

Thadomal Shahani Engineering College
Bandra (W.), Mumbai - 400 050.

© CERTIFICATE ©

Certify that Mr./Miss ADITYA APANDIKAR
of IT Department, Semester IV with
Roll No. 05 has completed a course of the necessary
experiments in the subject Cloud Security Lab under my
supervision in the **Thadomal Shahani Engineering College**
Laboratory in the year 2023 - 2024


Teacher In-Charge

Head of the Department

Date _____

Principal

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	Breaking shift cipher & mono-alphabetic substitution cipher using frequency analysis.	1	21/7/23	
2.	Cryptanalysis or decoding of polyalphabetic cipher ; Playfair, Vigenere ciphers.	7	28/7/23	
3.	Block cipher modes of operation using AES.	11	18/8/23	
4.	Implementation & analysis of RSA crypto system & Digital signature scheme using RSA.	19	28/7/23	
5.	To explore hashdeep tool in Kali Linux for generating, matching & auditing hash of files.	23	11/8/23	
6.	Study use of network reconnaissance tools like WHOIS, dig, traceroute, netlookup, dnsquery to gather information about network & domain registers.	29	4/8/23	Start from 10/7/23
7.	Study of packet sniffer tools, Wireshark & TCDUMP.	39	18/8/23	
8.	Installation of NMAP & using it with different options to scan open ports, perform OS fingerprinting, ping scan, TCP, UDP, etc.	49	1/9/23	
9.	Simulate DDoS attack using Hping3	55	8/9/23	
10.	To study & configure firewalls	57	6/10/23	
11.	Installing snort, setting intrusion, detection mode & writing rules for intrusion detection.	60	25/8/23	

CONTENTS

Thadomal Shahani Engineering College

Department of Information Technology

LABORATORY ASSIGNMENT LIST

Class: T.E.

Semester: V

Year: 2023 -2024

Subject: Security Lab (ITL502)

Practical: 2 Hrs/Week

Lab Outcomes:

- LO 1: Illustrate symmetric cryptography by implementing classical ciphers.
- LO 2: Demonstrate Key management, distribution and user authentication.
- LO 3: Explore the different network reconnaissance tools to gather information about networks.
- LO 4: Use tools like sniffers, port scanners and other related tools for analyzing packets in a network.
- LO 5: Use open-source tools to scan the network for vulnerabilities and simulate attacks.
- LO 6: Demonstrate the network security system using open source tools.

Lab Requirements:

Entire lab sessions will be accomplished with the help of following software / Tools

Windows or Linux/Ubuntu Desktop OS

Open source tools like Wireshark, TCPDUMP, ARPWATCH, Kismet, NetStumbler, NMAP, GPG, Hashdeep, nikto etc.

Sr. No.	Name of Experiment	LO
1	Breaking shift cipher and Mono-alphabetic Substitution Cipher using Frequency analysis method.	LO1
2	Cryptanalysis or decoding of polyalphabetic ciphers: Playfair, Vigenere cipher.	LO1
3	Block cipher modes of operation using Advanced Encryption Standard (AES)	LO2
4	Implementation and analysis of RSA cryptosystem and Digital signature scheme using RSA.	LO2
5	To explore hashdeep tool in kali linux for generating, matching and auditing hash of files.	LO2
6	Study the use of network reconnaissance tools like WHOIS, dig, traceroute, nslookup, nikto, dmitry to gather information about networks and domain registrars.	LO3
7	Study of packet sniffer tools Wireshark and TCPDUMP.	LO3
8	Installation of NMAP and using it with different options to scan open ports, perform OS fingerprinting, ping scan, TCP port scan, UDP port scan, etc.	LO4
9	Simulate DOS attack using Hping3.	LO5
10	To study and configure Firewalls using IP tables.	LO6
11	Installing snort, setting in Intrusion Detection Mode and writing rules for Intrusion Detection.	LO6
12	Explore the GPG tool of linux to implement email security.	LO6

Dr. Shachi Natu
(Subject Teacher)

Prof. Nagaveni Hebbar
(Subject Teacher)

Lab Assignment 1

Aim: Breaking shift cipher and Mono-alphabetic Substitution Cipher using Frequency analysis method.

Lab outcome: Illustrate symmetric cryptography by implementing classical ciphers

What is shift cipher?

A shift cipher, also known as a Caesar cipher, is a basic and straightforward encryption technique used in cryptography. It is one of the earliest and simplest methods of encryption. In a shift cipher, each letter in the plaintext is shifted a fixed number of positions down or up the alphabet. The number of positions shifted is called the "key" or "shift value." For example, with a shift of 3, "A" would be encrypted as "D," "B" would become "E," and so on. The shift wraps around the alphabet, so "Z" would be encrypted as "C" with a shift of 3.

The screenshot shows a web browser window with the URL <http://cse29-iitb.vlabs.ac.in/exp/shift-cipher/simulation.html>. The page title is "Breaking the Shift Cipher".

PART I: Decrypt the following ciphertext. You can use the tool beneath in PART III to simulate the Shift cipher.

Ciphertext to be decrypted:
heahicr he khdu

PART II: Do your rough work here:

heahicr he khdu - shift: 0
ezxajlo gq iact - shift: 1
tychho fy ihs - shift: 2
exwem ex heam - shift: 3
dowmifn du gdzu - shift: 4
cwecem cv fcvn - shift: 5
bawdfl loj egzo - shift: 6
attack at down - shift: 7

PART III:

Plaintext: attack at down shift: 7
 Encrypt Decrypt
Ciphertext: heahicr he khdu

PART IV:

Enter your solution Plaintext and shift key here:
attack at down Key: 7

CORRECT!!

How and why it can be broken using brute force attack?

A shift cipher (Caesar cipher) can be broken using a brute force attack due to its limited key space. Since there are only 26 possible keys in the case of the English alphabet (shifting 0 to

25 positions), an attacker can systematically try all possible keys to decrypt the ciphertext. Here's how a brute force attack works on a shift cipher: Enumerate all possible keys:

Since there are only 26 possible keys, the attacker tries all combinations, shifting the ciphertext by 0 to 25 positions.

Decrypt the ciphertext: For each key, the attacker shifts the ciphertext letters back by the corresponding number of positions to recover the plaintext.

Check for meaningful results: After decrypting with each key, the attacker examines the output to see if it resembles meaningful English text. The assumption is that the correct key will produce a message that is easily recognizable as a coherent sentence or phrase.

Verify the result: The attacker usually employs some form of automated or manual analysis to check the output for linguistic patterns, common words, or other recognizable features that indicate a successful decryption.

Select the correct key: Once the attacker finds the decryption that appears to be meaningful English text, they have likely found the correct key, and the message is deciphered.

The reason a shift cipher is vulnerable to brute force attacks is that it has a very small key space. With only 26 possible keys, trying all combinations is a relatively trivial task for a computer program or a determined attacker. This simplicity is why the shift cipher is considered weak and is not used for secure encryption purposes today.

The screenshot shows a web browser window titled "Virtual Labs" with the URL "cse29-iithlabs.ac.in/exp/shift-cipher/simulation.html". The page is titled "Breaking the Shift Cipher". It contains instructions: "Decrypt the following ciphertext. You can use the tool beneath in PART III to simulate the Shift cipher." Below this, under "PART I", there is a text input field containing the ciphertext "vhb srufxslah lv xqghu vkh vkhbwv" and a button labeled "Next Ciphertext". Under "PART II", there is a text area containing the following text:
Do your rough work here:
vhb srufxslah lv xqghu vkh vkhbwv - shift 0
vzb rotewckeg kv nafet vzb ulqewv - shift 1
uaf qapavxofit vzbfa vzb taffut - shift 2
the porcupine is under the sheets - shift 3

Aditya Apandkar / Roll No : 5 / T11

PART III

Plaintext:
the porcupine is under the sheets

shift: 3

Ciphertext:
wch scrufxslooh lv xqphu wch vkhbwv

PART IV

Enter your solution Plaintext and shift key here:
the porcupine is under the sheets

Key: 3

CORRECT!

What is monoalphabetic cipher?

A monoalphabetic cipher is a type of substitution cipher where each letter in the plaintext is replaced with a fixed corresponding letter in the ciphertext. In other words, the same substitution rule is applied consistently throughout the entire message. This means that each occurrence of a specific letter in the plaintext will always be replaced by the same letter in the ciphertext.

Virtual Labs

cse29-iith.vlabs.ac.in/exp/substitution-cipher/simulation.html

Gmail YouTube Google Groups - In... Introduction to Law... Online Courses - I... Adityapandkar Find icons with the...

Breaking the Mono-alphabetic Substitution Cipher

PART I

Decrypt the following cipher text. A tool to simulate the Mono-Alphabetic Substitution cipher is provided beneath for your assistance.

Here is the table of frequencies of English alphabets for your reference:

a	b	c	d	e	f	g	h	i	j	k	l	m
8.167	1.49	2.782	4.253	12.702	2.228	2.015	6.094	6.966	0.153	0.772	4.025	2.406
n	o	p	q	r	s	t	u	v	w	x	y	z
6.749	7.507	1.929	0.095	5.987	6.327	9.056	2.758	0.978	2.360	0.150	1.974	0.074

decypher 1 - qeqgt vkr hccowv keur: xuadlr un cehrq mowoup et vkr
hccowxto gavk khr manvrb, gkrt nkr twmdnr x vxuoxtp, diawkrq glown
hccowv gavk x yedev gavdk hit yxnv, nkr leuwegn wv (qeqgt x hccowv keur
gkrt nlgptub nkr lxxun x wotp gxb ve x dlhwein lxxu gavk fxtb uedrq
qeqgt el xun manen, nkr lxtgn x nfxau orb ve x qeqh vee nfxau leh khr
ve lxxv, clv vkhspk gavdk nkr nrrn xt xvvhxdwsr pxharts, nkr vkr

Ciphertext Frequencies:

a	b	c	d	e	f	g	h	i	j	k	l	m
0.000	1.037	2.282	3.942	8.091	1.452	3.112	5.602	2.075	0.000	8.506	1.452	0.415
n	o	p	q	r	s	t	u	v	w	x	y	z
7.469	1.867	1.452	3.32	11.618	0.622	4.979	5.602	9.959	6.639	7.584	0.622	0.000

PART II

Note that the *cipher text* is in lower case and when you replace any character, the final character of replacement, i.e., *plaintext* is changed to upper case automatically in the following scratchpad.

Scratchpad:

CHAPTER 1 - DOWN THE RABBIT HOLE: ALICE IS BORED SITTING ON THE RIVERBANK WITH HER SISTER, WHEN SHE NOTICES A TALKING, CLOTHED WHITE RABBIT WITH A POCKET WATCH RUN PAST. SHE FOLLOWS IT DOWN A RABBIT HOLE. WHEN SUDDENLY SHE FALLS A LONG WAY TO A CURIOUS HALL WITH MANY LOCKED DOORS OF ALL SIZES. SHE FINDS A SMALL KEY TO A DOOR TOO SMALL FOR HER TO FIT, BUT THROUGH WHICH SHE SEES AN ATTRACTIVE GARDEN. SHE THEN DISCOVERS A BOTTLE LABELLED 'DRINK ME', THE CONTENTS OF WHICH CAUSE HER TO SHRINK TOO SMALL TO REACH THE KEY. A CAKE WITH 'EAT ME' ON IT CAUSES HER TO GROW TO SUCH A TREMENDOUS SIZE HER HEAD HITS THE CEILING.

Modify the text above (in scratchpad):

This is case insensitive function and replaces only cipher text (lower case) by plain text (upper case).

Replace cipher character by plaintext character

Use the following function to undo your previous change by giving an uppercase character and a lower case. This is a case sensitive function.

Replace character by character

Your replacement history:

You replaced b by Y You replaced c by B You replaced d by C You replaced e by O You replaced f by M You replaced g by W You replaced h by R You replaced i by U You replaced k by H You replaced l by F You replaced m by Z You replaced n by S You replaced o by K You replaced p by G You replaced q by J

Enter your solution plaintext here:

TO FIT, BUT THROUGH WHICH SHE SEES AN ATTRACTIVE GARDEN. SHE THEN DISCOVERS A BOTTLE LABELLED 'DRINK ME', THE CONTENTS OF WHICH CAUSE HER TO SHRINK TOO SMALL TO REACH THE KEY. A CAKE WITH 'EAT ME' ON IT CAUSES HER TO GROW TO SUCH A TREMENDOUS SIZE HER HEAD HITS THE CEILING.

Solution Key =

CORRECT!!

PART IV

Plaintext:

chapter 1 - down the rabbit hole: alice is bored sitting on the riverbank with her sister, when she

key =

Encrypt Decrypt

Remove Punctuation

Ciphertext:

dcxvrbh 1 - degt vkr hawcav keur; xndre wo gefrha novvvtg et vkr hawrhcteo gauk kthi rawnvnh, gert ncr

How and why it can be broken using brute force attack?

A shift cipher (Caesar cipher) can be broken using a brute force attack due to its limited key space. Since there are only 26 possible keys in the case of the English alphabet (shifting 0 to 25 positions), an attacker can systematically try all possible keys to decrypt the ciphertext.

Here's how a brute force attack works on a shift cipher:

Enumerate all possible keys: Since there are only 26 possible keys, the attacker tries all combinations, shifting the ciphertext by 0 to 25 positions.

Decrypt the ciphertext: For each key, the attacker shifts the ciphertext letters back by the corresponding number of positions to recover the plaintext.

Check for meaningful results: After decrypting with each key, the attacker examines the output to see if it resembles meaningful English text. The assumption is that the correct key will produce a message that is easily recognizable as a coherent sentence or phrase.

Verify the result: The attacker usually employs some form of automated or manual analysis to check the output for linguistic patterns, common words, or other recognizable features that indicate a successful decryption.

Select the correct key: Once the attacker finds the decryption that appears to be meaningful English text, they have likely found the correct key, and the message is deciphered.

The reason a shift cipher is vulnerable to brute force attacks is that it has a very small key space. With only 26 possible keys, trying all combinations is a relatively trivial task for a computer program or a determined attacker. This simplicity is why the shift cipher is considered weak and is not used for secure encryption purposes today.

a	b	c	d	e	f	g	h	i	j	k	l	m
8.167	1.49	2.782	4.253	12.702	2.228	2.015	6.094	6.966	0.153	0.772	4.025	2.406
n	o	p	q	r	s	t	u	v	w	x	y	z
6.749	7.507	1.929	0.095	5.987	6.327	9.056	2.758	0.978	2.360	0.150	1.974	0.074

asibix lldzx kolz o dkzeolid atu zbjzbfm lf bj bz a rwox iajxpobnewp
zdjgbfa a elgac. jex iajxpobnewp ykxzjblfz audibix atu rex audijsz jl
exp lkpofjy bxofjbjt ipbziz, ilddolkfuu rt exp bfarbdjyt jl pdoddxp a
olxd. rxolpx ipshdhu aefat, jex iajxpobnewp jnacx audibix jcaj lfx zbox
ln jex dkzeolid lbox dags exp jnacx audibix jex ljeap zbox lbox dags exp
zalpjap. zex rpwagz lrm jhl audibix npid jex dkzeolid. lfx zbox dagsz

[Next Ciphertext](#)

[Calculate Frequencies in ciphertext](#)

Ciphertext Frequencies:

a	b	c	d	e	f	g	h	i	j	k	l	m
8.084	6.854	0.176	4.394	7.206	4.394	1.757	1.582	2.988	7.909	2.636	7.381	1.582
n	o	p	q	r	s	t	u	v	w	x	y	z
1.582	2.285	8.26	0.000	1.582	0.000	0.703	1.933	0.000	5.097	14.06	0.176	7.381

PART II

Note that the *cipher text is in lower case* and when you replace any character, the final character of replacement, i.e., *plaintext is changed to upper case* automatically in the following scratchpad.

Scratchpad:
ALICE CAME UPON A MUSHROOM AND SITTING ON IT IS A BLUE CATERPILLAR SMOKING A HOOKAH. THE CATERPILLAR QUESTIONS ALICE AND SHE ADMITS TO HER CURRENT IDENTITY CRISIS, COMPOUNDED BY HER INABILITY TO REMEMBER A POEM, BEFORE CRAWLING AWAY. THE CATERPILLAR TELLS ALICE THAT ONE SIDE OF THE MUSHROOM WILL MAKE HER TALLER AND THE OTHER SIDE WILL MAKE HER SHORTER. SHE BREAKS OFF TWO PIECES FROM THE MUSHROOM. ONE SIDE MAKES HER SHRINK SMALLER THAN EVER WHILE ANOTHER CAUSES HER NECK TO GROW HIGH INTO THE TREE WHERE SHE PEGGED HERSELF UP FOR A SNAKE. WITH GREAT EFFORT, ALICE BRINGS HERSELF BACK TO HER USUAL HEIGHT. SHE STUMBLES UPON A SMALL ESTATE AND USES THE MUSHROOM TO REACH A MORE APPROPRIATE HEIGHT.

Modify the text above (in scratchpad):

This is case insensitive function and replaces only cipher text (lower case) by plain text (upper case)

Replace cipher character by plaintext character [Modify](#)

Use the following function to make any wanted exchange by giving a uppercase character and a lower case. This is a case sensitive function.

Replace character by character [Replace these exact characters](#)

Your replacement history:

You replaced a by A

Aditya Apandkar / Roll No : 5 / T11

Enter your solution plaintext here:

ALICE COMES UPON A MUSHROOM AND SITTING ON IT IS A BLUE CATERPILLAR
SHE HAS BEEN DRINKING TOO MUCH PEPPER, WHICH HAS CAUSED
HER CURRENT IDENTITY CRISIS, COMPOUNDED BY HER INABILITY TO REMEMBER A
POEM. BEFORE CRAWLING AWAY, THE CATERPILLAR TELLS ALICE THAT ONE SIDE
OF THE MUSHROOM WILL MAKE HER TALLER AND THE OTHER SIDE WILL MAKE HER

Solution Key = arunommecogwdflolyzlkchstv

CORRECT!!

PART IV

Plaintext

alice comes upon a mushroom and sitting on it is a blue caterpillar smoking a hookah. the caterpillar

key = arunommecogwdflolyzlkchstv

Encrypt Decrypt

Remove Punctuation

Ciphertext

zlkdrwxz kolf a zdewm xriaiix atu kizx jex gkzeplld #
11 pxsie a diox sncoplhxix exbowl

Conclusion

We understood the working of shift and monoalphabetic cipher and successfully implemented the simulation of shift and monoalphabetic cipher using virtual labs.

Lab Assignment 2

Aim: Cryptanalysis or decoding of polyalphabetic ciphers: Playfair, Vigenere cipher.

Lab outcome: Illustrate symmetric cryptography by implementing classical ciphers.

How does vigenere cipher work?

In a Vigenère cipher, the key is a word or phrase known only to the sender and recipient of the message. The key is typically as long as the plaintext message. Each letter in the key corresponds to a shift value that determines the substitution for the corresponding letter in the plaintext.

Here's how the Vigenère cipher works:

Key Repetition: The sender repeats the key to match the length of the plaintext message. For example, if the plaintext is "HELLO" and the key is "KEY," the key is repeated as "KEYKE" to match the length of the plaintext.

Encryption: Each letter of the plaintext is then shifted according to the corresponding letter in the key using the Vigenère square. The Vigenère square is a table that consists of 26 Caesar ciphers (one for each letter of the alphabet). Each row in the square is a Caesar cipher with a shift of 1 position to the right compared to the previous row.

The screenshot shows a web-based tool for decoding Vigenère ciphers. On the left, there's a sidebar with a 'Results' section containing a logo for 'Vigenere' and a note that it automatically decodes Vigenère messages. Below this is an 'Ads by Google' banner with a 'Stop seeing this ad' button. In the main area, there's a large yellow panel titled 'VIGENERE DECODER'. It has a 'VIGENERE CIPHERTEXT' field containing the ciphertext 'yKuhr Qqmiazzk ehowsedkgpyt lIsqz Dokrimxi npbuqnoqieygg jGbym Bmtzvkvrv czszsvbognge'. Below this are 'PARAMETERS' fields for 'PLAINTEXT LANGUAGE' (set to English) and 'ALPHABET' (set to ABCDEFGHIJKLMNOPQRSTUVWXYZ). A '► AUTOMATIC DECRYPTION' button is present. At the bottom, there's a 'DECRIPTION METHOD' section with several radio buttons: 'KNOWING THE KEY/PASSWORD: VIGEN' (selected), 'KNOWING THE KEY-LENGTH/SIZE, NUMBER OF LETTERS: 5', 'KNOWING ONLY A PARTIAL KEY: VIG??', 'KNOWING A PLAINTEXT WORD: CODE', and 'VIGENERE CRYPTANALYSIS (KASISKI'S TEST)'. A final '► DECRYPT' button is at the bottom right.

Explain in brief how kasiki test is used to break the vigenere cipher

The Kasiski test is a technique used to break the Vigenère cipher by exploiting repetitions in the ciphertext that result from repeated occurrences of the same keyword or key segments. It is an example of a cryptanalysis method that can help determine the key length used in the Vigenère cipher.

Aditya Apandkar / Roll no : 05 / T11

Here's a brief explanation of how the Kasiski test works:

Find Repeated Substrings: The first step is to search for repeated substrings in the ciphertext.

Since the Vigenère cipher uses a repeating keyword, if the same plaintext segment is encrypted using the same part of the keyword, it will result in the same ciphertext segment. By identifying these repetitions, we can deduce potential key lengths.

Measure the Gaps: Once repeated substrings are found, measure the distances (gaps) between them. The gaps correspond to multiples of the key length.

Find Common Factors: Look for common factors among the measured gaps. These common factors are likely to be the lengths of the repeating segments of the key.

Test Potential Key Lengths: Use the identified potential key lengths to divide the ciphertext into groups. Each group represents characters encrypted using the same part of the key. Frequency Analysis on Groups: Analyze each group separately using frequency analysis. Since each group is encrypted using a simple Caesar cipher, the same frequency analysis methods applied to a monoalphabetic cipher can be used here.

Deduce Key: With enough characters analyzed, the most likely letters for each group can be identified. This information can be combined to deduce the full key and decrypt the entire ciphertext.

The screenshot shows the Vigenere Decoder interface. On the left, the 'Results' panel displays a table of Kasiski + IC test results for different key lengths (1 to 14). The table includes columns for 'lett.' (letter) and 't1' (gap). For a key length of 5, there are three occurrences of a gap of 5. On the right, the 'VIGENERE DECODER' panel shows the ciphertext 'yKuhn Qqmiazzk ehowsegdkgpyt lIsqz Dokrimxi npbuqnoqieygg JGbym Bmtzvkv vczszvbogngte'. Below it, the 'PARAMETERS' section is set to 'PLAINTEXT LANGUAGE: English' and 'ALPHABET: ABCDEFGHIJKLMNOPQRSTUVWXYZ'. Under 'AUTOMATIC DECRYPTION', the 'VIGENERE CRYPTANALYSIS (KASISKI'S TEST)' option is selected. At the bottom, the 'DECRYPT' button is visible.

How does playfair cipher work?

The Playfair cipher uses a 5x5 matrix, typically known as the Playfair square, to encrypt pairs of letters (digraphs) from the plaintext. The matrix is filled with a keyword, and any remaining letters of the alphabet are appended in order, excluding duplicates and the letter "J" (to make it a 25-letter matrix).

Aditya Apandkar / Roll no : 05 / T11

Here's how the Playfair cipher works:

Generating the Playfair Square: Choose a keyword without any duplicate letters. For example, "KEYWORD." Construct the 5x5 Playfair square by writing the letters of the keyword (without duplicates) in order, row by row. Fill the remaining spaces in the matrix with the letters of the alphabet, excluding "J." The resulting Playfair square might look like this (using "KEYWORD"):

K E Y W O

R D A B C

F G H I L

M N P Q S

T U V X Z

Handling the Message: Remove any characters other than letters from the plaintext message and convert all letters to uppercase. If there are any consecutive identical letters or "J"s in the message, insert a filler letter (e.g., "X") between them.

Encryption: Divide the message into pairs of letters (digraphs). For each digraph: If the two letters are in the same row of the Playfair square, replace each letter with the letter to its right (wrapping around to the left side if it's at the end of the row). If the two letters are in the same column, replace each letter with the letter below it (wrapping around to the top if it's at the bottom of the column). If the two letters form a rectangle (neither same row nor same column), replace each letter with the letter at the opposite corner of the rectangle.

Ciphertext: The resulting pairs of letters after encryption form the ciphertext message.

PLAYFAIR ENCODER

★ PLAYFAIR PLAIN TEXT [?](#)

```
harry potter and the cursed child
```

★ PLAYFAIR GRID

S	E	C	U	R
I	T	Y	A	B
D	F	G	H	K
L	M	N	O	P
Q	V	W	X	Z

SECURITYABDFGHJKLMNPQVWXZ

★ SHIFT IF SAME ROW Cell on the right → ▾

★ SHIFT IF SAME COLUMN Cell below ↓ ▾

★ ORDER OF LETTER ELSEWHERE Same row as letter 1 first ▾

► ENCRYPT

PLAYFAIR DECODER

★ PLAYFAIR CIPHERTEXT [?](#)

```
OHUZCBLPAVFTUBLGAFCURSECSDAQL
```

★ PLAYFAIR GRID

S	E	C	U	R
I	T	Y	A	B
D	F	G	H	K
L	M	N	O	P
Q	V	W	X	Z

SECURITYABDFGHJKLMNPQVWXZ

★ SHIFT IF SAME ROW Cell on the left ← (Encryption with right cell →) ▾

★ SHIFT IF SAME COLUMN Cell above ↑ (Encryption with below cell ↓) ▾

★ ORDER OF LETTER ELSEWHERE Same row as letter 1 first ▾

► DECRYPT PLAYFAIR

Results

```
HARXRYPOTXTERANDTHECURSEDCHILD
```

Ads by Google

Stop seeing this ad

Why this ad? ⓘ

How Cryptanalysis of the Playfair cipher can be done?

Aditya Apandkar / Roll no : 05 / T11

Cryptanalysis of the Playfair cipher involves attempting to break the encryption and recover the original plaintext without knowing the keyword or the Playfair square. There are several techniques that can be used to cryptanalyze the Playfair cipher:

Frequency Analysis: Similar to other classical ciphers, frequency analysis can be applied to the Playfair cipher. The attacker looks for repeating digraphs in the ciphertext and compares their frequencies. Common digraphs in the language used (e.g., "TH," "EN," "ED") will have higher frequencies. This analysis may help to identify potential matches and make educated guesses about certain parts of the keyword or the Playfair square.

Known Plaintext Attack: If the attacker knows or can guess parts of the plaintext and their corresponding ciphertext, they can analyze these known pairs to deduce parts of the Playfair square or keyword. This information can be used to gain further insights into the encryption process.

Digraph Patterns: The Playfair cipher has specific patterns that occur during encryption. For example, when two letters in a digraph are in the same row or column of the Playfair square, they will encrypt to another digraph with the same property. Identifying these patterns in the ciphertext may help to narrow down the possible Playfair square configurations.

Kasiski Examination: Similar to the Kasiski test for the Vigenère cipher, the Kasiski examination can be applied to the Playfair cipher to identify repeating sequences in the ciphertext. These repetitions could indicate repeated parts of the keyword or Playfair square, aiding in the cryptanalysis process.

Brute Force: As the keyword is relatively short, the Playfair cipher is vulnerable to brute force attacks. The attacker can try all possible keywords, construct the Playfair square, and decrypt the ciphertext until a meaningful plaintext is obtained.
Repeated Digraph Attack: If the same digraph appears multiple times in the ciphertext, it implies that the same plaintext segment was encrypted with the same part of the keyword. By finding these repetitions, the attacker can deduce parts of the keyword and decrypt the message.

Conclusion:

We understood the working of Playfair and Vigenere cipher and successfully implemented the simulation of Playfair and Vigenere cipher using an online tool Dcode.

Lab Assignment No.3

Aim: Block cipher modes of operation using Advanced Encryption Standard (AES).

Lab Outcome Attained: LO2 – Demonstrate Key management, distribution and user authentication.

Theory:

- I. Briefly explain AES algorithm (What type of cipher it is? Number of rounds, key size, block size, operations in each round.)

AES stands for Advanced Encryption Standard, and it is a widely used symmetric encryption algorithm. It is a block cipher, meaning it operates on fixed-size blocks of data. AES replaces the older Data Encryption Standard (DES) due to its more robust security features. AES operates on a fixed-size block of 128 bits (16 bytes) and uses a key of varying lengths: 128, 192, or 256 bits.

The key features of the AES algorithm are as follows:

1. Number of Rounds: The number of rounds in AES varies based on the key size. For AES-128, there are 10 rounds; for AES-192, there are 12 rounds; and for AES-256, there are 14 rounds.
2. Key Size: AES supports three different key lengths: 128 bits, 192 bits, and 256 bits. The key length determines the number of rounds and the complexity of the algorithm.
3. Block Size: The block size in AES is fixed at 128 bits (16 bytes). This means that plaintext data is divided into 16-byte blocks for processing.
4. Operations in Each Round: Each round of AES consists of several operations, including SubBytes, ShiftRows, MixColumns, and AddRoundKey. These operations involve substitution, permutation, and mixing of the data to achieve diffusion and confusion, which enhances the algorithm's security.
 - SubBytes: Non-linear substitution of each byte in the block using a fixed substitution table (S-box).
 - ShiftRows: Shifting the rows of the block to the left by varying offsets.
 - MixColumns: Mixing the columns of the block using a specific matrix multiplication operation.
 - AddRoundKey: XORing the block with a round key derived from the original encryption key.

Overall, AES is designed to provide a high level of security, efficiency, and versatility. Its use of multiple rounds and different operations in each round contributes to its resistance

against various cryptographic attacks. The choice of key size (128, 192, or 256 bits) allows users to balance security and performance based on their specific needs.

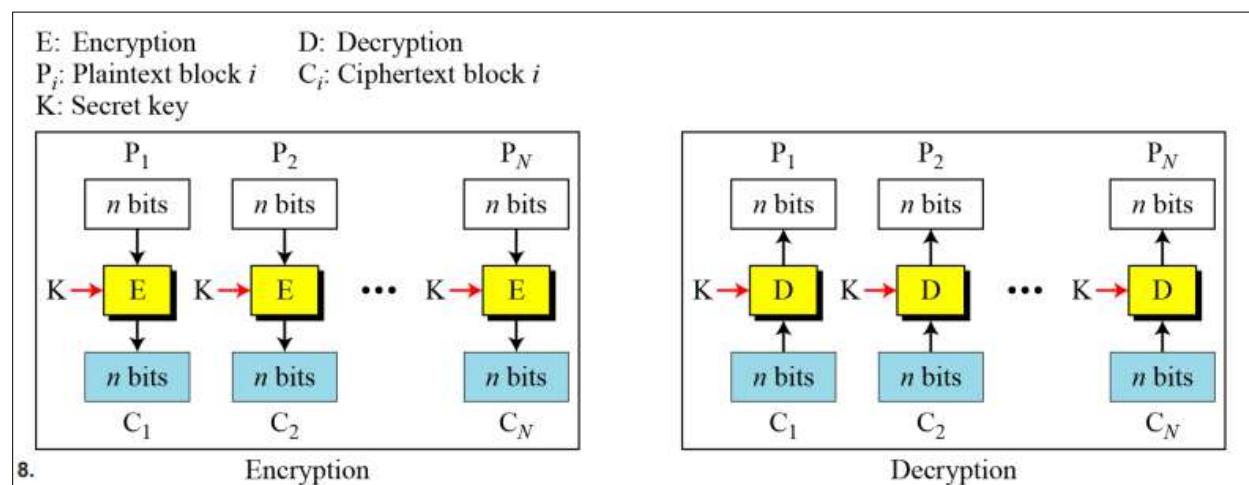
II. With diagram explain in brief block cipher modes of operation

- A. ECB mode
- B. CBC mode
- C. OFB mode
- D. Counter mode

A. ECB mode:

Block cipher modes of operation are techniques used to apply a block cipher, like AES, to encrypt larger amounts of data than a single block. One such mode is the Electronic Codebook (ECB) mode. However, it's worth noting that ECB mode has significant security issues and is not recommended for most practical applications due to its lack of proper diffusion and vulnerability to patterns in the data.

Here's a brief explanation of ECB mode along with a diagram:



ECB Mode (Electronic Codebook Mode):

In ECB mode, each block of plaintext is encrypted independently using the same encryption key. The resulting ciphertext blocks are then combined to create the encrypted message. This means that identical plaintext blocks will result in identical ciphertext blocks, which can leak information about the original data.

In the diagram:

- `P1`, `P2`, `P3`, `P4`, etc. represent consecutive plaintext blocks.
- Each plaintext block is individually encrypted using the same encryption key, resulting in `C1`, `C2`, `C3`, `C4`, etc. ciphertext blocks.

- The ciphertext blocks are combined to form the encrypted message.

Advantages of ECB Mode:

1. Simplicity: Easy to implement and parallelize for fast processing.
2. Low Latency: Minimal delay between block encryption.
3. Error Isolation: Errors in one block don't spread to others.

Disadvantages of ECB Mode:

1. Pattern Leakage: Identical inputs lead to identical outputs, exposing patterns.
2. Weak Security: Vulnerable to various attacks due to lack of diffusion.
3. Predictable: Same input produces same output, allowing attackers to predict.
4. Limited for Large Data: Not well-suited for securing large messages.
5. No Data Integrity: Doesn't ensure if data was modified or tampered.
6. Dictionary Attacks: Repeated blocks lead to repeated ciphertext.
7. No Randomness: Lacks randomness without initialization vector (IV).

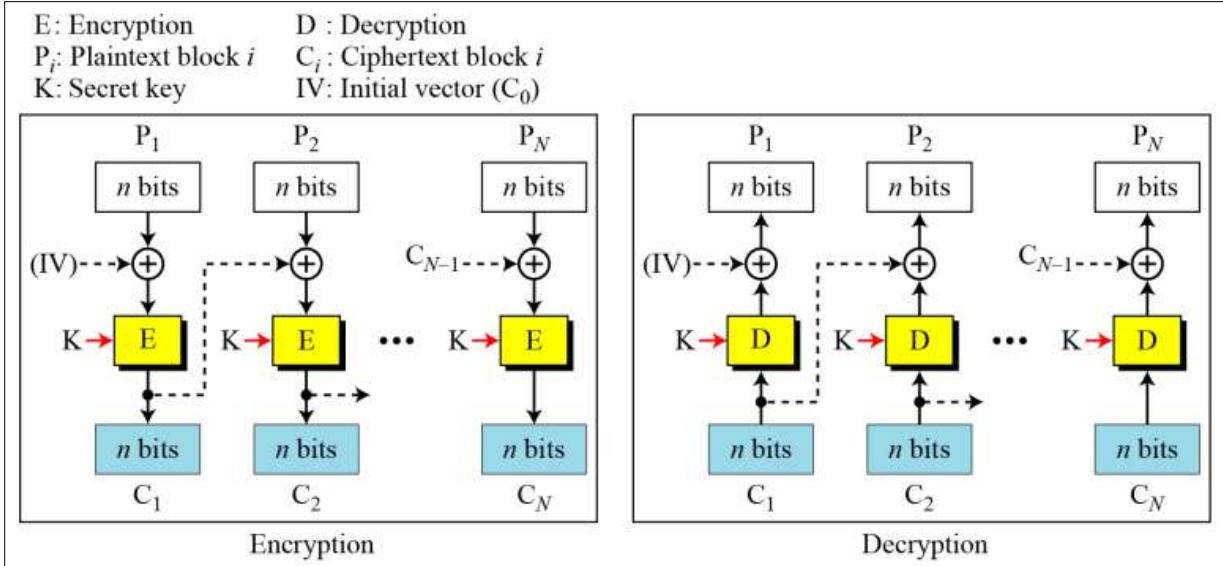
In summary, ECB mode's simplicity comes at the cost of security vulnerabilities. For stronger security, alternative modes like CBC, CFB, OFB, and GCM are preferred.

In summary, while ECB mode has its simplicity and efficiency advantages, its significant security vulnerabilities and inability to provide proper confidentiality and integrity make it unsuitable for most secure applications. For better security, other modes like CBC, CFB, OFB, and GCM are recommended, as they address many of the drawbacks associated with ECB mode.

B. CBC mode:

CBC Mode (Cipher Block Chaining Mode):

In CBC mode, each plaintext block is XORed with the previous ciphertext block before encryption. This introduces a level of diffusion, where the output of one block affects the encryption of the next block. Additionally, an initialization vector (IV) is used to XOR with the first plaintext block to provide randomness to the process.



In the diagram:

- The Initialization Vector (IV) is used to provide randomness to the encryption process. It is XORed with the first plaintext block before encryption.
- For subsequent plaintext blocks, the ciphertext of the previous block is XORed with the current plaintext block before encryption. This introduces diffusion, where the effect of one block spreads to the next.
- Each resulting block is individually encrypted using the encryption key.

Advantages of CBC Mode:

1. Confidentiality: Diffusion introduced by XORing the previous ciphertext block with the current plaintext block enhances confidentiality.
2. Random IV: The use of a random Initialization Vector helps prevent patterns from forming in the ciphertext.
3. Message Integrity: Changes to the ciphertext or IV will disrupt the decryption process, providing a degree of integrity checking.

Disadvantages of CBC Mode:

1. Sequential Processing: CBC mode requires sequential processing, making it less suitable for parallel computing.
2. IV Management: A unique IV must be generated for each message to maintain security, which can be cumbersome.

3. Padding: Irregular plaintext lengths require padding, which can complicate the implementation and introduce vulnerabilities if not done correctly.

4. Error Propagation: Errors in ciphertext block decryption propagate to subsequent blocks, potentially affecting the entire message.

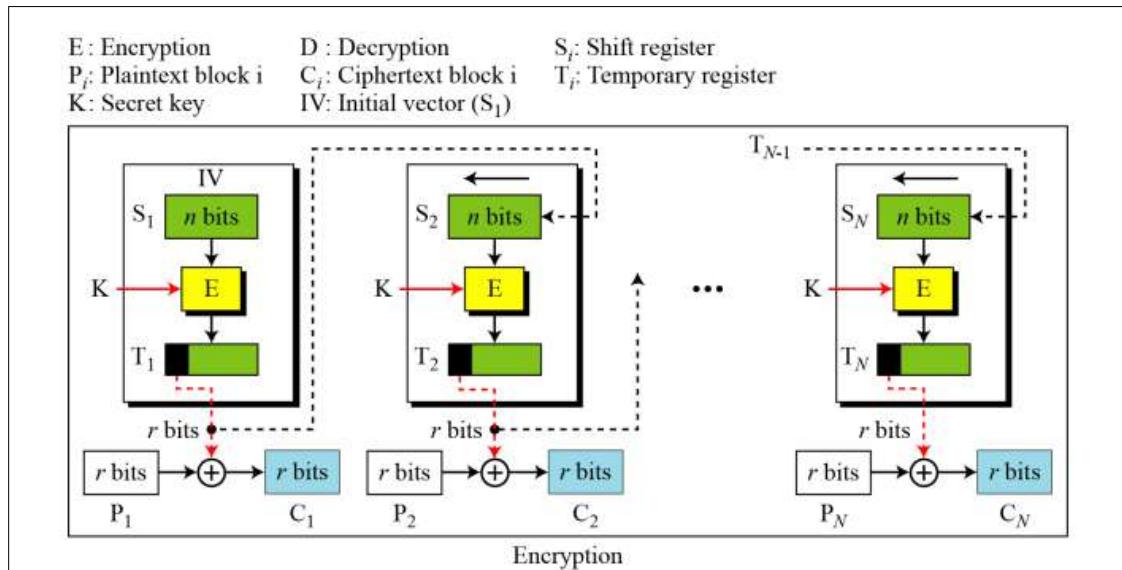
5. Susceptibility to Bit Flipping Attacks: Malicious tampering with the ciphertext can lead to predictable changes in the decrypted plaintext.

In summary, CBC mode improves upon the weaknesses of ECB mode by introducing diffusion and an Initialization Vector, making it a better choice for secure encryption of larger messages. However, it still has certain vulnerabilities that need to be carefully managed.

C. OFB mode:

OFB Mode (Output Feedback Mode):

In OFB mode, a feedback mechanism is employed where the output of the block cipher is fed back into the encryption process to generate a stream of key-like values. These values are then XORed with the plaintext to produce the ciphertext. OFB mode operates in a similar fashion regardless of whether encryption or decryption is being performed.



In the diagram:

- The Initialization Vector (IV) serves as the starting point for the encryption process.
- The Block Cipher (e.g., AES) is used to generate a sequence of key stream values.
- The key stream values are XORed with the plaintext to produce the ciphertext.

Advantages of OFB Mode:

1. Bit-Level Encryption: OFB mode provides bit-level encryption, allowing it to be used for stream ciphers as well as block ciphers.
2. Decoupled Encryption: Encryption and decryption processes are decoupled, allowing flexibility in implementation.
3. Error Propagation: Errors in ciphertext decryption do not affect subsequent blocks.
4. Random Access: Since encryption and decryption are not sequential, random access to the ciphertext is possible.

Disadvantages of OFB Mode:

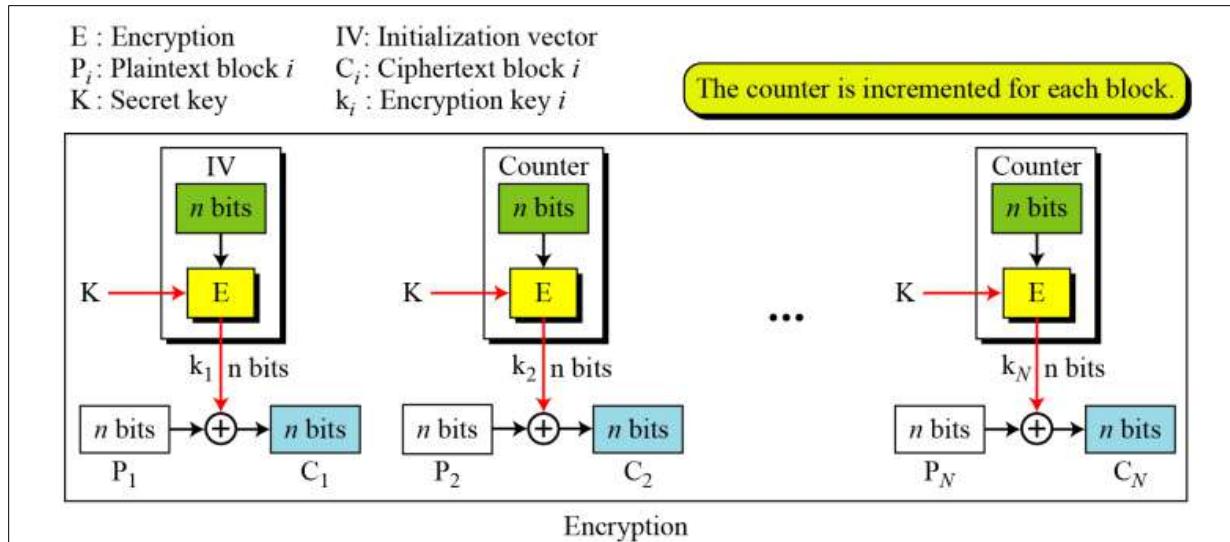
1. Lack of Message Integrity: Like other stream cipher modes, OFB mode doesn't inherently provide data integrity.
2. Key Stream Synchronization: Synchronization between the sender and receiver is critical; a lost or corrupted bit affects subsequent decryption.
3. Block Size Limitation: The key stream is generated in blocks, which can limit its performance for small block sizes.
4. Exposure to Bit Errors: Bit errors in the ciphertext can affect the decryption process and may not be localized.

In summary, OFB mode provides bit-level encryption and error isolation, but it requires careful synchronization between sender and receiver and doesn't inherently ensure data integrity. It's generally more suitable for streaming applications than traditional block cipher modes.

D. Counter mode:

CTR Mode (Counter Mode):

In CTR mode, a counter value is encrypted with a block cipher to generate a stream of pseudorandom key values. These values are then XORed with the plaintext to produce the ciphertext. The counter is incremented for each block, and the encryption of the counter values is done independently.



In the diagram:

- The Nonce is a unique value used as a starting point for the counter.
- The Block Cipher (e.g., AES) encrypts the counter values to generate the key stream.
- The key stream values are XORED with the plaintext to produce the ciphertext.

Advantages of CTR Mode:

1. Parallel Processing: CTR mode can be efficiently parallelized, making it suitable for high-performance scenarios.
2. Random Access: Random access to both encryption and decryption is possible, as counter values can be generated independently.
3. Decoupled Encryption: Encryption and decryption are decoupled, allowing flexibility in implementation.
4. Bit-Level Encryption: CTR mode provides bit-level encryption, allowing it to be used for stream ciphers as well.

Disadvantages of CTR Mode:

1. Nonce Requirement: The Nonce must be unique for each message and must never repeat with the same key.
2. Lack of Message Integrity: Like other stream cipher modes, CTR mode doesn't inherently provide data integrity.
3. Synchronization: Counter synchronization between sender and receiver is crucial; any mismatch results in errors.

4. Bit Errors: Bit errors in the ciphertext can affect decryption, potentially impacting multiple blocks.

In summary, CTR mode offers parallel processing and random access, but careful management of nonces and counter synchronization is necessary. It's well-suited for scenarios requiring high performance and supports both block and stream cipher functionality.

Output Screenshots:

The screenshot shows a web-based simulation for AES and Modes of Operation. The URL is https://cse29-liith.vlabs.ac.in/exp/aes/simulation.html. The interface is divided into several sections:

- PART I:** A dropdown menu labeled "Choose your mode of operation: [Electronic Code Book (ECB)]".
- PART II:** A section for key management:
 - "Key size in bits: [128]" dropdown.
 - A text area containing a 16x4 grid of hex values:

9ab54a21	c7017eab	e9ff20fa	70245cb0
9d41bd49	b5b359ba	0d1daa83	84e7e4e6
f6e2352d	2135d294	c2575113	a0ae0394
c657f73c	7ff2e403	463de1e5	c14bb016
2238659e	b4197de3	e4cfafcc5	ea39614e
 - "Plaintext:" input field with value "b03e3490 346f7b45 1504499a 50007406".
 - "Key:" input field with value "b03e3490 346f7b45 1504499a 50007406".
 - "Next Plaintext" and "Next Keytext" buttons.
- PART IV:** A section for encryption and decryption:
 - "Key in hex:" input field with value "b03e3490 346f7b45 1504499a 50007406".
 - "Plaintext in hex:" input field with value "223865de b4197de3 e4cfafcc5 ea39614e".
 - "Ciphertext in hex:" input field with value "f995ed58 889b7272 ee0aec22 b68498ad".
 - "Encrypt", "Decrypt", and "Clear" buttons.
- PART V:** An answer entry section:
 - "Enter your answer here:" input field with value "f2783708 ed578ab2 11bc47bd 8a65613a 805496d1 f995ed58 889b7272 ee0aec22 b68498ad".
 - "Check Answer!" button.

Aditya Apandkar / Roll no : 05 / T11

PART I
Choose your mode of operation: [Cipher Block Chaining]

PART II
Key size in bits: 128
Plaintext: 9ab54a21 c7017a4b nc9f28fa 70245cb0
9d41bd99 b5b359ba 0d3da03 8fe7e4e6
IV: 515675ef 250881bc 135cca3c b160c3f6

PART III
Calculate XOR:
XOR: 0714f7f8 72b223f1 e1a28279 f4c3b856

PART IV
Key in hex: b03e3490 346f7b45 1504499a 50007406
Plaintext in hex: 223885de b4197de3 e4cfac65 ea39614e
Ciphertext in hex: f995ed58 889b7272 ee0aec2 b684913d
[Encrypt] [Decrypt] [Clear]

PART V
Enter your answer here:
j2783708 ed578ab2 11bc47bd 8a65613a 805496d1 f995ed58 889b7272 ee0aec2 b68498ad [Check Answer]

Conclusion: In conclusion, this assignment aimed to explore block cipher modes of operation using Advanced Encryption Standard (AES), comprehending their features and implications for data security and integrity.

Lab Assignment 4

Aim: Implementation and analysis of RSA cryptosystem and Digital signature scheme using RSA

Lab Outcome: Implementation and analysis of RSA cryptosystem and Digital signature scheme using RSA

RSA key generation steps

Select Two Large Prime Numbers: Choose two distinct prime numbers, typically denoted as "p" and "q." These prime numbers should be large enough to provide security against factorization attacks. The security of RSA relies on the difficulty of factoring the product of these two primes.

Calculate the Modulus: Compute the modulus "n" by multiplying the two prime numbers: $n = p * q$. The modulus is used in both the public and private keys.

Calculate Euler's Totient Function: Calculate Euler's totient function (ϕ) of n. For two prime numbers, $\phi(n) = (p - 1) * (q - 1)$. Euler's totient function is important for ensuring the correctness of the RSA algorithm.

Choose the Public Exponent: Select a small odd integer "e" (usually prime) as the public exponent. Common choices include 3, 17, or 65537. The public exponent should be relatively prime to $\phi(n)$, meaning that their greatest common divisor (GCD) is 1.

Calculate the Private Exponent: Compute the private exponent "d" such that $(e * d) \% \phi(n) = 1$. This can be achieved using modular multiplicative inverse calculations, often using the extended Euclidean algorithm.

Generate Public Key: The public key consists of the modulus "n" and the public exponent "e." The public key is distributed widely and is used for encryption and verification of digital signatures.

Generate Private Key: The private key consists of the modulus "n" and the private exponent "d." The private key is kept secret and is used for decryption and signing.

Key Length and Security: The security of RSA depends on the length of the prime numbers used. Longer prime numbers result in stronger security, as they increase the difficulty of factorization attacks. Key length is typically measured in bits.

Key Pair Storage: Safely store the private key, as its compromise can lead to unauthorized decryption and impersonation. The public key can be shared openly.

Key Generation Verification: It's important to verify that the generated keys work correctly for encryption, decryption, and digital signatures before using them in production.

Digital Signature Generation

Message Hashing: The message that needs to be signed is first hashed using a cryptographic hash function (such as SHA-256). This produces a fixed-length hash value that uniquely represents the message.

Private Key Signing: The sender uses their private key to encrypt the hash value. This creates the digital signature. The encryption process involves applying a mathematical operation (often modular exponentiation) to the hash value using the private key.

Digital Signature Generation: The encrypted hash value is the digital signature. This signature is unique to the specific message and the sender's private key.

Digital Signature Verification

Message Hashing: The recipient of the signed message hashes the received message using the same cryptographic hash function that was used during signature generation. This produces a hash value.

Signature Decryption: The recipient uses the sender's public key to decrypt the digital signature. This process involves applying a mathematical operation (often modular exponentiation) to the signature using the sender's public key.

Compare Hashes: The recipient compares the decrypted signature (which is the hash value) with the hash value calculated from the received message. If the two hash values match, it indicates that the message has not been altered since it was signed, and the digital signature is valid.

Public Key Authentication: Before proceeding with verification, the recipient needs to ensure that the public key used for verification actually belongs to the sender. This might involve

Aditya Apandkar / Roll no : 05 / T11

checking the authenticity of the public key through a trusted certificate authority (CA) or another secure method.

Validity Checks: Additional checks may be performed to verify the validity of the digital signature, such as verifying the expiration date of the certificate associated with the public key, checking for certificate revocation status, and ensuring that the digital signature algorithm used is secure.

Virtual
Labs

Public-Key Cryptosystems (PKCSv1.5)

Plaintext (string):

Ciphertext (hex):

Decrypted Plaintext (string):

Status:

RSA private key

bits =

Modulus (hex):

Aditya Apandkar / Roll no : 05 / T11

Virtual Labs
An IEEE E-Learning Initiative

Public-Key Cryptosystems (PKCSv1.5)

Public exponent (hex, F4=0x10001):
10001

Private exponent (hex):
7cd1745aec69096129b1f42da52ac9cae0afbbe0bc2ec89253598dcf454960e
3e5e4ec9f8c87202b986601dd167253ee3fb3fa047e14F1dfd5cc37e931b29d

P (hex):
f0e4dd1eac5622bd3932860fc749bbc48662edabdf3d2826059acc0251ac0d3b

Q (hex):
d13cb38fbcd06ee9bca330b4000b3dae5dae12b27e5173e4d888c325cda61ab3

D mod (P-1) (hex):
b3d5571197fc31b0eb6b4153b425e24c033b054d22b9c8282254fe69d8c8c593

D mod (Q-1) (hex):
968ffe89e50d7b72585a79b65cfdb9c1da0963ccb56c3759e57334de5a0ac3f

1/Q mod P (hex):
d9bc4f420e93adad9f007d0e5744c2fe051c9ed9d3c9b65f439a18e13d6e3908

Aditya Apandkar / Roll no : 05 / T11

Digital Signatures Scheme

Digitally sign the plaintext with Hashed RSA.

Plaintext (string):

Hash output(hex):

Input to RSA(hex):

Digital Signature(hex):

Digital Signatures Scheme

Digital Signature(hex):

Digital Signature(base64):

Status:
Time: 1ms

RSA public key

Public exponent (hex, F4=0x10001):

Modulus (hex):

Conclusion:

We understood the generation of RSA key , generation and verification of digital signature and successfully implemented the same in virtual labs

Lab Assignment No.5

Aim: To explore hashdeep tool in kali linux for generating, matching and auditing hash of files.

Lab Outcome Attained: LO2 – Demonstrate Key management, distribution and user authentication.

Theory:

- I. What is the need of hashing? List different hashing algorithms.
Hashing serves several important purposes in computer science and information security. Here are some of the key reasons for using hashing:
 1. Data Integrity: Hashing ensures data integrity by generating a fixed-size hash value (digest) from the input data. Even a small change in the input data will result in a significantly different hash value, making it easy to detect alterations or corruption.
 2. Data Retrieval: Hashing is used in data structures like hash tables, which provide efficient data retrieval and storage. Hash functions help map keys to indexes in an array, enabling quick access to values associated with those keys.
 3. Cryptography: Hashing is a fundamental component of cryptographic algorithms. Hash functions are used in digital signatures, password storage, message authentication, and more. Hashing ensures that sensitive data like passwords is not stored in plaintext, enhancing security.
 4. Checksums: Hashing is used in checksums to verify the integrity of files during transmission. Hash values are compared before and after transmission to check for data corruption.
 5. Data Deduplication: Hashing helps in data deduplication by identifying duplicate data chunks and storing only one copy. This is useful for efficient storage and backup solutions.
 6. Load Balancing: Hashing is employed in load balancing algorithms to distribute incoming requests across multiple servers based on hash values.
 7. Digital Forensics: Hashing is crucial in digital forensics to verify the authenticity of files and evidence during investigations.
 8. Cryptocurrency and Blockchain: Hash functions are a foundational element in blockchain technology, providing the basis for consensus mechanisms and maintaining the integrity of the distributed ledger.

Common hashing algorithms include:

1. MD5 (Message Digest Algorithm 5): Despite its historical use, MD5 is now considered insecure due to vulnerabilities that allow collision attacks.
2. SHA-1 (Secure Hash Algorithm 1): Similar to MD5, SHA-1 is no longer considered secure due to vulnerabilities.

3. SHA-256 (Secure Hash Algorithm 256): Part of the SHA-2 family, SHA-256 is widely used for cryptographic applications and is considered secure for most purposes as of my knowledge cutoff in September 2021.
4. SHA-3 (Secure Hash Algorithm 3): Introduced as a replacement for SHA-2, SHA-3 is designed to provide enhanced security and resistance to certain types of attacks.
5. Blake2: A cryptographic hash function that offers improved performance and security compared to older algorithms.
6. RIPEMD (RACE Integrity Primitives Evaluation Message Digest): A family of hash functions that come in various bit lengths, offering an alternative to the SHA family.
7. Whirlpool: A cryptographic hash function that is often used in digital signatures and other security applications.
8. Bcrypt: A hash function specifically designed for securely hashing passwords, incorporating a salt and a cost factor to slow down brute-force attacks.

It's important to choose a hashing algorithm based on the specific requirements of your application, especially considering the trade-off between security, performance, and compatibility. Always use the latest and most secure algorithms available.

- II. Write the commands used for generating hash values, matching them with stored hash values and auditing using hashdeep tool.

1. To check the version of Hashdeep - *hashdeep -V*
2. To display help about hashdeep - *hashdeep -h or hashdeep -hh*
3. To display the manual page of hashdeep - *man hashdeep*
4. To display the manual page of any specific hash algorithm supported by hashdeep – *man md5deep*.

By default, hashdeep generates MD5 n SHA256 hash values.

5. To hash a file - *hashdeep filename*
6. If you don't want to display the full path of file in output hash record - *hashdeep -b filename*
7. To suppress any error messages - *hashdeep -s filename*
8. To apply multiple hash algorithms than default - *hashdeep -c md5,sha1,sha256,tiger filename*
9. To hash multiple files (say all text files) using md5 - *hashdeep -c md5 *.txt*

Aditya Apandkar / Roll no : 05 / T11

10. To hash multiple files (say all text files) using md5 and sha1 - *hashdeep -c md5,sha1 *.txt*

11. Hashing block of files - *hashdeep -c md5 -p 100 example.txt*

12. To recursively calculate hash (all files and subdirectories in a specified directory) -
hashdeep c md5 -r /home/shachi/myfiles

13. To redirect the output of md5 hash of files to another file -

*md5deep *.txt>hashset.txt*
*hashdeep *.txt>hashtext1.txt*

Check the content of output file –

cat hashset.txt
cat hashtext1.txt

14. To display output in matching mode -

*md5deep -m hashset.txt **
*hashdeep -m -k hashtext1.txt **

15. To suppress unwanted system msgs/error -

*md5deep -m hashset.txt **
*hashdeep -s -m hashtext1.txt **

No output is displayed if there is no matching hashed file is found.

16. To display all files which are negatively matching use -x option -

*Md5deep -s -x hashset.txt **
*hashdeep -s -x hashtext1.txt **

Forensic auditing can be done using hashdeep tool which means a check to determine if any files in the system are changed due to malware or any normal system operation like update patching.

17. To audit, first create a hashset file and then audit it against the files to be checked if they are modified.

hashdeep -c md5,sha1,sha256 -r /home/shachi/myfiles>hashtext1.txt
hashdeep -a -r -k hashtext1.txt /home/shachi/myfiles

18. Add new file to the directory and audit. It fails.

touch /home/shachi/myfiles/newfile.txt
hashdeep -a -r -k hashtext1.txt /home/shachi/myfiles

19. To get where it failed use the command with -v option

hashdeep -v -a -r -k hashtext1.txt /home/shachi/myfiles

20. Move one of the files to another directory and audit n see output

mv /home/shachi/myfiles/example.txt /tmp
hashdeep -v -a -r -k hashtext1.txt /home/shachi/myfiles

Aditya Apandkar / Roll no : 05 / T11

21. Rename one of the files and audit n see the output

```
mv /home/shachi/myfiles/shachi.txt /home/shachi/myfiles/shachi.bak  
hashdeep -v -a -r -k hashset1.txt /home/shachi/myfiles
```

22. For verbose output of audit

```
hashdeep -vv -a -r -k hashset1.txt /home/shachi/myfiles  
hashdeep -vvv -a -r -k hashset1.txt /home/shachi/myfiles
```

Output Screenshots:

The screenshot shows a terminal window titled "Terminal" with the following content:

```
File Edit View Search Terminal Help  
Fri 11:40  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -V  
4.4  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -h  
hashdeep version 4.4 by Jesse Kornblum and Stinson Garfinkel.  
S hashdeep [OPTION]... [FILE]...  
-c <alg1,[alg2]> - Compute hashes only. Defaults are MD5 and SHA-256  
    legal values: md5,sha1,sha256,tiger,whirlpool,  
-p <size> - piecewise mode. Files are broken into blocks for hashing  
-r - recursive mode. All subdirectories are traversed  
-d - output in DFXML (Digital Forensics XML)  
-k <file> - add a file of known hashes  
-a - audit mode. Validates FILES against known hashes. Requires -k  
-m - matching mode. Requires -k  
-x - negative matching mode. Requires -k  
-w - In -m mode, displays which known file was matched  
-M and -X act like -m and -x, but display hashes of matching files  
-e - compute estimated time remaining for each file  
-s - silent mode. Suppress all error messages  
-b - prints only the bare name of files; all path information is omitted  
-l - print relative paths for filenames  
-I/-T - only process files smaller than the given threshold  
-o - only process certain types of files. See README/manpage  
-v - verbose mode. Use again to be more verbose  
-d - output in DFXML: -W FILE - write to FILE.  
-j <nump> - use num threads (default 8)  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ man hashdeep  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ man md5deep  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep hashset1.txt  
XXXXX HASHDEEP-1.0  
XXXXX size_md5,sha256,filename  
## Invoked from: /home/lab1006  
## $ hashdeep hashset1.txt  
##  
1620,720ace78e0e475f896e6ebf659c67ddc,bbac8d47f085b79e0d7ae1c4cb744a3eba19F51b49c31533f9024f1fc54f6b7e,,/home/lab1006/hashset1.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -b hashset1.txt  
XXXXX HASHDEEP-1.0  
XXXXX size_md5,sha256,filename  
## Invoked from: /home/lab1006  
## $ hashdeep -b hashset1.txt  
##  
1620,720ace78e0e475f896e6ebf659c67ddc,bbac8d47f085b79e0d7ae1c4cb744a3eba19F51b49c31533f9024f1fc54f6b7e,hashset1.txt  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 hashset1.txt  
XXXXX HASHDEEP-1.0  
XXXXX size_md5,filename  
## Invoked from: /home/lab1006  
## $ hashdeep -c md5 -p 100 hashset1.txt  
##  
160_1054499268d57f428b62a5e21ddfd55e,,/home/lab1006/hashset1.txt offset 0-99  
160_c243b51826ae5e5a05536b9f4856rdg,,/home/lab1006/hashset1.txt offset 160-199  
160_d7e81fc4305784019e55e0d40881eb1,,/home/lab1006/hashset1.txt offset 260-299  
160_d5d914ec8e0df69a31c1cc6d268ba57,,/home/lab1006/hashset1.txt offset 360-399  
160_e8f67aa935e1f985492bc7c77bfff97,,/home/lab1006/hashset1.txt offset 460-499  
160_f1142a103d85942c27b76e0d727b6c,,/home/lab1006/hashset1.txt offset 560-599  
160_b0b4193b3c992d17f43d0d45e7a72z,,/home/lab1006/hashset1.txt offset 660-699  
160_0dc78fd518a45c0a493e4d32aa618ac,,/home/lab1006/hashset1.txt offset 760-799  
160_4014ea886214ebd876417c7085366e,,/home/lab1006/hashset1.txt offset 860-899  
160_f095dd3a2cb0d718d9c0609227ab2ee,,/home/lab1006/hashset1.txt offset 960-999  
160_91ecae5a39129e95618144e852392fe,,/home/lab1006/hashset1.txt offset 1600-1699  
160_595ad426113e97fca86f17629fbab787,,/home/lab1006/hashset1.txt offset 1100-1199  
160_aee27ccfae4921a01cd468245fcfcf,,/home/lab1006/hashset1.txt offset 1200-1299  
160_aee08b57fab72aa12c28e7c7eac17f7,,/home/lab1006/hashset1.txt offset 1300-1399  
160_36ed39c88a52ba418fa1eb270567cf7d,,/home/lab1006/hashset1.txt offset 1400-1499  
160_3ab4b595df1e80d6241032be1d114,,/home/lab1006/hashset1.txt offset 1500-1599  
160_ab966371b9329a913935adeb32fbab97,,/home/lab1006/hashset1.txt offset 1600-1699  
160_d4c4e62881c8e7d88641673527367423,,/home/lab1006/hashset1.txt offset 1700-1799  
26_14427522eb1a47e23244f8d6a101d35,,/home/lab1006/hashset1.txt offset 1800-1819  
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep c md5 -r [-]
```

Aditya Apandkar / Roll no : 05 / T11

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c sha1 hashset1.txt
XXXXX HASHDEEP-1.0
XXXXX size,sha1,filename
## Invoked from: /home/lab1006
## $ hashdeep -c sha1 hashset1.txt
##
1820,14e4f4846612d94f5ab12872ac7959bd70f8041e,/home/lab1006/hashset1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c sha256 hashset1.txt
XXXXX HASHDEEP-1.0
XXXXX size,sha256,filename
## Invoked from: /home/lab1006
## $ hashdeep -c sha256 hashset1.txt
##
1820,bbacd8d47f085b79e9d7ae1c4cb4a3e8a19f51b49c31533f9024f1fc54f6b7e,/home/lab1006/hashset1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c tiger hashset1.txt
XXXXX HASHDEEP-1.0
XXXXX size,tiger,filename
## Invoked from: /home/lab1006
## $ hashdeep -c tiger hashset1.txt
##
1820,4f68be70b645a9fb0b5dd4db1e1c51d921bf61e1295dc,/home/lab1006/hashset1.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 *hashset.txt hashset1.txt
XXXXX HASHDEEP-1.0
XXXXX size,md5,filename
## Invoked from: /home/lab1006
## $ hashdeep -c md5 hashset.txt hashset1.txt
##
1820,720ace78e0e475f896e6ebf659c67dc,/home/lab1006/hashset1.txt
1034,0c81874c5f3d8a03e9a899ba7focag7,/home/lab1006/hashset.txt
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5,sha1 *hashset.txt hashset1.txt
XXXXX HASHDEEP-1.0
XXXXX size,md5,sha1,filename
## Invoked from: /home/lab1006
## $ hashdeep -c md5,sha1 hashset.txt hashset1.txt
##
1034,0c81874c5f3d8a03e9a899ba7focag7,/home/lab1006/hashset1.txt
/home/lab1006/c: No such file or directory
/home/lab1006/md5: No such file or directory
XXXXX HASHDEEP-1.0
XXXXX size,md5,sha1,filename
## Invoked from: /home/lab1006
## $ hashdeep -c md5 /home/lab1006/Desktop
##
1820,720ace78e0e475f896e6ebf659c67dc,/home/lab1006/Desktop/hashset.txt
0,d41d8cd98f00b20e9800998ecf8427e,e3b0c4298fc1c149afbf4c899ef592427ae41e4b499f3c4e95991b7852855,/home/lab1006/Desktop/cns/hashset1.txt
4905,9d36f7dd11/9b9982ae42f399151e8d7,0b5bbf53c17e/0e8c9cb45a5017df8ce5b6d8be1ebf97f500504880d25f851b/,/home/lab1006/Desktop/cns/lab4
1828,720ace78e0e475f896e6ebf659c67dc,b0ac8d47f095b79e0d7ae1c4b7443eba19f51b49c31533f9024f1fc54f6b7e,/home/lab1006/Desktop/hashset1.txt
72,79222aefb490d8abccc4694754d785e25,7e8aa1c2edde849676la5f8895c46d78547fedfd6da22debea96f0852sd6,/home/lab1006/Desktop/cns/hashset.txt
20,7d46e232272fdcc852f4469d2f776249,1f49a1cfa8b5119ecbb7c152391d1ad54f67b51a0f289481b736f453bc86b03,/home/lab1006/Desktop/cns/f12
06553,0f10ee3e30b53a343430e15429f014/e5,28309f3cad0814940407010b/fe185bc2e500e90f2/c0540954217a0021f0,/home/lab1006/Desktop/Lab Assignment 5 Hashing.pdf
16,3e069fd4993384927a9865ac532e5f53,05029c9788aa1fb4315b74587b7b50d501568059e9c3226cc9ef284e830974,/home/lab1006/Desktop/cns/f13/f1e3
92267,f40074c6734b5909aee4882cde270fec,a7ef590919414702002f597f16e22e99605c4e10a2c6f2505babc7b8d8cd9ef,/home/lab1006/Desktop/Lab Assignment 7 TCPDUMP.pdf
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep hashset1.txt hashset1.txt hashset2.txt
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ cat hashset2.txt
720ace78e0e475f896e6ebf659c67dc,/home/lab1006/hashset1.txt
d41d8cd98f00b20e9800998ecf8427e,/home/lab1006/hashset.txt
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -m hashset.txt *
md5deep: hashset.txt: Unable to find any hashes in file, skipped.
Try `md5deep --help` for more information.
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -m hashset2.txt *
/home/lab1006/Desktop: Is a directory
/home/lab1006/Documents: Is a directory
/home/lab1006/Downloads: Is a directory
/home/lab1006/Hashset1.txt
/home/lab1006/Hashset.txt
/home/lab1006/Music: Is a directory
/home/lab1006/Pictures: Is a directory
/home/lab1006/Public: Is a directory
/home/lab1006/Templates: Is a directory
/home/lab1006/Videos: Is a directory
```

```
Activities Terminal Fri 11:41
File Edit View Search Terminal Help
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5 -r /home/lab1006/Desktop
/home/lab1006/c: No such file or directory
/home/lab1006/md5: No such file or directory
XXXXX HASHDEEP-1.0
XXXXX size,md5,sha1,filename
## Invoked from: /home/lab1006
## $ hashdeep -c md5 /home/lab1006/Desktop
##
1034,0c81874c5f3d8a03e9a899ba7focag7,/home/lab1006/Desktop/hashset.txt
0,d41d8cd98f00b20e9800998ecf8427e,e3b0c4298fc1c149afbf4c899ef592427ae41e4b499f3c4e95991b7852855,/home/lab1006/Desktop/cns/hashset1.txt
4905,9d36f7dd11/9b9982ae42f399151e8d7,0b5bbf53c17e/0e8c9cb45a5017df8ce5b6d8be1ebf97f500504880d25f851b/,/home/lab1006/Desktop/cns/lab4
1828,720ace78e0e475f896e6ebf659c67dc,b0ac8d47f095b79e0d7ae1c4b7443eba19f51b49c31533f9024f1fc54f6b7e,/home/lab1006/Desktop/hashset1.txt
72,79222aefb490d8abccc4694754d785e25,7e8aa1c2edde849676la5f8895c46d78547fedfd6da22debea96f0852sd6,/home/lab1006/Desktop/cns/hashset.txt
20,7d46e232272fdcc852f4469d2f776249,1f49a1cfa8b5119ecbb7c152391d1ad54f67b51a0f289481b736f453bc86b03,/home/lab1006/Desktop/cns/f12
06553,0f10ee3e30b53a343430e15429f014/e5,28309f3cad0814940407010b/fe185bc2e500e90f2/c0540954217a0021f0,/home/lab1006/Desktop/Lab Assignment 5 Hashing.pdf
16,3e069fd4993384927a9865ac532e5f53,05029c9788aa1fb4315b74587b7b50d501568059e9c3226cc9ef284e830974,/home/lab1006/Desktop/cns/f13/f1e3
92267,f40074c6734b5909aee4882cde270fec,a7ef590919414702002f597f16e22e99605c4e10a2c6f2505babc7b8d8cd9ef,/home/lab1006/Desktop/Lab Assignment 7 TCPDUMP.pdf
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep hashset1.txt hashset1.txt hashset2.txt
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ cat hashset2.txt
720ace78e0e475f896e6ebf659c67dc,/home/lab1006/hashset1.txt
d41d8cd98f00b20e9800998ecf8427e,/home/lab1006/hashset.txt
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -m hashset.txt *
md5deep: hashset.txt: Unable to find any hashes in file, skipped.
Try `md5deep --help` for more information.
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -m hashset2.txt *
/home/lab1006/Desktop: Is a directory
/home/lab1006/Documents: Is a directory
/home/lab1006/Downloads: Is a directory
/home/lab1006/Hashset1.txt
/home/lab1006/Hashset.txt
/home/lab1006/Music: Is a directory
/home/lab1006/Pictures: Is a directory
/home/lab1006/Public: Is a directory
/home/lab1006/Templates: Is a directory
/home/lab1006/Videos: Is a directory
```

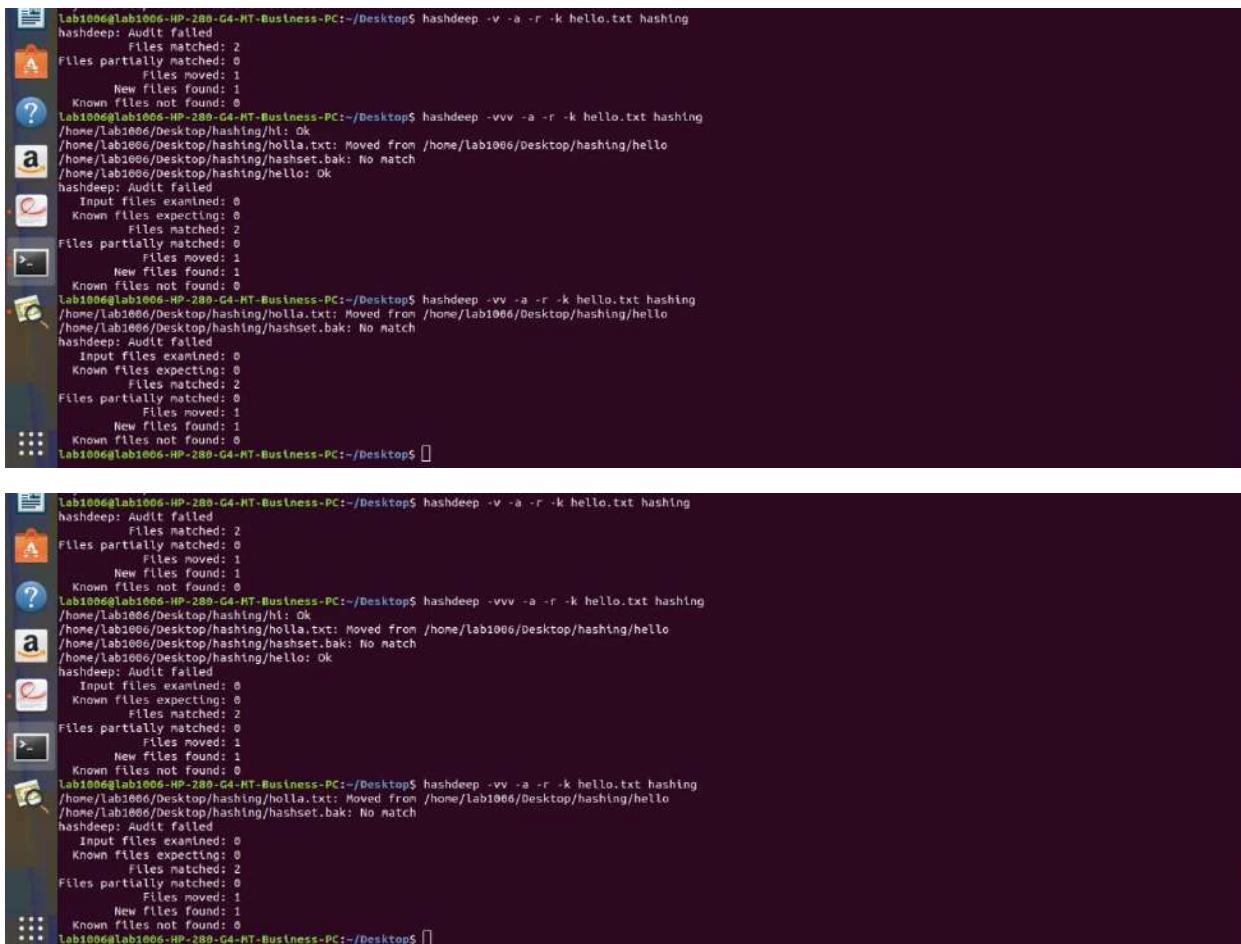
Aditya Apandkar / Roll no : 05 / T11

```
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -s m hashset2.txt *
/home/Lab1006/Desktop/ Is a directory
/home/Lab1006/Documents/ Is a directory
/home/Lab1006/Downloads/ Is a directory
/home/Lab1006/Music/ Is a directory
/home/Lab1006/Pictures/ Is a directory
/home/Lab1006/Public/ Is a directory
/home/Lab1006/Templates/ Is a directory
/home/Lab1006/Videos/ Is a directory
/home/Lab1006/hashset1.txt
/home/Lab1006/hashset2.txt
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -s -m hashset2.txt *
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5Deep -s -x hashset.txt *
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -s -x hashset2.txt *
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -s -x hashset3.txt *
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ md5deep -s -x hashset2.txt *
/home/Lab1006/a1.txt
/home/Lab1006/hashset2.txt
/home/Lab1006/Files3
/home/Lab1006/Lab4
/home/Lab1006/examples.desktop
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -s -x hashset2.txt *
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -s -x hashset3.txt *
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -s -x hashset.txt *
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5,sha1,sha256 -r ./home/Lab1006/hash/hashset3.txt
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -c md5,sha1,sha256 -r ./home/Lab1006/Desktop/Hashing>hello.txt
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -a -r -k hello.txt /home/Lab1006/Desktop/Hashing
hashdeep: Audit passed
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ touch ./home/Lab1006/Desktop/Hashing/holla.txt
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -a -r -k holla.txt /home/Lab1006/Desktop/Hashing
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ touch ./home/Lab1006/Desktop/Hashing/holla.txt
```



```
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -a -r -k hello.txt /home/Lab1006
/home/Lab1006/.mozilla/firefox/sogms0v5.default/lock: No such file or directory
/home/Lab1006/.thunderbird/jikai0k4.default-release/lock: No such file or directory
hashdeep: Audit Failed
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -v -a -r -k hello.txt /home/Lab1006
/home/Lab1006/.mozilla/firefox/sogms0v5.default/lock: No such file or directory
/home/Lab1006/.thunderbird/jikai0k4.default-release/lock: No such file or directory
hashdeep: Audit Failed
    Files matched: 2
    Files moved: 429
    New files Found: 189608
    Known files not Found: 0
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ mv ./home/Lab1006/hashset.txt ./home/Lab1006/Desktop/Hashing
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hashdeep -v -a -r -k hello.txt /home/Lab1006
/home/Lab1006/.mozilla/firefox/sogms0v5.default/lock: No such file or directory
/home/Lab1006/.thunderbird/jikai0k4.default-release/lock: No such file or directory
hashdeep: Audit Failed
    Files matched: 2
    Files moved: 429
    New files Found: 18976
    Known files not Found: 0
```

Aditya Apandkar / Roll no : 05 / T11



```
Lab1006@Lab1006-HP-280-G4-HT-Business-PC:~/Desktop$ hashdeep -v -a -r -k hello.txt hashing
hashdeep: Audit failed
    Files matched: 2
    Files partially matched: 0
    New files found: 1
    Known files not found: 0
Lab1006@Lab1006-HP-280-G4-HT-Business-PC:~/Desktop$ hashdeep -vvv -a -r -k hello.txt hashing
/home/Lab1006/Desktop/Hashing/hi: Ok
/home/Lab1006/Desktop/Hashing/holla.txt: Moved from /home/lab1006/Desktop/Hashing/hello
/home/Lab1006/Desktop/Hashing/hashset.bak: No match
/home/Lab1006/Desktop/Hashing/hello: Ok
hashdeep: Audit failed
    Input files examined: 0
    Known files expecting: 0
    Files matched: 2
    Files partially matched: 0
    Files moved: 1
    New files found: 1
    Known files not found: 0
Lab1006@Lab1006-HP-280-G4-HT-Business-PC:~/Desktop$ hashdeep -vv -a -r -k hello.txt hashing
/home/Lab1006/Desktop/Hashing/holla.txt: Moved from /home/lab1006/Desktop/Hashing/hello
/home/Lab1006/Desktop/Hashing/hashset.bak: No match
hashdeep: Audit failed
    Input files examined: 0
    Known files expecting: 0
    Files matched: 2
    Files partially matched: 0
    Files moved: 1
    New files found: 1
    Known files not found: 0
Lab1006@Lab1006-HP-280-G4-HT-Business-PC:~/Desktop$ hashdeep -v -a -r -k hello.txt hashing
hashdeep: Audit failed
    Files matched: 2
    Files partially matched: 0
    New files found: 1
    Known files not found: 0
Lab1006@Lab1006-HP-280-G4-HT-Business-PC:~/Desktop$ hashdeep -vvv -a -r -k hello.txt hashing
/home/Lab1006/Desktop/Hashing/hi: Ok
/home/Lab1006/Desktop/Hashing/holla.txt: Moved from /home/lab1006/Desktop/Hashing/hello
/home/Lab1006/Desktop/Hashing/hashset.bak: No match
/home/Lab1006/Desktop/Hashing/hello: Ok
hashdeep: Audit failed
    Input files examined: 0
    Known files expecting: 0
    Files matched: 2
    Files partially matched: 0
    Files moved: 1
    New files found: 1
    Known files not found: 0
Lab1006@Lab1006-HP-280-G4-HT-Business-PC:~/Desktop$ hashdeep -vv -a -r -k hello.txt hashing
/home/Lab1006/Desktop/Hashing/holla.txt: Moved from /home/lab1006/Desktop/Hashing/hello
/home/Lab1006/Desktop/Hashing/hashset.bak: No match
hashdeep: Audit failed
    Input files examined: 0
    Known files expecting: 0
    Files matched: 2
    Files partially matched: 0
    Files moved: 1
    New files found: 1
    Known files not found: 0
Lab1006@Lab1006-HP-280-G4-HT-Business-PC:~/Desktop$
```

Conclusion: In this lab assignment, we explored hashdeep in Kali Linux, learning hash generation, matching, and auditing. It ensures data integrity, aids forensics, and verifies changes. Valuable for cybersecurity and digital forensics.

Lab Assignment 6

Aim: Study the use of network reconnaissance tools like WHOIS, dig, traceroute, nslookup to gather information about networks and domain registrars.

Lab Outcome: Explore the different network reconnaissance tools to gather information about networks.

What is the important information that attackers look for using whois command and what attacks can be performed using this information?

The WHOIS command is used to retrieve domain registration information from a domain name registrar's database. While WHOIS data is intended to provide transparency and accountability for domain ownership, it can also be used by attackers to gather information that might be exploited for various malicious purposes. Here are some of the important pieces of information that attackers may look for using the WHOIS command, along with potential attacks that could be facilitated:

1. Domain Owner Information: Attackers may use WHOIS to find out the name, organization, email address, and physical address of the domain owner. This information can be used for social engineering attacks, spear phishing, or targeted cyberattacks.
2. Administrative and Technical Contacts: The contact details for administrative and technical personnel responsible for the domain can provide attackers with potential targets for social engineering and phishing attacks.
3. Domain Expiration Date: Knowing when a domain is set to expire can help attackers time their attacks. If a domain is about to expire and the owner is not paying attention, attackers might attempt to take over the domain or launch phishing campaigns.
4. Name Servers and DNS Information: Attackers might use this information to target the domain's DNS infrastructure and potentially conduct DNS hijacking or distributed denial-of-service (DDoS) attacks.
5. Registrar Information: Knowing the domain's registrar can give attackers insights into the domain's security posture. Some registrars might have weaker security practices than others, making domains registered with them potential targets.

Aditya Apandkar / Roll no : 05 / T11

6. Domain History: Information about domain transfers and changes in ownership can help attackers understand the domain's lifecycle and potentially exploit any gaps in security during these transitions.
7. IP Addresses Associated with the Domain: Attackers can use IP address information to gather potential targets for scanning and attacks, such as identifying potentially vulnerable systems.
8. Email Addresses: Email addresses listed in WHOIS records can be targeted for phishing attacks, spam, or other forms of social engineering.

Potential Attacks Using WHOIS Information:

Phishing Attacks: Attackers can use WHOIS information to craft convincing phishing emails that appear to come from legitimate sources, increasing the chances of luring victims into revealing sensitive information.

Social Engineering: Armed with detailed contact information, attackers can impersonate domain owners, administrators, or technical personnel to gain unauthorized access to systems or extract confidential information.

Domain Hijacking: Exploiting domain expiration dates or vulnerabilities, attackers might attempt to gain control of the domain, redirect traffic, or launch further attacks.

Spear Phishing: With detailed contact information, attackers can create targeted spear-phishing campaigns that are highly convincing and difficult to detect. Identity

Theft: WHOIS data can be used to steal identities or commit fraud by impersonating legitimate individuals or organizations.

Reconnaissance: Attackers can use WHOIS data to gather intelligence about a target's infrastructure, which can help them plan more sophisticated attacks.

To mitigate the risks associated with WHOIS information, domain owners and administrators should consider using domain privacy services or providing minimal contact details in their WHOIS records. Additionally, organizations should be cautious about sharing sensitive information in WHOIS records and should regularly monitor their domains for any unauthorized changes.

How traceroute command works in order to trace the route of given host?

The traceroute command is a network diagnostic tool used to trace the route that packets take from your local machine to a destination host over an IP network, typically the internet. It provides valuable information about the path that data packets follow, including the number of hops (routers) between your machine and the destination. Here's how the traceroute command works:

Packet TTL (Time to Live): When a packet is sent over a network, it has a Time to Live (TTL) value associated with it. This value represents the maximum number of hops (routers) the packet can traverse before being discarded. Each time a router receives a packet, it decrements the TTL by one. If the TTL reaches zero, the router discards the packet and sends an ICMP "Time Exceeded" error message back to the source.

Sending Packets: When you run the traceroute command, it sends a series of packets to the destination host with increasing TTL values, starting from 1. The first packet (TTL = 1) will be discarded by the first router encountered, which will then send an ICMP "Time Exceeded" message back to your machine.

Receiving ICMP Messages: As each router along the path discards the packet due to TTL expiration, it sends an ICMP "Time Exceeded" message back to your machine. This message includes information about the router's IP address.

Hop Identification: The traceroute tool records the IP address of each router that sends back an ICMP "Time Exceeded" message. It then calculates the round-trip time (RTT) for each packet, which gives an indication of the delay between your machine and each router. Mapping the

Route: By repeating the process with increasing TTL values, the traceroute command gradually builds a picture of the route that packets take to reach the destination host. It identifies each hop (router) along the way and displays the IP address and sometimes the hostname of each router, along with the RTT.

Displaying the Results: Once the traceroute command has sent packets with increasing TTL values and received ICMP messages from routers along the path, it displays the gathered information in the terminal. This output shows the sequence of routers and their corresponding IP addresses that the packets traveled through to reach the destination host.

Explain dig command with various options

Aditya Apandkar / Roll no : 05 / T11

The dig command (Domain Information Groper) is a powerful and versatile DNS (Domain Name System) tool used to query DNS servers and retrieve various types of DNS information. It's commonly used for troubleshooting DNS-related issues, analyzing DNS records, and gathering DNS-related data. Here's an explanation of the dig command along with some of its various options:

-h or --help: Displays a help message that provides a brief overview of dig options and usage.

-t type or --type=type: Specifies the type of DNS record to query. For example, -t A queries for IPv4 addresses, -t MX for mail exchange records, etc.

-c class or --class=class: Specifies the DNS class. The default is IN (Internet). Other classes include CH (Chaosnet) and HS (Hesiod).

@server: Specifies the DNS server to query. By default, dig uses the system's configured DNS servers.

-x addr or --xaddr=addr: Reverse DNS lookup. Provide an IP address, and dig will attempt to find the corresponding domain name.

-q name or --question=name: Perform a question/query for the specified domain name. **+short:** Outputs only the answer section of the DNS response, in a simplified format. **+trace:** Traces the delegation path from the root down to the authoritative name servers, showing each step along the way.

+all: Displays the full DNS response, including the question, answer, authority, and additional sections.

+noauthority and +noadditional: Suppresses the display of authority and additional sections in the DNS response.

+stats: Displays query and response statistics, including query time, server used, and query size.

+nocmd: Suppresses the command header in the output.

Explain any two vulnerabilities detected for the website that you have scanned using Which attacks are possible if these vulnerabilities are exploited? Hypothetical Vulnerability 1:

Aditya Apandkar / Roll no : 05 / T11

Cross-Site Scripting (XSS) Description: Cross-Site Scripting (XSS) occurs when a web application allows malicious scripts to be injected into user-generated content. These scripts then execute within the context of a user's browser.

Potential Attack: An attacker could inject a malicious script into a website's comment section. When other users view the comments, the malicious script executes in their browsers, stealing their session cookies. The attacker can then use these stolen cookies to impersonate users and gain unauthorized access to their accounts.

Hypothetical Vulnerability 2:

SQL Injection (SQLi) Description: SQL Injection is a vulnerability that allows attackers to manipulate an application's database by injecting malicious SQL queries through user inputs.

Potential Attack: An attacker could exploit a SQL Injection vulnerability in a login form. By inputting specially crafted SQL code, they could bypass authentication and gain unauthorized access to user accounts. Additionally, they could manipulate the database to extract, modify, or delete sensitive information.

It's important to note that these examples are purely hypothetical and provided for educational purposes. Actual vulnerabilities and their potential consequences can vary greatly based on the specific circumstances and security measures in place. Organizations should prioritize security measures, conduct regular security assessments, and follow best practices to mitigate these and other potential vulnerabilities.

What are different functionalities provided by dimtry. Write Dmitry command for whois lookup, an IP whois lookup, retrieve Netcraft info, search for subdomains, search for email addresses, do a TCP port scan, and save the output to example.txt for the domain example.com

"Dmitry" (Deepmagic Information Gathering Tool) is a command-line tool used for reconnaissance and information gathering. It provides various functionalities to gather information about a target domain. Here are some of the functionalities provided by Dmitry, along with example commands:

WHOIS Lookup: dmitry -
winse example.com

Aditya Apandkar / Roll no : 05 / T11

IP WHOIS Lookup: dmitry -

wips example.com Retrieve

Netcraft Info: dmitry -wnet

example.com Search for

Subdomains: dmitry -ws

example.com Search for

Email Addresses: dmitry -

we example.com TCP Port

Scan:

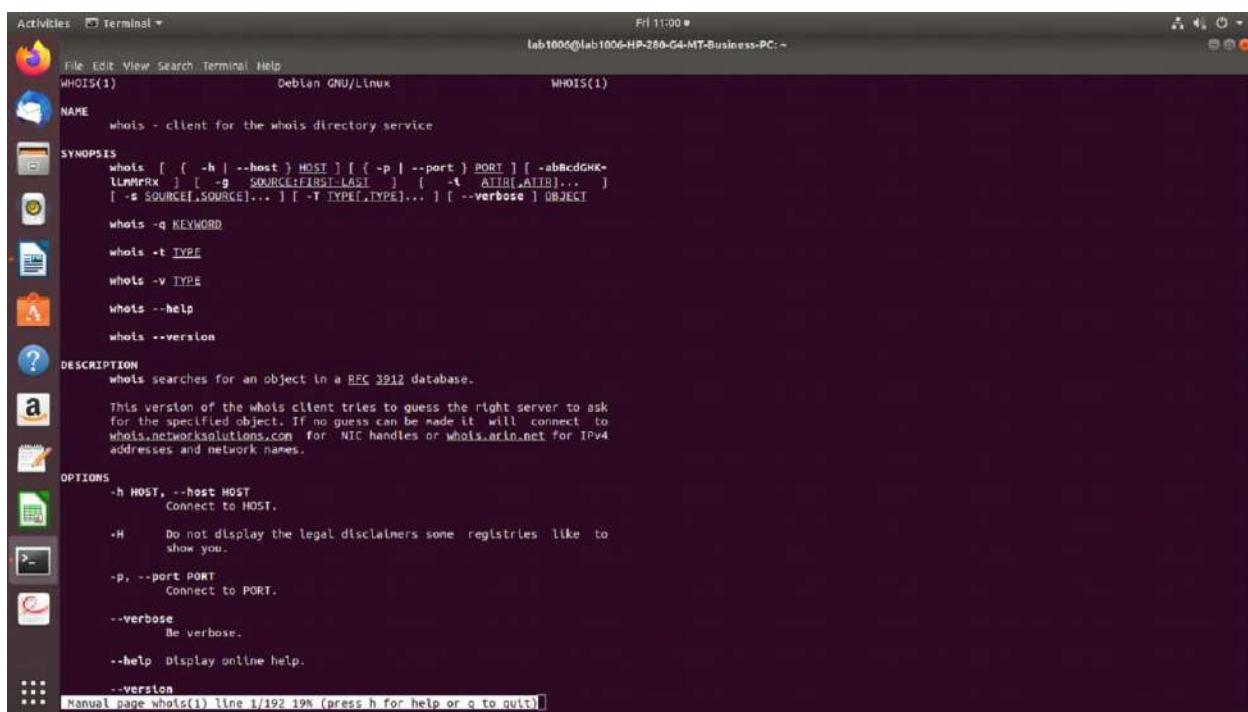
dmitry -wt example.com Save Output to File: To save the output to a file named "example.txt" for the domain "example.com," you can use output redirection.

For example, for a WHOIS lookup:

dmitry -winse example.com >

example.txt

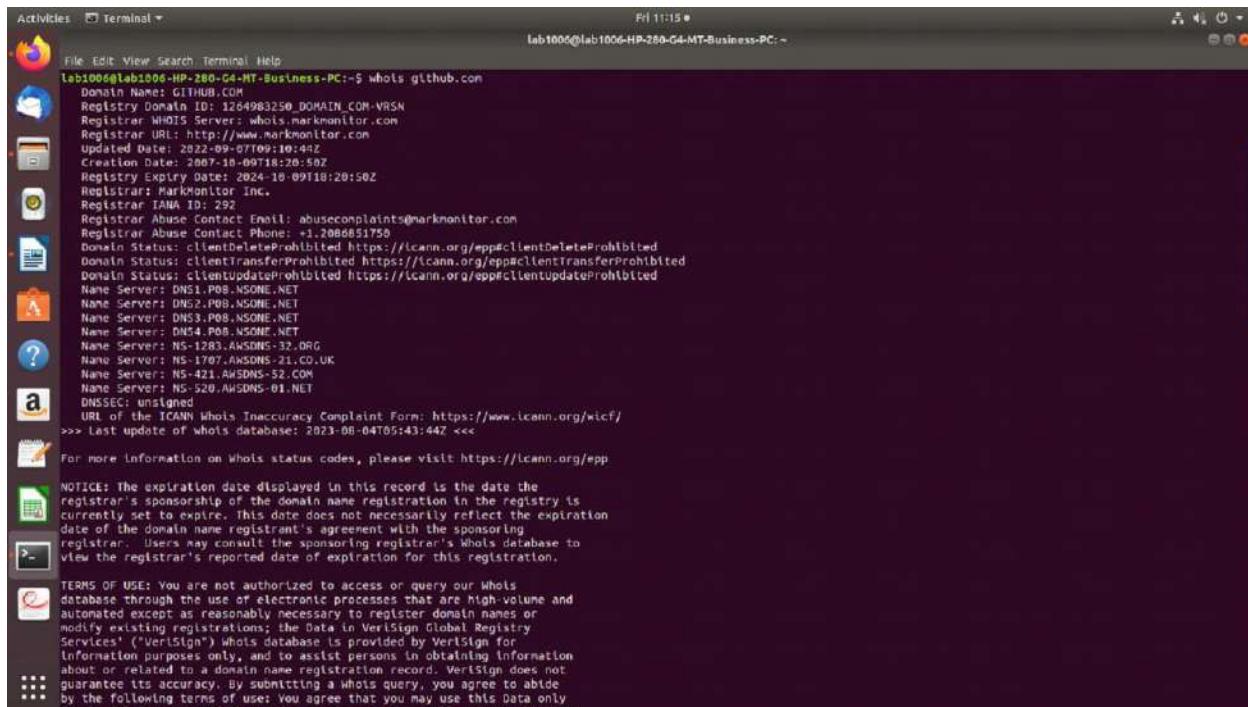
Remember to use tools like Dmitry ethically and responsibly, and only on systems and domains that you have proper authorization to scan. Unauthorized scanning or information gathering on systems you don't own or have explicit permission to scan is illegal and unethical. Always ensure you follow ethical guidelines and laws while using such tools.



The screenshot shows a terminal window on a Linux desktop environment. The title bar reads "Terminal". The window contains the man page for the WHOIS command. The text is as follows:

```
File Edit View Search Terminal Help
WHOIS(1) Debian GNU/Linux WHOIS(1)
NAME      whois - client for the whois directory service
SYNOPSIS
  whois [ ( -h | --host ) HOST ] [ { -P | --port } PORT ] [ -abBcdGHK-
  LLMMRrX ] [ -g SOURCE[:FIRST-LAST] ] [ -A ATTR[;ATTR]... ]
  [ -S SOURCE[;SOURCE]... ] [ -T TYPE[;TYPE]... ] [ --verbose ] OBJECT
  whois -q KEYWORD
  whois -t TYPE
  whois -v TYPE
  whois --help
  whois --version
DESCRIPTION
  whois searches for an object in a RFC 3912 database.
  This version of the whois client tries to guess the right server to ask
  for the specified object. If no guess can be made it will connect to
  whois.networksolutions.com for NIC handles or whois.arin.net for IPv4
  addresses and network names.
OPTIONS
  -h HOST, --host HOST
    Connect to HOST.
  -H      Do not display the legal disclaimers some registries like to
          show you.
  -P, --port PORT
    Connect to PORT.
  --verbose
    Be verbose.
  --help  Display online help.
  --version
  Manual page whois(1) line 1/192 19% (press h for help or q to quit)
```

Aditya Apandkar / Roll no : 05 / T11



Activities terminal Fri 11:15 lab1006@lab1006-HP-280-G4-MT-Business-PC: ~

```
File Edit View Search Terminal Help
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ whois github.com
Domain Name: GITHUB.COM
Registry Domain ID: 1264983250_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2022-09-07T09:10:42Z
Creation Date: 2007-10-09T18:20:50Z
Registry Expiry Date: 2024-10-09T18:20:50Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2086851759
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Name Server: DNS1.P08.NS0ME.NET
Name Server: DNS2.P08.NS0ME.NET
Name Server: DNS3.P08.NS0ME.NET
Name Server: DNS4.P08.NS0ME.NET
Name Server: NS-1707.AWS0NS-21.CO.UK
Name Server: NS-421.AWS0NS-52.COM
Name Server: NS-520.AWS0NS-01.NET
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2023-08-04T05:43:44Z <<
For more information on Whois status codes, please visit https://icann.org/epp

NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar. Users may consult the sponsoring registrar's Whois database to
view the registrar's reported date of expiration for this registration.

TERMS OF USE: You are not authorized to access or query our Whois
database through the use of electronic processes that are high-volume and
automated except as reasonably necessary to register domain names or
modify existing registrations; the Data in VeriSign Global Registry
Services' ("VeriSign") Whois database is provided by VeriSign for
information purposes only, and to assist persons in obtaining information
about or related to a domain name registration record. VeriSign does not
guarantee its accuracy. By submitting a Whois query, you agree to abide
by the following terms of use: You agree that you may use this Data only
for lawful purposes and that you will not use this Data to search for
purposes of selling, advertising, or promoting your own products or services.
```

Aditya Apandkar / Roll no : 05 / T11

Activities Terminal Fri 11:09 lab1006@lab1006-HP-280-G4-MT-Business-PC: ~

File Edit View Search Terminal Help

robot, please enter your email address to receive a link to a page that facilitates email communication with the relevant contact(s).

Web-based WHOIS:
https://domains.markmonitor.com/whois

If you have a legitimate interest in viewing the non-public WHOIS details, send your request and the reasons for your request to whoisrequest@markmonitor.com and specify the domain name in the subject line. We will review that request and may ask for supporting documentation and explanation.

The data in MarkMonitor's WHOIS database is provided for information purposes, and to assist persons in obtaining information about or related to a domain name's registration record. While MarkMonitor believes the data to be accurate, the data is provided "as is" with no guarantee or warranties regarding its accuracy.

By submitting a WHOIS query, you agree that you will use this data only for lawful purposes and that, under no circumstances will you use this data to:

- (1) allow, enable, or otherwise support the transmission by email, telephone, or facsimile of mass, unsolicited, commercial advertising, or spam; or
- (2) enable high volume, automated, or electronic processes that send queries, data, or email to MarkMonitor (or its systems) or the domain name contacts (or its systems).

MarkMonitor reserves the right to modify these terms at any time.

By submitting this query, you agree to abide by this policy.

MarkMonitor Domain Management(TM)
Protecting companies and consumers in a digital world.

Visit MarkMonitor at <https://www.markmonitor.com>
Contact us at +1.8007459229
In Europe, at +44.02032062226

Lab1006@Lab1006-HP-280-G4-MT-Business-PC:~\$ nslookup github.com

Server: 127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
Name: github.com
Address: 26.207.73.82

Lab1006@Lab1006-HP-280-G4-MT-Business-PC:~\$

Activities Terminal Fri 11:02 lab1006@lab1006-HP-280-G4-MT-Business-PC: ~

File Edit View Search Terminal Help

NSLOOKUP(1)

NAME nslookup - query Internet name servers interactively

SYNOPSIS nslookup [-option] [name | -] [server]

DESCRIPTION nslookup is a program to query Internet domain name servers. nslookup has two modes: interactive and non-interactive. Interactive mode allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain. Non-interactive mode is used to print just the name and requested information for a host or domain.

ARGUMENTS

Interactive mode is entered in the following cases:

1. when no arguments are given (the default name server will be used)
2. when the first argument is a hyphen (-) and the second argument is the host name or Internet address of a name server.

Non-interactive mode is used when the name or Internet address of the host to be looked up is given as the first argument. The optional second argument specifies the host name or address of a name server.

Options can also be specified on the command line if they precede the arguments and are prefixed with a hyphen. For example, to change the default query type to host information, and the initial timeout to 10 seconds, type:

nslookup -query=host -timeout=10

The -version option causes nslookup to print the version number and immediately exits.

INTERACTIVE COMMANDS

host [server]
Look up information for host using the current default server or using server, if specified. If host is an Internet address and the query type is A or PTR, the name of the host is returned. If host is a name and does not have a trailing period, the search list is used to qualify the name.

To look up a host not in the current domain, append a period to the name.

server domain
Change the default server to domain; lserver uses the initial server to look up information about domain, while server uses the current default server. If an authoritative answer can't be found, the names of servers that might have the answer are returned.

root

Manual page nslookup(1) line 1 [press h for help or q to quit]

Aditya Apandkar / Roll no : 05 / T11

```
activities terminal
File Edit View Search Terminal Help
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ traceroute(1) Traceroute For Linux
NAME traceroute - print the route packets trace to network host
SYNOPSIS traceroute [-addrtimerval] [-f first_ttl] [-g gate[,...]
[-L device] [-m max_ttl] [-p port] [-s src_addr]
[-q heuristics] [-N queries] [-t tos]
[-L flow_label] [-w waittime] [-z sendwait] [-UL] [-D]
[-P proto] [-sport:port] [-M method] [-o mod_options]
[-I interface]
host [packet_len]
traceroute [options]
tcptraceroute [options]
lft [options]
DESCRIPTION traceroute tracks the route packets taken from an IP network on their way to a given host. It utilizes the IP protocol's time to live (TTL) field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to the host.
traceroute is equivalent to tracert -e
tcptraceroute is equivalent to traceroute -T
lft, the Layer Four Traceroute, performs a TCP traceroute, like traceroute -T, but attempts to provide compatibility with the original such implementation, also called "lft".
The only required parameter is the name or IP address of the destination host. The optional packet_length is the total size of the probing packet (default 60 bytes for IPv4 and 80 for IPv6). The specified size can be ignored in some situations or increased up to a minimal value.
This program attempts to trace the route an IP packet would follow to some internet host by launching probe packets with a small ttl (time to live) then listening for an ICMP "time exceeded" reply from a gateway. We start our probes with a ttl of one and increase by one until we get an ICMP "port unreachable" reply. The ttl of the probe is printed before the address of the gateway. The round trip time of each probe is printed after the address. If the ttl setting and a time is printed showing the ttl, address of the gateway and round trip time of each probe. The address can be followed by additional information when requested. If the probe answers come from different gateways, the address of each responding system will be printed. If there is no response within a certain timeout, an "*" (asterisk) is printed for that probe.
After the trip time, some additional annotation can be printed: IN, IN, or IP (host, network or protocol unreachable), IS (source route failed), IF (fragmentation needed), IX (communication administratively prohibited), IV (host precedence violation), IC (precedence cutoff in effect), or Iunreachable (ICMP unreachable code <num>). If almost all the probes result in some kind of unreachable, traceroute will give up and exit.
We don't want the destination host to process the UDP probe packets, so the destination port is set to an unlikely value (you can change it with the -p flag). There is no such a problem for ICMP or TCP tracerouting (for TCP we use half-open technique, which prevents our probes to be seen by applications on the destination host).
Manual page traceroute(1) line 1 [press h for help or q to quit]
```

allows the user to print just the second argument default query every type is A or name.

```
activities terminal
File Edit View Search Terminal Help
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ traceroute github.com
traceroute to github.com (20.207.73.87), 30 hops max, 64 byte packets
1 * gateway (192.168.0.1) 0.553 ms 0.490 ms 0.543 ms
2 203.212.25.11 (203.212.25.1) 1.423 ms 1.493 ms 1.362 ms
3 203.212.24.53 (203.212.24.53) 2.100 ms 2.059 ms 2.127 ms
4 *
5 ac76-0.lcr01.bon02.ntwk.msn.net (104.44.6.69) 3.891 ms 8.568 ms 3.810 ms
6 *
7 be-20-6.lcr03.bon02.ntwk.msn.net (104.44.34.37) 8.066 ms *
8 * be-10-8.lcr03.bon30.ntwk.msn.net (104.44.7.121) 15.324 ms 7.854 ms
9 ae12d-0.rwd03.png21.ntwk.msn.net (104.44.21.22) 4.959 ms ae12d-0.rwd04.png21.ntwk.msn.net (104.44.22.22)
10 * be-1-8.lcr01.ng021.ntwk.msn.net (104.44.16.50) 6.835 ms
11 * ac182-0.rwd02.png21.ntwk.msn.net (104.44.11.234) 12.360 ms *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ lserver domain
Change the default server to domain; lserver uses the initial server to look up information about domain, while server uses the current default server.
If an authoritative answer can't be found, the names of servers that might have the answer are returned.
root Manual page nslookup(1) line 1 [press h for help or q to quit]
```

allows the user to print just the second argument default query every type is A or name.

Aditya Apandkar / Roll no : 05 / T11

The screenshot shows two terminal windows side-by-side. The left window displays the man page for the 'dig' command, which is a DNS lookup utility. The right window displays the man page for 'nslookup', which is used to print just the second argument of a default query. Both windows are running on a Linux desktop environment with a dark theme.

```
DIG(1) BIND9 DIG(1)
NAME
    dig - DNS lookup utility
SYNOPSIS
    dig {server} [-b address] [-c class] [-f filename] [-k filename] [-n] [-p port] [-q name] [-t type] [-v] [-x addr] [-y [hex]namekey] [{-e}] [-e]
    dig [-h]
    dig [global-queryopt...] [query...]
DESCRIPTION
    dig is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried. Most DNS administrators use dig to troubleshoot DNS problems because of its flexibility, ease of use and clarity of output. Other lookup tools tend to have less functionality than dig.
    Although dig is normally used with command-line arguments, it also has a batch mode of operation for reading lookup requests from a file. A brief summary of its command-line arguments and options is printed when the -h option is given. Unlike earlier versions, the BIND 9 implementation of dig allows multiple lookups to be issued from the command line.
    Unless it is told to query a specific name server, dig will try each of the servers listed in /etc/resolv.conf. If no usable server addresses are found, dig will send the query to the local host.
    When no command line arguments or options are given, dig will perform an NS query for "." (the root).
    It is possible to set per-user defaults for dig via ${HOME}/.digrc. This file is read and any options in it are applied before the command line arguments.
    The IN and CH class names overlap with the IN and CH top level domain names. Either use the -t and -c options to specify the type and class, use the -q option to specify the domain name, or use "IN." and "CH." when looking up these top level domains.
SIMPLE USAGE
    A typical invocation of dig looks like:
        dig @server name type
    where:
        server
            is the name or IP address of the name server to query. This can be an IPv4 address in dotted-decimal notation or an IPv6 address in colon-delimited notation. When the supplied server argument is a hostname, dig resolves that name before querying that name server.
        If no server argument is provided, dig consults /etc/resolv.conf; if an address is found there, it queries the name server at that address. If either of
        Manual page dig(1) line 1 (press h for help or q to quit)
    lserver domain
        change the default server to domain; lserver uses the initial server to look up information about domain, while server uses the current default server.
        If an authoritative answer can't be found, the names of servers that might have the answer are returned.
    root
        Manual page nslookup(1) line 1 (press h for help or q to quit)
```

```
NSLOOKUP(1)
NAME
    nslookup - prints just the second argument of a default query
DESCRIPTION
    nslookup allows the user to print just the second argument of a default query.
```

The screenshot shows two terminal windows side-by-side. The left window displays the man page for the 'traceroute' command, which is used to trace the route taken by a packet from the local host to a destination host. The right window displays the man page for 'dmitry', which is a Deepmagic Information Gathering Tool. Both windows are running on a Linux desktop environment with a dark theme.

```
File Edit View Search Terminal Help
;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Fri Aug 04 11:37:03 IST 2023
;; MSG SIZE rcvd: 55
Lab1006@Lab1006-HP-280-G4-MT-Business-PC:~$ man dig
Lab1006@Lab1006-HP-280-G4-MT-Business-PC:~$ traceroute tsec.edu
traceroute to tsec.edu (152.21.70.62), 30 hops max, 60 byte packets
  1  gateway (192.168.0.1)  0.642 ms  0.580 ms  0.725 ms
  2  281.212.25.1 (70.212.25.1)  2.118 ms  2.145 ms  2.035 ms
  3  203.212.24.53 (203.212.24.53)  2.193 ms  2.350 ms  1.983 ms
  4  175.100.177.53 (175.100.177.53)  2.788 ms  2.878 ms  2.706 ms
  5  * 172.16.2.101 (172.16.2.101)  3.391 ms  3.349 ms
  6  * 121.241.42.57.static-mumbai.vsnl.net.in (121.241.42.57)  2.703 ms  3.142 ms
  7  172.23.78.237 (172.23.78.237)  3.173 ms  2.621 ms  2.567 ms
  8  ix-06-0-100.tcore1.mly-mumbai.as6453.net (188.87.38.5)  2.800 ms  3.101 ms  2.872 ms
  9  * 1f-be-6-2.ecore1.enrs2.marseille.as6453.net (195.219.174.16)  124.131 ms
  10  * 1f-ae-7-2.tcore1.pye-paris.as6453.net (195.219.174.9)  125.073 ms  124.004 ms
  11  *
  12  *
  13  50-6-131-2.unifiedlayer.com (50.6.131.2)  238.519 ms  236.465 ms  232.736 ms
  14  *
  15  162.241.78.62.webhostbox.net (162.241.78.62)  241.039 ms IX *
Lab1006@Lab1006-HP-280-G4-MT-Business-PC:~$ dig tsec.edu CNAME
; <> DIG 9.11.3-ubuntu1.18-Ubuntu <> tsec.edu CNAME
;; global options: +cmd
;; Got answer:
;; >>>HEADER<< opcode: QUERY, status: NOERROR, id: 51664
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;tsec.edu.           IN      CNAME
;;
;; Query time: 6 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Fri Aug 04 11:57:26 IST 2023
;; MSG SIZE rcvd: 37
Lab1006@Lab1006-HP-280-G4-MT-Business-PC:~$ man dmitry
Lab1006@Lab1006-HP-280-G4-MT-Business-PC:~$ dmitry tsec.edu
Deepmagic Information Gathering Tool
```

Aditya Apandkar / Roll no : 05 / T11

```
activities terminal
File Edit View Search Terminal Help
Lab1006@Lab1006-HP-Z80-G4-MT-Business-PC:~$ man dmitry
Lab1006@Lab1006-HP-Z80-G4-MT-Business-PC:~$ dmitry -p tsec.edu
Deepmagic Information Gathering Tool
"There be some deep magic going on"
HOSTIP: 103.34.78.62
Hostnametsec.edu
Gathered TCP Port Information for 162.241.78.62
Port State
22/tcp open
25/tcp open
53/tcp open
80/tcp open
110/tcp open
143/tcp open
Ports scan finished: Scanned 150 ports, 143 ports were in state closed
All scans completed, exiting
Lab1006@Lab1006-HP-Z80-G4-MT-Business-PC:~$ dmitry -f tsec.edu
Deepmagic Information Gathering Tool
"There be some deep magic going on"
error: No '-p' flag passed with TTL, assuming -p
HOSTIP:162.241.78.62
HOSTNAME:tsec.edu
Gathered TCP Port Information for 162.241.78.62
Port State
22/tcp open
25/tcp open
53/tcp open
80/tcp open
110/tcp open
+C
Lab1006@Lab1006-HP-Z80-G4-MT-Business-PC:~$ man nikto
Lab1006@Lab1006-HP-Z80-G4-MT-Business-PC:~$
```

```
activities terminal
File Edit View Search Terminal Help
Lab1006@Lab1006-HP-Z80-G4-MT-Business-PC:~$ nikto -h 162.241.78.62
- Nikto v2.1.5
=====
+ Target IP: 162.241.78.62
+ Target Hostname: 162.241.78.62
+ Target Port: 80
+ Start Time: 2022-08-04 12:11:41 (GMT+5)
+ Server: Microsoft-IIS/7.5
+ SSL Version: TLS 1.2
+ SSL Cipher Suite: ECDHE-RSA-AES256-SHA384
+ OSVDB-27077: /phpImageview.php?pic=javascript:alert(0754): PHP Image View 1.0 is vulnerable to Cross Site Scripting (XSS). http://www.cirt.org/advisories/CA-2000-02.html.
+ OSVDB-3931: /myphorum/links.php?op=search&query=[script]alert('Vulnerable');[/script]&query=: myphorum is vulnerable to Cross Site Scripting (XSS). http://www.cirt.org/advisories/CA-2000-02.html.
+ OSVDB-3931: /myphorum/links.php?op=PopularAratumen:[script]alert(document.cookie);[/script]&rettype=percent: myphorum is vulnerable to Cross Site Scripting (XSS). http://www.cirt.org/advisories/CA-2000-02.html.
+ modules.php?op=loadAdminFile&file=index&mynewsid_cat=1&category=1&src=javascript:alert(9480);X3&parent_id=0: Post Nuke 0.7.2.3-Phoenix is vulnerable to Cross Site Scripting (XSS). http://www.cirt.org/advisories/CA-2000-02.html.
+ /modules.php?op=loadAdminFile&file=index&mynewsid_cat=1&category=1&src=javascript:alert(9480);X3&parent_id=0: Post Nuke 0.7.2.3-Phoenix is vulnerable to Cross Site Scripting (XSS). http://www.cirt.org/advisories/CA-2000-02.html.
+ OSVDB-4598: /members.asp?if=N22;alert(223344);functionN20x();{vN20-N22: Web Wiz Forums ver. 7.01 and below is vulnerable to Cross Site Scripting (XSS). http://www.cert.org/advisories/CA-2000-02.html.
+ OSVDB-7948: /index.php?if=N22;alert(9833);functionN20x();{vN20-N22: Web Wiz Forums ver. 7.01 and below is vulnerable to Cross Site Scripting (XSS). http://www.cert.org/advisories/CA-2000-02.html.
+ OSVDB-31239: /mainman/listInfo: Mainman was found on the server.
***: ~Lab1006@Lab1006-HP-Z80-G4-MT-Business-PC:~$
```

Conclusion:

We understood the different commands and its types for reconnaissance and implemented the same in ubuntu

Lab Assignment No.7

Aim: Study of Packet Sniffer tool TCPDUMP. Use it to capture and analyze the packet.

Lab Outcome Attained: LO3 – Explore the different network reconnaissance tools to gather information about networks.

Theory:

A. What is TCPDUMP and how to install it?

TCPDUMP is a command line utility that allows you to capture and analyze network traffic going through your system. It is often used to help troubleshoot network issues, as well as a security tool.

Below are a few options you can use when configuring TCPDUMP:

- -i any: Listen on all interfaces just to see if you're seeing any traffic.
- -i eth0: Listen on the eth0 interface.
- -D: Show the list of available interfaces
- -n: Don't resolve hostnames.
- -nn: Don't resolve hostnames or port names.
- -q: Be less verbose (more quiet) with your output.
- -t: Give human-readable timestamp output.
- -ttt: Give maximally human-readable timestamp output.
- -X: Show the packet's contents in both hex and ASCII.
- -XX: Same as -X, but also shows the ethernet header.
- -v, -vv, -vvv: Increase the amount of packet information you get back.
- -c: Only get x number of packets and then stop.
- -s: Define the snaplength (size) of the capture in bytes. Use -s0 to get everything, unless you are intentionally capturing less.
- -S: Print absolute sequence numbers.
- -e: Get the ethernet header as well.

To install tcpdump

\$ sudo apt-get install tcpdump

B. Explain various commands in tcpdump to capture different types of packets.

1. Choosing an interface:

By default, tcpdump captures packets on all interfaces. To view a summary of available interfaces, run the command:

tcpdump -D

2. Basic command for sniffing

tcpdump -n

The -n parameter is given to stop tcpdump from resolving ip addresses to hostnames.

Now lets increase the display resolution of this packet, or get more details about it. The verbose switch comes in handy. Here is a quick example

3. `tcpdump -v -n`

4. Getting the ethernet header (link layer headers)

In the above examples details of the ethernet header are not printed. Use the `-e` option to print the ethernet header details as well.

Filtering packets using expressions

5. selecting packets with specific protocol

```
# tcpdump -n tcp  
#tcpdump -n icmp
```

6. Capturing traffic from particular host or port

Expressions can be used to specify source ip, destination ip, and port numbers. The next example picks up all those packets with source address 172.16.92.1

```
# tcpdump -n src 172.16.92.1  
# tcpdump -n dst 172.16.92.1  
# tcpdump -n port 80  
# tcpdump port 80
```

7. Specific Packets from specific port

```
# tcpdump udp and src port 53
```

observing packets within a specific port range

```
# tcpdump -n portrange 1-80
```

It shows all packets whose source or destination port is between 1 to 80

```
tcpdump -n src port 443
```

8. Writing Captures to a File

It's often useful to save packet captures into a file for analysis in the future. These files are known as PCAP (PEE-cap) files, and they can be processed by hundreds of different applications, including network analyzers, intrusion detection systems, and of course by `tcpdump` itself. Here we're writing to a file called `capture_file` using the `-w` switch.

```
# tcpdump port 80 -w capture_file
```

9. Reading PCAP files

You can read PCAP files by using the `-r` switch. Note that you can use all the regular commands within `tcpdump` while reading in a file; you're only limited by the fact that you can't capture and process what doesn't exist in the file already.

```
# tcpdump -r capture_file
```

10. From specific IP and destined for a specific Port

Let's find all traffic from 10.5.2.3 going to any host on port 3389.

```
tcpdump -nnvvS src 10.5.2.3 and dst port 3389
```

11. From One Network to Another

Let's look for all traffic coming from 192.168.x.x and going to the 10.x or 172.16.x.x networks, and we're showing hex output with no hostname resolution and one level of extra verbosity.

```
tcpdump -nvX src net 192.168.0.0/16 and dst net 10.0.0.0/8 or 172.16.0.0/16
```

12. Non ICMP Traffic Going to a Specific IP

This will show us all traffic going to 192.168.0.2 that is not ICMP.

```
tcpdump dst 192.168.0.2 and src net and not icmp
```

13. Complex Grouping and Special Characters

Also keep in mind that when you're building complex queries you might have to group your options using single quotes. Single quotes are used in order to tell tcpdump to ignore certain special characters—in this case below the “()” brackets. This same technique can be used to group using other expressions such as host, port, net, etc. Take a look at the command below.

(incorrect)

```
# tcpdump src 10.0.2.4 and (dst port 3389 or 22)
```

If you tried to run this otherwise very useful command, you'd get an error because of the parenthesis. You can either fix this by escaping the parenthesis (putting a before each one), or by putting the entire command within single quotes:

(correct)

```
# tcpdump 'src 10.0.2.4 and (dst port 3389 or 22)'
```

14. Isolating Specific TCP Flags:

You can also capture traffic based on specific TCP flag(s).

The filters below find these various packets because tcp[13] looks at offset 13 in the TCP header, the number represents the location within the byte, and the !=0 means that the flag in question is set to 1, i.e. it's on.

Show me all URGENT (**URG**) packets...

```
# tcpdump 'tcp[13] & 32!=0'
```

Show me all ACKNOWLEDGE (**ACK**) packets...

```
# tcpdump 'tcp[13] & 16!=0'
```

Show me all PUSH (**PSH**) packets...

```
# tcpdump 'tcp[13] & 8!=0'
```

```
Show me all RESET (RST) packets...
# tcpdump 'tcp[13] & 4!=0'
Show me all SYNCHRONIZE (SYN) packets...
# tcpdump 'tcp[13] & 2!=0'
Show me all FINISH (FIN) packets...
# tcpdump 'tcp[13] & 1!=0'
Show me all SYNCHRONIZE/ACKNOWLEDGE (SYNACK) packets...
# tcpdump 'tcp[13]=18'
```

Only the PSH, RST, SYN, and FIN flags are displayed in tcpdump's flag field output. URGs and ACKs are displayed, but they are shown elsewhere in the output rather than in the flags field.

As with most powerful tools, however, there are multiple ways to do things. The example below shows another way to capture packets with specific TCP flags set.

```
# tcpdump 'tcp[tcpflags] == tcp-syn'
tcpflags option...
# tcpdump 'tcp[tcpflags] == tcp-rst'
tcpflags option...
# tcpdump 'tcp[tcpflags] == tcp-fin
```

Output Screenshots:

Aditya Apandkar / Roll no : 05 / T11

Aditya Apandkar / Roll no : 05 / T11

```
Activities Terminal Fri 11:51 * lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -e
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:51:19.661809 a:1e:12:b8:80:d8 (oui Unknown) > 01:0e:5e:7f:ff:fe (oui Unknown), ethertype IPv4 (0x0800), length 218: 192.168.0.161.57489 > 239.255.255.250.1900: UDP, length 176
11:51:19.663530 0:0e:3c:1a:64:30 (oui Unknown) > ac:15:a2:b9:9e:29 (oui Unknown), ethertype IPv4 (0x0800), length 88: lab1006-HP-280-G4-MT-Business-PC.59149 > _gateway.domain., PTR? 27825.255.255.239.in-addr.arpa. (46)
11:51:19.665124 ac:15:a2:b9:9e:29 (oui Unknown) > 04:0e:3c:1a:64:30 (oui Unknown), ethertype IPv4 (0x0800), length 145: _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.59149; PTR? 27825.NXDomain# 0/1 (103)
11:51:19.667206 0:0e:3c:1a:64:30 (oui Unknown) > ac:15:a2:b9:9e:29 (oui Unknown), ethertype IPv4 (0x0800), length 86: lab1006-HP-280-G4-MT-Business-PC.47336 > _gateway.domain., PTR? 161.0.168.192.in-addr.arpa. (44)
11:51:19.669729 ac:15:a2:b9:9e:29 (oui Unknown) > 04:0e:3c:1a:64:30 (oui Unknown), ethertype IPv4 (0x0800), length 145: _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.47336; PTR? 62331.NXDomain# 0/1 (103)
11:51:19.715214 0:0e:3c:1a:64:30 (oui Unknown) > Broadcast, ethertype ARP (0x0806), length 68: Request who-has 192.168.0.141 tell 192.168.0.226, length 46
11:51:19.715844 0:0e:3c:1a:64:30 (oui Unknown) > ac:15:a2:b9:9e:29 (oui Unknown), ethertype IPv4 (0x0800), length 86: lab1006-HP-280-G4-MT-Business-PC.48798 > _gateway.domain., PTR? 141.0.168.192.in-addr.arpa. (44)
11:51:19.720583 ac:15:a2:b9:9e:29 (oui Unknown) > 04:0e:3c:1a:64:30 (oui Unknown), ethertype IPv4 (0x0800), length 145: _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.48798; PTR? 7220583.NXDomain# 0/1 (103)
11:51:19.722674 0:0e:3c:1a:64:30 (oui Unknown) > ac:15:a2:b9:9e:29 (oui Unknown), ethertype IPv4 (0x0800), length 86: lab1006-HP-280-G4-MT-Business-PC.55541 > _gateway.domain., PTR? 226.0.168.192.in-addr.arpa. (44)
11:51:19.723625 ac:15:a2:b9:9e:29 (oui Unknown) > 04:0e:3c:1a:64:30 (oui Unknown), ethertype IPv4 (0x0800), length 145: _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.55541; PTR? 226.NXDomain# 0/1 (103)
11:51:19.726348 0:0e:3c:1a:60:02 (oui Unknown) > 01:0e:5e:7f:ff:fe (oui Unknown), ethertype IPv4 (0x0800), length 217: 192.168.0.190.56582 > 239.255.255.250.1900: UDP, length 175
11:51:19.727077 0:0e:3c:1a:64:30 (oui Unknown) > ac:15:a2:b9:9e:29 (oui Unknown), ethertype IPv4 (0x0800), length 86: lab1006-HP-280-G4-MT-Business-PC.40875 > _gateway.domain., PTR? 190.0.168.192.in-addr.arpa. (44)
11:51:19.728488 ac:15:a2:b9:9e:29 (oui Unknown) > 04:0e:3c:1a:64:30 (oui Unknown), ethertype IPv4 (0x0800), length 145: _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.40875; PTR? 190.0.168.192.in-addr.arpa. (44)
11:51:19.817736 ac:15:a2:b8:80:d8 (oui Unknown) > Broadcast, ethertype IPv4 (0x0800), length 176: 192.168.0.161.17500 > 255.255.255.255.17500: UDP, length 134
11:51:19.822264 ac:15:a2:b8:80:d8 (oui Unknown) > Broadcast, ethertype IPv4 (0x0800), length 176: 192.168.0.161.17500 > 192.168.0.255.17500: UDP, length 134
11:51:19.822283 ac:15:a2:b8:80:d8 (oui Unknown) > Broadcast, ethertype IPv4 (0x0800), length 176: 192.168.0.161.17500 > 255.255.255.255.17500: UDP, length 134
11:51:19.822285 ac:15:a2:b8:80:d8 (oui Unknown) > Broadcast, ethertype IPv4 (0x0800), length 176: 192.168.0.161.17500 > 255.255.255.255.17500: UDP, length 134
11:51:19.822286 ac:15:a2:b8:80:d8 (oui Unknown) > Broadcast, ethertype IPv4 (0x0800), length 176: 192.168.0.161.17500 > 255.255.255.255.17500: UDP, length 134
11:51:19.822288 ac:15:a2:b8:80:d8 (oui Unknown) > Broadcast, ethertype IPv4 (0x0800), length 176: 192.168.0.161.17500 > 255.255.255.255.17500: UDP, length 134
11:51:19.822952 ac:15:a2:b8:90:30 (oui Unknown) > ac:15:a2:b9:9e:29 (oui Unknown), ethertype IPv4 (0x0800), length 86: lab1006-HP-280-G4-MT-Business-PC.54936 > _gateway.domain., PTR? 23728.255.168.192.in-addr.arpa. (44)
11:51:19.824419 ac:15:a2:b9:9e:29 (oui Unknown) > 04:0e:3c:1a:64:30 (oui Unknown), ethertype IPv4 (0x0800), length 145: _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.54936; PTR? 23728.NXDomain# 0/1 (103)
11:51:19.833651 ac:15:a2:b8:80:d4 (oui Unknown) > 01:0e:5e:7f:ff:fe (oui Unknown), ethertype IPv4 (0x0800), length 217: 192.168.0.140.55902 > 239.255.255.250.1900: UDP, length 175
11:51:19.834366 0:0e:3c:1a:64:30 (oui Unknown) > ac:15:a2:b9:9e:29 (oui Unknown), ethertype IPv4 (0x0800), length 86: lab1006-HP-280-G4-MT-Business-PC.34636 > _gateway.domain., PTR? 41241.NXDomain# 0/1 (103)
11:51:19.835862 ac:15:a2:b9:9e:29 (oui Unknown) > 04:0e:3c:1a:64:30 (oui Unknown), ethertype IPv4 (0x0800), length 145: _gateway.domain > lab1006-HP-280-G4-MT-Business-PC.34636; PTR? 41241.NXDomain# 0/1 (103)
11:51:20.203218 dt:b9:cc:03:2f (oui Unknown) > 01:0e:5e:09:00:fb (oui Unknown), ethertype IPv4 (0x0800), length 79: 192.168.0.173.mdns > 224.0.0.251.mdns: 27532 PTR
```

Aditya Apandkar / Roll no : 05 / T11

```
Fri 11:48 #  
File Edit View Search Terminal Help  
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n -v  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes  
11:05:34.740864 IP 192.168.0.198.53928 > 52.215.37.57.443: Flags [P..], seq 231237102:231237204, ack 680700016, win 501, options [nop,nop,TS val 4046105926 ecr 2810503602], length 352  
11:05:34.740919 IP 192.168.0.198.53928 > 52.215.37.57.443: Flags [P..], seq 102:454, ack 1, win 501, options [nop,nop,TS val 4046105926 ecr 2810503602], length 352  
11:05:34.873602 IP 52.215.37.57.443 > 192.168.0.198.53928: Flags [P..], seq 1:195, ack 454, win 425, options [nop,nop,TS val 2810534650 ecr 4046105926], length 194  
11:05:34.873605 IP 192.168.0.198.53928 > 52.215.37.57.443: Flags [P..], seq 1:195, ack 195, win 425, options [nop,nop,TS val 2810534651 ecr 4046105926], length 194  
11:05:38.296137 IP 192.168.0.198.40800 > 91.189.91.48.80: Flags [S.], seq 2178220167, win 64240, options [mss 1440,sackOK,TS val 3397411090 ecr 0,nop,wscale 7], length 0  
11:05:38.296137 IP 91.189.91.48.80 > 192.168.0.198.40800: Flags [S.], seq 3981380951, ack 2178210168, win 28960, options [mss 1440,sackOK,TS val 2296614634 ecr 3397411090  
90,nop,wscale 9], length 0  
11:05:38.3553431 IP 91.189.91.48.80 > 192.168.0.198.40800: Flags [., ack 1, win 502, options [nop,nop,TS val 3397411345 ecr 2296614634], length 0  
11:05:38.3551758 IP 192.168.0.198.40800 > 91.189.91.48.80: Flags [P..], seq 1:198, ack 1, win 502, options [nop,nop,TS val 3397411345 ecr 2296614634], length 87: HTTP: GET  
/ HTTP/1.1  
11:05:38.3553431 IP 91.189.91.48.80 > 192.168.0.198.40800: Flags [., ack 88, win 57, options [nop,nop,TS val 2296614634 ecr 3397411345], length 0  
11:05:38.867378 IP 91.189.91.48.80 > 192.168.0.198.40800: Flags [P..], seq 1:199, ack 88, win 57, options [nop,nop,TS val 2296614635 ecr 3397411345], length 189: HTTP: H  
TTP/1.1 294 No Content  
11:05:38.867378 IP 91.189.91.48.80 > 192.168.0.198.40800: Flags [P..], seq 190, ack 190, win 501, options [nop,nop,TS val 3397411601 ecr 2296614659], length 0  
11:05:38.867660 IP 192.168.0.198.40800 > 91.189.91.48.80: Flags [F.], seq 88, ack 190, win 501, options [nop,nop,TS val 3397411601 ecr 2296614659], length 0  
11:05:38.866599 IP 91.189.91.48.80 > 192.168.0.198.40800: Flags [F.], seq 190, ack 88, win 57, options [nop,nop,TS val 4046106059 ecr 2810534651], length 0  
11:05:38.866646 IP 192.168.0.198.40800 > 91.189.91.48.80: Flags [., ack 191, win 501, options [nop,nop,TS val 3397411602 ecr 2296614659], length 0  
11:05:38.8669281 IP 91.189.91.48.80 > 192.168.0.198.40800: Flags [., ack 89, win 57, options [nop,nop,TS val 2296614659 ecr 3397411601], length 0  
^C  
16 packets captured  
16 packets received by filter  
0 packets dropped by kernel  
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n -i enp3s0 -n -v  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes  
11:06:51.337780 IP 192.168.0.198 > 192.168.0.1: ICMP 192.168.0.198 udp port 137 unreachable, length 86  
11:06:51.343632 IP 192.168.0.198 > 192.168.0.1: ICMP 192.168.0.198 udp port 137 unreachable, length 86  
11:07:51.304347 IP 192.168.0.198 > 192.168.0.1: ICMP 192.168.0.198 udp port 137 unreachable, length 86  
11:07:51.320091 IP 192.168.0.198 > 192.168.0.1: ICMP 192.168.0.198 udp port 137 unreachable, length 86  
^C  
4 packets captured  
4 packets received by filter  
0 packets dropped by kernel  
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n -src 172.34.76.2  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

```
Fri 11:48 #  
File Edit View Search Terminal Help  
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n -src 168.195.218.201  
tcpdump: verbose output suppressed, use -v or -vv for Full protocol decode  
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes  
11:19:26.969221 IP 168.195.218.201.80 > 192.168.0.198.42836: Flags [S.], seq 210078969, ack 2555517200, win 28960, options [mss 1440,sackOK,TS val 2296697469 ecr 410949  
6370,nop,wscale 9], length 0  
11:19:26.912698 IP 168.195.218.201.80 > 192.168.0.198.42836: Flags [., ack 348, win 59, options [nop,nop,TS val 2296697470 ecr 4109496726], length 0  
11:19:26.915337 IP 168.195.218.201.80 > 192.168.0.198.42842: Flags [S.], seq 1163014742, ack 48896756, win 28960, options [mss 1440,sackOK,TS val 2296697470 ecr 410949  
372,nop,wscale 9], length 0  
11:19:27.149802 IP 168.195.218.201.80 > 192.168.0.198.42846: Flags [S.], seq 1919087986, ack 1597564558, win 28960, options [mss 1440,sackOK,TS val 2296697493 ecr 41094  
96623,nop,wscale 9], length 0  
11:19:27.289708 IP 168.195.218.201.80 > 192.168.0.198.42836: Flags [P..], seq 1:1429, ack 348, win 59, options [nop,nop,TS val 2296697507 ecr 4109496726], length 1428: H  
TTP: HTTP/1.1 208 OK  
11:19:27.297639 IP 168.195.218.201.80 > 192.168.0.198.42836: Flags [., seq 1429:4285, ack 348, win 59, options [nop,nop,TS val 2296697508 ecr 4109497105], length 2856:  
HTTP  
11:19:27.297169 IP 168.195.218.201.80 > 192.168.0.198.42836: Flags [., seq 4285:5713, ack 348, win 59, options [nop,nop,TS val 2296697508 ecr 4109497105], length 1428:  
HTTP  
11:19:27.297250 IP 168.195.218.201.80 > 192.168.0.198.42836: Flags [., seq 5713:7141, ack 348, win 59, options [nop,nop,TS val 2296697508 ecr 4109497105], length 1428:  
HTTP  
11:19:27.297316 IP 168.195.218.201.80 > 192.168.0.198.42836: Flags [., seq 7141:8569, ack 348, win 59, options [nop,nop,TS val 2296697508 ecr 4109497105], length 1428:  
HTTP  
11:19:27.297395 IP 168.195.218.201.80 > 192.168.0.198.42836: Flags [P..], seq 8569:9620, ack 348, win 59, options [nop,nop,TS val 2296697508 ecr 4109497105], length 1051  
: HTTP  
11:19:27.297531 IP 168.195.218.201.80 > 192.168.0.198.42836: Flags [., seq 9620:11948, ack 348, win 59, options [nop,nop,TS val 2296697508 ecr 4109497105], length 1428  
: HTTP  
11:19:27.297629 IP 168.195.218.201.80 > 192.168.0.198.42836: Flags [., seq 11048:12476, ack 348, win 59, options [nop,nop,TS val 2296697508 ecr 4109497105], length 142  
8: HTTP  
11:19:27.297733 IP 168.195.218.201.80 > 192.168.0.198.42836: Flags [FP.], seq 12476:13195, ack 348, win 59, options [nop,nop,TS val 2296697508 ecr 4109497105], length 7  
19: HTTP  
11:19:27.360113 IP 168.195.218.201.80 > 192.168.0.198.42836: Flags [., ack 349, win 59, options [nop,nop,TS val 2296697508 ecr 4109497113], length 0  
11:19:27.383208 IP 168.195.218.201.80 > 192.168.0.198.42842: Flags [., ack 338, win 59, options [nop,nop,TS val 2296697517 ecr 4109497197], length 0  
11:19:27.748835 IP 168.195.218.201.80 > 192.168.0.198.42842: Flags [P..], seq 1:416, ack 338, win 59, options [nop,nop,TS val 2296697553 ecr 4109497197], length 415: HTT  
P: HTTP/1.1 404 Not Found  
11:19:27.751623 IP 168.195.218.201.80 > 192.168.0.198.42846: Flags [., ack 341, win 59, options [nop,nop,TS val 2296697554 ecr 4109497565], length 6  
P: HTTP/1.1 404 Not Found  
11:19:28.169834 IP 168.195.218.201.80 > 192.168.0.198.42846: Flags [P..], seq 1:412, ack 341, win 59, options [nop,nop,TS val 2296697589 ecr 4109497565], length 411: HTT  
P: HTTP/1.1 404 Not Found  
11:19:28.249803 IP 168.195.218.201.80 > 192.168.0.198.42842: Flags [., ack 340, win 61, options [nop,nop,TS val 2296697603 ecr 4109498064], length 0  
11:19:28.886415 IP 168.195.218.201.80 > 192.168.0.198.42842: Flags [P..], seq 416:1844, ack 640, win 61, options [nop,nop,TS val 2296697667 ecr 4109498864], length 1428:  
HTTP  
11:19:28.895762 IP 168.195.218.201.80 > 192.168.0.198.42842: Flags [., seq 1844:4706, ack 640, win 61, options [nop,nop,TS val 2296697668 ecr 4109498702], length 2856:  
HTTP  
11:19:28.895949 IP 168.195.218.201.80 > 192.168.0.198.42842: Flags [., seq 4706:7556, ack 640, win 61, options [nop,nop,TS val 2296697668 ecr 4109498702], length 2856:  
HTTP  
11:19:28.896093 IP 168.195.218.201.80 > 192.168.0.198.42842: Flags [P..], seq 7556:10035, ack 640, win 61, options [nop,nop,TS val 2296697668 ecr 4109498702], length 247  
9: HTTP  
11:19:28.896093 IP 168.195.218.201.80 > 192.168.0.198.42842: Flags [P..], seq 10035:12476, ack 640, win 61, options [nop,nop,TS val 2296697668 ecr 4109498702], length 247  
9: HTTP
```

Aditya Apandkar / Roll no : 05 / T11

```
Activities Terminal Fri 11:49 * lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
96 packets captured
96 packets received by filter
0 packets dropped by kernel
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n dst 168.195.218.201
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enps3, link-type EN10MB (Ethernet), capture size 262144 bytes
11:20:35.867798 IP 192.168.0.198.42846 > 168.195.218.201.80: Flags [.], ack 1919089224, win 501, options [nop,nop,T5 val 4109565623 ecr 2296703335], length 0
11:20:45.290373 IP 192.168.0.198.42846 > 168.195.218.201.80: Flags [.], ack 1, win 501, options [nop,nop,T5 val 4109575106 ecr 2296704359], length 0
11:20:46.048879 IP 192.168.0.198.42846 > 168.195.218.201.80: Flags [.], ack 1, win 501, options [nop,nop,T5 val 4109575064 ecr 2296704359], length 0
11:20:56.288806 IP 192.168.0.198.42846 > 168.195.218.201.80: Flags [.], ack 1, win 501, options [nop,nop,T5 val 4109580104 ecr 2296705383], length 0
11:21:01.1508051 IP 192.168.0.198.52194 > 168.195.218.201.80: Flags [S], seq 39337/16145, win 64240, options [mss 1400,sackOK,T5 val 4109591396 ecr 0,nope,wscale 7], length 0
11:21:01.61.831263 IP 192.168.0.198.52208 > 168.195.218.201.80: Flags [S], seq 562770641, win 64240, options [mss 1400,sackOK,T5 val 4109591647 ecr 0,nope,wscale 7], length 0
11:21:01.938588 IP 192.168.0.198.52194 > 168.195.218.201.80: Flags [.], ack 1454118551, win 502, options [nop,nop,T5 val 4109591754 ecr 2296706972], length 0
11:21:01.938979 IP 192.168.0.198.52194 > 168.195.218.201.80: Flags [P.], seq 6:398, ack 1, win 502, options [nop,nop,T5 val 4109591754 ecr 2296706972], length 398: HTTP
: GET / HTTP/1.1
11:21:01.175844 IP 192.168.0.198.52208 > 168.195.218.201.80: Flags [.], ack 3949569501, win 502, options [nop,nop,T5 val 4109591991 ecr 2296706996], length 0
11:21:02.61.321627 IP 192.168.0.198.52194 > 168.195.218.201.80: Flags [.], ack 1429, win 501, options [nop,nop,T5 val 4109592126 ecr 2296707816], length 0
11:21:02.61.315559 IP 192.168.0.198.52194 > 168.195.218.201.80: Flags [.], ack 4285, win 496, options [nop,nop,T5 val 4109592131 ecr 2296707816], length 0
11:21:02.61.315667 IP 192.168.0.198.52194 > 168.195.218.201.80: Flags [.], ack 5713, win 489, options [nop,nop,T5 val 4109592131 ecr 2296707816], length 0
11:21:02.61.315788 IP 192.168.0.198.52194 > 168.195.218.201.80: Flags [.], ack 741, win 481, options [nop,nop,T5 val 4109592131 ecr 2296707810], length 0
11:21:02.61.315883 IP 192.168.0.198.52194 > 168.195.218.201.80: Flags [.], ack 8569, win 495, options [nop,nop,T5 val 4109592131 ecr 2296707810], length 0
11:21:02.61.315962 IP 192.168.0.198.52194 > 168.195.218.201.80: Flags [.], ack 9997, win 501, options [nop,nop,T5 val 4109592131 ecr 2296707810], length 0
11:21:02.61.316663 IP 192.168.0.198.52194 > 168.195.218.201.80: Flags [.], ack 11425, win 501, options [nop,nop,T5 val 4109592132 ecr 2296707810], length 0
11:21:02.61.316156 IP 192.168.0.198.52194 > 168.195.218.201.80: Flags [.], ack 13195, win 496, options [nop,nop,T5 val 4109592132 ecr 2296707810], length 0
11:21:02.61.83397 IP 192.168.0.198.52194 > 168.195.218.201.80: Flags [F.], seq 398, ack 1316, win 501, options [nop,nop,T5 val 4109592134 ecr 2296707816], length 0
11:21:02.62.325405 IP 192.168.0.198.52208 > 168.195.218.201.80: Flags [F.], seq 6:388, ack 3, win 502, options [nop,nop,T5 val 4109592141 ecr 2296706990], length 388: HTTP
: GET /Bluehost.sr20-x20e-ruby.ee_files/bootstrap_rnln.css HTTP/1.1
11:21:02.62.326753 IP 192.168.0.198.52222 > 168.195.218.201.80: Flags [S], seq 716280989, win 64240, options [mss 1400,sackOK,T5 val 4109592142 ecr 0,nope,wscale 7], length 0
24 packets captured
24 packets received by filter
0 packets dropped by kernel
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enps3, link-type EN10MB (Ethernet), capture size 262144 bytes
11:30:38.286634 IP 192.168.0.198.35470 > 91.189.91.48.80: Flags [S], seq 3639913378, win 64240, options [mss 1400,sackOK,T5 val 3398911874 ecr 0,nope,wscale 7], length 0
11:30:38.529336 IP 91.189.91.48.80 > 192.168.0.198.35470: Flags [S.], seq 1460955278, ack 3639913371, win 28960, options [mss 1400,sackOK,T5 val 2296706431 ecr 33989118
```

```
Activities Terminal Fri 11:49 * lab1006@lab1006-HP-280-G4-MT-Business-PC:~
```

File Edit View Search Terminal Help

```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 202144 bytes
11:30:38.280634 IP 192.168.0.198.35476 > 91.189.91.48.80: Flags [S], seq 30399113370, win 64240, options [mss 1460,sackOK,TS val 3398911074 ecr 0,nop,wscale 7], length 8
11:30:38.529336 IP 91.189.91.48.80 > 192.168.0.198.35476: Flags [S.], seq 1460955278, ack 30399113371, win 28960, options [mss 1440,sackOK,TS val 2296764031 ecr 3398911074
74,nop,wscale 9], length 0
11:30:38.529405 IP 192.168.0.198.35476 > 91.189.91.48.80: Flags [., ack 1, win 502, options [nop,nop,TS val 3398911332 ecr 2296764031]], length 6
11:30:38.529607 IP 192.168.0.198.35476 > 91.189.91.48.80: Flags [P.], seq 1:88, ack 1, win 502, options [nop,nop,TS val 3398911323 ecr 2296764031], length 87: HTTP: GET / HTTP/1.1
11:30:38.531091 IP 91.189.91.48.80 > 192.168.0.198.35476: Flags [., ack 88, win 57, options [nop,nop,TS val 2296764032 ecr 3398911323]], length 0
11:30:38.778985 IP 91.189.91.48.80 > 192.168.0.198.35476: Flags [P.], seq 1:190, ack 88, win 57, options [nop,nop,TS val 2296764056 ecr 3398911323], length 189: HTTP: H
TP/1.1 0 Content
11:30:38.778126 IP 192.168.0.198.35476 > 91.189.91.48.80: Flags [., ack 190, win 501, options [nop,nop,TS val 3398911572 ecr 2296764056]], length 8
11:30:38.779720 IP 192.168.0.198.35476 > 91.189.91.48.80: Flags [F.], seq 89, ack 190, win 501, options [nop,nop,TS val 3398911572 ecr 2296764056], length 0
11:30:38.779764 IP 91.189.91.48.80 > 192.168.0.198.35476: Flags [F.], seq 190, ack 88, win 57, options [nop,nop,TS val 2296764056 ecr 3398911323], length 0
11:30:38.779782 IP 192.168.0.198.35476 > 91.189.91.48.80: Flags [., ack 191, win 501, options [nop,nop,TS val 3398911573 ecr 2296764056]], length 0
11:30:38.781291 IP 91.189.91.48.80 > 192.168.0.198.35476: Flags [., ack 89, win 57, options [nop,nop,TS val 2296764057 ecr 3398911572]], length 0
^C
11 packets captured
11 packets received by filter
0 packets dropped by kernel


```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump udp and src port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 202144 bytes
11:35:37.358874 IP _gateway.domain> lab1006-HP-280-G4-MT-Business-PC.54691: 28275 9/3/1 A 185.125.198.18, A 185.125.198.49, A 34.122.121.32, A 35.224
.178.0, A 91.189.91.48, A 35.232.111.17, A 185.125.198.17, A 91.189.91.49 (266)
11:35:37.358883 IP _gateway.domain> lab1006-HP-280-G4-MT-Business-PC.48692: 1745 6/3/1 AAAA 2620:2d:4800:1::2a, AAAA 2620:2d:4800:1::23, AAAA 2001:67c:1562::24, AAAA 2
.602:2d:4006:1::2b, AAAA 2001:67c:1562::23, AAAA 2620:2d:4000:1::22 (296)
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel


```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n portrange 1-80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 202144 bytes
11:36:31.246837 IP 0.0.0.0.68 > 255.255.255.255.68: BOOTP/DHCP, Request From 04:0e:3c:1a:5c:74, length 306
11:36:31.249113 IP 192.168.0.1.0.67 > 255.255.255.255.68: BOOTP/DHCP, Reply, length 306
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel


```
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n src port 443
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 202144 bytes
11:37:57.962988 IP 157.248.242.34.443 > 192.168.0.1.0.46:55996: Flags [S.], seq 4027348222:4027348268, ack 2796636781, win 269, options [nop,nop,TS val 514656924 ecr 19798
```


```


```


```

Aditya Apandkar / Roll no : 05 / T11

```
Activities Terminal Fri 11:49 ●
File Edit View Search Terminal Help
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n src port 443
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:37:57.962988 IP 157.240.242.34.443 > 192.168.0.146.55596: Flags [P.], seq 4027348222:4027348268, ack 2798636781, win 269, options [nop,nop,T5 val 514656924 ecr 1978829422], length 46
11:37:57.962995 IP 157.240.242.34.443 > 192.168.0.146.55596: Flags [P.], seq 0:46, ack 1, win 269, options [nop,nop,T5 val 514656979 ecr 1978829422], length 46
11:37:58.014280 IP 157.240.242.34.443 > 192.168.0.146.55596: Flags [P.], seq 0:46, ack 1, win 269, options [nop,nop,T5 val 514657035 ecr 1978829422], length 46
11:37:58.131395 IP 157.240.242.34.443 > 192.168.0.146.55596: Flags [P.], seq 0:46, ack 1, win 269, options [nop,nop,T5 val 514657152 ecr 1978829422], length 46
11:37:58.354385 IP 157.240.242.34.443 > 192.168.0.146.55596: Flags [P.], seq 0:46, ack 1, win 269, options [nop,nop,T5 val 514657375 ecr 1978829422], length 46
11:37:58.866492 IP 157.240.242.34.443 > 192.168.0.146.55596: Flags [P.], seq 0:46, ack 1, win 269, options [nop,nop,T5 val 514657824 ecr 1978829422], length 46
11:37:59.766705 IP 157.240.242.34.443 > 192.168.0.146.55596: Flags [P.], seq 0:46, ack 1, win 269, options [nop,nop,T5 val 514658727 ecr 1978829422], length 46
11:38:01.498663 IP 157.240.242.34.443 > 192.168.0.146.55596: Flags [P.], seq 0:46, ack 1, win 269, options [nop,nop,T5 val 514668519 ecr 1978829422], length 46
^C
8 packets captured
8 packets received by filter
9 packets dropped by kernel
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump port 80 -w tcpsol
tcpdump: listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C3569 packets captured
3571 packets received by filter
0 packets dropped by kernel
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -r tcpsol
reading from file tcpsol, link-type EN10MB (Ethernet)
11:40:38.313770 IP lab1006-HP-280-G4-MT-Business-PC.37882 > fracktail.canonical.com.http: Flags [S.], seq 2843880018, win 64240, options [mss 1400,sackOK,T5 val 37565146
53 ecr 0,nop,wscale 7], length 0
11:40:38.433298 IP fracktail.canonical.com.http > lab1006-HP-280-G4-MT-Business-PC.37882: Flags [S.], seq 3445806576, ack 2843880019, win 28960, options [mss 1440,sack0
,K,T5 val 2296824622 ecr 3756514653,nop,wscale 9], length 0
11:40:38.433357 IP lab1006-HP-280-G4-MT-Business-PC.37882 > fracktail.canonical.com.http: Flags [.], ack 1, win 502, options [nop,nop,T5 val 3756514773 ecr 2296824622],
length 0
11:40:38.433541 IP lab1006-HP-280-G4-MT-Business-PC.37882 > fracktail.canonical.com.http: Flags [P.], seq 1:80, ack 1, win 502, options [nop,nop,T5 val 3756514773 ecr 2
296824622], length 87: HTTP: GET /HTTP/1.1
11:40:38.434988 IP fracktail.canonical.com.http > lab1006-HP-280-G4-MT-Business-PC.37882: Flags [.], ack 88, win 57, options [nop,nop,T5 val 2296824634 ecr 37
56514773], length 6
11:40:38.557462 IP fracktail.canonical.com.http > lab1006-HP-280-G4-MT-Business-PC.37882: Flags [.], seq 1:190, ack 88, win 57, options [nop,nop,T5 val 2296824634 ecr 37
56514773], length 190
11:40:38.557465 IP fracktail.canonical.com.http > lab1006-HP-280-G4-MT-Business-PC.37882: Flags [F.], seq 190, ack 88, win 57, options [nop,nop,T5 val 2296824634 ecr 37
56514773], length 0
11:40:38.557491 IP lab1006-HP-280-G4-MT-Business-PC.37882 > fracktail.canonical.com.http: Flags [.], ack 190, win 501, options [nop,nop,T5 val 3756514897 ecr 2296824634
], length 0
11:40:38.557725 IP lab1006-HP-280-G4-MT-Business-PC.37882 > fracktail.canonical.com.http: Flags [F.], seq 88, ack 191, win 501, options [nop,nop,T5 val 3756514897 ecr 2
296824634], length 0
11:40:38.562557 IP fracktail.canonical.com.http > lab1006-HP-280-G4-MT-Business-PC.37882: Flags [F.], seq 198, ack 88, win 57, options [nop,nop,T5 val 2296824635 ecr 37
56514773], length 0
11:40:38.562560 IP fracktail.canonical.com.http > lab1006-HP-280-G4-MT-Business-PC.37882: Flags [.], ack 89, win 57, options [nop,nop,T5 val 2296824635 ecr 37
56514773], length 0
```

```
Activities Terminal Fri 11:59 ●
File Edit View Search Terminal Help
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ 
File Edit View Search Terminal Help
kyuhokyulkuyu
^C
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ echo "Hello test" | nc 192.168.46.19
13389
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hostname-I
hostname-I: command not found
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hostname-i
hostname-i: command not found
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ hostname -I
192.168.0.198
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ nc 192.168.0.208 3389
Hello
SHARITTTTTT
^C
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ nc -l -p 3389
Test
Hacked
Hacked
[acdhilfc]
[acdhilfc]

Reading PCAP files
You can read PCAP files by using the -r switch. Note that you can use all the regular commands within tcpdump while reading in a file; you're only limited by the fact that you can't capture and process what doesn't exist in the file already.
#tcpdump -r capture_file

From specific IP and destined for a specific Port
Let's look for all traffic from 10.5.2.3 going to any host on port 3389.
tcpdump -nnvv src 10.5.2.3 and dst port 3389

From One Network to Another
Let's look for all traffic coming from 192.168.x.x and going to the 10.x or 172.16.x.x networks, and we're showing hex output with no hostname resolution and one level of extra verbosity.
tcpdump -nvX src net 192.168.0.0/16 and dst net 10.0.0.0/8 or 172.16.0.0/16

Non ICMP Traffic Going to a Specific IP
This will show us all traffic going to 192.168.0.2 that is not ICMP.
tcpdump dst 192.168.0.2 and src net and not icmp

Complex Grouping and Special Characters
```

Aditya Apandkar / Roll no : 05 / T11

```
Activities Terminal Fri 12:31 * lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[13] & 32!<0'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[13] & 16!=0'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:24:07.369619 IP ec2-52-209-0-233.eu-west-1.compute.amazonaws.com.https : Flags [.], ack 3726312221, win 2096, length 0
12:24:07.369619 IP lab1006-HP-280-G4-MT-Business-PC.53836 :> ec2-52-209-0-233.eu-west-1.compute.amazonaws.com.https: Flags [.], ack 1, win 501, length 8
12:24:08.254667 IP lab1006-HP-280-G4-MT-Business-PC.57588 :> 76.237.128.34.bc.googleusercontent.com.https: Flags [P..], seq 3836136614:3836136818, ack 3902321872, win 985
, options [nop,nop,TS val 2183652865 ts r3 2041279313], length 204
12:24:08.256789 IP 76.237.128.34.bc.googleusercontent.com.https :> lab1006-HP-280-G4-MT-Business-PC.57588: Flags [.], ack 284, win 417, options [nop,nop,TS val 2183652865 ts r3 2041279313], length 6
12:24:08.257613 IP 76.237.128.34.bc.googleusercontent.com.https :> lab1006-HP-280-G4-MT-Business-PC.57588: Flags [P..], seq 1:85, ack 284, win 417, options [nop,nop,TS va l. 2183652658 ts r3 2041279313], length 84
12:24:08.258157 IP lab1006-HP-280-G4-MT-Business-PC.57588 :> 76.237.128.34.bc.googleusercontent.com.https: Flags [P..], seq 264:243, ack 85, win 985, options [nop,nop,TS val 2041279313 ts r3 2183652658], length 39
12:24:08.261518 IP 76.237.128.34.bc.googleusercontent.com.https :> lab1006-HP-280-G4-MT-Business-PC.57588: Flags [.], ack 243, win 417, options [nop,nop,TS val 2183652666 ts r3 2041279317], length 0
12:24:09.255369 IP lab1006-HP-280-G4-MT-Business-PC.57588 :> 76.237.120.34.bc.googleusercontent.com.https: Flags [P..], seq 243:448, ack 85, win 905, options [nop,nop,TS val 2041280314 ts r3 2041280314], length 205
12:24:09.257108 IP 76.237.120.34.bc.googleusercontent.com.https :> lab1006-HP-280-G4-MT-Business-PC.57588: Flags [.], ack 448, win 422, options [nop,nop,TS val 218365365 ts r3 2041280314], length 6
12:24:09.258598 IP 76.237.120.34.bc.googleusercontent.com.https :> lab1006-HP-280-G4-MT-Business-PC.57588: Flags [P..], seq 85:173, ack 448, win 422, options [nop,nop,TS val 2183653658 ts r3 2041280314], length 86
12:24:09.259885 IP lab1006-HP-280-G4-MT-Business-PC.57588 :> 76.237.120.34.bc.googleusercontent.com.https: Flags [P..], seq 448:487, ack 171, win 985, options [nop,nop,TS val 2041280318 ts r3 2183653658], length 39
12:24:09.266247 IP 76.237.120.34.bc.googleusercontent.com.https :> lab1006-HP-280-G4-MT-Business-PC.57588: Flags [.], ack 487, win 422, options [nop,nop,TS val 2041280318], length 6
12:24:09.695585 IP lab1006-HP-280-G4-MT-Business-PC.39846 :> 52.95.119.153.https: Flags [.], ack 214778342, win 501, length 8
12:24:09.695579 IP lab1006-HP-280-G4-MT-Business-PC.39884 :> 52.95.119.153.https: Flags [.], ack 261762129, win 501, length 8
12:24:09.695888 IP lab1006-HP-280-G4-MT-Business-PC.39856 :> 52.95.119.153.https: Flags [.], ack 271762387, win 501, length 8
12:24:09.824773 IP 52.95.119.153.https :> lab1006-HP-280-G4-MT-Business-PC.39884: Flags [.], ack 1, win 942, length 8
12:24:09.824776 IP 52.95.119.153.https :> lab1006-HP-280-G4-MT-Business-PC.39856: Flags [.], ack 1, win 918, length 8
12:24:09.833819 IP 52.95.119.153.https :> lab1006-HP-280-G4-MT-Business-PC.39846: Flags [.], ack 1, win 918, length 8
12:24:09.951859 IP lab1006-HP-280-G4-MT-Business-PC.39858 :> 52.95.119.153.https: Flags [.], ack 1251739929, win 501, length 8
12:24:09.951875 IP lab1006-HP-280-G4-MT-Business-PC.39898 :> 52.95.119.153.https: Flags [.], ack 3320293981, win 501, length 8
12:24:09.951881 IP lab1006-HP-280-G4-MT-Business-PC.39878 :> 52.95.119.153.https: Flags [.], ack 827353120, win 501, length 8
12:24:10.087611 IP 52.95.119.153.https :> lab1006-HP-280-G4-MT-Business-PC.39890: Flags [.], ack 1, win 942, length 8
12:24:10.087631 IP 52.95.119.153.https :> lab1006-HP-280-G4-MT-Business-PC.39858: Flags [.], ack 1, win 918, length 8
12:24:10.090895 IP 52.95.119.153.https :> lab1006-HP-280-G4-MT-Business-PC.39879: Flags [.], ack 1, win 942, length 8
12:24:10.259787 IP lab1006-HP-280-G4-MT-Business-PC.57588 :> 76.237.120.34.bc.googleusercontent.com.https: Flags [P..], seq 487:692, ack 171, win 995, options [nop,nop,TS val 218365366 ts r3 2041280318], length 204
```

```
Activities Terminal Fri 12:31 * lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[13] & 8!=0'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:24:33.844595 IP lab1006-HP-280-G4-MT-Business-PC.41886 > whatsapp-cdn-shv-02-bomi.firebaseio.net.https: Flags [P.], seq 1516425139:1516425208, ack 2125073013, win 501, options [nop,nop,TS val 1983406747 ecr 2028579252], length 69
12:24:34.026941 IP whatsapp-cdn-shv-02-bomi.firebaseio.net.https > lab1006-HP-280-G4-MT-Business-PC.41886: Flags [P.], seq 1:72, ack 69, win 270, options [nop,nop,TS val 80867538 ecr 1983406747], length 71
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[13] & 4!=0'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:24:50.965593 IP 119.9.106.191.https > lab1006-HP-280-G4-MT-Business-PC.53794: Flags [R], seq 3658212265, win 0, length 0
12:24:50.965593 IP 119.9.106.191.https > lab1006-HP-280-G4-MT-Business-PC.53794: Flags [R], seq 3658212265, win 0, length 0
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[13] & 2!=0'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:25:38.297345 IP lab1006-HP-280-G4-MT-Business-PC.46282 > 17.111.232.35.bc.googleusercontent.com.http: Flags [S], seq 1049381640, win 64240, options [mss 1460,sackOK, TS val 31111494561 ecr 0,nop,wscale 7], length 0
12:25:38.536732 IP 17.111.232.35.bc.googleusercontent.com.http > lab1006-HP-280-G4-MT-Business-PC.46282: Flags [S.], seq 368966158, ack 1049381641, win 64768, options [mss 1420,sackOK, TS val 898771948 ecr 31111494561,nop,wscale 7], length 0
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[13] & 1!=0'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:26:39.814493 IP lab1006-HP-280-G4-MT-Business-PC.47156 > server-18-66-56-171.bom78.r.cloudfront.net.https: Flags [F.], seq 3270048631, ack 3622424537, win 501, options [nop,nop,TS val 122865534 ecr 4183642191], length 0
12:26:39.814493 IP lab1006-HP-280-G4-MT-Business-PC.48548 > 151.101.153.51.https: Flags [F.], seq 3412498398, ack 41732237, win 501, options [nop,nop,TS val 3304048563 ecr 99382658], length 0
12:26:39.815329 IP lab1006-HP-280-G4-MT-Business-PC.48544 > 151.101.153.51.https: Flags [F.], seq 3169162883, ack 510392666, win 501, options [nop,nop,TS val 3304648564 ecr 2386215411], length 0
12:26:39.815746 IP lab1006-HP-280-G4-MT-Business-PC.47820 > server-18-66-53-3.bom78.r.cloudfront.net.https: Flags [F.], seq 4639715831, ack 2832577795, win 501, options [nop,nop,TS val 4139865384 ecr 4162659999], length 0
12:26:39.816294 IP server-18-66-56-171.bom78.r.cloudfront.net.https > lab1006-HP-280-G4-MT-Business-PC.47156: Flags [F.], seq 1, ack 0, win 135, options [nop,nop,TS val 4139865333 ecr 1226695533], length 0
12:26:39.817379 IP server-18-66-56-171.bom78.r.cloudfront.net.https > lab1006-HP-280-G4-MT-Business-PC.47020: Flags [F.], seq 1, ack 0, win 146, options [nop,nop,TS val 4
```

Aditya Apandkar / Roll no : 05 / T11

```
Activities Terminal Fri 12/31 40
File Edit View Search Terminal Help
19 packets captured
46 packets received by filter
21 packets dropped by kernel
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[13]=18'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:27:49.368328 IP 123.26.120.34.bc.googleusercontent.com.https > lab1006-HP-280-G4-MT-Business-PC.53298: Flags [S.], seq 2503443221, ack 3916996776, win 65535, options [mss 1412,sackOK,T5 val 97346372 ecr 3379345315,nop,wscale 8], length 0
12:27:49.538954 IP server-108-159-56-32.bom78.r.cloudfront.net.https > lab1006-HP-280-G4-MT-Business-PC.32832: Flags [S.], seq 196105776, ack 2921971945, win 65535, options [mss 1412,sackOK,T5 val 4155621541 ecr 3299662687,nop,wscale 9], length 0
12:27:49.538958 IP server-108-159-56-32.bom78.r.cloudfront.net.https > lab1006-HP-280-G4-MT-Business-PC.32838: Flags [S.], seq 1962105846, ack 198221063, win 65535, options [mss 1412,sackOK,T5 val 754630931 ecr 3299662687,nop,wscale 9], length 0
12:27:49.557281 IP server-108-159-56-32.bom78.r.cloudfront.net.https > lab1006-HP-280-G4-MT-Business-PC.32840: Flags [S.], seq 1406577553, ack 824697447, win 65535, options [mss 1412,sackOK,T5 val 1859848018 ecr 3299662714,nop,wscale 9], length 0
12:27:50.167899 IP server-108-159-58-51.bom78.r.cloudfront.net.https > lab1006-HP-280-G4-MT-Business-PC.35386: Flags [S.], seq 1916072751, ack 3475858950, win 65535, options [mss 1440,sackOK,T5 val 1387521373 ecr 828466162,nop,wscale 9], length 0
12:27:50.254622 IP server-18-66-53-3.bon78.r.cloudfront.net.https > lab1006-HP-280-G4-MT-Business-PC.40488: Flags [S.], seq 1937510214, ack 487951999, win 65535, options [mss 1440,sackOK,T5 val 3986566261 ecr 4135156620,nop,wscale 9], length 0
12:27:50.259570 IP server-108-159-46-96.bom78.r.cloudfront.net.https > lab1006-HP-280-G4-MT-Business-PC.37800: Flags [S.], seq 799399648, ack 550412823, win 65535, options [mss 1440,sackOK,T5 val 1819952951 ecr 3919178882,nop,wscale 9], length 0
12:27:50.277869 IP 151.161.151.51.https > lab1006-HP-280-G4-MT-Business-PC.41690: Flags [S.], seq 3018251566, ack 866497868, win 65535, options [mss 1440,sackOK,T5 val 268284387 ecr 3394719824,nop,wscale 9], length 0
12:27:50.286118 IP ec2-52-229-30-166.eu-west-1.compute.amazonaws.com.https > lab1006-HP-280-G4-MT-Business-PC.37818: Flags [S.], seq 2098788224, ack 4808592788, win 26847, options [mss 1440,sackOK,T5 val 716934785 ecr 3304719937,nop,wscale 8], length 0
12:27:50.310885 IP 67-226-224.106.https > lab1006-HP-280-G4-MT-Business-PC.38026: Flags [S.], seq 1601343639, ack 329951764, win 8190, options [mss 1440,nop,wscale 6,no p,nop,sackOK], length 0
12:27:50.316899 IP server-108-159-59-226.bom78.r.cloudfront.net.https > lab1006-HP-280-G4-MT-Business-PC.60334: Flags [S.], seq 443886901, ack 792632056, win 65535, options [mss 1440,sackOK,T5 val 1463204132 ecr 633923446,nop,wscale 9], length 0
12:27:50.371189 IP 151.161.151.51.https > lab1006-HP-280-G4-MT-Business-PC.41702: Flags [S.], seq 4117814626, ack 10492846, win 65535, options [mss 1440,sackOK,T5 val 200778395 ecr 3304719917,nop,wscale 9], length 0
12:27:50.387331 IP 67-226-224.106.https > lab1006-HP-280-G4-MT-Business-PC.38044: Flags [S.], seq 3767804753, ack 4263773101, win 8190, options [mss 1440,nop,wscale 6,nop,op,sackOK], length 0
12:27:50.394985 IP 67-226-224.106.https > lab1006-HP-280-G4-MT-Business-PC.38038: Flags [S.], seq 3011425597, ack 847932931, win 8190, options [mss 1440,nop,wscale 6,no p,nop,sackOK], length 6
12:27:50.461502 IP ec2-52-289-8-233.eu-west-1.compute.amazonaws.com.https > lab1006-HP-280-G4-MT-Business-PC.57732: Flags [S.], seq 2123115401, ack 233922537, win 26883, options [mss 1440,nop,wscale 5], length 0
12:27:50.663734 IP 67-226-224.106.https > lab1006-HP-280-G4-MT-Business-PC.38068: Flags [S.], seq 3026615746, ack 4046318232, win 8190, options [mss 1440,nop,wscale 6,nop,op,sackOK], length 0
12:27:50.664139 IP 67-226-224.106.https > lab1006-HP-280-G4-MT-Business-PC.38074: Flags [S.], seq 1822509215, ack 3764298877, win 8190, options [mss 1440,nop,wscale 6,nop,op,sackOK], length 0
12:27:50.683762 IP server-108-159-80-74.bom78.r.cloudfront.net.https > lab1006-HP-280-G4-MT-Business-PC.57520: Flags [S.], seq 3731408285, ack 776804246, win 65535, options [mss 1440,sackOK,T5 val 226243831 ecr 1469261394,nop,wscale 9], length 0
12:27:50.748586 IP 67-226-224.106.https > lab1006-HP-280-G4-MT-Business-PC.38084: Flags [S.], seq 2461978166, ack 251027220, win 8190, options [mss 1440,nop,wscale 6,no p,nop,sackOK], length 0
```

```
Activities Terminal Fri 12/31 40
File Edit View Search Terminal Help
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[tcpflags]==tcp-syn'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:28:30.768772 IP lab1006-HP-280-G4-MT-Business-PC.59802 > 76.237.120.34.bc.googleusercontent.com.https: Flags [S.], seq 2590485627, win 64240, options [mss 1460,sackOK,T5 val 2045141827 ecr 0,nop,wscale 7], length 0
12:28:34.244410 IP lab1006-HP-280-G4-MT-Business-PC.43380 > 20.207.73.82.https: Flags [S.], seq 2490752872, win 64240, options [mss 1460,sackOK,T5 val 2633959014 ecr 0,nop,op,sackOK], length 0
12:28:34.362673 IP lab1006-HP-280-G4-MT-Business-PC.37904 > cdn-185-199-109-154.github.com.https: Flags [S.], seq 2585262118, win 64240, options [mss 1460,sackOK,T5 val 2313739940 ecr 0,nop,wscale 7], length 0
12:28:34.387879 IP lab1006-HP-280-G4-MT-Business-PC.37914 > cdn-185-199-109-154.github.com.https: Flags [S.], seq 1102515088, win 64240, options [mss 1460,sackOK,T5 val 2313739965 ecr 0,nop,wscale 7], length 0
12:28:34.667608 IP lab1006-HP-280-G4-MT-Business-PC.49318 > lb-140-82-113-21-lab.github.com.https: Flags [S.], seq 1975083633, win 64240, options [mss 1460,sackOK,T5 val 2323260391 ecr 0,nop,wscale 7], length 0
12:28:34.918486 IP lab1006-HP-280-G4-MT-Business-PC.49334 > lb-140-82-113-21-lab.github.com.https: Flags [S.], seq 1038676714, win 64240, options [mss 1460,sackOK,T5 val 2323260642 ecr 0,nop,wscale 7], length 0
12:28:34.936921 IP lab1006-HP-280-G4-MT-Business-PC.34766 > 20.207.73.85.https: Flags [S.], seq 1066565829, win 64240, options [mss 1460,sackOK,T5 val 4136118630 ecr 0,nop,op,scale 7], length 0
^C
7 packets captured
0 packets received by Filter
0 packets dropped by kernel
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[tcpflags]==tcp-rst'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:28:59.611444 IP lab1006-HP-280-G4-MT-Business-PC.56670 > 104.17.24.14.https: Flags [R], seq 2584235300, win 0, length 0
12:28:59.614399 IP lab1006-HP-280-G4-MT-Business-PC.56670 > 104.17.24.14.https: Flags [R], seq 2584235325, win 0, length 0
12:29:00.496422 IP lab1006-HP-280-G4-MT-Business-PC.42140 > bom7s32-1n-f3.ie100.net.https: Flags [R], seq 1817942852, win 0, length 0
12:29:00.496322 IP lab1006-HP-280-G4-MT-Business-PC.42140 > bom7s32-1n-f3.ie100.net.https: Flags [R], seq 1817942852, win 0, length 0
12:29:00.496444 IP lab1006-HP-280-G4-MT-Business-PC.42140 > bom7s32-1n-f3.ie100.net.https: Flags [R], seq 1817942852, win 0, length 0
12:29:00.496445 IP lab1006-HP-280-G4-MT-Business-PC.42140 > bom7s32-1n-f3.ie100.net.https: Flags [R], seq 1817942852, win 0, length 0
12:29:00.511165 IP lab1006-HP-280-G4-MT-Business-PC.42144 > bom7s32-1n-f3.ie100.net.https: Flags [R], seq 961112086, win 0, length 0
12:29:00.511241 IP lab1006-HP-280-G4-MT-Business-PC.42144 > bom7s32-1n-f3.ie100.net.https: Flags [R], seq 961112086, win 0, length 0
12:29:00.511244 IP lab1006-HP-280-G4-MT-Business-PC.42144 > bom7s32-1n-f3.ie100.net.https: Flags [R], seq 961112086, win 0, length 0
^C
9 packets captured
0 packets received by Filter
0 packets dropped by kernel
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump 'tcp[tcpflags]==tcp-fin'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by Filter
0 packets dropped by kernel
Lab1006@lab1006-HP-280-G4-MT-Business-PC:~$
```

Conclusion: In this lab assignment, we engaged with the TCPDUMP packet sniffer tool, mastering the capture and analysis of network packets. This hands-on experience deepened our comprehension of network intricacies and security considerations.

Lab Assignment No.8

Aim: Installation of NMAP and using it with different options to scan open ports, perform OS fingerprinting, do a ping scan, tcp port scan, udp port scan, etc.

Lab Outcome Attained: LO4 – Use tools like sniffers, port scanners and other related tools for analyzing packets in a network.

Theory:

1. What is Port Scanning? What is NMAP?

Port Scanning:

Port scanning is a technique employed in computer networking to examine the state of network ports on a target system. By systematically probing a range of ports, typically using TCP or UDP packets, port scanning aims to uncover open ports where services or applications are listening for incoming connections. The information gleaned from port scans is crucial for network administrators and security professionals to understand a system's security posture and potential vulnerabilities. However, it's essential to use port scanning responsibly and ethically, as indiscriminate or malicious port scanning can disrupt networks and violate security policies.

Nmap (Network Mapper):

Nmap, short for Network Mapper, is a robust and open-source network scanning tool widely used for network exploration and security auditing. With its extensive set of features, Nmap enables users to discover hosts, identify open ports, determine running services, and even perform OS fingerprinting. It empowers network administrators to assess the security of their systems and assists security professionals in penetration testing and vulnerability assessment. Nmap's flexibility and scripting capabilities make it a versatile tool for in-depth network analysis, ensuring that networks are well-protected and resilient against potential threats.

2. Explain in brief different states of port. (open, closed, filtered, unfiltered, open | filtered and closed | filtered).

Different states of ports in the context of port scanning are crucial for understanding the accessibility and security of network services on a target system. Here's a brief explanation of each state:

1. Open: An "open" port means that the target system is actively listening for incoming connections on that port. This suggests that a service or application is running and available to accept connections, making it accessible to potential clients.
2. Closed: A "closed" port indicates that there is no service actively listening on the specified port. The system has responded to the port scan with a TCP RST (Reset) packet, signifying that no application is waiting for connections. Closed ports are typically considered safe from an attack perspective.

3. Filtered: A "filtered" port means that the target system has actively blocked access to the port. This block can occur due to a firewall, router, or some other security mechanism. The scanning tool did not receive a response from the port, making it unclear whether the port is open or closed.

4. Unfiltered: An "unfiltered" port suggests that the scanning tool received a response from the target system, but it remains inconclusive whether the port is open or closed. Unfiltered ports are typically more informative than filtered ones but still require further investigation.

5. Open | Filtered: This state indicates that the scanning tool could not definitively determine whether the port is open or filtered. It implies that there may be a service running on the port, but the system is also actively blocking access.

6. Closed | Filtered: This state signifies that the scanning tool could not conclusively determine whether the port is closed or filtered. It suggests that the system may respond to the port scan with a closed indication but could also be employing filtering mechanisms that affect the result.

Understanding these port states is essential for network administrators and security professionals to assess the security posture of a system, configure firewalls, and identify potential vulnerabilities or misconfigurations in network services.

3. Write the commands for following type of port scanning techniques using NMAP.

The commands are as follows -

1. TCP Connect Scan:

- Description: TCP Connect Scan, `‐sT`, establishes a full TCP connection to target ports. Ports are marked as open if a successful connection is made, and closed if the connection attempt fails. It's straightforward but easily detectable.

- Syntax: `nmap -sT <ipaddress>`

2. TCP SYN Scan:

- Description: TCP SYN Scan, `‐sS`, sends SYN packets to target ports and analyzes responses. It marks ports as open if a SYN/ACK response is received, and closed if an RST response is received. This scan is stealthier as it avoids completing the TCP handshake.

- Syntax: `nmap -sS <ipaddress>`

3. TCP FIN Scan:

- Description: TCP FIN Scan, `‐sF`, sends TCP packets with only the FIN flag set to identify open ports based on their response behavior. Closed ports usually respond with an RST, while open ports often drop the packet. Effective on systems following RFC 793, except Microsoft Windows.

- Syntax: `nmap -sF <ipaddress>`

4. TCP NULL Scan:

- Description: TCP NULL Scan, `‐sN‐, sends TCP packets with no set flags to identify open and closed ports based on their response behavior. Open ports typically don't respond, while closed ports usually send an RST. Can evade certain firewall and intrusion detection systems.

- Syntax: `nmap -sN <ipaddress>`

5. XMAS Scan:

- Description: The XMAS Tree Scan sets the FIN, URG, and PUSH flags in TCP packets. It aims to identify open ports based on kernel response behavior. Syntax is similar to other scans with flag options.

- Syntax: `nmap -sX <ipaddress>`

6. Ping Sweep:

- Description: Ping Sweep, `‐sP‐, lists hosts within a specified IP range that respond to ICMP ping requests. Useful for identifying online hosts without probing individual ports.

- Syntax: `nmap -sP <IP address range>`

7. ACK Scan:

- Description: The ACK Scan sends ACK packets to target ports and categorizes them as unfiltered (RST response) or filtered (no response). Helps determine if a firewall is stateless or stateful.

- Syntax: `nmap -sA <IP address range>`

8. Scanning Range of Ports:

- Description: Nmap can scan specific ports or a range of ports on a target IP address. Use `‐p <port>` for a specific port or `‐p <start port>‐<end port>` for a range.

- Syntax for Specific Port: `nmap -p <port> <ipaddress>`

- Syntax for Port Range: `nmap -p <start port>‐<end port> <ipaddress>`

9. Version Detection:

- Description: Version Detection, `‐sV‐, collects information about the specific service running on open ports. Identifies product names and version numbers.

- Syntax: `nmap -sV -p <port range> <ipaddress>`

10. Comprehensive Scan with OS Fingerprinting:

- Description: This command performs a comprehensive scan that includes OS fingerprinting, service identification, and scans all ports within a specified IP range.

- Syntax: `nmap -sV -p <port range> <IP address range>`

11. Operating System Fingerprinting:

- **Description:** Operating System Fingerprinting, ` -O `, determines the remote host's operating system by analyzing responses to various TCP and UDP packets. Nmap compares the results to its extensive database of known OS fingerprints.

- Syntax: `nmap -O <ipaddress>`

Output Screenshots:

Activities Terminal Fri 11:43 lab1006@lab1006-HP-280-G4-MT-Business-PC: ~

```
File Edit View Search Terminal Help
11:25:44.666623 IP 34.117.65.55.443 > 192.168.0.153.34564: Flags [.], ack 29, win 27
2, options [nop,nop,T5 val 2307436602 ecr 1912084142], length 0
^C
3 packets captured
3 packets received by filter
0 packets dropped by kernel
labi1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:27:44.865881 IP 192.168.0.153.64955 > 192.168.0.153.80: Flags [.], ack 3460778370
, win 128, length 0
11:27:44.865933 IP 192.168.0.153.80 > 192.168.0.153.64955: Flags [R], seq 3460778370
, win 0, length 0
11:30:06.797185 IP 192.168.0.153.54162 > 91.189.91.49.80: Flags [S], seq 1381206552
, win 64248, options [mss 1460,sockOK,T5 val 3027565668 ecr 0,nop,wscale 7], length 0
11:30:07.176846 IP 91.189.91.49.80 > 192.168.0.153.54162: Flags [S.], seq 1397315163
, ack 120210553, win 65100, options [mss 1440,sockOK,T5 val 4200720725 ecr 30275056
68,nop,wscale 7], length 0
11:30:07.178787 IP 192.168.0.153.54162 > 91.189.91.49.80: Flags [P.], ack 1, win 502
, options [nop,nop,T5 val 3027565871 ecr 4200720725], length 0
11:30:07.178891 IP 192.168.0.153.54162 > 91.189.91.49.80: Flags [P.], seq 1:198, ack
1, win 502, options [nop,nop,T5 val 3027565871 ecr 4200720725], length 80
11:30:07.380913 IP 91.189.91.49.80 > 192.168.0.153.54162: Flags [P.], seq 1:198, ack
88, win 509, options [nop,nop,T5 val 4200720725 ecr 3027565871], length 189: HTTP:
HTTP/1.1 204 No Content
11:30:07.380964 IP 192.168.0.153.54162 > 91.189.91.49.80: Flags [.], ack 190, win 50
1, options [nop,nop,T5 val 3027565871 ecr 4200720725], length 0
11:30:07.381712 IP 192.168.0.153.54162 > 91.189.91.49.80: Flags [F.], seq 88, ack 19
0, win 501, options [nop,nop,T5 val 3027565697 ecr 4200720725], length 0
11:30:07.381885 IP 91.189.91.49.80 > 192.168.0.153.54162: Flags [F.], seq 190, ack 8
9, win 509, options [nop,nop,T5 val 4200720725 ecr 3027565871], length 0
11:30:07.384915 IP 192.168.0.153.54162 > 91.189.91.49.80: Flags [.], ack 191, win 50
1, options [nop,nop,T5 val 3027565694 ecr 4200720725], length 0
11:30:07.583148 IP 91.189.91.49.80 > 192.168.0.153.54162: Flags [.], ack 89, win 509
1, options [nop,nop,T5 val 4200721130 ecr 3027565073], length 0
^C
12 packets captured
12 packets received by filter
0 packets dropped by kernel
labi1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo tcpdump -n port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:34:01.141382 IP 192.168.0.153.49068 > 192.168.0.1:80: Flags [S], seq 224886266
, win 64246, options [mss 1460,sockOK,T5 val 3210394179 ecr 0,nop,wscale 7], length 0
File Edit View Search Terminal Help
labi1006@lab1006-HP-280-G4-MT-Business-PC: ~
```

File Edit View Search Terminal Help
labi1006@lab1006-HP-280-G4-MT-Business-PC:~\$ sudo nmap -sS 192.168.0.1
sudo password for labi1006:

```
starting Nmap 7.60 ( https://nmap.org ) at 2023-09-01 11:22 IST
nmap scan report for _gateway (192.168.0.1)
lost is up (0.01ms latency).
192.168.0.1      PORT      STATE SERVICE
3/TCP open       domain
80/TCP open      http
443/TCP open     https
800/TCP open    upnp
MAC Address: AC:15:A2:89:9E:29 (Unknown)

Map done: 1 IP address (1 host up) scanned in 1.62 seconds
labi1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo nmap -sT 192.168.0.1
starting Nmap 7.60 ( https://nmap.org ) at 2023-09-01 11:33 IST
nmap scan report for _gateway (192.168.0.1)
lost is up (0.0092s latency).
192.168.0.1      PORT      STATE SERVICE
3/TCP open       domain
80/TCP open      http
443/TCP open     https
800/TCP open    upnp
MAC Address: AC:15:A2:89:9E:29 (Unknown)

Map done: 1 IP address (1 host up) scanned in 1.62 seconds
labi1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo nmap -sT 192.168.0.1
starting Nmap 7.60 ( https://nmap.org ) at 2023-09-01 11:34 IST
nmap scan report for _gateway (192.168.0.1)
lost is up (0.01ms latency).
192.168.0.1      PORT      STATE SERVICE
3/TCP open       domain
80/TCP open      http
443/TCP open     https
800/TCP open    upnp
MAC Address: AC:15:A2:89:9E:29 (Unknown)

Map done: 1 IP address (1 host up) scanned in 0.38 seconds
labi1006@lab1006-HP-280-G4-MT-Business-PC:~$ sudo nmap -sN 192.168.0.1
```

Aditya Apandkar / T11 / Roll no : 05

Aditya Apandkar / T11 / Roll no : 05

```
Activities Terminal lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
11:53:12.521376 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [P..], seq 1:18, ack 1, win 227, options [nop,nop,TS val 215543230 ecr 3211545537], length 17: HTTP: HTT
11:53:12.521424 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [.], ack 18, win 502, options [nop,nop,TS val 3211545533 ecr 215543330], length 0
11:53:12.521853 IP 192.168.0.1.80 > 192.168.0.1.80: Flags [P..], seq 18:109, ack 19, win 227, options [nop,nop,TS val 215543230 ecr 3211545538], length 91: HTTP: HTT
11:53:12.521874 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [.], ack 109, win 503, options [nop,nop,TS val 3211545533 ecr 215543230], length 6
11:53:12.522027 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [FP.], seq 19:53, ack 19, win 227, options [nop,nop,TS val 215543230 ecr 3211545538], length 394: HTT
11:53:12.523756 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [F.], seq 19, ack 50, options [nop,nop,TS val 3211545539 ecr 215543328], length 8
11:53:12.523769 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [S], seq 2328736679, win 64248, options [mss 1460,sackOK,TOS val 3211545530 ecr 0,nop,wscale 7], length 0
11:53:12.533356 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [.], ack 20, win 227, options [nop,nop,TS val 215543233 ecr 3211545530], length 6
11:53:12.533384 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [S.], seq B10160005, ack 2328736680, win 14488, options [mss 1460,sackOK,TOS val 215543233 ecr 321154553], length 0
11:53:12.533392 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [.], ack 1, win 502, options [nop,nop,TS val 3211545531 ecr 215543233], length 8
11:53:12.533634 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [P..], seq 1:23, ack 1, win 502, options [nop,nop,TS val 3211545531 ecr 215543233], length 22: HTTP: OPT IONS /HTTP/1.0
11:53:12.534150 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [.], ack 23, win 227, options [nop,nop,TS val 215543233 ecr 3211545531], length 0
11:53:12.534226 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [.], ack 23, win 227, options [nop,nop,TS val 215543242 ecr 3211545531], length 101: HTTP: HTT
11:53:12.571763 IP 192.168.0.1.80 > 192.168.0.1.80: Flags [.], ack 102, win 58, options [nop,nop,TS val 3211545539 ecr 215543242], length 0
11:53:12.571913 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [FP.], seq 182:123, ack 23, win 227, options [nop,nop,TS val 3211545424 ecr 3211545531], length 21: HTTP: HTT
11:53:12.587020 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [F.], seq 23, ack 12, options [nop,nop,TS val 3211545084 ecr 215543242], length 0
11:53:12.587410 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [.], ack 24, win 227, options [nop,nop,TS val 215543246 ecr 3211545604], length 0
`c
39 packets captured
39 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
File Edit View Search Terminal Help
11:53:12.521376 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [P..], seq 1:18, ack 1, win 227, options [nop,nop,TS val 215543230 ecr 3211545537], length 17: HTTP: HTT
11:53:12.521424 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [.], ack 18, win 502, options [nop,nop,TS val 3211545533 ecr 215543330], length 0
11:53:12.521853 IP 192.168.0.1.80 > 192.168.0.1.80: Flags [P..], seq 18:109, ack 19, win 227, options [nop,nop,TS val 215543230 ecr 3211545538], length 91: HTTP: HTT
11:53:12.521874 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [.], ack 109, win 503, options [nop,nop,TS val 3211545533 ecr 215543230], length 6
11:53:12.522027 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [FP.], seq 19:53, ack 19, win 227, options [nop,nop,TS val 215543230 ecr 3211545538], length 394: HTT
11:53:12.523756 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [F.], seq 19, ack 50, options [nop,nop,TS val 3211545539 ecr 215543328], length 8
11:53:12.523769 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [S], seq 2328736679, win 64248, options [mss 1460,sackOK,TOS val 215543233 ecr 321154553], length 0
11:53:12.533356 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [.], ack 20, win 227, options [nop,nop,TS val 215543233 ecr 3211545530], length 6
11:53:12.533384 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [S.], seq B10160005, ack 2328736680, win 14488, options [mss 1460,sackOK,TOS val 215543233 ecr 321154553], length 0
11:53:12.533392 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [.], ack 1, win 502, options [nop,nop,TS val 3211545531 ecr 215543233], length 8
11:53:12.533634 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [P..], seq 1:23, ack 1, win 502, options [nop,nop,TS val 3211545531 ecr 215543233], length 22: HTTP: OPT IONS /HTTP/1.0
11:53:12.534150 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [.], ack 23, win 227, options [nop,nop,TS val 215543233 ecr 3211545531], length 0
11:53:12.534226 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [.], ack 23, win 227, options [nop,nop,TS val 215543242 ecr 3211545531], length 101: HTTP: HTT
11:53:12.571763 IP 192.168.0.1.80 > 192.168.0.1.80: Flags [.], ack 102, win 58, options [nop,nop,TS val 3211545539 ecr 215543242], length 0
11:53:12.571913 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [FP.], seq 182:123, ack 23, win 227, options [nop,nop,TS val 3211545424 ecr 3211545531], length 21: HTTP: HTT
11:53:12.587020 IP 192.168.0.153.37760 > 192.168.0.1.80: Flags [F.], seq 23, ack 12, options [nop,nop,TS val 3211545084 ecr 215543242], length 0
11:53:12.587410 IP 192.168.0.1.80 > 192.168.0.153.37760: Flags [.], ack 24, win 227, options [nop,nop,TS val 215543246 ecr 3211545604], length 0
`c
39 packets captured
39 packets received by filter
0 packets dropped by kernel
lab1006@lab1006-HP-280-G4-MT-Business-PC: ~
```

Aditya Apandkar / T11 / Roll no : 05

Conclusion: In this lab assignment, we successfully installed Nmap and explored its various options, including scanning open ports, performing OS fingerprinting, conducting ping scans, TCP port scans, and UDP port scans, enabling comprehensive network reconnaissance and analysis.

Lab Assignment No.9

Aim: Simulate DOS attack using Hping3.

Lab Outcome Attained: LO5 – To use Continuous Monitoring Tools to resolve any system errors (low memory, unreachable server etc.) before they have any negative impact on the business productivity.

Theory:

1. What is Denial of Service Attack?

A Denial of Service (DoS) attack is a malicious attempt to disrupt the normal functioning of a target server, network, or service by overwhelming it with a flood of illegitimate requests or traffic. The goal of a DoS attack is to make the targeted resource or service unavailable to its intended users, causing disruption or downtime.

2. Explain SYN flood, ICMP flood and SMURF attack.

1. SYN Flood Attack:

In a SYN Flood attack, an attacker sends a flood of TCP connection requests (SYN packets) to a target server but never completes the three-way handshake (SYN, SYN-ACK, ACK) to establish a connection.

This causes the target server to allocate resources for each incomplete connection request, leading to resource exhaustion and potentially rendering the server inaccessible to legitimate users.

2. ICMP Flood Attack:

An ICMP (Internet Control Message Protocol) Flood attack involves sending a high volume of ICMP echo request packets (ping requests) to a target server or network.

The target server responds to each ping request with a ping reply (ICMP echo reply) and can become overwhelmed when receiving a flood of these requests, leading to network congestion or resource exhaustion.

3. SMURF Attack:

A SMURF attack is a type of amplification attack that exploits the ICMP protocol. The attacker sends ICMP echo requests (pings) with a forged source IP address of the victim to multiple hosts on a network.

These hosts, believing the victim's IP address is the source, respond to the ICMP echo requests by sending ICMP echo replies (pings) to the victim's IP address.

This can result in a massive amount of network traffic directed at the victim, potentially causing a DoS condition.

3. Write the Hping3 commands used for performing SYN flood and ICMP flood.

Hping3 SYN Flood Command:

To perform a SYN Flood attack using hping3, you can use the following command:

```
sudo hping3 -c 10000 -d 120 -S -w 64 -p 80 --flood <target_ip>
```

In this command:

- c 10000: Send 10,000 packets.
- d 120: Set the data size in bytes to 120.
- S: Send TCP SYN packets.
- w 64: Set the TCP window size to 64.
- p 80: Specify the target port (port 80 in this example).
- flood: Flood mode, sending packets as fast as possible.
- <target_ip>: Replace this with the IP address of the target.

Hping3 ICMP Flood Command:

To perform an ICMP Flood attack using hping3, you can use the following command:

```
sudo hping3 --icmp --flood -a 192.168.103.1 192.168.1.255
```

In this command:

- icmp: Specifies ICMP mode for sending ICMP echo requests.
- flood: Activates flood mode, sending packets as quickly as possible.
- a 192.168.103.1: Sets the source IP address as 192.168.103.1 (you should replace this with the actual source IP address you want to use).
- 192.168.1.255: Specifies the target address as a broadcast address on the local network (you should replace this with the target IP address or network you want to target).

Output Screenshots:

```
Preparing to unpack .../hping3_3.a2.ds2-10_and64.deb ...
Unpacking hping3 (3.a2.ds2-10) ...
Setting up hping3 (3.a2.ds2-10) ...
Processing triggers for man-db (2.18.2-1) ...
[altaf@LAPTOP-DGNIK4U9:~]$ man hping3
[altaf@LAPTOP-DGNIK4U9:~]$ hping3 -1 --flood -a 192.168.1.103 192.168.1.255
[open_sockraw] socket(): Operation not permitted
[main] can't open raw socket
[altaf@LAPTOP-DGNIK4U9:~]$ sudo hping3 -1 --flood -a 192.168.1.103 192.168.1.255
[sudo] password for altaf:
HPING 192.168.1.255 (eth0 192.168.1.255): icmp mode set, 28 headers + 8 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.255 hping statistic ---
352510 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
[altaf@LAPTOP-DGNIK4U9:~]$ sudo hping3 -c 15000 -d 120 -w 64 -p 80 --flood --rand-source 192.168.1.159
[sudo] password for altaf:
hping3: you must specify only one target host at a time
[altaf@LAPTOP-DGNIK4U9:~]$ sudo hping3 -c 15000 -d 120 -w 64 -p 80 --flood --rand-source 192.168.1.159
hping3: you must specify only one target host at a time
[altaf@LAPTOP-DGNIK4U9:~]$ sudo hping3 -c 15000 -d 120 -w 64 -p 80 --flood --rand-source 192.168.1.159
HPING 192.168.1.159 (eth0 192.168.1.159): NO FLAGS are set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.159 hping statistic ---
#8761 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
[altaf@LAPTOP-DGNIK4U9:~]$ sudo hping3 -c 15000 -d 120 -w 64 -p 80 --flood --rand-source 192.168.1.159
HPING 192.168.1.159 (eth0 192.168.1.159): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.159 hping statistic ---
232030 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
[altaf@LAPTOP-DGNIK4U9:~]$ ^C
[altaf@LAPTOP-DGNIK4U9:~]$ ^C
[altaf@LAPTOP-DGNIK4U9:~]$ ^C
[altaf@LAPTOP-DGNIK4U9:~]$ ^C
```

The terminal window shows the following sequence of commands and output:

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  hping3
0 upgraded, 1 newly installed, 0 to remove and 140 not upgraded.
Need to get 106 kB of archives.
After this operation, 263 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 hping3 amd64 3.a2.ds2-10 [106 kB]
Fetched 106 kB in 2s (64.4 kB/s)
Selecting previously unselected package hping3.
(Reading database ... 53952 files and directories currently installed.)
Preparing to unpack .../hping3_3.a2.ds2-10_amd64.deb ...
Unpacking hping3 (3.a2.ds2-10) ...
Setting up hping3 (3.a2.ds2-10) ...
Processing triggers for man-db (2.18.2-1) ...
altaf@LAPTOP-DGNIIH4U9:~$ man hping3
altaf@LAPTOP-DGNIIH4U9:~$ man hping3
altaf@LAPTOP-DGNIIH4U9:~$ hping3 -1 --flood -a 192.168.1.103 192.168.1.255
[open_sockraw] socket(): Operation not permitted
[main] can't open raw socket
altaf@LAPTOP-DGNIIH4U9:~$ sudo hping3 -1 --flood -a 192.168.1.103 192.168.1.255
[sudo] password for altaf:
HPING 192.168.1.255 (eth0 192.168.1.255): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.255 hping statistic ---
352610 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
altaf@LAPTOP-DGNIIH4U9:~$
```

The desktop environment visible at the bottom includes icons for weather (27°C Heavy rain), search, file manager, and various application icons. The system tray shows network status, battery level (ENG IN), and the date/time (08-09-2023).

Conclusion: In this experiment, we simulated Denial of Service (DoS) attacks using Hping3, exploring SYN flood and ICMP flood attacks. We learned about their destructive potential and the importance of safeguarding against such threats in network security.

Lab Assignment No.11

Aim: Installing snort, setting in Intrusion Detection Mode and writing rules for Intrusion Detection.

Lab Outcome Attained: LO6 – Demonstrate the network security system using open source tools.

Theory:

A. What is Intrusion Detection System?

An Intrusion Detection System (IDS), in the context of computer network security, is a specialized technology or software application designed to monitor and analyze network traffic and system activities for signs of unauthorized or malicious activities. The primary goal of an IDS is to identify and respond to potential security breaches or attacks in real-time or near-real-time, helping to safeguard computer networks and systems from various threats.

An IDS operates by examining network packets, log files, and system events to identify patterns or behaviors that deviate from established norms or known attack signatures. There are two main categories of IDS:

1. Network-based IDS (NIDS): This type of IDS focuses on monitoring network traffic, analyzing data packets passing through network segments or devices. NIDS sensors are strategically placed within a network infrastructure and inspect traffic to detect suspicious or unauthorized activities. NIDS can identify various types of attacks, including port scans, denial-of-service (DoS) attacks, and attempts to exploit vulnerabilities in network services.

2. Host-based IDS (HIDS): HIDS operates on individual host systems and monitors activities taking place within those systems. It inspects log files, system files, and operating system activities to detect signs of unauthorized access or malicious activities that might not be visible at the network level. HIDS is particularly effective at identifying attacks that target specific vulnerabilities on a single system.

Modern IDS systems often incorporate machine learning and anomaly detection techniques to identify new, previously unseen attack patterns. They can also utilize signature-based detection, where known attack patterns are compared against network traffic or system activities. Some advanced IDS systems can even respond to detected threats by automatically blocking or isolating the compromised systems from the network.

It's important to note that an Intrusion Detection System differs from an Intrusion Prevention System (IPS). While IDS focuses on detecting and alerting about potential threats, IPS takes a more proactive approach by not only identifying but also actively blocking or mitigating those threats in real-time.

In summary, an Intrusion Detection System is a crucial component of network security that helps organizations detect and respond to potential security breaches and attacks, thereby enhancing the overall security posture of their computer networks and systems.

B. What are different modes in which Snort works?

Snort is a popular open-source Intrusion Detection and Prevention System (IDPS) that monitors network traffic for signs of malicious activity or potential security threats. It has three primary modes of operation:

1. Sniffer Mode: In this mode, Snort acts as a passive network sniffer. It monitors and captures network traffic without taking any active actions based on the contents of the packets. Sniffer mode is useful for gaining insight into network traffic patterns and potential anomalies. However, it doesn't actively analyze packets for intrusion detection.
2. Packet Logger Mode: Packet Logger mode extends the functionality of Sniffer mode by not only capturing network packets but also logging their contents. This mode is useful for detailed forensic analysis. It allows security teams to review the actual payload of captured packets for investigative purposes, which can be particularly helpful when examining potential security incidents or breaches.
3. Network Intrusion Detection System Mode: This is the primary and most active mode of Snort. In Network Intrusion Detection System (NIDS) mode, Snort actively inspects incoming network traffic against a set of predefined rules and signatures. When it identifies packets that match known attack patterns or signatures, it generates alerts or notifications. NIDS mode is designed to detect potential security threats and malicious activities in real-time and is commonly used to enhance network security by identifying and alerting on suspicious traffic.

These modes provide different levels of functionality and usage for Snort, ranging from passive observation and logging to active intrusion detection and prevention. It's important to note that while Sniffer and Packet Logger modes focus on observing and capturing network traffic, the Network Intrusion Detection System mode actively analyzes traffic for security threats based on the provided rules and signatures.

C. Write the commands used for installing snort, editing its configuration file and configuring it in intrusion detection mode.

Installing Snort and Configuring Intrusion Detection Mode:

1. Check the name of the interface using command: `ifconfig`
2. Install Snort on Ubuntu: `sudo apt-get install snort`

3. During installation, specify the interface to listen on.

4. Open Snort's configuration file: *sudo gedit /etc/snort/snort.conf*

5. Modify the configuration file:

```
ipvar HOME_NET 192.168.44.0/24 # Update the HOME_NET IP range
```

6. Open a new terminal and edit a rule file (optional): *sudo gedit /etc/snort/rules/ftp.rules*

7. Open another new terminal and validate configuration:

```
sudo snort -T -c /etc/snort/snort.conf -i ens33
```

8. Start Snort in NIDS mode:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens33
```

9. On Kali Linux, run port scanning on the Ubuntu machine: *nmap 192.168.44.128*

10. Observe output on the Snort terminal for alerts.

11. Ping the Ubuntu machine from Kali: *ping 192.168.44.128*

12. To add a rule for detecting ping activity:

a. Create local.rules file: *sudo gedit /etc/snort/rules/local.rules*

b. Add the rule in local.rules:

```
alert icmp any any -> $HOME_NET any (msg:"ICMP test detected"; GID:1;  
sid:10000001; rev:001; classtype:icmp-event;)
```

c. Save and close the file.

d. Comment out these lines in snort.conf:

```
#include $RULE_PATH/icmp.rules  
#include $RULE_PATH/icmp-info.rules
```

e. Include local.rules in snort.conf: *include \$RULE_PATH/local.rules*

f. Validate changes: *sudo snort -T -c /etc/snort/snort.conf -i ens33*

g. Set Snort in Intrusion Detection Mode:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens33
```

h. Ping the Ubuntu machine from Kali and observe generated alerts.

- i. Compare alerts when using icmp.rules, icmp-info.rules, and local.rules to detect ping activity.

Conclusion: In this experiment, we successfully installed Snort, configured it in Intrusion Detection Mode, and crafted custom rules for precise threat detection. This hands-on experience enhances our understanding of network security and intrusion prevention strategies.

Lab Assignment No.13

Aim: Explore the GPG tool of linux to implement email security.

Lab Outcome Attained: LO5 – Demonstrate the network security system using open source tools.

Theory:

1. What is private key ring and public key ring?

In GPG, a "private key ring" and a "public key ring" are used to store cryptographic keys, specifically for asymmetric encryption and digital signatures.

- Private Key Ring: This is a collection of private keys associated with a user or entity.

Private keys are used for decrypting messages that were encrypted with the corresponding public key and for creating digital signatures. The private key ring should be kept secret and secure, as anyone with access to the private key can perform decryption and sign messages on behalf of the associated user or entity.

- Public Key Ring: This is a collection of public keys that can be freely shared with others.

Public keys are used by others to encrypt messages intended for the holder of the corresponding private key. Public keys are also used to verify digital signatures created with the corresponding private key. Public keys can be distributed widely and openly because they do not expose any sensitive information.

In summary, a private key ring contains private keys that should be kept confidential, while a public key ring contains public keys that can be shared openly for encryption and verification purposes. These concepts are fundamental to secure communication and data protection in public-key cryptography systems like GPG.

2. Write the commands used for key generation, export and import of keys and signing and encrypting the message in gpg tool.

Step 1: Generate Key Pairs for Sender and Receiver

- Generate private key and public key pairs for both sender and receiver.

gpg --gen-key or gpg -full-generate-key

Step 2: Create a File Containing Sender's Public Key

- Create a file containing the sender's public key to share with others.

gpg --export -a username > filename

or

gpg --output filename --armor --export user's_email

Step 3: Create a File Containing Sender's Private Key

- Create a file containing the sender's private key (optional, for backup purposes).

gpg --export-secret-key -a username > filename

Step 4: Create a Fingerprint of Receiver's Key

- Generate a fingerprint of the receiver's key to verify its authenticity.
`gpg --fingerprint receiver's_email`

Step 5: Import Receiver's Public Key

- Add the receiver's public key to the sender's public key ring.

Step 6: List Public Keys

- List public keys in the keyring.
 - To list public keys from the keyrings of all users:
`gpg --list-keys`
 - To list public keys from the keyring of a specific user (e.g., shachi_natu@yahoo.com):
`gpg --list-keys shachi_natu@yahoo.com`

Step 7: Sign the Receiver's Public Key

- Sender can sign the receiver's public key to indicate trust.
`gpg --sign-key receiver_email`

Step 8: Encrypt Data for Sending

- Encrypt the data to send. Create the file to be encrypted beforehand.
 - To encrypt and create a .gpg file:
`gpg --encrypt -r receiver_email name_of_file`
 - To encrypt, sign, and create an ASCII-armored file:
`gpg --encrypt --sign --armor -r receiver_email name_of_file`
 - To encrypt, sign, and create a .gpg file:
`gpg --encrypt --sign -r receiver_email name_of_file`

Step 9: Decrypt the Encrypted File

- Decrypt the received encrypted file (e.g., myfile.txt.gpg) to obtain the original content.
`gpg -o myfile_decrypted -d myfile.txt.gpg`

These steps provide a guide for using GnuPG (GNU Privacy Guard) to manage keys, encrypt and decrypt data, and establish trust in public keys within a secure communication framework.

Output Screenshots

```

rudrani@RuduHP:~$ sudo apt-get install gpg
[sudo] password for rudrani:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gpg is already the newest version (2.2.27-3ubuntu2.1).
gpg set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 135 not upgraded.
rudrani@RuduHP:~$ gpg --v
option "--v" is ambiguous
rudrani@RuduHP:~$ gpg --version
gpg (GnuPG) 2.2.27
libgcrypt 1.9.4
Copyright (C) 2021 Free Software Foundation, Inc.
License GNU GPL-3.0-or-later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: /home/rudrani/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2
rudrani@RuduHP:~$ man gpg

rudrani@RuduHP:~$ gpg --full-generate-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/home/rudrani/.gnupg' created
gpg: keybox '/home/rudrani/.gnupg/pubring.kbx' created
Please select what kind of key you want:
  (1) RSA and RSA (default)
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 1024
Requested keysize is 1024 bits
Please specify how long the key should be valid.
      0 = key does not expire
  <n> = key expires in n days
  <n>w = key expires in n weeks
  <n>m = key expires in n months
  <n>y = key expires in n years
Key is valid for? (0) 1

```

```
Key expires at Sat Sep  9 10:58:59 2023 IST
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: rudrani
Email address: zoolzoo@abc.com
Comment: sender key
You selected this USER-ID:
  "rudrani (sender key) <zoolzoo@abc.com>"

Change (N)ame, (C)omment, (E)m ail or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: agent_genkey failed: Timeout
Key generation failed: Timeout
rudrani@RuduHP:~$ gpg --full-generate-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
(1) RSA and RSA (default)
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
(14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 1024
Requested keysize is 1024 bits
Please specify how long the key should be valid.
      0 = key does not expire
      <n> = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 1
Key expires at Sat Sep  9 11:02:11 2023 IST
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: rudrani
Email address: zoolzoo@abc.com
Comment: sender key
You selected this USER-ID:
  "rudrani (sender key) <zoolzoo@abc.com>"
```

```

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /home/rudrani/.gnupg/trustdb.gpg: trustdb created
gpg: key F438FBBF921BFDFC marked as ultimately trusted
gpg: directory '/home/rudrani/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/rudrani/.gnupg/openpgp-revocs.d/3825F3768D742E547AC67FE9F438FBBF921BFDFC.rev'
public and secret key created and signed.

pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
  3825F3768D742E547AC67FE9F438FBBF921BFDFC
uid          rudrani (sender key) <zoozoo@abc.com>
sub  rsa1024 2023-09-08 [E] [expires: 2023-09-09]

rudrani@RuduHP:~$ gpg --gen-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: rudrani
Email address: zoolzoo@abc.com
You selected this USER-ID:
  "rudrani1 <zoolzoo@abc.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 71E78D71BF4CFDCC marked as ultimately trusted
gpg: revocation certificate stored as '/home/rudrani/.gnupg/openpgp-revocs.d/DCC7F886783263E5E87E417271E78D71BF4CFDCC.rev'
public and secret key created and signed.

pub  rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
  DCC7F886783263E5E87E417271E78D71BF4CFDCC
uid        rudrani1 <zoolzoo@abc.com>
sub  rsa3072 2023-09-08 [E] [expires: 2025-09-07]
```

```

rudrani@RuduHP:~$ gpg --export -a rudrani>senderpublickey
rudrani@RuduHP:~$ gpg --export-secret-key -a rudrani>senderprivatekey
rudrani@RuduHP:~$ gpg --fingerprint zoolzoo@abc.com
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid: 2  signed: 0  trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2023-09-09
pub  rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
      DCC7 F886 7832 63E5 E87E 4172 71E7 8D71 BF4C FDCC
uid        [ultimate] rudrani1 <zoolzoo@abc.com>
sub  rsa3072 2023-09-08 [E] [expires: 2025-09-07]

rudrani@RuduHP:~$ gpg --export -a rudrani1>receiverpublickey
rudrani@RuduHP:~$ gpg --import receiverpublickey
gpg: key 71E78D71BF4CFDCC: "rudrani1 <zoolzoo@abc.com>" not changed
gpg: Total number processed: 1
gpg:                      unchanged: 1
rudrani@RuduHP:~$ gpg --list-keys
/home/rudrani/.gnupg/pubring.kbx
```

```
pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
3825F3768D742E547AC67FE9F438FBBF921BFDFC
uid [ultimate] rudrani (sender key) <zoozoo@abc.com>
sub rsa1024 2023-09-08 [E] [expires: 2023-09-09]

pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
DCC7F886783263E5E87E417271E78D71BF4CFDCC
uid [ultimate] rudrani1 <zoolzoo@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]

rudrani@RuduHP:~$ gpg --list-keys zoozoo@abc.com
pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
3825F3768D742E547AC67FE9F438FBBF921BFDFC
uid [ultimate] rudrani (sender key) <zoozoo@abc.com>
sub rsa1024 2023-09-08 [E] [expires: 2023-09-09]

rudrani@RuduHP:~$ gpg --list-keys zoolzoo@abc.com
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
DCC7F886783263E5E87E417271E78D71BF4CFDCC
uid [ultimate] rudrani1 <zoolzoo@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]

rudrani@RuduHP:~$ gpg --encrypt -r zoolzoo@abc.com sample.txt
rudrani@RuduHP:~$ gpg --encrypt --sign -armor -r zoolzoo@abc.com sample.txt
gpg: mor: skipped: No public key
gpg: sample.txt: sign+encrypt failed: No public key
rudrani@RuduHP:~$ gpg --encrypt --sign -armor -r zoolzoo@abc.com try.txt
gpg: mor: skipped: No public key
gpg: try.txt: sign+encrypt failed: No public key
rudrani@RuduHP:~$ gpg --encrypt --sign --armor -r zoolzoo@abc.com try.txt
```

```
rudrani@RuduHP:~$ gpg --encrypt -r zoolzoo@abc.com sample.txt
rudrani@RuduHP:~$ gpg --encrypt --sign -armor -r zoolzoo@abc.com sample.txt
gpg: mor: skipped: No public key
gpg: sample.txt: sign+encrypt failed: No public key
rudrani@RuduHP:~$ gpg --encrypt --sign -armor -r zoolzoo@abc.com try.txt
gpg: mor: skipped: No public key
gpg: try.txt: sign+encrypt failed: No public key
rudrani@RuduHP:~$ gpg --encrypt --sign --armor -r zoolzoo@abc.com try.txt
rudrani@RuduHP:~$ gpg --encrypt --sign --armor -r zoolzoo@abc.com sample.txt
rudrani@RuduHP:~$ gpg -o sampledecrypted -d sample.txt.gpg
gpg: encrypted with 3072-bit RSA key, ID 8644E9D1F6E917D7, created 2023-09-08
    "rudrani1 <zoolzoo@abc.com>"
rudrani@RuduHP:~$
```

Conclusion: In this experiment, we explored the GPG tool on Linux to enhance email security. We learned how to manage encryption keys, encrypt email content, and ensure the confidentiality and integrity of email communications.

Lab Theory Assignment No. 1

Explain the padding scheme used in RSA. Why it is used? What is its limitation?

The padding scheme used in RSA is known as PKCS (Public-Key Cryptography Standards) version 1.5 padding. It is used to enhance the security of RSA encryption and address a fundamental limitation of basic RSA encryption, which lacks a notion of plaintext security. The primary goal of this padding scheme is to add randomness to the plaintext message before encryption to prevent the same plaintext from always producing the same ciphertext.

Here's a summary of how the PKCS 1.5 padding scheme works:

1. Random Padding: Before encrypting a message `m` using RSA, a random number `r` is generated.
2. Concatenation: The random number `r` is concatenated with the original message `m`.
3. Special Bytes: Special bytes are added at the beginning and end of the concatenated message. Specifically, it starts with `0x00 0x02`, followed by the random number `r`, and ends with another `0x00` byte.
4. RSA Encryption: The padded message (now called `m_cap`) is then encrypted using RSA.

The purpose of this padding scheme is to ensure that even if the same message is encrypted multiple times, it will produce different ciphertexts due to the random `r`. This prevents an attacker from easily identifying identical messages based on their ciphertexts, which is essential for achieving chosen-plaintext security (CPA security). CPA security means that an attacker cannot distinguish between ciphertexts of different messages even if they can choose the messages to be encrypted.

Limitation:

The main limitation of PKCS 1.5 padding is that it is not entirely secure against certain types of attacks, particularly attacks that exploit the padding oracle vulnerability (such as the Bleichenbacher attack). This means that there are situations where an attacker can still gain information about the plaintext, even when PKCS 1.5 padding is used. As a result, it is generally recommended to use more secure padding schemes like OAEP (Optimal Asymmetric Encryption Padding) or RSA-PSS (RSA Probabilistic Signature Scheme) for RSA encryption to mitigate these vulnerabilities.

Lab Outcome Attained: LO2 - Demonstrate Key management, distribution and user authentication.

Lab Theory Assignment No. 2

What is Intrusion Detection System? Explain different types of intrusion detection systems with their working. State the advantages and limitations of each.

An Intrusion Detection System (IDS) is a security technology designed to monitor network traffic or system activities to detect and respond to suspicious or malicious behavior that may indicate unauthorized access, attacks, or security breaches. IDS helps organizations identify potential security threats and take appropriate actions to protect their systems and data. There are different types of IDS, each with its own working principles, advantages, and limitations. Here's an overview of the main types:

1. Network-based Intrusion Detection System (NIDS):

- Working: NIDS monitors network traffic in real-time by inspecting packets and analyzing network protocols. It uses predefined signatures or behavioral patterns to identify potential threats or attacks. NIDS can be placed at strategic points within a network to monitor all traffic passing through those points.

- Advantages:

- Comprehensive coverage: NIDS can monitor all traffic within a network segment, providing a broad view of potential threats.
- Detects external threats: Effective at detecting attacks coming from outside the network, such as DDoS attacks or external intrusion attempts.
- Low impact on host resources: Since it doesn't run on individual hosts, it doesn't consume host resources.

- Limitations:

- Limited visibility: NIDS can't detect attacks or anomalies within encrypted traffic (e.g., HTTPS) without decryption capabilities.
- False positives: Signature-based NIDS may generate false alarms if traffic patterns change, causing legitimate traffic to trigger alerts.
- Can't detect insider threats: NIDS primarily focuses on external threats and may miss attacks originating from within the network.

2. Host-based Intrusion Detection System (HIDS):

- Working: HIDS is installed on individual host systems and monitors activities at the host level. It looks for deviations from established baselines, unauthorized access, and unusual system behavior. HIDS can analyze log files, system calls, and file integrity.

- Advantages:

- Granular visibility: HIDS provides detailed insights into activities on individual hosts, making it effective for detecting insider threats and host-specific attacks.

- Can monitor encrypted traffic: HIDS can detect malicious activity on the host, even within encrypted traffic.

- Low false positives: Since it operates on the host level, it has a better understanding of normal system behavior.

- Limitations:

- Limited coverage: HIDS only monitors the host it's installed on, making it less suitable for detecting network-wide attacks or lateral movement.

- Resource overhead: Installing and maintaining HIDS on numerous hosts can be resource-intensive and complex.

- May not detect zero-day attacks: HIDS relies on known patterns or behaviors and may miss previously unseen attack techniques.

3. **Behavior-based Intrusion Detection System:**

- Working: Behavior-based IDS focuses on identifying patterns of behavior that deviate from the normal baseline, rather than relying on predefined signatures. It uses machine learning and heuristics to detect anomalies.

- Advantages:

- Effective against unknown threats: Behavior-based IDS can detect previously unseen attacks or zero-day vulnerabilities.

- Reduced false positives: It's less likely to generate false alarms when normal behavior evolves or changes.

- Adaptable: Can adapt to changing network or system environments.

- Limitations:

- Training required: Behavior-based IDS needs a training period to establish a baseline of normal behavior, and it may produce false negatives during this phase.

- Resource-intensive: Analyzing behavior patterns in real-time can be computationally expensive.

- May require more sophisticated tuning and maintenance.

Each type of IDS has its strengths and weaknesses, and organizations often use a combination of these systems to achieve a more comprehensive security posture. Additionally, organizations should consider factors such as the scale of their network, the nature of their assets, and the specific threats they face when choosing and deploying IDS solutions.

Lab Outcome Attained: LO6 - Demonstrate the network security system using open source tools.

Detecting Phishing Sites Using ChatGPT

Takashi Koide
takashi.koide@global.ntt
NTT Security Japan (KK)
Tokyo, Japan

Naoki Fukushi
naoki.fukushi@global.ntt
NTT Security Japan (KK)
Tokyo, Japan

Hiroki Nakano
hi.nakano.sec@gmail.com
NTT Security Japan (KK)
Tokyo, Japan

Daiki Chiba
daiki.chiba@ieee.org
NTT Security Japan (KK)
Tokyo, Japan

ABSTRACT

The rise of large language models (LLMs) has had a significant impact on various domains, including natural language processing and artificial intelligence. While LLMs such as ChatGPT have been extensively researched for tasks such as code generation and text synthesis, their application in detecting malicious web content, particularly phishing sites, has been largely unexplored. To combat the rising tide of automated cyber attacks facilitated by LLMs, it is imperative to automate the detection of malicious web content, which requires approaches that leverage the power of LLMs to analyze and classify phishing sites.

In this paper, we propose a novel method that utilizes ChatGPT to detect phishing sites. Our approach involves leveraging a web crawler to gather information from websites and generate prompts based on this collected data. This approach enables us to detect various phishing sites without the need for fine-tuning machine learning models and identify social engineering techniques from the context of entire websites and URLs. To evaluate the performance of our proposed method, we conducted experiments using a dataset. The experimental results using GPT-4 demonstrated promising performance, with a precision of 98.3% and a recall of 98.4%. Comparative analysis between GPT-3.5 and GPT-4 revealed an enhancement in the latter's capability to reduce false negatives. These findings not only highlight the potential of LLMs in efficiently identifying phishing sites but also have significant implications for enhancing cybersecurity measures and protecting users from the dangers of online fraudulent activities.

KEYWORDS

Phishing Sites, Social Engineering, and ChatGPT.

1 INTRODUCTION

The emergence of large language models (LLMs) [15, 30, 31, 35] has revolutionized various domains, including natural language processing and artificial intelligence. These models have gained significant popularity due to their remarkable performance across a wide range of tasks. Among them, ChatGPT [5] has consistently achieved state-of-the-art results, demonstrating its ability to generate coherent and contextually relevant responses in conversational settings. While extensive research has focused on exploring the capabilities of LLMs for tasks such as code generation and text synthesis, little attention has been paid to leveraging these models for the analysis and detection of malicious web content, particularly phishing sites. Phishing sites, which employ social engineering techniques [28, 41, 45], pose a severe threat to Internet users by masquerading as legitimate platforms and tricking them into revealing sensitive information or causing financial harm. To effectively combat the growing threat of automated cyber-attacks facilitated by LLMs, it is important to automate the detection of malicious

web content. This requires the exploration of methodologies that leverage the power of LLMs to analyze and classify phishing sites.

In this research, we propose a novel method that utilizes ChatGPT to detect phishing sites. Our approach involves leveraging a web crawler to gather information from websites and generate prompts [23, 43] based on this collected data. These prompts are then presented to ChatGPT, which determines whether a given website is a phishing site or not. The integration of web crawling and ChatGPT's contextual understanding enables us to make informed decisions concerning the legitimacy or suspiciousness of websites. By employing ChatGPT, we can (1) detect various phishing sites without the need for fine-tuning machine learning models and (2) identify social engineering (SE) techniques from the context of entire websites and URLs. To the best of our knowledge, this is the first study to analyze the ability of LLMs to identify phishing sites. To evaluate the performance of our proposed method, we conducted experiments using a carefully curated dataset for phishing site detection. The experimental results using GPT-4 [30] showed promising performance, with a precision of 98.3% and a recall of 98.4%. Moreover, a comparative analysis between GPT-3.5 and GPT-4 revealed a significant improvement in the latter's capabilities, particularly in terms of reducing false negatives. GPT-4 outperformed GPT-3.5 in its ability to determine the suspiciousness of domain names, identify SE techniques from the website content, and provide comprehensive phishing detection by considering multiple factors. The results of this study emphasize the potential of LLMs in efficiently detecting phishing sites, particularly in uncovering SE techniques aimed at psychologically manipulating users. These findings have significant implications for enhancing automated cybersecurity measures and mitigating the risks of online fraudulent activities faced by users.

In summary, we make the following contributions:

- We propose a novel method for detecting phishing sites using ChatGPT. By utilizing web crawling techniques to gather information from websites and generate prompts, our approach enables ChatGPT to classify websites as phishing or not.
- We present an experimental evaluation of our proposed method using a dataset designed for phishing site detection. Our results show promising performance with a precision of 98.3% and a recall of 98.4%, demonstrating the effectiveness of LLMs in identifying phishing sites.
- We performed a comparative analysis between GPT-3.5 and GPT-4 and found notable improvements in GPT-4's ability to identify phishing sites, particularly in minimizing false negatives. It excelled at identifying suspicious domains, detecting SE techniques in web content, and providing comprehensive phishing detection by considering multiple factors.

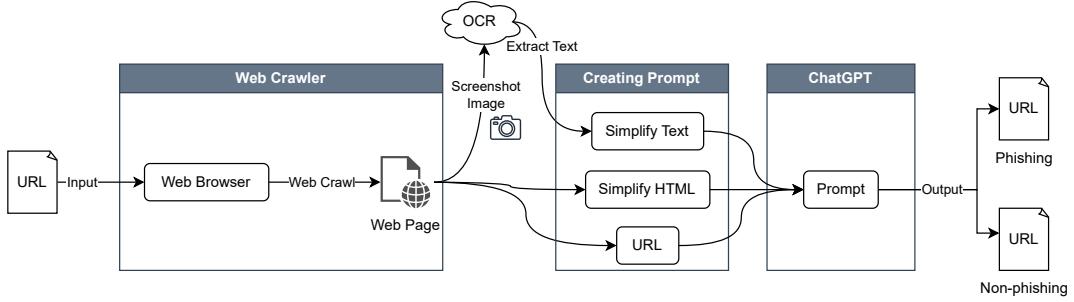


Figure 1: Overview of Proposed Method.

2 BACKGROUND

Phishing sites are fraudulent websites that aim to steal personal information, money, or cause malware infections by psychologically luring users. These sites use email, short message service (SMS), and web advertisements to attract users and redirect them to phishing sites by having them click on malicious links [37, 38]. There are two key components that constitute phishing sites, which either have one or both of these elements: (1) Imitation of domain names and the use of official logos and branding to deceive users into believing they are interacting with legitimate services, and (2) Employing SE techniques to deceive users and manipulate their actions. By impersonating legitimate services, phishing sites deceive users and trust them [24, 25]. These websites often imitate legitimate platforms such as online banking [13], e-commerce sites [14], and social [39] media. In addition, they can create a sense of urgency or fear by displaying fake malware infection warnings or account problems, or generate interest by displaying fake rewards [22]. Through these SE techniques, users are misled into providing sensitive information such as login credentials and credit card numbers. Furthermore, various user actions may be induced, such as sending cryptocurrency, calling fake technical support centers [26], or downloading apps and executable files [28].

Previous studies have been conducted to understand the characteristics, techniques, and fundamental mechanisms used by attackers in phishing sites. Researchers analyze the design, structure, and content of these sites, identify common patterns, and develop methods for detecting phishing sites. These studies can contribute to improving security practices, educating users about potential threats, and devising effective strategies to mitigate online fraud and associated risks. For example, studies exist on identifying websites that abuse legitimate branding based on their appearance [11, 24, 25], as well as on discovering brand information in domain names and URLs [12, 20, 29]. Some studies have also been conducted on identifying phishing sites based on information contained in certificates and domain names [16, 19], and on capturing the context of social engineering in technical support scams, fake infection warnings, and fake rewards [18, 21, 22].

Although various methods have been employed to detect phishing sites, there are two main problems. One is the need for learning targeted brands and modifying algorithms depending on the phishing sites. For example, it is necessary to learn in advance the logo images that phishing sites abuse, or to create rules to detect them

according to domain squatting techniques. The other is the inability to analyze the context of psychological manipulation induced by social engineering in detail. While some studies have been conducted on keyword matching and deep learning-based analysis [18, 44], no attempt has been made to automatically analyze and understand the various contexts of psychological manipulation by examining the entire content of a website.

3 METHOD

We propose a method for detecting phishing sites by using ChatGPT, to analyze the contextual information of websites and their respective URLs. Our proposed method takes advantage of ChatGPT's highly accurate contextual understanding to precisely identify textual representations involving SE techniques, as well as inconsistencies between brands deceived by the websites and domain names. By employing ChatGPT, which is trained on extensive text data, we can detect various phishing sites that disguise themselves as legitimate services without learning from collected phishing site data. In this manner, the proposed method addresses two problems present in previous studies. An overview of the proposed method is illustrated in Figure 1. Our method uses a web crawler to access the input URL and obtain information from the visited website, such as screenshot images, HTML, and URLs. This information is used to create a prompt for input into ChatGPT, which then determines whether the site is a phishing site or a non-phishing site.

3.1 Web Crawling

We implement a web crawler that automates Google Chrome with Chrome DevTools Protocol [4] to visit websites and collect information. Given an input URL, the web crawler retrieves the URL of the reached web page (the final destination after any redirects), the HTML after JavaScript execution, and captures a screenshot image. The reason for obtaining the HTML after JavaScript execution (Browser-rendered HTML) is that some phishing sites employ obfuscated JavaScript to generate DOM elements as a means to evade analysis. As a result, it becomes challenging to analyze any traces of phishing solely based on the HTML before JavaScript execution. The web crawler is configured to emulate two different environments: Windows + Chrome, and iPhone + Safari. These configurations included specifying the *UserAgent* and browser size for each environment.

3.2 Prompt

We describe the process of generating a prompt for detecting phishing sites. The template for the prompt is shown in Prompt 1. The purpose of this prompt is to provide website information to ChatGPT and determine whether it is a phishing site or not. We create the template based on the Chain of Thought (CoT) prompting technique [23, 42]. CoT is a prompting technique that encourages LLMs to explain their reasoning. CoT has been shown to be effective in improving performance on a variety of reasoning tasks, such as arithmetic and symbolic reasoning. The task of phishing site detection is divided into four subtasks, which facilitate the execution of specific reasoning processes. These subtasks are as follows:

- (1) Analyze whether the website contains SE techniques that deceive or attract users. The prompt provides typical SE techniques commonly used by phishing sites, such as cash prizes, fake malware infection warnings, account problems, and postal parcel issues. The LLM is instructed that the presence of these phrases without context in a non-login state indicates a high likelihood of a phishing site.
- (2) Extract the brand name of the website. Phishing sites may be created by copying resources such as HTML and images from legitimate sites, making it difficult to determine their authenticity based on HTML alone. Therefore, the LLM is prompted to confirm whether the URL corresponds to the legitimate site.
- (3) Determine if the site is a phishing site or not and explicitly state the rationale for the decision. Providing detailed descriptions improves the accuracy of responses and makes it easier for humans to analyze them.
- (4) Generate output in JSON format. If the website uses SE techniques, the LLM is expected to return *phishing* as true. If the brand name can be identified and its domain name differs from the legitimate one, *suspicious_domain* is expected to be true. Additionally, the LLM should produce a *phishing_score* ranging from 0 to 10.

The prompt is provided with website information in the form of HTML, URL, and text extracted from screenshot images using optical character recognition (OCR). Some phishing sites evade HTML-based phishing site analysis by displaying brand information (logos and headings) in image files or canvas elements without including specific brand names or text in the HTML. To input text that cannot be extracted from HTML into the prompt, we employ OCR for text extraction.

Some websites may exceed the token limit of ChatGPT when the entered data is large. Tokens in ChatGPT refer to individual units of text that represent characters or words and are used for processing and analysis by LLMs. The GPT-3.5 model has a token limit of 4,096, whereas GPT-4 (8K) has a limit of 8,000 tokens, and GPT-4 (32k) has a limit of 32,000 tokens. In fact, the median token count for HTML in our dataset (see Section 4) is 30,398 (with a median of 74,937 for non-phishing pages), indicating that a large number of web pages exceed this token limit. To compare the performance of the GPT-3.5 model and GPT-4 model, we simplify the HTML and OCR-extracted text to fit within 4,096 tokens in this study. Table 1 presents the maximum token counts for each data category. This prompt template consumes 362 tokens, with a maximum limit of

Prompt Template 1

You are a web programmer and security expert tasked with examining a web page to determine if it is a phishing site or a legitimate site. To complete this task, follow these sub-tasks:

1. Analyze the HTML, URL, and OCR-extracted text for any SE techniques often used in phishing attacks. Point out any suspicious elements found in the HTML, URL, or text.
2. Identify the brand name. If the HTML appears to resemble a legitimate web page, verify if the URL matches the legitimate domain name associated with the brand, if known.
3. State your conclusion on whether the site is a phishing site or a legitimate one, and explain your reasoning. If there is insufficient evidence to make a determination, answer "unknown".
4. Submit your findings as JSON-formatted output with the following keys:
 - *phishing_score*: int (indicates phishing risk on a scale of 0 to 10)
 - *brands*: str (identified brand name or None if not applicable)
 - *phishing*: boolean (whether the site is a phishing site or a legitimate site)
 - *suspicious_domain*: boolean (whether the domain name is suspected to be not legitimate)

Limitations:

- The HTML may be shortened and simplified.
- The OCR-extracted text may not always be accurate.

Examples of social engineering techniques:

- Alerting the user to a problem with their account
- Offering unexpected rewards
- Informing the user of a missing package or additional payment required
- Displaying fake security warnings

URL:

{URL}

HTML:

~~~

{Browser-rendered HTML}

~~~

Text extracted using OCR:

~~~

{OCR-extracted text}

~~~

Table 1: Token Counts for Prompt Template, HTML, OCR-extracted Text, and URL

Data	Maximum Token Count
Prompt Template	362 (fix)
HTML	3,000
OCR-extracted Text	500
URL	234
Total	4,096

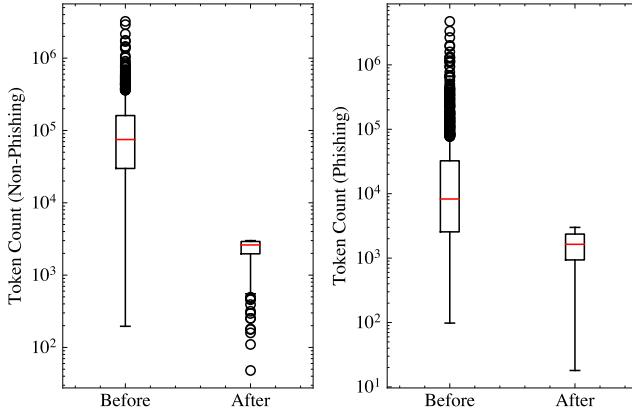


Figure 2: Number of Tokens Before and After Simplified HTML

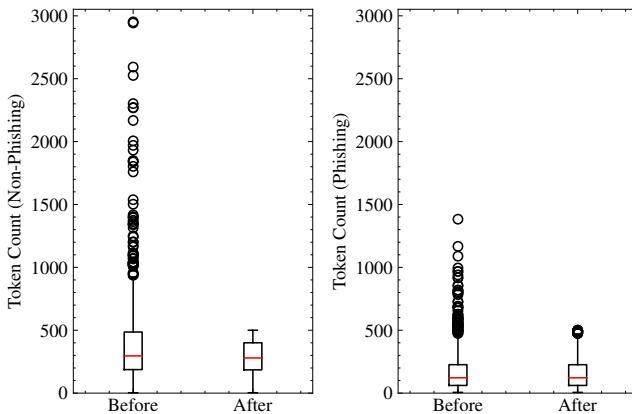


Figure 3: Number of Tokens Before and After Simplified OCR-extracted Text

3,000 tokens for HTML and 500 tokens for OCR-extracted text. Since all generated prompts in this dataset fit within the 4,096-token limit, we did not simplify URLs. The median token count for URLs in the dataset is 9, with the longest URL consisting of 2,209 tokens. In the following sections, we describe the specific processes to simplify the HTML and OCR-extracted text.

Algorithm 1 Simplifying HTML

Require: inputHTML: Input HTML
Ensure: processedHTML: Processed HTML

```

1: function SIMPLIFYHTML(inputHTML)
2:   Remove style, script, and comment elements from
   inputHTML
3:   processedHTML ← Result after removal
4:   if lengthToken(processedHTML) < 3000 then
5:     return processedHTML
6:   end if
7:   Unwrap elements except for important tags
8:   Remove elements without text content
9:   Shorten href in a tags and src in img tags
10:  processedHTML ← Result after removal
11:  if lengthToken(processedHTML) < 3000 then
12:    return processedHTML
13:  end if
14:  while lengthToken(processedHTML) > 3000 do
15:    Remove an HTML element from the midpoint of the
   processedHTML
16:    processedHTML ← Result after removal
17:  end while
18:  return processedHTML
19: end function

```

3.3 Simplifying HTML

The proposed method simplifies the HTML while preserving essential information for phishing detection and brand identification. Elements located at the top of the HTML, such as the title and the meta description element, primarily contain cues for determining the services provided by the website. Additionally, the form element used for inputting login information and text displayed on the page are crucial for analyzing the role of the website. To achieve maximum retention of such information while minimizing its length, we follow the steps outlined in Algorithm 1.

The proposed method removes HTML elements, such as style, script, and comment tags, that have low relevance and a high number of tokens. Subsequently, we unwrap HTML elements other than the important tags listed as head, title, meta, body, h1, h2, h3, h4, h5, h6, p, strong, a, img, hr, table, tbody, tr, th, td, ol, ul, li, ruby, and label. Unimportant elements are eliminated, and their child elements are incorporated into their parent elements. Then, our method removes HTML elements that do not contain text enclosed within tags. We also shorten the src attribute img elements encoded in base64 and the href attribute of a elements containing lengthy URLs. The proposed method repetitively removes intermediate HTML elements until the token count falls below 3,000.

3.4 Simplifying OCR-extracted Text

Some websites may have a large amount of text on the screen, and OCR-extracted text as well as HTML may exceed the token limit. The proposed method also simplifies OCR-extracted text.

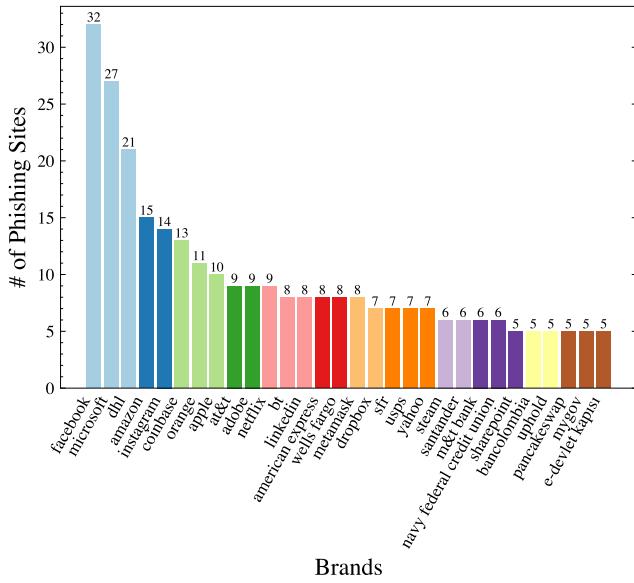


Figure 4: Top 30 Brands Targeted by Phishing Sites.

Specifically, it examines the font size of the identified text from OCR and removes sentences starting from the smallest font size. This process is repeated until the number of tokens in the text falls below 500 tokens.

4 DATASET

We describe how we created the dataset used in our experiment. The dataset includes a total of 1,000 phishing sites and an equal number of non-phishing sites. To gather these websites, we conducted web crawling starting from the collected seed URLs.

4.1 Phishing Sites

To collect phishing sites, we used OpenPhish [8] and PhishTank [9], which are phishing intelligence sources, and CrowdCanary [27], a method for extracting phishing-related posts from Twitter using machine learning models. By using keyword-based filters such as “phishing” and “scam,” CrowdCanary extracts URLs from posts associated with reports of phishing encountered by victims and observed by experts. We collected URLs from March to April 2023 and used them as seed URLs. This enabled us to obtain a wide range of phishing sites that are not limited to incoming channels such as email, SMS, and web advertisements. Subsequently, we accessed the seed URLs using the web crawler described in Section 3.1. To ensure the integrity of our dataset, we manually excluded websites with incomplete rendering and image loading. We conducted a thorough examination of the screenshot images of websites, focusing on two factors:

- Websites that display logos or brand names associated with well-known services or that visually resemble legitimate sites.

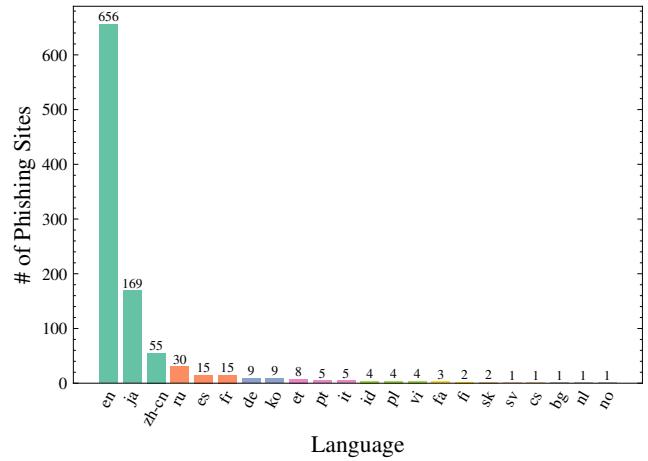


Figure 5: Language Distribution of Phishing Sites.

- Websites employing SE techniques, such as presenting false information (e.g., fake rewards, virus infection alerts, account issues), to generate a sense of urgency or interest.

From the gathered website information, we excluded those meeting the following conditions:

- Websites that contain R18 (adult, gambling, dating) content due to potential conflicts with ChatGPT’s content filter.
- In cases where similar screenshots were found among the extracted phishing sites, all but one were excluded. The criteria for determining similarity included the Fully Qualified Domain Name (FQDN), the page title, the website’s appearance, and the text content. However, if websites have the same appearance but different languages, we keep them.

As a result of the above analysis, we obtained a dataset of 1,000 phishing sites with unique 1,000 FQDNs. These phishing sites targeted a total of 147 legitimate service brands (see Appendix B). We identified 32 phishing sites that did not contain any brand information. Figure 4 shows the top 30 brands used by phishing sites in the dataset. In addition, we extracted text from the screenshot images using OCR. In this paper, we used Azure Cognitive Service [2] to streamline the OCR process for multilingual websites. We identified the languages of the OCR-extracted text using langdetect [7], a language detection library, revealing that the 1,000 phishing sites were distributed across 22 different languages. Figure 5 illustrates the number of phishing sites in each language.

4.2 Non-phishing Sites

We collected an equal number of websites as phishing sites for non-phishing sites. The seed URLs consisted of legitimate websites for the 153 brands targeted by phishing sites, as well as the top 2k domain names from the Tranco list [33]. From the legitimate sites of the 153 brands, we extracted the URLs of their homepages and, if available, the login pages (a total of 236 URLs with 196 FQDNs). We accessed the seed URLs using a web crawler. After crawling the Tranco top 2k, we successfully accessed 1,661 URLs. The Tranco top sites include certain categories such as adult content, illegal downloading/streaming, and gambling. Since these categories can trigger

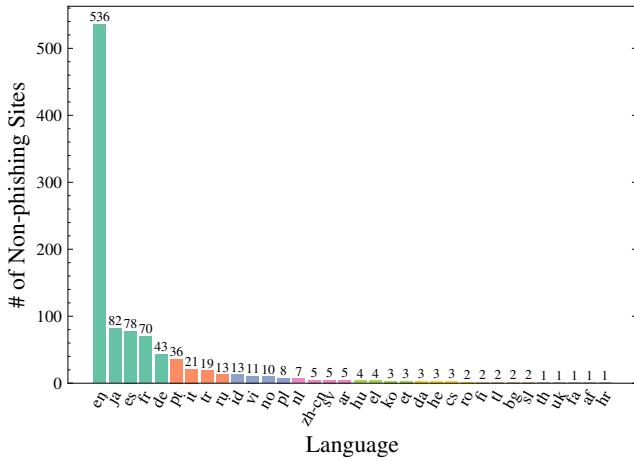


Figure 6: Language Distribution of Non-phishing Sites.

content filters in ChatGPT or lead to phishing sites through malicious advertisements, we excluded such websites. The breakdown of excluded sites is as follows: 28 porn, 3 gambling, and 9 illegal sites. The remaining 764 websites (out of 1,000 non-phishing sites) were randomly selected from the crawled Tranco top sites. The non-phishing sites are distributed across 34 languages. Figure 6 presents the distribution of non-phishing sites across different languages.

4.3 Simplifying HTML and OCR-Extracted Text

We simplified the collected HTML and OCR-extracted text of phishing sites and non-phishing sites using the process explained in Section 3.3 and 3.4. Figure 2 shows the comparison of token counts for the HTML content before and after simplification. The median token count for non-phishing sites was 74,937 before simplification, while it was 8,275 for phishing sites. In general, non-phishing sites have a higher token count. One reason for this difference is that non-phishing sites often implement a variety of sophisticated features through complex JavaScript code or use large platforms, while phishing sites have only the minimum functionality necessary to deceive users or steal sensitive information. Of the 2,000 sites in the dataset, 980 exceed the current ChatGPT model’s maximum token limit of 32,000. Therefore, simplification is required for website inspections under the existing ChatGPT model. Figure 3 compares the token counts of OCR-extracted text before and after simplification. The median token count for non-phishing sites was 296, whereas for phishing sites, it was 122. Similar to the HTML comparison, non-phishing sites tend to have a higher token count in this case as well.

5 EVALUATION

To assess the detection accuracy of the proposed method for phishing sites, we conducted an evaluation experiment using the Azure OpenAI API [3] with our own dataset. We used two models: gpt-4-0314 (referred to as GPT-4) and gpt-3.5-turbo-0301 (referred to as GPT-3.5). To compare their detection accuracy, we analyzed the responses from both models for each URL in the dataset using the same prompt. The dataset and experimental results cannot be made

Table 2: Confusion Matrix for GPT-3.5

		Predicted	
		Phishing	Non-phishing
Actual	Phishing	TP	FN
	Non-phishing	867	133
		FP	TN
		15	985

Table 3: Confusion Matrix for GPT-4

		Predicted	
		Phishing	Non-phishing
Actual	Phishing	TP	FN
	Non-phishing	984	16
		FP	TN
		17	983

Table 4: Performance Metrics Comparison

	GPT-3.5	GPT-4
Precision	98.3%	98.3%
Recall	86.7%	98.4%
Accuracy	92.6%	98.4%
F-measure	92.1%	98.4%

publicly available due to the inclusion of commercially licensed data and copyright issues, but we can provide them to researchers upon request.

5.1 Summary of Result

To extract detection results from the API responses, we defined a response as phishing if either the value of the *phishing* key or the *suspicious_domain* key in the JSON response was true. Conversely, a response was labeled as non-phishing if both keys were false. Due to probabilistic fluctuations in ChatGPT’s output, some responses did not contain machine-readable JSON text, such as using a single quote instead of a double quote. Among the responses, 2 (0.2%) for GPT-4 and 13 (1.3%) for GPT-3.5 resulted in JSON parse errors. However, since all responses included the corresponding answers for each key, we heuristically parsed and aggregated them.

Table 2 and Table 3 present the phishing detection results of GPT-3.5 and GPT-4. GPT-3.5 had 15 false positives (FPs), while GPT-4 had 17 FPs, indicating similar accuracy for non-phishing sites. On the other hand, GPT-3.5 had a higher number of false negatives (FNs) with 133 websites, whereas GPT-4 improved with only 16 websites. These results indicate that GPT-4 demonstrates an enhanced predictive accuracy for identifying actual phishing sites compared to GPT-3.5. Out of the 867 true positives (TPs) for GPT-3.5, 759 responses marked *phishing* as true, 796 marked *suspicious_domain* as true, and 688 had both marked as true. For GPT-4, out of the 984 TPs, 906 marked *phishing* as true, 983 marked *suspicious_domain* as true, and 905 had both marked as true.

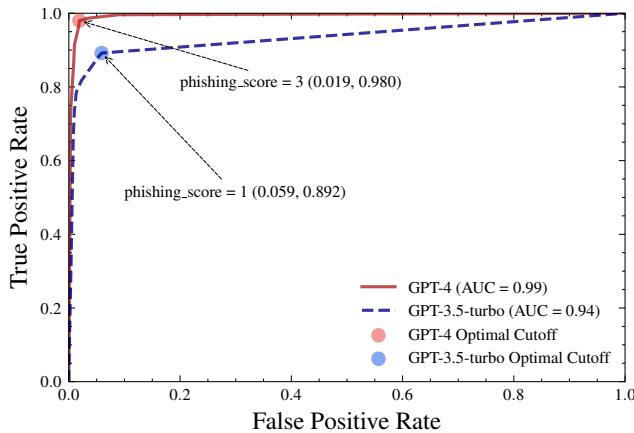


Figure 7: ROC Curve for the *phishing_score*.

Table 4 shows the results of the performance metrics comparison. Precision ($= \frac{TP}{TP+FP}$), Recall ($= \frac{TP}{TP+FN}$), Accuracy ($= \frac{TP+TN}{TP+TN+FP+FN}$), and F-measure ($= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$) were used as performance metrics. The capability of non-phishing site classification (FP and TN) was comparable between GPT-3.5 and GPT-4, resulting in similar precision values. However, GPT-4 outperforms GPT-3.5 by approximately 5.8% to 11.7% in the other metrics. Despite the inclusion of non-English local brands in the dataset, both GPT-3.5 and GPT-4 achieved accurate non-phishing site classification. On the other hand, for phishing site classification, GPT-4 significantly improved with a recall of 98.4% compared to GPT-3.5's 86.7%. Based on these results, the subsequent sections will provide a detailed analysis of the detection capability of the proposed method.

Additionally, we can classify the responses by selecting an appropriate threshold for the *phishing_score* value. By incrementing the threshold by 1, we calculated the true positive rate and false positive rate and plotted the receiver-operating characteristics (ROC) curve, as illustrated in Figure 7. The optimal cut-off values for *phishing_score*, calculated using Youden's J statistic ($J = \frac{TP}{TP+FN} + \frac{TN}{TN+FP} - 1$), were 3 for GPT-4 and 1 for GPT-3.5. The area under the curve (AUC) was 0.99 for GPT-4 and 0.94 for GPT-3.5. The detection accuracy based on the threshold setting for *phishing_score* also demonstrated the superior performance of GPT-4 compared to GPT-3.5. The corresponding performance metrics for this case can be found in Appendix A.

5.2 Examples of ChatGPT's Responses

In this section, we explain specific examples of ChatGPT's response when it correctly detected phishing sites.

Phishing Example A

Figure 8 displays a screenshot image of the phishing site ([https://securtypages.start\[.\]page/](https://securtypages.start[.]page/)), and Response 1 represents the response generated by ChatGPT. This website impersonates Facebook and attempts to create a sense of urgency regarding account issues for the user. ChatGPT correctly identified this website as being associated with the "Meta Facebook" brand and determines that the domain name is not legitimate Facebook. Furthermore, ChatGPT recognized the presence of SE techniques and highlights

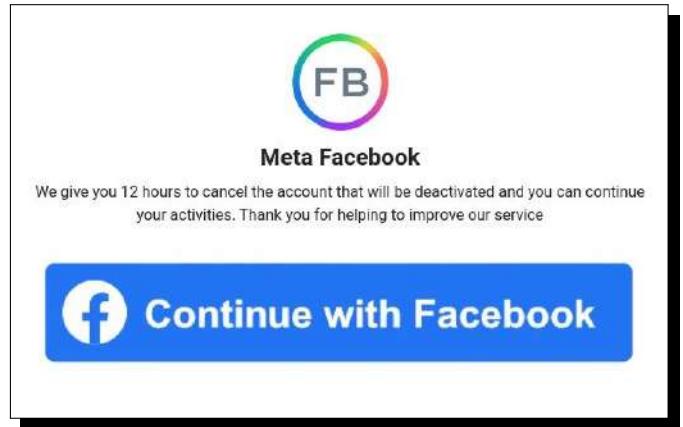


Figure 8: An Example of Facebook Phishing Site.

the existence of suspicious links within the HTML. ChatGPT set the *phishing_score* to 9, marking both *phishing* and *suspicious_domain* as true. Below, we have defanged some of the URLs included in the response.

Response 1

1. Suspicious elements:
 - The URL contains a subdomain "securtypages", which is unrelated to the legitimate Facebook domain ([www.facebook\[.\]com](http://www.facebook[.]com)).
 - The extracted text alerts the user about having 12 hours to cancel the account. This is a social engineering technique often used in phishing attacks.
 - Furthermore, the link provided redirects to a suspicious, non-legitimate URL: "[http://centrenotificationvice.s.duckdns\[.\]org/cntre.html](http://centrenotificationvice.s.duckdns[.]org/cntre.html)".
 2. The brand name identified is Meta Facebook.
 3. The site appears to be a phishing site because of the suspicious URL and social engineering tactics in the text.
 4. JSON-formatted output:
- ```

```
{
  "phishing_score": 9,
  "brands": "Meta Facebook",
  "phishing": true,
  "suspicious_domain": true
}
```

```

### Phishing Example B

Figure 9 illustrates a phishing site ([https://siges.com\[.\]br/service/](https://siges.com[.]br/service/)) impersonating DHL, attempting to deceive users into providing their financial information under the pretext of a

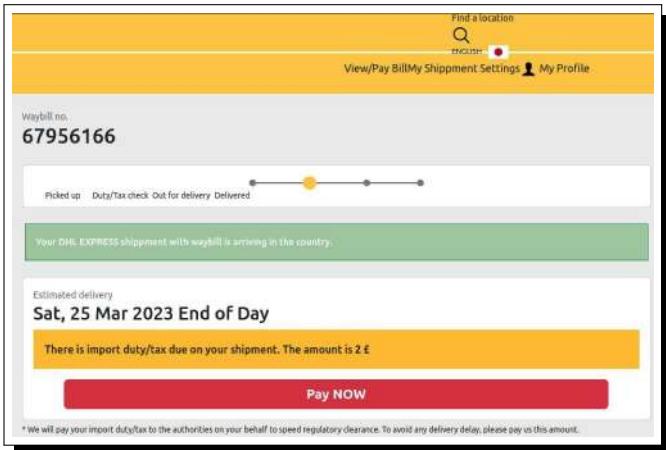


Figure 9: An Example of DHL Phishing Site.

payment requirement for receiving a package. ChatGPT accurately recognized the brand name as DHL and correctly identified the SE technique of requesting payment for import duty/tax expenses. It also determined that the domain name does not belong to the legitimate DHL website. Furthermore, by analyzing the HTML structure, ChatGPT provided detailed insights into the structure of the phishing site, which attempts to steal credit card information such as card numbers and CVC numbers.

#### Response 2

Based on my analysis of the HTML, URL, and OCR-extracted text, here are my findings:

1. Suspicious elements in the HTML and OCR-extracted text:
  - a. The site prompts the user to pay an import duty/tax amounting to 2€ on their shipment, which could be a social engineering technique to trick the user into providing their payment information.
  - b. The HTML structure has a card data input form where the user is expected to enter their card details, such as card number, expiry date, CVC number, and cardholder's name. This could be an attempt to steal sensitive card information.
2. The brand name identified is "DHL EXPRESS".
3. The URL ([https://siges.com\[.\]br/service/](https://siges.com[.]br/service/)) does not match the legitimate domain name associated with the DHL brand, which is "dhl.com". This mismatch is a red flag as it is not coming from the legitimate source.

Considering my findings, I conclude that this site is a phishing site attempting to steal user payment information by posing as a DHL EXPRESS page.

Here's the JSON-formatted output of my findings:

```

```
{
    "phishing_score": 9,
    "brands": "DHL EXPRESS",
    "phishing": true,
    "suspicious_domain": true
}
```

```

### 5.3 Evidence for Identifying Phishing Sites

We analyze the evidence used by GPT-4 to identify phishing sites. We provide a detailed explanation of each of the seven types of evidence used to detect phishing sites, including the characteristics of the corresponding SE techniques and how ChatGPT identified them.

**Domain Name is Not Legitimate** The most crucial evidence for identifying phishing sites is when the inspected website impersonates a legitimate brand but has a domain name that does not match the official domain name. The prompt of our proposed method instructs ChatGPT to extract the brand name from the HTML or OCR-extracted text and compare it with the legitimate domain name. In many cases, ChatGPT accurately detected phishing sites by identifying inconsistencies between the extracted brand and its official domain name. ChatGPT can also detect if the domain name included in the provided URL is a fake domain name (domain squatting [20, 40]) attempting to deceive users. For instance, ChatGPT correctly recognized that `disc0rd[.]pro`, which displayed a download page for Discord's installer, was not the legitimate domain name `discord[.]com`. It also identified that `amazon761[.]com`, which displayed the installation page for Amazon's mobile app, did not match `amazon[.]com`. Moreover, ChatGPT was able to discern that `aib[.]monster`, which showed the login page for Allied Irish Bank, although having the same second-level domain as the legitimate one (`aib[.]ie`), was not legitimate due to the different top-level domain (TLD). The explanation is as follows: *The URL provided in the question is "[http://aib\[.\]monster/](http://aib[.]monster/)", but the actual legitimate domain for AIB is "[https://www.aib\[.\]ie/](https://www.aib[.]ie/)".*

**Fake Virus Infection Warning** We also observed the effectiveness of ChatGPT in identifying fake malware infection warnings, a commonly used social engineering attack on the web. For instance, the website `landiingpages[.]beauty`, which displayed a screen resembling Microsoft Windows Defender's virus scan, was analyzed to display fake security warnings such as *このPCへのアクセスはセキュリティ上の理由でブロックされています (Access to this PC is blocked for security reasons)* and *脅威を発見 (Threat found) - Trojan Spyware App: Ads.financetrack(1).exe*. ChatGPT determined that these warnings aim to make the user believe their computer is in danger. Furthermore, the website `p1fkuyacu.duckdns[.]org`, which displayed the logo of mobile carrier SoftBank, was recognized for a series of social engineering tactics. After a fake security warning stating *マルウェアが検出されました (Malware detected)*, the user was urged to install the *SoftBankセキュリティ無料版アプリ (SoftBank Security App Free Edition)*.

**Fake Account Issue** Phishing sites often display warnings claiming that there are issues with the user's account, aiming to steal their login credentials. An example of this is the website `m4gence.web[.]app`, which ChatGPT analyzed as a phishing site targeting the bank Société Générale. ChatGPT identified that it employed a SE technique as follows: *The text extracted using OCR alerts the user to a problem with their account, which can be a typical technique of phishing attacks.* Furthermore, ChatGPT explained the presence of suspicious HTML elements on this website as follows: *The HTML contains a link to a different domain than the one specified in the visible text: "https://dev-sgwebnetauth.pantheonsite[.]io/wp-content/sysconnect" instead of "www.societe generale[.]fr/synchronisation-agence."* Thus, ChatGPT has a high capability of HTML analysis, such as detecting inconsistencies between the text of the `a` element and the link specified in the `href`.

**Urgent Payment Request** As shown in Section 5.2, phishing sites that impersonate courier services and generate alerts about package issues, while creating a sense of urgency for payment, are targeting brands worldwide. An example is `royalmail-online[.]com`, which disguised as Royal Mail and requested additional payment due to the overweight of the shipment. ChatGPT identified this as a phishing site based on the mismatch in the domain name and the presence of phrases such as *Interrupted delivery* and *Additional charges of 0.76€*, indicating the employment of SE techniques. In the case of a phishing site (`www.caaarem[.]mx`) impersonating UPS, ChatGPT detected the suspicious phrase *790,45 HUF fizetése: Szál-lítási költség (Pay 790,45 HUF: Shipping cost)*. Additionally, ChatGPT provided an explanation stating, *The request for the user's credit card information, including card number, expiration date, and security code.* By analyzing the HTML, ChatGPT gained a detailed understanding of the elements prompting user input. Moreover, `tlmcjohsvz.duckdns[.]org` was identified as a phishing site demanding payment for outstanding tax obligations. ChatGPT provided the following statement as evidence: *The HTML contains alerting text that seems like a phishing attempt, such as "差押最終通知" (Final Attachment Notice), informing the user about an unpaid tax and urging them to pay using specific methods.*

**Fake Login Error** As demonstrated in Section 5.2, some phishing sites display misleading error messages, such as indicating that the login credentials are wrong even when no input has been provided or alerting users that their credit card number is invalid. These SE techniques aim to deceive users and extract sensitive information. For instance, the website `www.interceptionbookingconfirmation[.]com`, pretending to be Booking.com, prompted users to input their card information within 24 hours, accompanied by an alert stating, *Your credit card was marked as invalid.* ChatGPT explained as follows: *The presence of a "Wrong Credential" error message in the HTML text is a common social engineering technique used to trick users into re-entering their login credentials.* Similarly, ChatGPT identified the website `mail.sikkimrajshree[.]in`, which impersonated the logistics company SF Express. Based on the displayed text on the website, ChatGPT categorized it as a phishing site, providing the following statement as evidence: *The HTML content contains an error message '无效的用户名或密码' which translates to 'Invalid username or password', indicating that the user's account*

*might have a problem, which can be a social engineering technique used in phishing attacks.*

**Fake Reward** Phishing sites often employ deceptive tactics to capture users' interest, and one common initial step is to offer fake rewards, such as monetary prizes, cryptocurrencies, gift cards, or popular products like smartphones. For instance, the website `www.mobilegoodies4you[.]com` impersonates SFR, a telecommunication company, and displayed a fraudulent cash prize. ChatGPT identified the phrase *Votre adresse IP a été tirée au sort et vous avez une chance de gagner un 460,00 € en espèces! (Your IP address has been entered into a draw for a chance to win €460.00 in cash!)* within the website as an example of an unexpected reward. Similarly, ChatGPT identified the phishing site `allesettlemenie[.]top`, which impersonates the e-commerce brand Allegro, as using SE techniques to captivate users' interest such as *Sign up or log in today to shop and earn cash! Withdrawal to bank account (limited to one mobile number).* In the case of a phishing site (`(me7q1.vetedeb[.]xyz)` targeting Sberbank, users are prompted to participate in a survey in exchange for cash. ChatGPT recognized the use of a cash reward to attract users' interest, as indicated by the statement: *The text contains social engineering techniques such as enticing users with rewards ('ПОЛУЧИТЬ ВОЗМОЖНОСТЬ ПОЛУЧИТЬ НА БАНКОВСКОЙ КАРДО 600 000 РУБ'), which means "get reward up to 600,000 RUB."*

**SMS Authentication Request** Phishing sites not only target users' login credentials but also attempt to obtain SMS authentication codes from them. For instance, the website `demo.crustncakes[.]com` impersonates the payment service BenefitPay and deceived users into inputting an SMS code for the purpose of verifying a transaction. ChatGPT explained that this technique as follows: *The OCR-extracted text informs about an expired SMS code that needs to be re-entered, which can be considered as a social engineering technique to alert users.*

## 5.4 False Positives (GPT-4)

GPT-4 identified phishing sites with high accuracy, however, it contained a small number of false positives (17). We analyzed the causes of these false positives by categorizing them into seven factors based on the responses generated by ChatGPT.

**Misidentifying SE Techniques** As discussed in Section 5.3, ChatGPT demonstrated precise identification of various SE techniques commonly used in phishing sites. However, it falsely flagged some non-phishing sites as phishing. In our experiment, GPT-4 misidentified `cfspart.impots.gouv[.]fr` as containing SE techniques in HTML elements. This element, initially hidden using the `display: none` property, triggers an alert message upon receiving incorrect input in the form. GPT-4 recognized `impots.gouv[.]fr` as a legitimate domain name, however, it misidentified `cfspart.impots.gouv[.]fr` as a different domain name because of the existence of the subdomain "cfspart". Although there were other cases where GPT-4 reported that some legitimate sites contained elements using SE techniques, they were all classified as non-phishing because GPT-4 correctly verified that they matched legitimate domain names.

**Multiple Domain Names of Legitimate Brands** ChatGPT sometimes misclassifies legitimate brands that operate multiple domains as phishing sites. This occurs when ChatGPT can identify one or

more of these domain names but does not have the knowledge of the specific domain name that is being checked. In our experiment, GPT-4 produced four false positives for the following domain names: m.botw[.]com (an alternate domain of bankofthewest[.]com), aplusfcu.balancepro[.]org (aplusfcu[.]org), hb.redlink.c om[.]ar (bancodelapampa.com[.]ar), and cloudflare[.]net (cloudflare[.]com). Additionally, it falsely flagged the law firm Pinsent Masons' website site www.aboutcookies[.]org as phishing. This false positive occurred because the extracted brand name Pinsent Masons was not present in the domain name.

**Global Brands with Local Domain Names** False positives can occur when global brands use local domain names specific to different countries. GPT-4 produced false positives for three domain names due to their mismatch with the most recognized legitimate domain names: www.aliexpress[.]us, sube.garantibbva.com[.]tr, and www.sicredi.com[.]br.

**Numerous Subdomains** ChatGPT sometimes fails to recognize a legitimate domain name if it has multiple subdomains, even if the domain name is under a legitimate domain name. The following domain names were false positives in this category: cloud.walletconnect[.]com, entry11.bk.mufg[.]jp, and www.my.commbank.com[.]au.

**Non-English Websites** While ChatGPT demonstrated high accuracy in identifying SE techniques across a wide range of languages, false positives can occur due to insufficient knowledge of non-English websites. The following two domain names were falsely flagged as they were not well-known: justhost[.]ru and www.jb51[.]net.

**Long URL Query String** ChatGPT correctly recognized the website auth.talktalk.co[.]uk as legitimate and did not detect any SE techniques within the web page content. However, it falsely flagged this site as phishing due to an extremely long URL query string (360 characters) in the URL path.

## 5.5 False Negatives (GPT-4)

GPT-4 demonstrated an improvement in reducing false negatives compared to GPT-3.5, achieving an 11.7% reduction. However, there are cases where GPT-4 erroneously classifies certain phishing sites as non-phishing (16 cases), necessitating a thorough analysis of the underlying causes. Out of the 16 FNs, except for 2 cases, GPT-4 correctly identified the brand names of the remaining websites. Among these FNs, 8 websites lacked any descriptive text other than JSON-formatted content in their responses, making it impossible to analyze the basis of their classification. There were 4 FNs where GPT-3.5 correctly identified the phishing sites, but GPT-4 did not.

**Falsely Identified Domain Names as Legitimate** Despite correctly identifying the targeted brand names for the following 5 phishing sites, GPT-4 erroneously classified the domain names as legitimate. For instance, GPT-4 misclassified the phishing site www.phototan-push[.]de (disguised as commerzbank[.]com) by stating *there's no suspicion surrounding the domain name*, thus incorrectly labeling it as a non-phishing site. The phishing site www.gmendororacingteam[.]it was identified as non-phishing by GPT-4, because the legitimate URL of the brand ([https://www.mooney\[.\]it/](https://www.mooney[.]it/)) that this phishing site deceived was included as

a link in the HTML, even though the actual domain name is different. The following three sites were identified as false negatives by GPT-4, but correctly identified by GPT-3.5. A website impersonating www.gov[.]uk (claim.redundancy-payments.org[.]uk) was classified as phishing by GPT-3.5 due to its SE technique of requesting the national insurance numbers and bank details. In contrast, GPT-4 accurately identified the brand as "GOV.UK" but indicated *it is likely that the site is a legitimate one*. GPT-4 also suggested *it is always better to verify the authenticity of the URL by visiting the official government website*. etmmetaverse[.]com, a phishing site for Office 365, was correctly classified as phishing by GPT-3.5 based on the mismatch in the domain name and links to other pages. However, GPT-4 offered a neutral explanation, stating *there is not enough evidence supporting that this is a phishing site*, despite assigning a *phishing\_score* of 5 and labeling both *phishing* and *suspicious\_domain* as *unknown*. Furthermore, both GPT-3.5 and GPT-4 correctly identified the brand SI-DEP (France's national Covid-19 screening system) for the phishing site ca8567c7fa0141658f08b0dabe13d5ee.v1.radwarecloud.net. While GPT-3.5 correctly pointed out *The domain name "radware-cloud.net" does not appear to be associated with the brand name "SI-DEP"*, GPT-4 failed to detect the phishing attempt.

**Failure to Identify Domain Squatting** The website www.americanexpressseguros[.]com was a phishing site for American Express, offering insurance solicitations. Despite containing the phrase *Recibe hasta \$2,000.00 M.N de bonificación al contratar* (*Receive up to \$2,000.00 M.N of bonus when contracting*), GPT-4 failed to recognize the SE technique employed. Although it was a case of domain squatting, GPT-4 incorrectly identified it as a legitimate domain.

**Failure to Identify SE Techniques** The phishing site phpstack-197144-1061735.cloudwaysapps[.]com presented an insurance survey, without specifically targeting any particular brand. Since it lacks prominent elements such as rewards or a sense of urgency, it was mistakenly classified as a legitimate site. Similarly, the e-commerce site lojanewgeneration[.]com attempted to attract users with the phrase *daily offers*, yet GPT-4 failed to identify the SE technique employed, leading to its classification as a legitimate site.

## 5.6 Comparative Analysis of Phishing Site Detection: GPT-4 vs. GPT-3.5

We conducted a comparative analysis of the capabilities of GPT-4 and GPT-3.5 in accurately identifying phishing sites. Specifically, we thoroughly analyzed the phishing sites that were successfully detected by GPT-4 but failed by GPT-3.5. Among the 133 FNs of GPT-3.5, a total of 121 phishing sites were correctly identified by GPT-4. Upon analyzing these 121 phishing sites, we discovered three abilities in which GPT-4 exhibited superior performance compared to GPT-3.5:

- Ability to determine the authenticity of domain names.
- Ability to recognize SE techniques.
- Ability to detect phishing sites comprehensively by using multiple factors.

**Ability to Determine the Suspiciousness of Domain Names** GPT-4 and GPT-3 showed differences in their ability to assess

the suspiciousness of domain names. This involves determining whether the domain name matches a legitimate domain name or has suspicious characteristics. For instance, when analyzing the OpenAI phishing site `openai-gpt-4[.]com`, GPT-3 classified it as non-phishing due to the presence of the term “`openai`.” In contrast, GPT-4 correctly identified it as phishing since the actual domain name, `openai[.]com`, differs from the suspicious one. In the case of a phishing site (`krakken-logi.mystrikingly[.]com`) impersonating the cryptocurrency exchange Kraken, both GPT-3 and GPT-4 successfully recognized the brand name. However, GPT-4 identified the misspelling of “`krakken-logi`” as suspicious, whereas GPT-3 erroneously classified it as a legitimate domain name based on the presence of “`krakken`”. Furthermore, both GPT-3.5 and GPT-4 accurately identified the phishing site `firstcitizncb[.]com` as distinct from the legitimate domain name `firstcitizens[.]com`. However, GPT-3.5 considered it to be a variation of the legitimate domain name and not a phishing site. Another example is the phishing site `correos-es.firebaseioapp[.]com`, which impersonates Spain’s national postal service, Correos. While GPT-3.5 classified it as a legitimate domain name, GPT-4 correctly recognized it as a suspicious domain due to its hosting on Google Firebase.

**Ability to Identify SE Techniques** GPT-4 successfully identified SE techniques that were missed by GPT-3.5 in some phishing sites. For example, in the case of `verify.vodafone-uk[.]com`, which GPT-3.5 explained that *The page warns users about the potential termination of their phone number*. However, GPT-4 recognized the presence of SE techniques by explaining that the websites warned users about the potential termination of their phone number. Additionally, GPT-4 noted that *The domain name in the URL is "verify.vodafone-uk.com", which seems odd with the hyphen between "vodafone" and "uk".* In the phishing site (`aktivere-bnkid.web[.]app`) impersonating Sweden’s ID system, BankID, GPT-4 identified the SE technique by highlighting that it requested personal information such as the user’s personal number, phone number, and user-ID. In contrast, GPT-3 incorrectly states that the website *does not contain any obvious social engineering techniques*.

**Ability to Detect Phishing Sites Comprehensively by Using Multiple Factors** In some cases, ChatGPT may have incorrectly determined that legitimate text on the website was suspicious. For example, it identified legitimate error messages (e.g., “wrong password”) displayed after submitting a form as indicative of social engineering. It also flagged the inclusion of the year 2023 in a copyright notice as suspicious. This is because ChatGPT’s training data only goes back to September 2021, and it lacks knowledge of the current year, which could lead to inaccurate results. However, GPT-4, while occasionally providing irrelevant observations, tends to make overall accurate determinations by combining them with more reliable information. On the other hand, GPT-3.5 may classify a phishing site as benign even though it correctly recognizes messages using SE techniques in the HTML, or the domain name differs from the legitimate one. The following are some examples of non-phishing sites. For instance, the website `b2meguy.com` displayed an attention-grabbing statement at the top of the website: *“Obtenga acceso instantáneo al software de Quantum AI, y gane de 3200€ al día!”* (*Get instant access to Quantum AI’s software, and earn from 3200€ per day!*). Both GPT-3.5 and GPT-4 identified this statement as suspicious, however, while GPT-3.5 did not classify it as a phishing site,

GPT-4 correctly identified it as phishing based on the explanation that *it presents unrealistic promises of gains*. Another example is the phishing site `verification-appeal-code.firebaseioapp[.]com`, which pretended to be Facebook and asked users to respond to claims of intellectual property infringement. Both GPT-3.5 and GPT-4 detected the request for sensitive information from visitors, however, GPT-3.5 incorrectly classified it as a legitimate site.

## 6 LIMITATIONS

ChatGPT provides flexible and varied responses because its output is determined probabilistically. However, this also means that detection results can change depending on the experiment. In this study, we used default parameters for ChatGPT and prepared a dataset consisting of various websites to statistically evaluate the detection accuracy. In order to obtain reliable results for the classification of phishing sites, some parameters such as *Top\_k* and *Temperature* may require adjustments for specific purposes.

For the dataset used in this experiment, we excluded phishing sites and non-phishing sites containing R-18 content to eliminate the influence of ChatGPT’s content filter. To examine websites with adult or aggressive content, it will be necessary to modify or select an appropriate content filter for ChatGPT.

Methods that use LLMs, including the proposed approach, may be susceptible to prompt injection attacks [17, 32], where the original prompt is overwritten and malicious content is inserted into the response. Our proposed method employs simple countermeasures by simplifying the HTML and clarifying text sections through the placement of triple backticks. However, phishing sites may use prompt injections in the future to avoid analysis by LLMs. More advanced defenses against prompt injection attacks will be needed.

Because ChatGPT has been trained on data up to September 2021, it may not be able to accurately classify phishing and non-phishing sites associated with services created after that date. The classification ability could potentially be improved by externally referencing a list of domain names corresponding to brands or by using a fine-tuned LLM.

## 7 RELATED WORK

An effective method for detecting phishing sites is to compare the appearance of a website with that of a legitimate one. Previous studies such as Abdelnabi et al. [11], Lin et al. [24], and Liu et al. [25] detect the abuse of logo images by identifying and comparing them with legitimate ones, or extract features from the overall appearance of web pages to analyze their similarity. While these deep learning-based methods are effective in detecting phishing sites created by copying HTML or reusing logos, they cannot detect phishing sites that use their own logos or do not impersonate brands. Our proposed method aims to identify phishing sites that use SE techniques by analyzing the context of the web page, allowing for the detection of different types of phishing sites, not limited to the misuse of branding.

Methods have been proposed to detect domain squatting, where attackers obtain domain names similar to legitimate service domain names. Nikiforakis et al. [29], Quinkert et al. [34], Kintis et al. [20], Agten et al. [12], and dnstwist [1] have proposed techniques that use rule-based approaches or machine learning models to identify

malicious domain names by considering various factors such as minor variations, the inclusion of brand names, and changes in top-level domains. ChatGPT showed high capabilities against domain squatting, as demonstrated in our evaluation experiments, by detecting minor character differences and fake domain names containing brand names. There are also methods that use TLS certificates to detect phishing sites. Kim et al.[19], Bijnmans et al.[13], and Drichel et al. [16] have proposed such approaches that use machine learning algorithms to analyze features derived from TLS certificates or certificate transparency logs to detect phishing sites.

In recent years, there have been attempts to apply LLMs to cybersecurity. One such example is VirusTotal Code Insight [10], which is based on Google's security-focused LLM, Sec-PaLM. This model can explain the purpose and functionality of malware code in natural language, allowing security experts to gain a general understanding of the code's intentions. An article [6] analyzed phishing URLs using ChatGPT with a simple prompt. The experiment showed a detection rate of 87.2% and a false positive rate of 23.2%, indicating that while it can block phishing sites, it can also mistakenly block legitimate sites. In contrast, the method proposed in this paper includes not only the URL, but also HTML, text extracted from screenshots using OCR, and carefully designed prompts to achieve superior detection accuracy. Roy et al. [36] explored the ability of ChatGPT to automatically generate phishing sites. Their findings suggest that these generated phishing sites can mimic popular brands and employ various evasion tactics to evade anti-phishing systems. To counteract such automated phishing campaigns, it becomes important to employ security measures using LLMs, such as our proposed method.

## 8 CONCLUSION

In this study, we have presented a novel method for detecting phishing sites using ChatGPT, a state-of-the-art large language model. Our approach combines web crawling techniques with ChatGPT's contextual understanding to classify websites as phishing sites or not. Through detailed experiments, our method achieved remarkable performance with a precision of 98.3% and a recall of 98.4%, demonstrating the potential of LLMs to efficiently detect phishing sites and uncover social engineering techniques. Furthermore, our comparative analysis between GPT-3.5 and GPT-4 revealed significant improvements in GPT-4's ability to minimize false negatives. GPT-4 excelled at identifying suspicious domains, detecting social engineering techniques in web content, and considering multiple factors in its evaluation. These findings have important implications for enhancing automated cybersecurity measures and mitigating the risks of online fraudulent activities faced by users. This research provides new directions for exploring the application of LLMs in various cybersecurity domains and further optimizing their capabilities for analyzing and detecting malicious web content.

## REFERENCES

- [1] [n. d.]. GitHub - elceef/dnstwist. <https://github.com/elceef/dnstwist>
- [2] 2023. Azure Cognitive Service for Vision with OCR and AI | Microsoft Azure. <https://azure.microsoft.com/en-us/products/cognitive-services/vision-services>
- [3] 2023. Azure OpenAI Service – Advanced Language Models | Microsoft Azure. <https://azure.microsoft.com/en-us/products/cognitive-services/openai-service>
- [4] 2023. Chrome DevTools Protocol. <https://chromedevtools.github.io/devtools-protocol/>.
- [5] 2023. Introducing ChatGPT. <https://openai.com/blog/chatgpt>
- [6] 2023. Investigating ChatGPT phishing detection capabilities | Securelist. <https://securelist.com/chatgpt-anti-phishing/109590/>
- [7] 2023. Mimino666/langdetect: Port of Google's language-detection library to Python. <https://github.com/Mimino666/langdetect>
- [8] 2023. OpenPhish. <https://openphish.com/>
- [9] 2023. PhishTank. <https://phishtank.org/>
- [10] 2023. VirusTotal Code Insight. <https://blog.virustotal.com/2023/04/introducing-virustotal-code-insight.html>
- [11] Sahar Abdelnabi, Katharina Krombholz, and Mario Fritz. 2020. VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity. In *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, Jay Ligatti, Xinning Ou, Jonathan Katz, and Giovanni Vigna (Eds.). ACM, 1681–1698.
- [12] Pieter Agten, Wouter Joosen, Frank Piessens, and Nick Nikiforakis. 2015. Seven Months Worth of Mistakes: A Longitudinal Study of Typosquatting Abuse. In *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015*. The Internet Society.
- [13] Hugo L. J. Bijnmans, Tim M. Booij, Anneke Schewdersky, Aria Nedgabat, and Rolf van Wegberg. 2021. Catching Phishers By Their Bait: Investigating the Dutch Phishing Landscape through Phishing Kit Detection. In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, Michael Bailey and Rachel Greenstadt (Eds.). USENIX Association, 3757–3774. <https://www.usenix.org/conference/usenixsecurity21/presentation/bijnmans>
- [14] M. Bitaab, H. Cho, A. Oest, Z. Lyu, W. Wang, J. Abraham, R. Wang, T. Bao, Y. Shoshtaishvili, and A. Doupé. 2023. Beyond Phish: Toward Detecting Fraudulent e-Commerce Websites at Scale. In *2023 2023 IEEE Symposium on Security and Privacy (SP) (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 2566–2583. <https://doi.org/10.1109/SP46215.2023.00152>
- [15] Tom B. Brown, Benjamin Mann, Nick Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *CoRR* abs/2005.14165 (2020). arXiv:2005.14165 <https://arxiv.org/abs/2005.14165>
- [16] Arthur Drichel, Vincent Drury, Justus von Brandt, and Ulrike Meyer. 2021. Finding Phish in a Haystack: A Pipeline for Phishing Classification on Certificate Transparency Logs. In *ARES 2021: The 16th International Conference on Availability, Reliability and Security, Vienna, Austria, August 17-20, 2021*, Delphine Reinhardt and Tilo Müller (Eds.). ACM, 59:1–59:12.
- [17] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. *arXiv:2302.12173 [cs.CR]*
- [18] Amin Kharraz, William K. Robertson, and Engin Kirda. 2018. SurveyLane: Automatically Detecting Online Survey Scams. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*. IEEE Computer Society, 70–86. <https://doi.org/10.1109/SP.2018.00044>
- [19] Doowon Kim, Haehyun Cho, Yonghwi Kwon, Adam Doupé, Sooel Son, Gail-Joon Ahn, and Tudor Dumitras. 2021. Security Analysis on Practices of Certificate Authorities in the HTTPS Phishing Ecosystem. In *ASIA CCS '21: ACM Asia Conference on Computer and Communications Security, Virtual Event, Hong Kong, June 7-11, 2021*, Jianrong Cao, Man Ho Au, Zhiqiang Lin, and Moti Yung (Eds.). ACM, 407–420.
- [20] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. 2017. Hiding in Plain Sight: A Longitudinal Study of Combosquatting Abuse. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, Bhavani Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu (Eds.). ACM, 569–586.
- [21] Takashi Koide, Daiki Chiba, and Mitsuaki Akiyama. 2020. To Get Lost is to Learn the Way: Automatically Collecting Multi-step Social Engineering Attacks on the Web. In *ASIA CCS '20: The 15th ACM Asia Conference on Computer and Communications Security, Taipei, Taiwan, October 5-9, 2020*, Hung-Min Sun, Shiu-Hyng Shieh, Guofei Gu, and Giuseppe Ateniese (Eds.). ACM, 394–408. <https://doi.org/10.1145/3320269.3384714>
- [22] Takashi Koide, Daiki Chiba, Mitsuaki Akiyama, Katsunari Yoshioka, and Tsutomu Matsumoto. 2020. It Never Rains but It Pours: Analyzing and Detecting Fake Removal Information Advertisement Sites. In *The 17th Conference on Detection of Intrusions and Malware & Vulnerability Assessment, DIMVA 2020*. 171–191.
- [23] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. In *NeurIPS*.

- [http://papers.nips.cc/paper\\_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html)
- [24] Yun Lin, Ruofan Liu, Dinil Mon Divakaran, Jun Yang Ng, Qing Zhou Chan, Yiwen Lu, Yuxuan Si, Fan Zhang, and Jin Song Dong. 2021. Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify Phishing Webpages. In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, Michael Bailey and Rachel Greenstadt (Eds.). USENIX Association, 3793–3810.
- [25] Ruofan Liu, Yun Lin, Xianglin Yang, Siang Hwee Ng, Dinil Mon Divakaran, and Jin Song Dong. 2022. Inferring Phishing Intention via Webpage Appearance and Dynamics: A Deep Vision Based Approach. In *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, Kevin R. B. Butler and Kurt Thomas (Eds.). USENIX Association, 1633–1650.
- [26] Najmeh Miramirkhani, Oleksii Starov, and Nick Nikiforakis. 2017. Dial One for Scam: A Large-Scale Analysis of Technical Support Scams. In *24th Annual Network and Distributed System Symposium, NDSS*.
- [27] Hiroki Nakano, Daiki Chiba, Takashi Koide, Naoki Fukushi, Takeshi Yagi, Takeo Hariu, Katsunari Yoshioka, and Tsutomu Matsumoto. 2023. Canary in Twitter Mine: Collecting Phishing Reports from Experts and Non-experts. arXiv:2303.15847 [cs.CR]
- [28] Terry Nelms, Roberto Perdisci, Manos Antonakakis, and Mustaque Ahamad. 2016. Towards Measuring and Mitigating Social Engineering Software Download Attacks. In *25th USENIX Security Symposium, USENIX Security 16*. 773–789.
- [29] Nick Nikiforakis, Steven Van Acker, Wannes Meert, Lieven Desmet, Frank Piessens, and Wouter Joosen. 2013. Bitsquatting: exploiting bit-flips for fun, or profit? In *22nd International World Wide Web Conference, WWW ’13, Rio de Janeiro, Brazil, May 13-17, 2013*, Daniel Schwabe, Virgilio A. F. Almeida, Hartmut Glaser, Ricardo Baeza-Yates, and Sue B. Moon (Eds.), 989–998.
- [30] OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023). <https://doi.org/10.48550/arXiv.2303.08774> arXiv:2303.08774
- [31] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*. [http://papers.nips.cc/paper\\_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html)
- [32] Fábio Perez and Ian Ribeiro. 2022. Ignore Previous Prompt: Attack Techniques For Language Models. arXiv:2211.09527 [cs.CL]
- [33] Victor Le Pochat, Tom van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczynski, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society. <https://www.ndss-symposium.org/ndss-paper/tranco-a-research-oriented-top-sites-ranking-hardened-against-manipulation/>
- [34] Florian Quinkert, Martin Degeling, Jim Blythe, and Thorsten Holz. 2020. Be the Phisher - Understanding Users' Perception of Malicious Domains. In *ASIA CCS ’20: The 15th ACM Asia Conference on Computer and Communications Security, Taipei, Taiwan, October 5-9, 2020*, Hung-Min Sun, Shiu-Pyng Shieh, Guofei Gu, and Giuseppe Ateniese (Eds.). ACM, 263–276.
- [35] Alec Radford and Karthik Narasimhan. 2018. Improving Language Understanding by Generative Pre-Training.
- [36] Sayak Saha Roy, Krishna Vamsi Naragam, and Shirin Nilizadeh. 2023. Generating Phishing Attacks using ChatGPT. arXiv:2305.05133 [cs.CR]
- [37] Janos Szurdi, Meng Luo, Brian Kondracki, Nick Nikiforakis, and Nicolas Christin. 2021. Where are you taking me?Understanding Abusive Traffic Distribution Systems. In *WWW ’21: The Web Conference 2021, Virtual Event /Ljubljana, Slovenia, April 19-23, 2021*, Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM / IW3C2, 3613–3624.
- [38] Siyuan Tang, Xianghang Mi, Ying Li, XiaoFeng Wang, and Kai Chen. 2022. Clues in Tweets: Twitter-Guided Discovery and Analysis of SMS Spam. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi (Eds.). ACM, 2751–2764.
- [39] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, Daniel Margolis, Vern Paxson, and Elie Bursztein. 2017. Data Breaches, Phishing, or Malware?: Understanding the Risks of Stolen Credentials. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, Bhavani Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu (Eds.). ACM, 1421–1434. <https://doi.org/10.1145/3133956.3134067>
- [40] Ke Tian, Steve T. K. Jan, Hang Hu, Danfeng Yao, and Gang Wang. 2018. Needle in a Haystack: Tracking Down Elite Phishing Domains in the Wild. In *Proceedings of the Internet Measurement Conference 2018, IMC 2018, Boston, MA, USA, October 31 - November 02, 2018*. ACM, 429–442. <https://dl.acm.org/citation.cfm?id=3278569>
- [41] Phani Vadrevu and Roberto Perdisci. 2019. What You See is NOT What You Get: Discovering and Tracking Social Engineering Attack Campaigns. In *Proceedings of the Internet Measurement Conference, IMC 2019*. 308–321.
- [42] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs.CL]
- [43] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *NeurIPS*. [http://papers.nips.cc/paper\\_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html)
- [44] Peng Yang, Guangzhen Zhao, and Pengjie Zeng. 2019. Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning. *IEEE Access* 7 (2019), 15196–15209.
- [45] Zheng Yang, Joey Allen, Matthew Landen, Roberto Perdisci, and Wenke Lee. 2023. TRIDENT: Towards Detecting and Mitigating Web-based Social Engineering Attacks. In *32st USENIX Security Symposium, USENIX Security 2023, 2023*. USENIX Association, 1681–1698.

## A DETECTION ACCURACY USING PHISHING SCORE

**Table 5: Performance Metrics Comparison Using phishing\_score.**

|           | GPT-3.5 | GPT-4        |
|-----------|---------|--------------|
| Precision | 0.938   | <b>0.981</b> |
| Recall    | 0.892   | <b>0.980</b> |
| Accuracy  | 0.916   | <b>0.981</b> |
| F-measure | 0.914   | <b>0.980</b> |

Table 5 shows the results of calculating the detection accuracy metrics - Precision, Recall, Accuracy, and F-1 measure - using the optimal threshold values for *phishing\_score* in GPT-3.5 and GPT-4.

## B PHISHING BRANDS

Table 6 is a list of 147 brands that were targeted by phishing sites in our dataset.

**Table 6: Brands Targeted by Phishing Sites.**

|                           |                                                  |                              |
|---------------------------|--------------------------------------------------|------------------------------|
| A+ Federal Credit Union   | AEON CARD                                        | Alpha Web                    |
| Amazon                    | American Express                                 | Apple                        |
| Ardoiz                    | BECU Online Banking                              | BNL                          |
| BRImo                     | Banca Sella                                      | Banco Desio                  |
| Banco La Pampa            | Bancoagrícola                                    | Bancolombia                  |
| Bank of America           | Barclays                                         | Battle.net                   |
| Bendigo Bank              | Bradesco                                         | CAJA                         |
| CIMB Ni                   | Chunghua Post Co.                                | Citizens Bank                |
| CodeSquare                | Coin Wallet                                      | Colissimo                    |
| Commonwealth Bank         | Correos                                          | Credomatic                   |
| Crypto.com                | DHL                                              | DKB                          |
| Deutsche Bank             | Deutsche Post DHL Group                          | Disney                       |
| Docomo                    | DocuSign                                         | Dropbox                      |
| EATA                      | EClick PORTAL                                    | ETC                          |
| EXPRESSPAY                | Ebay                                             | Emirates Post                |
| Entrust                   | Facebook                                         | Fifth Third Momentum Banking |
| First Citizens Bank       | GLOBAL PASS                                      | GO online                    |
| GOV.UK                    | Garanti BBVA                                     | Garena                       |
| Gazprom                   | Gemini                                           | GitLab                       |
| Google                    | Gruppo BNP Paribas                               | IBC                          |
| ING                       | ImToken                                          | Impots.gouv.fr               |
| Instagram                 | Involta                                          | JUHACHI-SHINWA BANK          |
| JUNO                      | Kakao                                            | Kusainon                     |
| La Banque Postale         | Livelo                                           | METAMASK                     |
| MIR VISA                  | M&T Bank                                         | Ma Banque                    |
| Mashreq Online Banking    | MasterCard                                       | Mercari                      |
| Microsoft                 | Minnesota Unemployment Insurance Benefits System | Mitsubishi UFJ Bank          |
| Mitsui Sumitomo Card      | Mygrow                                           | NAB Internet Banking         |
| NAVY FEDERAL Credit Union | NETFLIX                                          | NLB Banka                    |
| Nexi                      | Nordea                                           | ORLEN                        |
| OakPay                    | OneDrive                                         | Orange                       |
| OurTime                   | Outlook.com                                      | PNC Online Banking           |
| PRESTÍA                   | PancakeSwap                                      | PayPal                       |
| PayPay                    | Post canada                                      | QUOTY                        |
| Qatar Post                | RENNER                                           | REVOLUT INTERNATIONAL BANK   |
| Ronin Wallet              | SAISONCARD                                       | SBJ Bank                     |
| SDCCU                     | SFR                                              | SMBC                         |
| SMBC Trust Bank           | SPANKKI                                          | STEAM                        |
| Santander                 | Satang                                           | Schwab Safe                  |
| Scotiabank                | Security Bank                                    | SharePoint                   |
| Sicredi                   | Siemens                                          | Slovenská POŠTA              |
| Spotify                   | Swiss                                            | Swiss Post                   |
| Swisscom                  | T-Mobile                                         | TD Ameritrade                |
| TESCO Bank                | THEWEST                                          | TalkTalk                     |
| Theta                     | Transcash                                        | Trust Wallet                 |
| USAA                      | USPS Tracking                                    | Ubisoft                      |
| UniCredit                 | Uphold                                           | UPS                          |
| VALVE                     | VISA                                             | Verizon                      |
| Vietcombank               | WELLS FARGO                                      | WalletConnect                |
| WeTransfer                | Weebly                                           | Yahoo!                       |

# DETECTING PHISHING SITES USING CHATGPT

WRITTEN BY :  
Takashi Koide  
Naoki Fukushi  
Hiroki Nakano  
Daiki Chiba

PRESENTED BY :  
ADITYA APANDKAR (05)  
ABHIGYAN BAFNA (08)  
RUDRANI CHAVARKAR (13)  
MARYAM CHOWDHRY (14)



# INTRODUCTION

Large language models like ChatGPT have impacted AI, yet their potential in **detecting phishing sites** against automated cyberattacks remains **underexplored**. This study introduces an innovative approach, combining ChatGPT with a web crawler to identify and expose social engineering tactics. This research can strengthen cybersecurity, safeguarding users from online fraud.



# BACKGROUND

Phishing sites use tricks like fake emails and ads to deceive users into sharing personal info or installing malware.

They impersonate trusted brands and use psychology to manipulate users, often mimicking services like online banking and creating urgency or fake rewards.

Previous research focuses on understanding phishing site design, patterns, and detection methods, improving security, and educating users.

Challenges in detection include adapting to evolving brands targeted by phishing sites and a lack of comprehensive analysis of social engineering contexts across entire websites.





# EVALUATION

The phishing detection method using ChatGPT's evaluation is based on a range of critical metrics, including Precision, Recall, Accuracy, and the F-1 measure, to assess the method's performance.

It also compares the performance of GPT-4 and GPT-3.5, and discusses the potential applications of LLMs in enhancing cybersecurity. The method demonstrates high precision, recall, and accuracy, making it an effective tool for identifying phishing sites and social engineering techniques.

# METHOD

The methods propose utilizing ChatGPT for the detection of fraudulent websites, leveraging its proficiency in contextual comprehension to identify dubious language and discrepancies between brand identities and domains. Unlike prior methodologies, this approach does not necessitate a database of known phishing sites. Instead, it capitalizes on ChatGPT's broad knowledge of text. The process involves employing a web crawler to navigate websites, collecting various elements including images, code, and hyperlinks. Subsequently, this data is inputted into ChatGPT to determine the authenticity of the site.



# WEB CRAWLING

In this tool, a tool has been developed that operates autonomously in Google Chrome to navigate websites and amass data. When provided with a web address, the tool acquires the ultimate destination URL (accounting for any redirects), takes a screenshot, and retrieves the website's code after JavaScript execution. This is crucial as deceptive websites often employ complex JavaScript to mask their true intent. Evaluating the code prior to JavaScript execution can yield inaccurate results. The tool can emulate two configurations - one simulating a Windows and Chrome environment, and the other an iPhone and Safari setup - by adjusting its user behavior accordingly.



# PROMPT

This method streamline the HTML code while ensuring vital information for phishing detection and brand identification remains intact. Their emphasis is on preserving elements such as title, meta description, login input forms, and visible on-page text, all of which are pivotal for analysis. They implement a systematic approach to achieve this.

Analyze if the website uses deceptive techniques to attract users, like offering cash prizes or fake malware warnings. If these phrases appear without context in a non-login state, it's likely a phishing site.

Extract the brand name of the website. Phishing sites sometimes copy resources from legitimate sites, making it hard to verify authenticity based on HTML alone. So, the model is prompted to confirm if the URL matches the real site.

Decide if the site is phishing or not, and provide a clear rationale for the decision. Detailed explanations improve accuracy and make it easier for humans to review.

Generate output in JSON format. If the website uses deceptive techniques, it should be flagged as phishing. If the brand name is identified and the domain name is different from the legitimate one, it's considered suspicious. Additionally, a phishing score from 0 to 10 should be given

# SIMPLIFYING HTML



Here, the method employs a process to simplify the HTML code while ensuring vital information for phishing detection and brand identification remains intact. Their focus is on preserving elements like title, meta description, login input forms, and visible on-page text, all of which are crucial for analysis.

1. They eliminate less relevant HTML elements such as style, script, and comment tags, which are often lengthy and less informative.
2. They unwrap HTML elements, retaining only essential ones like head, title, meta, body, and specific tags for headings, paragraphs, links, images, and more.
3. They remove unimportant elements and integrate their child elements into their parent elements.
4. They get rid of HTML elements lacking text within tags.
5. They shorten the source attribute for images encoded in base64 and the link attribute for lengthy URLs.

# SIMPLIFYING OCR EXTRACTED TEXT

In cases where websites have a lot of text, both from OCR extraction and HTML, it's possible to exceed the token limit. To address this, our method also simplifies the OCR-extracted text. It does this by looking at the font sizes of the identified text and gradually removing sentences starting from the smallest font size. This process continues until the text contains fewer than 500 tokens. This helps ensure that the text remains within the processing limits.



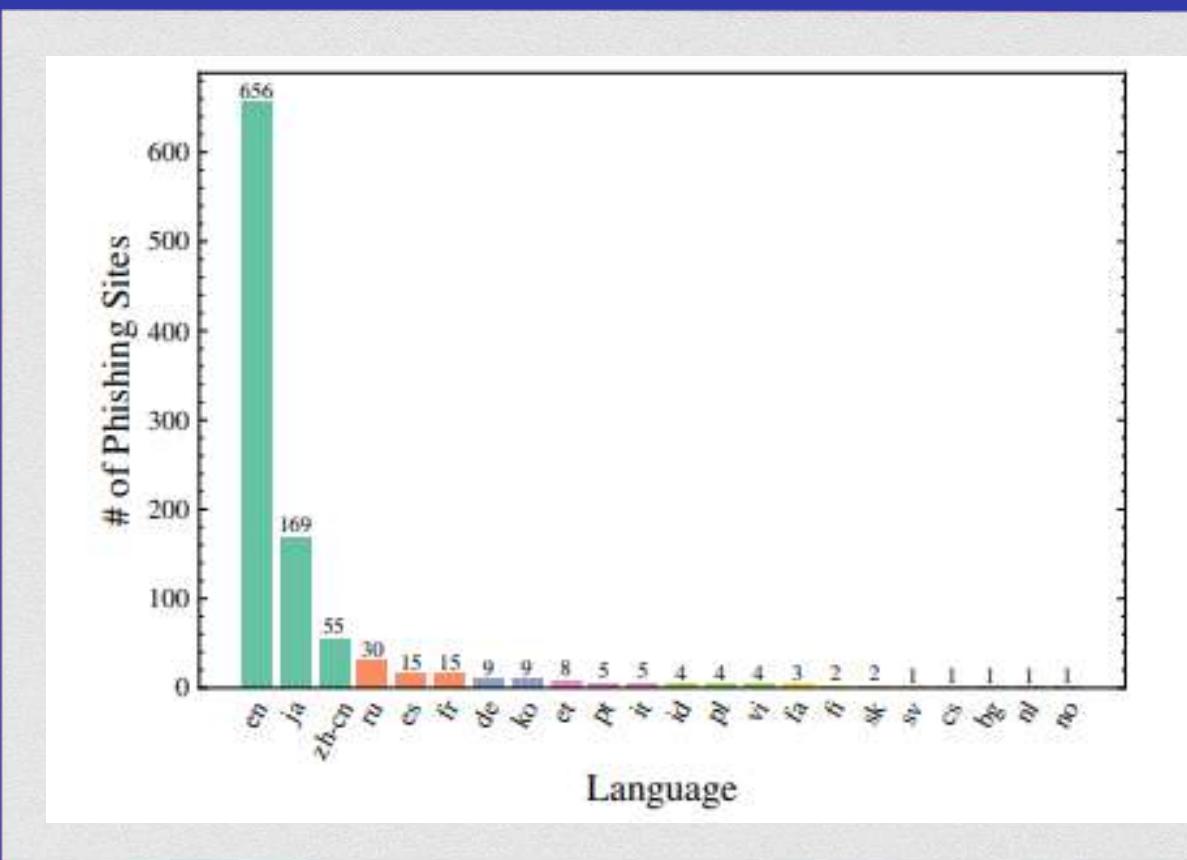
# DATA SET CREATION

- Dataset includes 1,000 phishing sites and an equal number of non-phishing sites.
- Phishing sites collected from OpenPhish, PhishTank, and CrowdCanary using keywords.
- Non-phishing sites gathered from legitimate sources and top domain names.
- Data collection took place from March to April 2023.



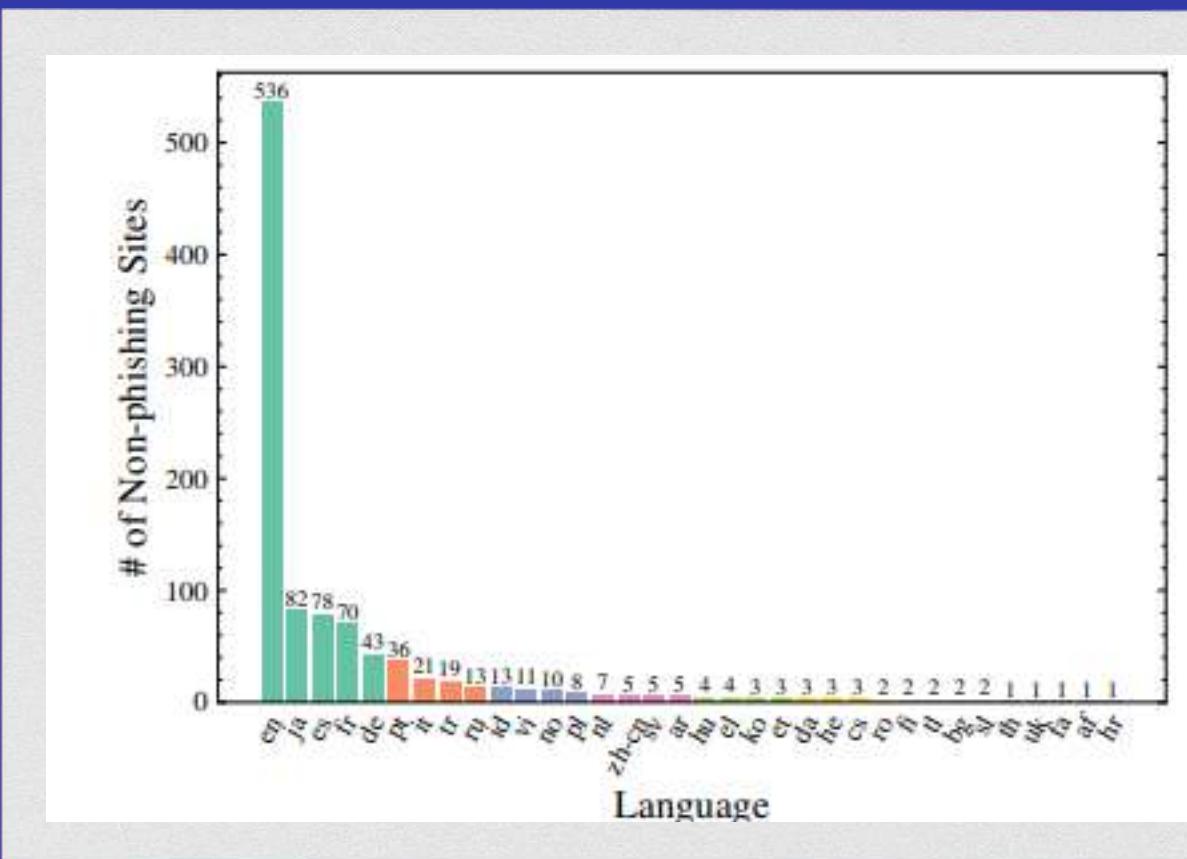
# PHISING SITES

- Phishing sites mimic legitimate services and create urgency.
  - Excluded sites with adult, gambling, or dating content.
  - Removed duplicates unless they were in different languages.
  - Resulted in 1,000 unique phishing sites in 22 languages



# NON - PHISING SITES

- Collected 764 non-phishing sites, balancing the dataset.
  - Excluded adult, illegal, and gambling-related sites.
  - Non-phishing sites written in 34 different languages.



# SIMPLIFYING DATA

- Simplified HTML and text to manage complexity.
- Non-phishing sites tend to be more complex.
- Data simplification was needed to fit within model's token limit.



# LIMITATIONS

1. ChatGPT's variable responses impact detection results.
2. Adjusting parameters may be necessary for specific needs.
3. The right content filter is crucial for certain websites with aggressive content.
4. Advanced defenses may be needed against evolving phishing techniques.
5. ChatGPT's training data is limited to pre-September 2021; improving accuracy may require external references or fine-tuned models.



# **98.3% PRECISION**

# **98.4% RECALL**

## **USING GPT-4**

The research introduced a novel method using ChatGPT to detect phishing sites, achieving remarkable results. GPT-4 showed significant improvements over GPT-3.5, excelling in spotting suspicious domains and social engineering techniques. This research enhances automated cybersecurity and offers insights into leveraging LLMs for detecting malicious web content.

**THANK  
YOU VERY  
MUCH!**

