    "scan_date": "2019-06-16 14:04:18",
    "permalink": "https://www.virustotal.com/file/275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f/analysis/1560693858/",
    "verbose_msg": "Scan finished, information embedded",
    "total": 63,
    "positives": 61,
    "sha256": "275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f",
    "md5": "44d88612fea8a8f36de82e1278abb02f"
},
"response_code": 200
```


Message not sent, found infected and results were dumped

CLI Email Client Demo:

Now, let’s try to receive a few emails, one with an infected attachment and one with a clean attachment.

CLI Email Client Demo:

```python
def cmd_receive(cmd):
    args = parse_receive(cmd)
    server = imaplib.IMAP4_SSL(_imap_ssl_host, _imap_ssl_port)
    if login(server) == -1:
        return
    result, data = fetch(server, args.filter)
    i = len(data[0].split())

    for x in range(i):
        latest_email_uid = data[0].split()[x]
        result, email_data = server.uid('fetch', latest_email_uid, '(RFC822)')
        raw_email = email_data[0][1]
        raw_email_string = raw_email.decode('utf-8')
        email_message = email.message_from_string(raw_email_string)

        # Header Details
        date_tuple = email.utils.parsedate_tz(email_message['Date'])
        if date_tuple:
            local_date = datetime.datetime.fromtimestamp(email.utils.mktime_tz(date_tuple))
            local_message_date = "%s" % (str(local_date.strftime("%a, %d %b %Y %H:%M:%S")))
        email_from = str(email.header.make_header(email.header.decode_header(email_message['From'])))
        email_to = str(email.header.make_header(email.header.decode_header(email_message['To'])))
        subject = str(email.header.make_header(email.header.decode_header(email_message['Subject'])))
        print(f"\t{Fore.CYAN}From:{Style.RESET_ALL}      %s\n\t{Fore.CYAN}To:{Style.RESET_ALL}        "
              f"%s\n\t{Fore.CYAN}Date:{Style.RESET_ALL}      %s\n\t{Fore.CYAN}Subject:{Style.RESET_ALL}   %s\n"
              % (email_from, email_to, local_message_date, subject))
```

```python
for part in email_message.walk():
    # Attachment details
    fileName = part.get_filename()
    if bool(fileName) and args.attachment:
        if not os.path.isdir(os.path.join(os.getcwd(), "downloads")):
            os.mkdir(os.path.join(os.getcwd(), "downloads"))
        filePath = os.path.join(os.getcwd(), "downloads", fileName)
        downloaded = part.get_payload(decode=True)
        if vt_scan(downloaded) != -1:
            fp = open(filePath, 'wb')
            fp.write(downloaded)
            fp.close()
            print('Downloaded {color}"{file}"{reset} from email titled '
                  '{color}"{subject}"{reset} with UID {color}{uid}{reset}.'
                  .format(color=Fore.CYAN, reset=Style.RESET_ALL, file=fileName,
                          subject=subject, uid=latest_email_uid.decode('utf-8')))
        else:
            continue
    if part.get_content_type() == "text/plain":
        if part.get('Content-Disposition') is not None:
            continue
        # Body details
        if args.dump_body:
            body = part.get_payload(decode=True)
            print(f"\t{Fore.CYAN}Body:{Style.RESET_ALL}\n    %s\n"
                  % (body.decode('utf-8')))
        else:
            print("\n")
```

cmd_receive

CLI Email Client Demo:

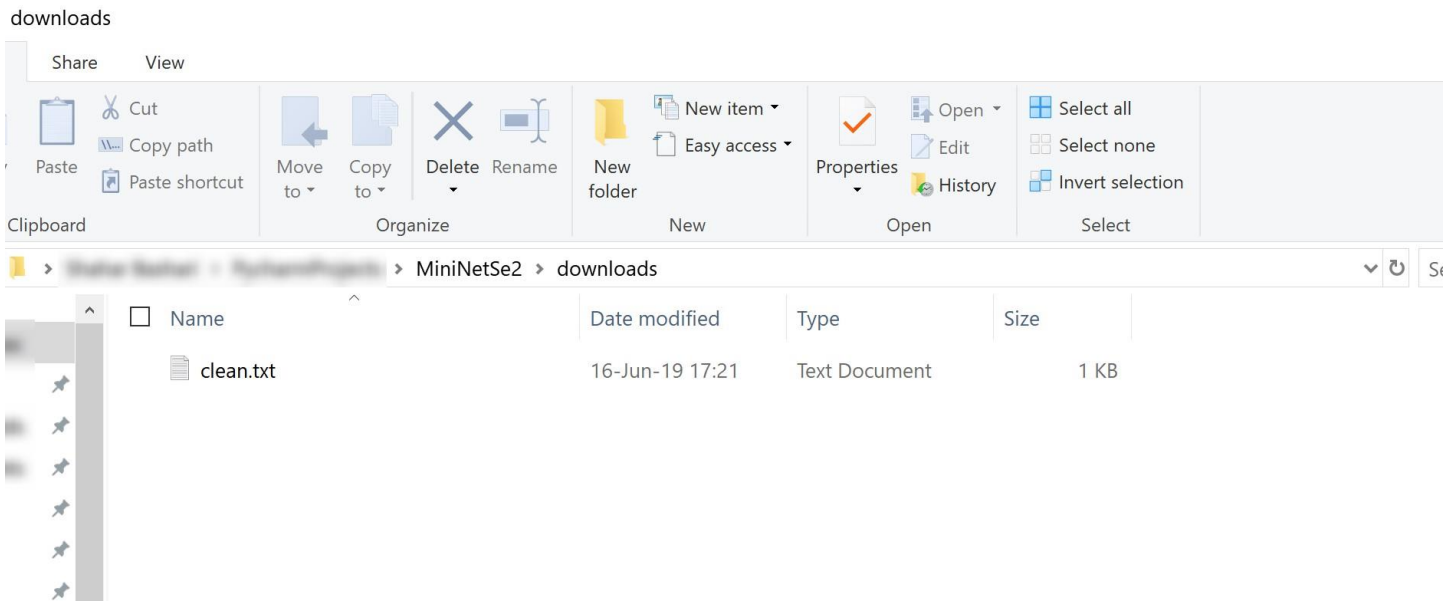
```
<NetworkSecurityMini192> receive -d -f ALL -a
Attempting to login... Login successful!
Fetching emails from inbox... Fetch successful!
    From:      Shahar Bashari <shahar.bashari@gmail.com>
    To:        testytest11simankriaa2@gmail.com
    Date:      Sun, 16 Jun 2019 09:23:57
    Subject:   testsubject1

    Body:
    testbody1


    From:      Shahar Bashari <shahar.bashari@gmail.com>
    To:        testytest11simankriaa2@gmail.com
    Date:      Sun, 16 Jun 2019 09:24:24
    Subject:   subject2
```

Fetching all messages in the INBOX

```
    From:      Shahar Bashari <shahar.bashari@gmail.com>
    To:        testytest11simankriaa2@gmail.com
    Date:      Sun, 16 Jun 2019 11:55:17
    Subject:   attached file subject

    Body:
    attached file body


Scanning file via VirusTotal.com... Clean!
Downloaded "clean.txt" from email titled "attached file subject" with UID 4.
    From:      Shahar Bashari <shahar.bashari@gmail.com>
    To:        testytest11simankriaa2@gmail.com
    Date:      Sun, 16 Jun 2019 16:21:29
    Subject:   This is a virus!

    Body:
    hey


Scanning file via VirusTotal.com... Infected :(
Dump results (y/n)? n
```

clean.txt was downloaded, but infected.rar was caught

CLI Email Client Demo:



Only clean.txt in the downloads folder

Full Capabilities of the Client:

```
[illegible] MiniNetSe2/email_client.py -h
usage: email_client.py [-h] [-hst {GMAIL,HOTMAIL}] -usr USER -pwd PASSWORD
                       -api VT_API_KEY

optional arguments:
  -h, --help            show this help message and exit
  -hst {GMAIL,HOTMAIL}, --host {GMAIL,HOTMAIL}
                        Email Host (GMAIL/HOTMAIL)
  -usr USER, --user USER
                        Email Username (e.g username@gmail.com)
  -pwd PASSWORD, --password PASSWORD
                        Email Password
  -api VT_API_KEY, --virus_total_api_key VT_API_KEY
                        VirusTotal.com Public API Key
```

Full Capabilities of the Client:

```
<NetworkSecurityMini192> help
     send
           usage: send -r RECIPIENT [-s SUBJECT] [-b BODY] [-a ATTACHMENT_PATH]
            optional arguments:
              -r RECIPIENT, --recipient RECIPIENT                      Recipient of this email (e.g. jdoe@gmail.com)
              -s SUBJECT, --subject SUBJECT                            Subject of the message
              -b BODY, --body BODY                                     Body of the message
              -a ATTACHMENT_PATH, --attachment_path ATTACHMENT_PATH    Path to Attached File
    receive
            usage: receive [-d] [-f {ALL,UNSEEN}]
            optional arguments:
              -d, --dump_body                                          Dump messages with bodies to stdout
              -f {ALL,UNSEEN}, --filter {ALL,UNSEEN}                   Get ALL/UNSEEN emails
              -a, --attachment                                         Download attachment if available
    exit
            usage: exit
```

In our days, the amount of spam and malicious messages that we receive through our emails nowadays is enormous.

A tool to help us to filter out infected files can help us ease our minds when doing everyday tasks that involve email.

The choice to send the files to be inspected at VirusTotal was made to ensure the files are 100% clean, as this tool makes use of most of the antivirus services out there.

If even one of them returns a positive result on a file, my client will reject it.

Development Tools & Dependencies:

- Python 3.7 with modules:
  - smtplib (send)
  - imaplib (receive)
  - emails.mime
  - virus_total_apis
  - colorama, argparse, etc.

Link to the project's GitHub repository