POORNIMA
COLLEGE OF ENGINEERING

Affiliated to RTU, Kota • Approved by AICTE & UGC under 2(f) • NAAC A+ Accredited

**ISI-6, RIICO Institutional Area, Sitapura, Jaipur-302022, Rajasthan**

**Phone/Fax: 0141-2770790-92, www.pce.poornima.org**

# Security Lab

**(Lab Code: 7IT4-22)**

**7th Semester, 4th Year**

**Department of Information Technology**

# TABLE OF CONTENT

## INSTITUTE VISION & MISSION

**VISION**

To create knowledge based society with scientific temper, team spirit and dignity oflabor to face the global competitive challenges

**MISSION**

To evolve and develop skill based systems for effective delivery of knowledge so asto equip young professionals with dedication & commitment to excellence in all spheres of life.

## POORNIMA COLLEGE OF ENGINEERING, JAIPUR
## DEPARTMENT OF INFORMATION TECHNOLOGY

## VISION

To attain distinction in education to enable students for their establishment as globally competent professional and empowering them with proficiency, knowledge and research ability required to be successful in field of Information Technology.

## MISSION

1. To provide state-of-the-art facilities with modern IT tools to students and faculty thereby enabling them to develop sustainable solutions for real world problems.
2. To create and propagate knowledge in field of Information Technology through research, teaching and learning for meeting societal challenges.
3. To inculcate analytical, leadership and team working skills with ethical behavior in students and provide an environment for continuous learning.

# RTU SYLLABUS AND MARKING SCHEME

| **7IT4-22: Security Lab** | |
|---|---|
| **Credit: 2** | **Max. Marks:100(IA:60, ETE:40)** |
| **0L+0T+4P** | **End Term Exam: 2 Hours** |

| **S. No.** | **List of Experiments** |
|---|---|
| | Implement the following Substitution & Transposition Techniques concepts: a) Caesar Cipher b) Rail fence row & Column Transformation |
| 2. | Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob). |
| 3. | Implement the following Attack: a) Dictionary Attack b) Brute Force Attack |
| 4. | Installation of Wire shark, tcpdump, etc. and observe data transferred in client server communication using UDP/TCP and identify the UDP/TCP datagram. |
| 5. | Installation of rootkits and study about the variety of options. |
| 6. | Perform an Experiment to Sniff Traffic using ARP Poisoning. |
| 7. | Demonstrate intrusion detection system using any tool (snort or any other s/w). |
| 8. | Demonstrate how to provide secure data storage, secure data transmission and for creating digital signatures |

## EVALUATION SCHEME

| **I+II Mid Term Examination** | | | **Attendance and performance** | | | **End Term Examination** | | | **Total Marks** |
|---|---|---|---|---|---|---|---|---|---|
| **Experiment** | **Viva** | **Total** | **Attendance** | **Performance** | **Total** | **Experiment** | **Viva** | **Total** | |
| 30 | 10 | 40 | 10 | 30 | 40 | 30 | 10 | 40 | 100 |

## DISTRIBUTION OF MARKS FOR EACH EXPERIMENT

| **Attendance** | **Record** | **Performance** | **Total** |
|---|---|---|---|
| 2 | 3 | 5 | 10 |

# LAB OUTCOME AND ITS MAPPING WITH PO & PSO

## LAB OUTCOMES

| | |
|---|---|
| 7IT4-22.1 | Analyze the data transferred and protocol using different security-based tools like Wireshark, tcpdump, rootkits, snort etc. |
| 7IT4-22.2 | Design the substitution and transposition techniques for plain text Encryption and decryption. |
| 7IT4-22.3 | Observe ARP Poisoning, encryption and decryption techniques for secure data transmission across network using snort and digital signatures |
| 7IT4-22.4 | Install appropriate tools for network protocol analyze security-based tools like Wireshark, tcpdump etc. |
| 7IT4-22.5 | Identify and describe a variety of ethical factors that may be relevant for understanding and assessing the cyber space. |
| 7IT4-22.6 | Improve team working skill for designing a solution for Key Exchange problem and general attacks on system like Diffie-Hellman Key Exchange, Brute Force Attack etc. |
| 7IT4-22.7 | Implement a small project for Server-Client technology using a File Transfer Protocol mechanism and through socket programming and makereport. |

## LO-PO-PSO MAPPING MATRIX OF COURSE

| LO/PO/PSO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7IT4-22.1 | - | 2 | - | - | - | - | - | - | - | - | - | - | 2 | - | 2 |
| 7IT4-22.2 | - | - | 2 | - | - | - | - | - | - | - | - | - | 2 | - | 2 |
| 7IT4-22.3 | - | - | - | 2 | - | - | - | - | - | - | - | - | 2 | - | - |
| 7IT4-22.4 | - | - | - | - | 2 | - | - | - | - | - | - | - | 2 | - | 2 |
| 7IT4-22.5 | - | - | - | - | - | - | - | 2 | - | - | - | - | 2 | 2 | - |
| 7IT4-22.6 | - | - | - | - | - | - | - | - | 2 | - | - | - | 2 | 2 | - |
| 7IT4-22.7 | - | - | - | - | - | - | 2 | - | - | 2 | 2 | 2 | - | 2 | 2 |

**PROGRAM OUTCOMES (POs)**

| | |
|---|---|
| **PO1** | **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems |
| **PO2** | **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineeringproblems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. |
| **PO3** | **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. |
| **PO4** | **Conduct investigations of complex problems:** Use research-based knowledge and research methodsincluding design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| **PO5** | **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with anunderstanding of the limitations. |
| **PO6** | **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professionalengineering practice. |
| **PO7** | **Environment and sustainability:** Understand the impact of the professional engineering solutions insocietal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. |
| **PO8** | **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. |
| **PO9** | **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. |
| **PO10** | **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports anddesign documentation, make effective presentations, and give and receive clear instructions. |

| | |
|---|---|
| **PO11** | **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, tomanage projects and in multidisciplinary environments. |
| **PO12** | **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. |

**PROGRAM SPECIFIC OUTCOMES (PSOs)**

| | |
|---|---|
| **PSO1** | The ability to understand and apply knowledge of mathematics, system analysis & design, Data Modelling, Cloud Technology, and latest tools to develop computer based solutions in the areas of system software, Multimedia, Web Applications, Big data analytics, IOT, Business Intelligence and Networking systems |
| **PSO2** | The ability to understand the evolutionary changes in computing, apply standards and ethical practices in project development using latest tools & Technologies to solve societal problems and meet the challenges of the future. |
| **PSO3** | The ability to employ modern computing tools and platforms to be an entrepreneur, lifelong learning and higher studies |

# LAB CONDUCTION PLAN

**Total number of experiment:** 08 **Total number of turns required:** 10 **Number of turns required for:**

| Experiment Number | Turns | Scheduled Week |
|---|---|---|
| Exp. 1 | 1 | Week 1,2 |
| Exp. 2 | 1 | Week 3,4 |
| Exp. 3 | 1 | Week 5 |
| Exp. 4 | 1 | Week 6 |
| Exp. 5 | 1 | Week 7 |
| Exp. 6 | 1 | Week 8 |
| Exp. 7 | 1 | Week 9 |
| Exp. 8 | 1 | Week 10 |

**DISTRIBUTIONOF LAB HOURS**

| S. No. | Activity | Distribution of Lab Hours |
|---|---|---|
| | | Time(120minute) |
| 1 | Attendance | 5 |
| 2 | Explanation of features of language | 15 |
| 3 | Explaining the Experiment | 15 |
| 4 | Performance of experiment | 70 |
| 5 | Viva/Quiz/Queries | 15 |

## GENERAL LAB INSTRUCTIONS

**DO'S**

- Enter the lab on time and leave at proper time.
- Wait for the previous class to leave before the next class enters.
- Keep the bag outside in the respective racks.
- Utilize lab hours in the corresponding.
- Turn off the machine before leaving the lab unless a member of lab staff has specificallytold you not to do so.
- Leave the labs at least as nice as you found them.
- If you notice a problem with a piece of equipment (e.g. a computer doesn't respond) or theroom in general (e.g. cooling, heating, lighting) please report it to lab staff immediately. Do not attempt to fix the problem yourself.

**DON'TS**

- Don't abuse the equipment.
- Do not adjust the heat or air conditioners. If you feel the temperature is not properly set, inform lab staff; we will attempt to maintain a balance that is healthy for people and machines.
- Do not attempt to reboot a computer. Report problems to lab staff.
- Do not remove or modify any software or file without permission.
- Do not remove printers and machines from the network without being explicitly told to do so by lab staff.
- Don't monopolize equipment. If you're going to be away from your machine for more than 10 or 15 minutes, log out before leaving. This is both for the security of your account, and to ensure that others are able to use the lab resources while you are not.
- Don't use internet, internet chat of any kind in your regular lab schedule.
- Do not download or upload of MP3, JPG or MPEG files.
- No games are allowed in the lab sessions.
- No hardware including USB drives can be connected or disconnected in the labs without prior permission of the lab in-charge.
- No food or drink is allowed in the lab or near any of the equipment. Aside from the factthat it leaves a mess and attracts pests, spilling anything on a keyboard or other piece of computer equipment could cause permanent, irreparable, and costly damage. (and in fact *has*) If you need to eat or drink, take a break and do so in the canteen.
- Don't bring any external material in the lab, except your lab record, copy and books.
- Don't bring the mobile phones in the lab. If necessary then keep them in silence mode.
- Please be considerate of those around you, especially in terms of noise level. While labsare a natural place for conversations of all types, kindly keep the volume turned down.
- If you are having problems or questions, please go to either the faculty, lab in-charge or the lab supporting staff. They will help you. We need your full support and cooperation for smooth functioning of the lab.

## LAB SPECIFIC SAFETY RULES

### Before entering in the lab

- All the students are supposed to prepare the theory regarding the next experiment/Program.
- Students are supposed to bring their lab records as per their lab schedule.
- Previous experiment/program should be written in the lab record.
- If applicable trace paper/graph paper must be pasted in lab record with proper labeling.
- All the students must follow the instructions, failing which he/she may not be allowed inthe lab.

### While working in the lab

- Adhere to experimental schedule as instructed by the lab in-charge/faculty.
- Get the previously performed experiment/ program signed by the faculty/ lab in charge.
- Get the output of current experiment/program checked by the faculty/ lab in charge in thelab copy.
- Each student should work on his/her assigned computer at each turn of the lab.
- Take responsibility of valuable accessories.

# Zero Lecture

## Topic: Cyber Security

- Cyber Security is the technique of protecting your systems, digital devices, networks, and all of the data stored in the devices from cyber-attacks. By acquiring knowledge of cyber-attacks and cyber security we can secure and defend ourselves from various cyber-attacks like phishing and DDoS attacks. It uses tools like firewalls and antivirus software to protect your devices from hackers and malware.
- Encryption is the technique that helps to keep your personal information private, you can only read it. Cyber security also teaches you how to spot tricks like phishing, where bad guys try to steal your info by pretending to be someone you trust. In short, cyber security keeps your online world safe and secure.

## Different Types of Cyber Security

1. **Network Security**

   Focuses on securing computer networks from unauthorized access, data breaches, and other network-based threats. It involves technologies such as Firewalls, Intrusion detection systems (IDS), Virtual private networks (VPNs), and Network segmentation.

2. **Application Security**

   Concerned with securing software applications and preventing vulnerabilities that could be exploited by attackers. It involves secure coding practices, regular software updates and patches, and application-level firewalls.

3. **Information or Data Security**

   Focuses on protecting sensitive information from unauthorized access, disclosure, alteration, or destruction. It includes Encryption, Access controls, Data classification, and Data loss prevention (DLP) measures.

4. **Cloud Security**

   It involves securing data, applications, and infrastructure hosted on cloud platforms, and ensuring appropriate access controls, data protection, and compliance. It uses various cloud service providers such as AWS, Azure, Google Cloud, etc., to ensure security against multiple threats.

5. **Mobile Security**

   It involves securing the organizational and personal data stored on mobile devices such as cell phones, tablets, and other similar devices against various malicious threats. These threats are unauthorized access, Device loss or Theft, Malware, etc.

## 6. Endpoint Security

Refers to securing individual devices such as computers, laptops, smartphones, and IoT devices. It includes antivirus software, intrusion prevention systems (IPS), device encryption, and regular software updates.

# Experiment–1

**OBJECTIVE**

**Implement the following Substitution & Transposition Techniques concepts:**

**a) Caesar Cipher b) Rail fence row & Column Transformation**

**Program:**

**a) Caesar Cipher:**

```
def encrypt_fun(text,shift):
    result = ""

    for i in range(len(text)):
        character = text[i]

        # Encrypt uppercase characters
        if (character.isupper()):
            result += chr((ord(character) + shift-65) % 26 + 65)

        # Encrypt lowercase characters
        else:
            result += chr((ord(character) + shift - 97) % 26 + 97)

    return result

text = "CaesarCipher"
shift = 3
print ("Text  : " + text)
print ("Shift : " + str(shift))
print ("Cipher: " + encrypt_fun(text,shift))
```

**OUTPUT:**

```
Text  : CaesarCipher
Shift : 3
Cipher: FdhvduFlskhu
```

## b) Rail fence row & Column Transformation :

```
def encryptRailFence(text, key):

    rail = [['\n' for i in range(len(text))]
              for j in range(key)]

    dir_down = False
    row, col = 0, 0

    for i in range(len(text)):

        if (row == 0) or (row == key - 1):
            dir_down = not dir_down

        rail[row][col] = text[i]
        col += 1

        if dir_down:
            row += 1
        else:
            row -= 1

    result = []
    for i in range(key):
        for j in range(len(text)):
            if rail[i][j] != '\n':
                result.append(rail[i][j])
    return("" . join(result))

def decryptRailFence(cipher, key):

    rail = [['\n' for i in range(len(cipher))]
              for j in range(key)]

    dir_down = None
```

```
row, col = 0, 0


for i in range(len(cipher)):
    if row == 0:
        dir_down = True
    if row == key - 1:
        dir_down = False


    rail[row][col] = '*'
    col += 1


    if dir_down:
        row += 1
    else:
        row -= 1


index = 0
for i in range(key):
    for j in range(len(cipher)):
        if ((rail[i][j] == '*') and
        (index < len(cipher))):
            rail[i][j] = cipher[index]
            index += 1


result = []
row, col = 0, 0
for i in range(len(cipher)):


    if row == 0:
        dir_down = True
    if row == key-1:
        dir_down = False


    if (rail[row][col] != '*'):
        result.append(rail[row][col])
        col += 1


    if dir_down:
        row += 1
```

```
        else:
            row -= 1
    return("".join(result))


if __name__ == "__main__":
    print(encryptRailFence("encryptRailFence", 2))
    print(encryptRailFence("encryptRailFence", 3))


    print(decryptRailFence("ecytalecnrpRiFne", 2))
    print(decryptRailFence("eyaenrpRiFnectlc", 3))
```

**OUTPUT:**

```
ecytalecnrpRiFne
eyaenrpRiFnectlc
encryptRailFence
encryptRailFence
```

# Experiment–2

**OBJECTIVE**

**Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob).**

**Program:**

```
<html>

<body>
<h1> Diffie-Hellman Key Exchange mechanism </h1>
</body>
<script>
function power(a, b, p)
{
   if (b == 1)
      return a;
   else
      return((Math.pow(a, b)) % p);
}

var P, G, x, a, y, b, ka, kb;

P = 23;
document.write("The value of P:" + P + "<br>");

G = 9;
document.write("The value of G:" + G + "<br>");

a = 4;
document.write("The private key a for Alice:" +
            a + "<br>");

x = power(G, a, P);

b = 3;
document.write("The private key b for Bob:" +
            b + "<br>");

y = power(G, b, P);

ka = power(y, a, P); // Secret key for Alice
```

```
kb = power(x, b, P); // Secret key for Bob

document.write("Secret key for the Alice is:" +
        ka + "<br>");
document.write("Secret key for the Bob is:" +
        kb + "<br>");


</script>
  </html>
```

## Output:

**Diffie-Hellman Key Exchange mechanism**
The value of P:23
The value of G:9
The private key a for Alice:4
The private key b for Bob:3
Secret key for the Alice is:9
Secret key for the Bob is:9

# Experiment–3

**OBJECTIVE**

**Implement the following Attack: a) Dictionary Attack b) Brute Force Attack**

**Program:**

## a) Dictionary Attack

```
import hashlib
# List of commonly used passwords and their variations
common_passwords = ["password", "password123", "letmein", "qwerty", "123456", "abc123", "admin", "welcome", "monkey", "sunshine"]
password_variations = ["", "123", "1234", "12345", "123456", "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", "-", "_", "+", "=", "/", "\\", "|", "[", "]", "{", "}", "<", ">"]
# Hash of the password to be attacked
hashed_password = hashlib.sha256(b"mypass12#@").hexdigest()
print("Hashed Password is:")
print(hashed_password)
# Try out all possible combinations of common passwords and their variations
for password in common_passwords:
    for variation in password_variations:
        possible_password = password + variation
        hashed_possible_password = hashlib.sha256(possible_password.encode()).hexdigest()
       #print(hashed_possible_password)
        if hashed_possible_password == hashed_password:
            print(f"Password found: {possible_password}")
            break
    else:
        continue
    break
else:
    print("Password not found")
```

**OUTPUT:**

```
Hashed Password is:
7914d20d1a95f83e938ebc0d0a731b86f1352e49d5aa1d27f1cf1011d32a4528

Password not found
```

## b) Brute Force Attack

```
import itertools
import string

def bruteforce_attack(password):
    chars = string.printable.strip()
    attempts = 0
    for length in range(1, len(password) + 1):
        for guess in itertools.product(chars, repeat=length):
            attempts += 1
            guess = ''.join(guess)
            if guess == password:
                return (attempts, guess)
    return (attempts, None)

password = input("Input the password to crack: ")
attempts, guess = bruteforce_attack(password)
if guess:
    print(f"Password cracked in {attempts} attempts. The password is {guess}.")
else:
    print(f"Password not cracked after {attempts} attempts.")
```

**OUTPUT:**

```
Input the password to crack: abcd
Password cracked in 9243692 attempts. The password is abcd.
```

# Experiment–4

**OBJECTIVE**

**Installation of Wire shark, tcpdump, etc and observe data transferred in client server communication using UDP/TCP and identify the UDP/TCP datagram.**

**Program:**

1. **Step Wireshark :**

**a. Download and Install Wireshark:**

- Go to the Wireshark website to download the version that is compatible with your operating system.
- Adhere to the website's installation instructions.

**b. Begin Packet Capturing:**

- Open Wireshark and choose the network interface from which to begin collecting packets.
- The display filter in Wireshark. The display filter in Wireshark's default configuration is a bar that sits right above the column display. Here is where we enter expressions to narrow down what we can see in a pcap file, be it Ethernet frames, IP packets, or TCP segments.
- There are several local interfaces available; please choose one.
- Press the Start button.
- In essence, you are recording and intercepting data packets as they pass through a network interface when you capture packets.

**c. Analyze Packets:**

- Wireshark will show packets as they come through the chosen interface in real time.To limit the packets that are shown based on parameters such as source, destination, protocol, etc., you can apply filters.

**2. Step tcpdump :**

a. **Launch a Terminal or Command Prompt:**
On Unix-based systems, open a terminal window. As an administrator, run the Command Prompt on Windows.

b. **Begin Packet Capturing:**
In the first case, run dumpcap -i <interface>-w<output_file>, where <interface> is the network interface that you choose to start capturing from.

c. **View Captured Packets:**
tcpdump will present captured packets in a readable format on the terminal window.

# Experiment–5

## OBJECTIVE

**Installation of rootkits and study about the variety of options.**

**Program:**

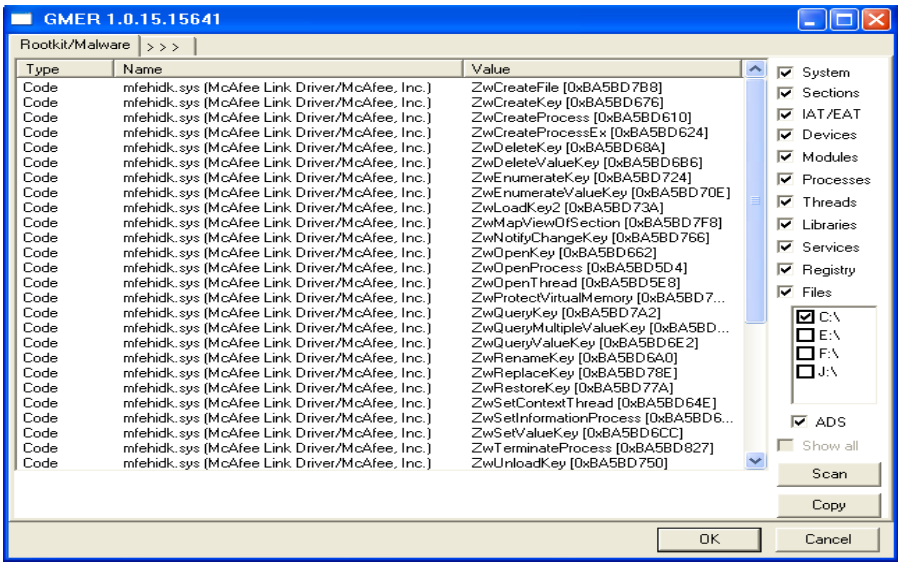
A **rootkit** is a stealthy type of malicious software (malware) designed to hide the existence of certain processes or programs from normal methods of detection and enables continued privileged access to a computer. The term *rootkit* is a concatenation of "root" (the traditional name of the privileged account on Unix operating systems) and the word "kit" (which refers to the software components that implement the tool). The term "rootkit" has negative connotations through its association with malware.

A rootkit is a collection of tools (programs) that enable administrator-level access to a computer or computer network. Typically, a cracker installs a rootkit on a computer after first obtaining user-level access, either by exploiting a known vulnerability or cracking a password. Once the rootkit is installed, it allows the attacker to mask intrusion and gain root or privileged access to the computer and, possibly, other machines on the network.

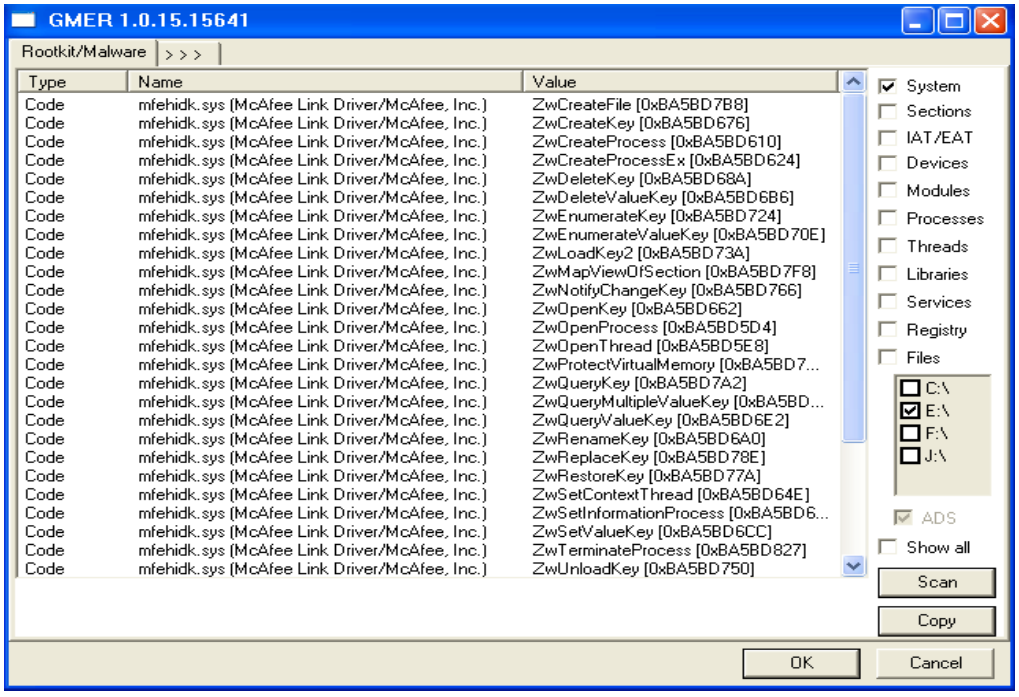
A rootkit may consist of spyware and other programs that: monitor traffic and keystrokes; create a "backdoor" into the system for the hacker's use; alter log files; attack other machines on the network; and alter existing system tools to escape detection.

**Steps:**

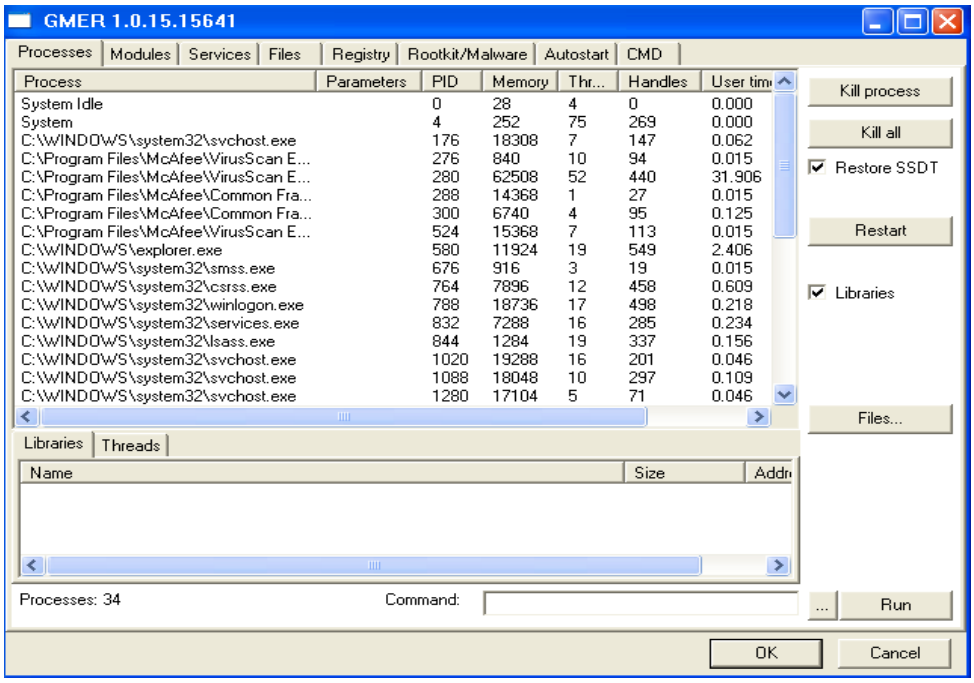
1. Double click on rootkit folder.
2. Double click on the GMER rootkit application.
3. Now the rootkit screen will be displayed.



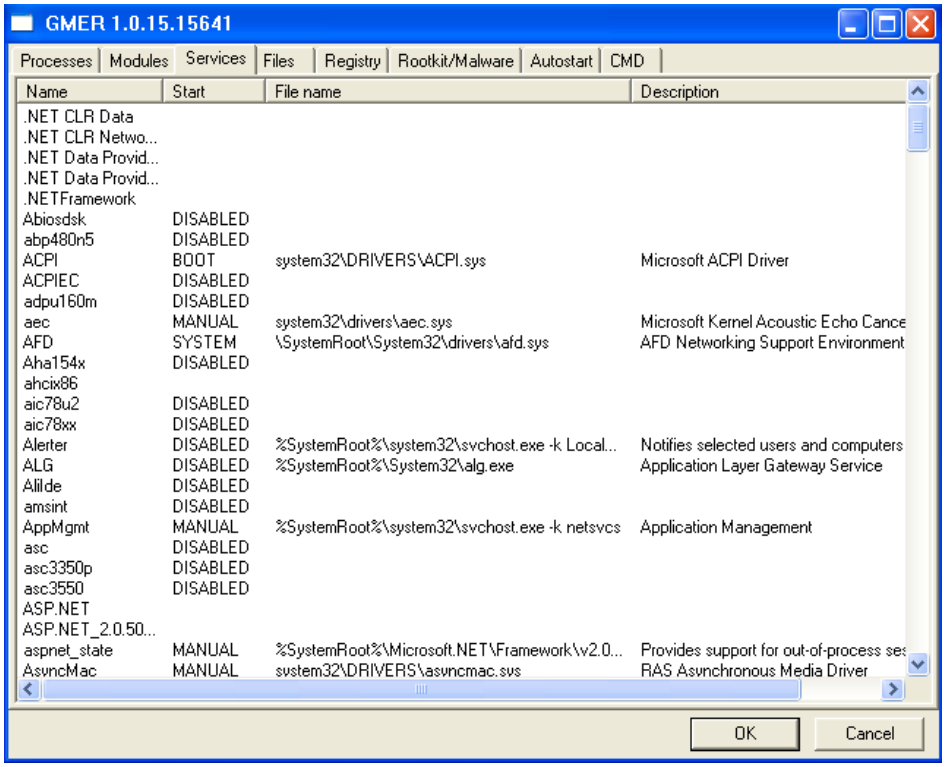
4. Select anyone of the drive which is shown at right side of the screen.
5. After selecting the drive click on scan button




6. Click on the option processes the screen will be displayed



7. Click on the option services.




8. Now click on different options to perform different actions.

# Experiment-6

**OBJECTIVE**

**Perform an Experiment to Sniff Traffic using ARP Poisoning.**

**Program:**

**RP poisoning attack steps:**

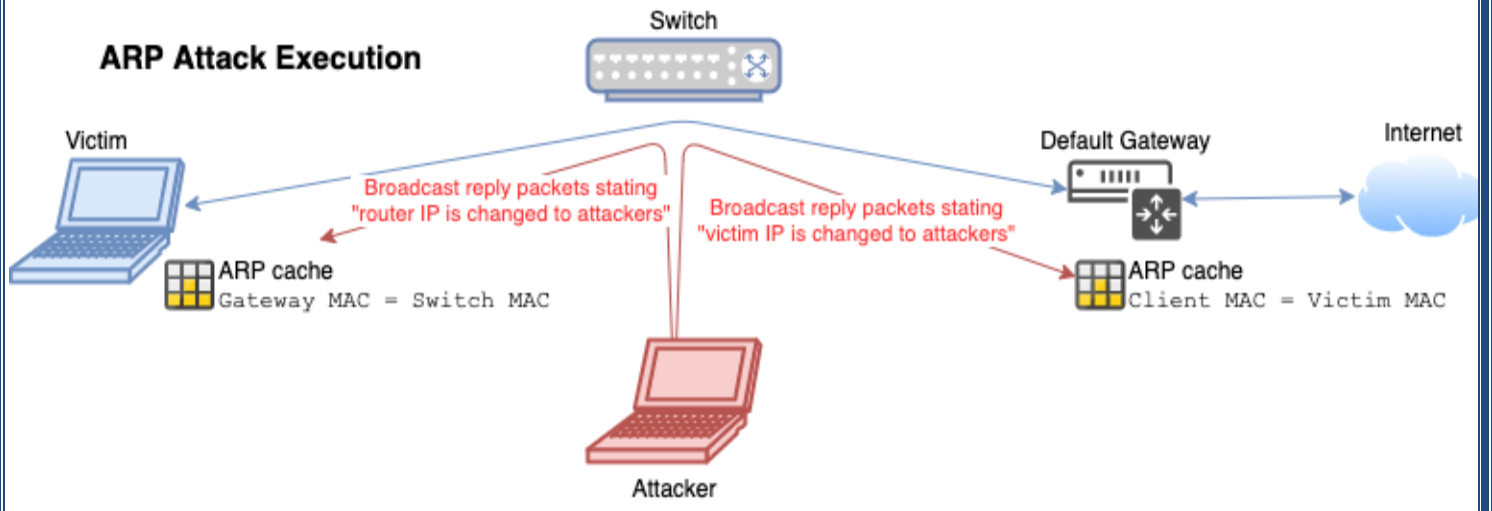
1. **Gather information**

i. Get victim IP address, e.g. 192.168.122.183
- E.g. through host discovery using nmap e.g. nmap -sn 192.168.0.0/24

ii. Get default gateway IP, e.g. 192.168.122.1
- Usually IP of the machine ending with .1
- Usually same for everyone on same network
- Default gateway is the forwarding host (router) to internet when no other specification matches the destination IP address of a packet.

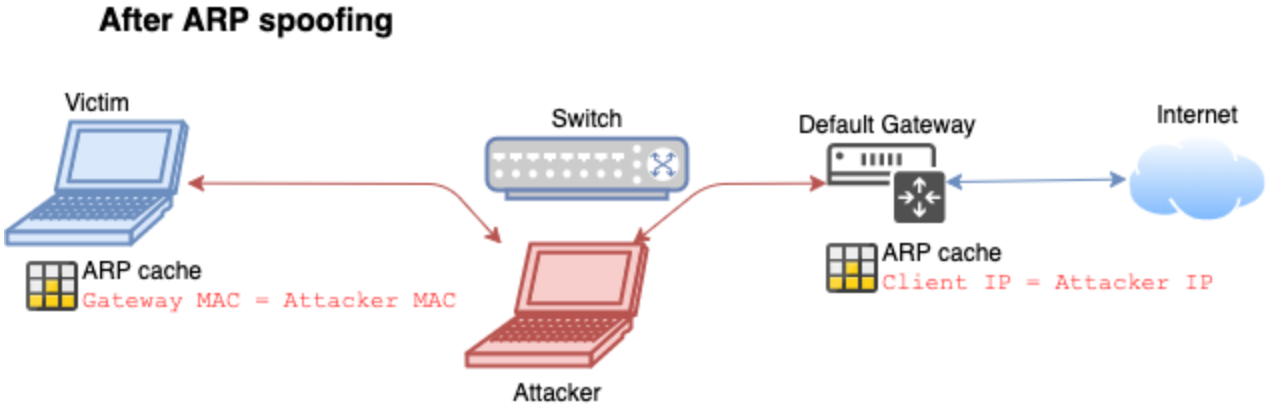
2. **Enable forwarding mode to sniff the traffic**
- echo 1 > /proc/sys/net/ipv4/ip_forward in Linux.
- **!**Otherwise no traffic is going through and you're just DOSing.
-

3. **Attack**




- Deceive the victim device through flooding ARP reply packets to it.
- Change gateways MAC address is to the attackers
- Use an ARP spoofing tool e.g.
- arpspoof
- arpspoof -t <victim-machine-ip> <default-gateway-ip>

- arpspoof -t <default-gateway-ip> <victim-machine-ip>
- ettercap
- Also sniffs passwords automatically
- ettercap -NaC <default-gateway-ip> <victim-machine-ip>
- N: make it non-interactive
- a: arp posion
- c: parse out passwords and usernames.
- Cain and Abel (Cain & Abel) on Windows




4. **Sniff**
   - Now you sniff the traffic between two devices.
   - If through HTTPS & SSL you can only see basic data such as User Agent and domain names.
   - Can use e.g. wireshark or dsniff

# Experiment-7

**OBJECTIVE**

**Demonstrate intrusion detection system using any tool (snort or any other s/w).**

**Program:**

**A. Configure and Use Snort IDS on Windows**

Steps to configure Snort on Widnows machine and how to use it for detection of attacks.

**B. Steps:**

1. Download Snort from "http://www.snort.org/" website.

2. Also download Rules from the same website. You need to sign up to get rules for registered users.

3. Click on the Snort_(version-number)_Installer.exe file to install it. By-default it will install snort in the "C:\Snort" directory.

4. Extract downloaded Rules file: snortrules-snapshot-(number).tar.gz

5. Copy all files from the "rules" directory of the extracted folder and paste them into "C:\Snort\rules" directory.

6. Copy "snort.conf" file from the "etc" directory of the extracted folder and paste it into "C:\Snort\etc" directory. Overwrite existing file if there is any.

7. Open command prompt (cmd.exe) and navigate to directory "C:\Snort\bin" directory.

8. To execute snort in sniffer mode use following command:

snort -dev -i 2

-i indicate interface number.

-dev is used to run snort to capture packets.

To check interface list use following command: snort -W

9. To execute snort in IDS mode, we need to configure a file "snort.conf" according to our network environment.

10. Set up network address we want to protect in snort.conf file. To do that look for "HOME_NET" and add your IP address.

var HOME_NET 10.1.1.17/8

11. You can also set addresses or DNS_SERVERS, if you have any. otherwise go to the next step.

12. Change RULE_PATH variable with the path of rules directory. var RULE_PATH c:\snort\rules

13. Change the path of all libraries with the name and path on your system. or change path of snort_dynamicpreprocessor variable.

sor file C:\Snort\lib\snort_dynamiccpreprocessor\sf_dcerpc.dll

You need to do this to all library files in the "C:\Snort\lib" directory. The old path might be something like: "/usr/local/lib/...". you need to replace that path with you system path.

14. Change path of the "dynamicengine" variable value in the "snort.conf" file with the path of your system. Such as:

dynamicengine C:\Snort\lib\snort_dynamicengine\sf_engine.dll

15 Add complete path for "include classification.config" and "include reference.config" files. include c:\snort\etc\classification.config

include c:\snort\etc\reference.config

16. Remove the comment on the line to allow **ICMP** rules, if it is alredy commented. include $RULE_PATH/icmp.rules

17. Similary, remove the comment of ICMP-info rules comment, if it is already commented. include $RULE_PATH/icmp-info.rules

18 To add log file to store alerts generated by snort, search for "output log" test and add following line:

output alert_fast: snort-alerts.ids

19. Comment whitelist $WHITE_LIST_PATH/white_list.rules and blacklist $BLACK_LIST_PATH/black_list.rules lines. Also ensure that you add change the line above $WHITE_LIST_PATH

Change nested_ip inner , \ to nested_ip inner #, \

20. Comment following lines:

#preprocessor normalize_ip4

#preprocessor normalize_tcp: ips ecn stream

#preprocessor normalize_icmp4

#preprocessor normalize_ip6

#preprocessor normalize_icmp6

21. Save the "snort.conf" file and close it.

22. Go to the "C:\Snort\log" directory and create a file: snort-alerts.ids

23. To start snort in IDS mode, run following command:

snort -c c:\snort\etc\snort.conf -l c:\snort\log -i 2

Above command will generate log file that will not be readable without using a tool. To read it use following command:

C:\Snort\Bin\> snort -r ..\log\log-filename

To generate Log files in ASCII mode use following command while running snort in IDS mode:

snort -A console -i2 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii

**24. Scan the computer running snort from another computer using PING or launch attack. Then check snort-alerts.ids file the log**

# Experiment-8

**OBJECTIVE**

**Demonstrate how to provide secure data storage, secure data transmission and for creatingdigital signatures.**

**Program:**

```
  import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.Signature;
import java.util.Scanner;

public class CreatingDigitalSignature {
   public static void main(String args[]) throws Exception {
      //Accepting text from user
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter some text");
      String msg = sc.nextLine();

      //Creating KeyPair generator object
      KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("DSA");

      //Initializing the key pair generator
      keyPairGen.initialize(2048);

      //Generate the pair of keys
      KeyPair pair = keyPairGen.generateKeyPair();

      //Getting the private key from the key pair
```

```
    PrivateKey privKey = pair.getPrivate();

    //Creating a Signature object
    Signature sign = Signature.getInstance("SHA256withDSA");

    //Initialize the signature
    sign.initSign(privKey);
    byte[] bytes = "msg".getBytes();

    //Adding data to the signature
    sign.update(bytes);

    //Calculating the signature
    byte[] signature = sign.sign();

    //Printing the signature
    System.out.println("Digital signature for given text: "+new String(signature, "UTF8"));
  }
}
```

**OUTPUT:**

```
Enter some text
Hi how are you
Digital signature for given text: 0=@gRD???-?.???? /yGL?i??a!?
```

# Beyond the Syllabus Experiment-1

**OBJECTIVE**

**To develop a program to implement Advanced Encryption Standard for encryption and decryption**

**Program:**

**AES Encryption and Decryption:**

```
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;public class AES {

private static SecretKeySpec secretKey; private static byte[] key;

public static void setKey(String myKey)
{
MessageDigest sha = null; try { key = myKey.getBytes("UTF-8");
sha = MessageDigest.getInstance("SHA-1"); key = sha.digest(key);key = Arrays.copyOf(key, 16);
secretKey = new SecretKeySpec(key, "AES");
}
catch (NoSuchAlgorithmException e) { e.printStackTrace();
}
catch (UnsupportedEncodingException e) { e.printStackTrace();
}
}

public static String encrypt(String strToEncrypt, String secret)
{
try
{
setKey(secret);
Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
cipher.init(Cipher.ENCRYPT_MODE, secretKey);
return
Base64.getEncoder().encodeToString(cipher.doFinal(strToEncrypt.getBytes("UTF-  8")));
```

```
}
catch (Exception e)
{
System.out.println("Error while encrypting: " + e.toString());
}
return null;
}

public static String decrypt(String strToDecrypt, String secret)
{
try
{
setKey(secret);
Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
cipher.init(Cipher.DECRYPT_MODE, secretKey);
return new String(cipher.doFinal(Base64.getDecoder().decode(strToDecrypt)));
}
catch (Exception e)
{
System.out.println("Error while decrypting: " + e.toString());
}
return null;
}
}
```

**Encryption and decryption example:**

```
public static void main(String[] args)
{
final String secretKey = "ssshhhhhhhhhhh!!!!";

String originalString = "howtodoinjava.com";
String encryptedString = AES.encrypt(originalString, secretKey) ; String decryptedString =
AES.decrypt(encryptedString, secretKey) ;

System.out.println(originalString); System.out.println(encryptedString);
System.out.println(decryptedString);
}
```