E-Content (Network Security Practical)

(Comp Engg VI-Sem 2025-2026)

SL. No.	Experiment
1	Installation and comparison of various anti virus software
2	Installation and study of various parameters of firewall
3	Writing program in C to Encrypt/ Decrypt using XOR key
4	Study of VPN
5	Study of various hacking tools
6	Practical applications of digital signature

Prepare by:

Munish Kumar
Sr. Lectuter in CE
GD Govt Polytechnic Hisar

1: Installation and comparison of various antivirus Software

Introduction

Antivirus software is a security tool designed to detect, prevent, and remove malicious programs such as viruses, worms, Trojans, spyware, ransomware, and other threats. With increasing cyber-attacks, installing reliable antivirus software is essential to protect data and ensure system security.

Objectives of the Practical

- To understand the installation procedure of antivirus software.
- To learn how to update antivirus definitions.
- To perform a basic system scan.
- To compare features, performance, and usability of different antivirus applications.
- To evaluate advantages and limitations of each antivirus.

Requirements

- A computer system with internet connectivity
- Access to at least **two antivirus software**
- Administrative rights for installation
- Practical observation sheet

Theory:

What is Antivirus Software?

Antivirus software monitors system activities, scans files, and removes malicious code by using:

- Signature-based detection
- Heuristic analysis
- Real-time protection
- Cloud-based scanning

Types of Scans

- Quick Scan – Checks common infection areas
- Full System Scan – Scans all files and drives
- Custom Scan – User-selected folders or drives
- Boot-time Scan – Detects threats before the OS loads

Step-by-Step Procedure

Installation Procedure

1. Download the antivirus setup file from the official website.
2. Run the installer (.exe file).
3. Read and accept the license agreement.
4. Choose installation type:
   (i) Typical (recommended) (ii) Custom (advanced settings)
5. Allow antivirus to install required components.
6. Restart the system if prompted.

Updating Antivirus Definitions

1. Open antivirus dashboard/console.
2. Navigate to Updates or Virus Definitions section.
3. Click Check for updates.
4. Ensure updates are successfully installed.

Performing System Scans

1. Open antivirus interface.
2. Select Scan -> Quick / Full / Custom.
3. Allow scanning to complete.
4. View and record detected threats (if any).
5. Take recommended actions:
    - Quarantine
    - Remove
    - Repair

Comparison of Various Antivirus Software

Below is a generic comparison format (you can fill with specific software used in your practical):

Feature	Antivirus A	Antivirus B	Antivirus C
Installation size			
User interface			
Update speed			
Types of scans available			
Real-time protection			
System performance impact			
Additional features (Firewall, VPN, Email protection etc.)			
Pricing (Free/Paid)			
Overall rating			

Observations

- Note ease of installation.
- Record scan times.
- Record number of threats detected (if any).
- Note resource usage (CPU/RAM during scan).
- Record additional tools (password manager, parental control, etc.).

Results

Based on comparison and observations, identify:

- Which antivirus is more user-friendly?
- Which gives better protection features?
- Which consumes fewer system resources?
- Which offers better value (free vs. paid)?

Conclusion

In this practical, you learned how to install antivirus software, update virus definitions, and perform system scans. By comparing different antivirus applications, you understood how features, performance, and protection levels vary across software. This helps in selecting the most suitable antivirus for personal or organizational use.

2: Installation and Study of Various Parameters of Firewall

Introduction

A firewall is a security system that monitors and controls incoming and outgoing network traffic based on predefined security rules. It acts as a barrier between trusted internal networks and untrusted external networks such as the internet. Firewalls can be implemented as hardware**,** software**,** or both**.**

Objectives

- To understand the concept and purpose of firewalls.
- To install and configure a firewall application.
- To study important firewall parameters such as rules, policies, logging, NAT, and filtering.
- To analyze how firewall rules affect network traffic and system security.

Requirements

- A computer system with internet access
- Any one firewall software/hardware such as:
  - Windows Defender Firewall
  - pfSense
  - FortiGate
  - Sophos Firewall
- Administrative access to install/configure firewall
- Practical observation sheet

Theory:

What is a Firewall?

A firewall is a network security device/software that filters traffic using a set of rules. It protects systems from unauthorized access, malware attacks, and data breaches.

Types of Firewalls

1. Packet-Filtering Firewall – Filters based on IP, port, protocol.
2. Stateful Inspection Firewall – Monitors state of active connections.
3. Proxy Firewall – Acts as an intermediary between user and internet.
4. Next-Generation Firewall (NGFW) – Includes deep packet inspection, IDS/IPS, application control.

Key Firewall Concepts

- Rules/Policies – Define what traffic is allowed or blocked.
- Ports & Protocols – TCP, UDP, ICMP control.
- NAT (Network Address Translation) – Maps private to public IP.
- Inbound/Outbound Filtering – Controls incoming & outgoing packets.
- Logging and Monitoring – Tracks firewall events and traffic.
- Default Deny Approach **–** Blocks everything except allowed rules.

Procedure for Installation and Configuration

Installation Steps (Example for software firewall)

1. Download firewall installation file from official website.
2. Run the installer (.exe/.msi).
3. Accept license agreement and choose installation location.
4. Select components to install (Core firewall, logs, monitoring tools).
5. Complete installation and restart system if required.
6. Open the firewall dashboard.

Basic Configuration

1. Open the Firewall Control Panel.
2. Navigate to Rules/Policies section.
3. Create inbound rule (example: allow HTTP port 80).
4. Create outbound rule (example: block unused ports).
5. Enable NAT if firewall is used as gateway.
6. Turn on logging to record blocked/allowed traffic.
7. Enable real-time protection or stateful inspection (if supported).
8. Test rules by accessing websites or using ping/port-scan utilities.

Important Firewall Parameters to Study

Parameter	**Description**
Rule Set	Defines allowed and blocked traffic based on source, destination, port, protocol.
Protocol Filtering	TCP, UDP, ICMP traffic control.
Stateful Inspection	Tracks connection states (Established, SYN, ACK, etc.).
Port Management	Opening and closing ports as per requirement.
NAT (Network Address Translation)	Converts private IPs to public IPs.
Application Control	Allows/blocks specific apps (NGFW only).
Logging & Alerts	Records traffic and security events.
IP/Domain Blocking	Prevents access to malicious or unwanted domains.
Bandwidth Control/QoS	Allocates network priority to applications.
VPN Support	Many firewalls support secure remote connectivity.

Observations

- Ease of installation and interface friendliness.
- Default rules applied after installation.

- Behavior of system after enabling/disabling rules.
- Log entries created for blocked or suspicious traffic.
- CPU and memory usage of firewall service.
- NAT behavior when connecting multiple devices.

Result

After installation and parameter study, the student should be able to:

- Install and configure a firewall.
- Create and modify traffic rules.
- Understand how firewall parameters impact network behavior.
- Analyze logs and identify blocked activities.

Conclusion

In this practical, we installed and studied various firewall parameters. Firewalls play a critical role in network security by controlling traffic, preventing unauthorized access, and monitoring real-time communication. Understanding firewall configurations helps in designing secure networks and managing cyber threats effectively.

3: Writing program in C to Encrypt/ Decrypt using XOR key

Introduction

Data security is an important aspect of communication and storage. One of the simplest and fastest encryption techniques is XOR (Exclusive OR) encryption. It uses a key to transform the original data (plaintext) into unreadable text (ciphertext). The same key is used again to decrypt the ciphertext back to plaintext.

Objectives

- To understand XOR-based encryption and decryption.
- To write a C program that uses XOR operation for securing data.
- To demonstrate that the same XOR key can both encrypt and decrypt.
- To analyze how XOR logic forms the basis of many lightweight cryptographic algorithms.

Requirements

- C compiler (GCC / Turbo C / CodeBlocks / Online compiler).
- Basic knowledge of C programming.
- Text input for encryption.

Theory:

XOR Operation

XOR stands for Exclusive OR**.**
Truth Table:

A	**B**	**A XOR B**
0	0	0
0	1	1
1	0	1
1	1	0

Why XOR for Encryption?

XOR has a special reversible property:

cipher = plaintext XOR key
plaintext = cipher XOR key

This means the same key can encrypt and decrypt data, making it simple and efficient.

Advantages

- Fast and lightweight
- Symmetric (same key for both operations)
- Easy to implement

Algorithm

Encryption

1. Take input string (plaintext) from user.
2. Take key value (a character or integer).
3. For each character in plaintext:
    - encrypted[i] = plaintext[i] XOR key
4. Display encrypted text.

Decryption

1. For each character in encrypted text:
    - decrypted[i] = encrypted[i] XOR key
2. Display decrypted text (should match original plaintext).

C Program for XOR Encryption/Decryption

```
#include <stdio.h>
#include <string.h>
int main() {
   char text[100], encrypted[100], decrypted[100];
   int key, i;

   printf("Enter text: ");
   gets(text);

   printf("Enter XOR key (integer): ");
   scanf("%d", &key);

   // Encryption
   for(i = 0; i < strlen(text); i++) {
      encrypted[i] = text[i] ^ key;
   }
   encrypted[i] = '\0';

   // Decryption
   for(i = 0; i < strlen(encrypted); i++) {
      decrypted[i] = encrypted[i] ^ key;
   }
   decrypted[i] = '\0';

   printf("\nEncrypted Text: %s", encrypted);
   printf("\nDecrypted Text: %s", decrypted);
   return 0;
}
```

Sample Output

```
Enter text: HELLO
Enter XOR key: 5

Encrypted Text: M@IIG
Decrypted Text: HELLO
```

Observations

- Encryption and decryption times are very fast.
- Even small key changes produce different ciphertext.
- XOR operation is reversible and symmetric.
- Output may contain unreadable ASCII characters (normal for XOR encryption).

Conclusion

In this practical, we successfully implemented XOR-based encryption and decryption using C. The reversible nature of XOR makes it useful in simple cryptography, data masking, and checksum operations. Although basic, this technique forms the foundation of more advanced encryption algorithms.

4: Study of VPN

Introduction

A **Virtual Private Network (VPN)** is a technology that creates a secure, encrypted connection over a public network such as the internet. It ensures privacy, confidentiality, and secure communication by routing data through a protected tunnel. VPNs are widely used for remote access, secure browsing, and bypassing geo-restrictions.

Objectives

- To understand the concept and working of VPN.
- To study different types of VPNs (Remote Access, Site-to-Site, SSL/IPSec).
- To configure and analyze VPN behavior using VPN software.
- To examine encryption, tunneling, authentication, and protocol parameters.

Requirements

- A computer system with internet connectivity
- Any one VPN application such as:
    - OpenVPN
    - NordVPN
    - Cisco AnyConnect
- Administrative privileges for configuration
- Observation sheet for practical notes

Theory:

What is a VPN?

A VPN creates a secure communication tunnel between a user and a remote server. All transmitted data is encrypted, ensuring secure access even over unsafe networks like public Wi-Fi.

Why VPN is Important?

- Protects data privacy
- Secures communication
- Masks user's IP address
- Allows safe remote access to corporate networks
- Prevents cyber-attacks like sniffing, man-in-the-middle, etc.

Types of VPN

1. Remote Access VPN – For individual users accessing remote servers.
2. Site-to-Site VPN – Connects two networks (e.g., branch office to HQ).
3. Client-Based VPN – Requires installation of a client application.
4. Clientless VPN – Access via web browser (SSL VPN).

VPN Technologies

- Tunneling Protocols**:** PPTP, L2TP, IPSec, OpenVPN
- Encryption Methods**:** AES-128, AES-256, 3DES
- Authentication**:** Password, certificates, multi-factor authentication

---

Procedure for VPN Study:

Installation of VPN Client

1. Download VPN client from official website.

2. Run installer and follow setup instructions.
3. Login or import VPN configuration file (if provided by organization).
4. Choose server location or gateway.

Establishing VPN Connection

1. Launch VPN application.
2. Select server (e.g., USA, India, Singapore).
3. Click Connect**.**
4. Wait for tunnel establishment confirmation.

Verifying VPN Functioning

1. Check new IP address using "What is my IP" service.
2. Observe encryption status in VPN client dashboard.
3. Test access to secured corporate resources.
4. Try accessing geo-restricted sites to test redirection.

Disconnecting VPN

1. Open VPN dashboard.
2. Click Disconnect**.**
3. Verify that your original IP address is restored.

Observations

- Speed difference before and after VPN connection
- New IP address reflected after VPN connection
- Encryption protocol used
- Change in latency and download/upload speed
- Server switching time and stability
- Changes in accessibility of restricted websites

Result

After performing the practical, the student is able to:

- Understand the working of VPN technology
- Install, configure, and use VPN client software
- Identify encryption, tunneling protocols, and IP masking
- Analyze the effect of VPN on network performance

Conclusion

In this practical, we studied the working and configuration of a VPN. Virtual Private Networks provide secure communication channels by encrypting data and masking user identity. VPNs are essential for secure remote work, protecting online privacy, and preventing cyber threats. Understanding VPN parameters helps strengthen network security and improve safe communication practices.

5: Study of various hacking tools

Introduction

Hacking tools are software programs used for penetration testing, vulnerability assessment, network analysis, and ethical hacking. These tools help cybersecurity professionals discover security weaknesses before malicious attackers exploit them. Studying these tools provides practical knowledge of cybersecurity defense, attack models, and system protection.

Objectives

- To understand the purpose and role of hacking/penetration testing tools.
- To study different categories of hacking tools such as scanning, enumeration, exploitation, sniffing, and password cracking.
- To gain hands-on knowledge of commonly used cybersecurity tools.
- To analyze how these tools are used ethically for security testing.

Requirements

- Computer system (Windows/Linux)
- Internet connectivity
- Cyber security toolkit/live OS such as:
    - Kali Linux
- Common hacking/penetration testing tools such as:
    - Nmap
    - Metasploit
    - Wireshark
    - John the Ripper
- Administrative/Root privileges
- Observation sheet

Theory:

What Are Hacking Tools?

Hacking tools are software utilities used for security auditing, penetration testing, and ethical hacking. They help identify:

- Vulnerabilities
- Weak passwords
- Misconfigurations
- Open ports
- Network traffic patterns
- Security loopholes

These tools are used legally and ethically by security professionals.

Types of Hacking Tools and Their Purpose

Network Scanning Tools

Used to discover hosts, open ports, and services running on a network.
Examples: Nmap, Zenmap
Purpose: Vulnerability scanning, port scanning, mapping network topology.

Exploitation Frameworks

Used to exploit vulnerabilities and test system defenses.
Example: Metasploit
Purpose: Launch exploits, payload generation, penetration testing automation.

Packet Sniffers / Network Analyzers

Capture and analyze network traffic.
Example: Wireshark
Purpose: Analyze packets, detect intrusions, troubleshoot network issues.

Password Cracking Tools

Used to test password strength.
Examples: John the Ripper, Hashcat
Purpose: Brute force, dictionary, and hybrid attacks.

Web Vulnerability Scanners

Scan web applications for security flaws.
Examples: Burp Suite, OWASP ZAP
Purpose: Identify SQL injection, XSS, CSRF, misconfigurations.

Wireless Hacking Tools

Used for wireless network auditing.
Examples: Aircrack-ng suite
Purpose: Analyze Wi-Fi security, crack WEP/WPA keys.

Reverse Engineering Tools

Used to analyze malware and binaries.
Examples: Ghidra, OllyDbg
Purpose: Malware analysis, vulnerability research.

Procedure (for Practical Demonstration)

Network Scanning with Nmap

1. Open terminal.
2. Run command:
   nmap <target IP>
3. Observe open ports, services, OS detection.

Packet Analysis Using Wireshark

1. Start packet capture.
2. Choose network interface.
3. Analyze captured packets (HTTP, DNS, TCP, UDP).

Exploitation Using Metasploit

1. Launch Metasploit console.
2. Search for an exploit.
3. Set target and payload.
4. Run exploit.

Password Cracking Using John the Ripper

1. Provide a hashed password file.
2. Run John to crack the hash.
3. Observe cracking speed and password complexity.

Important Parameters to Study

Category	**Parameter**	**Description**
Scanning	Open Ports	Identifies vulnerable services.
Sniffing	Packet Details	IP, MAC, protocol, payload.
Exploitation	Vulnerability Type	CVE used, exploit success.
Password Tools	Hash Type	MD5, SHA, bcrypt, NTLM.
Wireless	Encryption	WEP/WPA/WPA2 strength.
Web Testing	Payloads	XSS, SQLi vectors.

Observations

- Details of open ports and services discovered
- Amount and type of captured traffic
- Exploit behavior and system response
- Time taken for password cracking
- Identified vulnerabilities in systems/web applications

Result

After studying various hacking tools, students will be able to:

- Identify and categorize cybersecurity tools.
- Understand their role in ethical hacking and penetration testing.
- Use common tools to analyze networks, crack passwords, capture traffic, and exploit vulnerabilities.
- Recognize potential security weaknesses in systems.

Conclusion

This practical provides hands-on learning of widely used hacking and cyber security tools. Understanding these tools helps students perform ethical penetration testing and enhances their ability to secure systems and networks. Knowledge of scanning, sniffing, exploitation, and cracking tools is fundamental for any cyber security professional.

6: Practical applications of digital signature

Introduction

A **Digital Signature** is a cryptographic technique used to validate the authenticity, integrity, and non-repudiation of electronic documents or messages. It ensures that the document is signed by the rightful owner and has not been altered during transmission. Digital signatures rely on public key cryptography (PKI) and are widely used in government, business, banking, and legal processes.

Objectives

- To understand the concept and purpose of digital signatures.
- To study various applications of digital signatures in real environments.
- To explore how digital signatures ensure security, authenticity, and integrity.
- To learn about digital-signature tools and platforms used in industries.

Requirements

- Computer system with internet connectivity
- Sample documents (PDF, Word, text)
- Access to digital-signature tools such as:
    - DocuSign
    - Adobe Acrobat Sign
    - Aadhaar-based eSign service (India)
- Practical observation sheets

Theory:

What is a Digital Signature?

A digital signature is an encrypted electronic stamp that verifies:

- Authenticity – signer identity is valid
- Integrity – content is unchanged
- Non-repudiation – signer cannot deny signing

It uses **private key** to sign and **public key** to verify.

How Digital Signature Works

1. A hash of the document is generated.
2. Hash is encrypted using the sender's **private key** → produces digital signature.
3. Receiver decrypts it using sender's **public key**.
4. If decrypted hash matches the document hash → signature is valid.

Applications of Digital Signatures

Digital signatures are used in various sectors. Major applications include:

E-Governance and Government Services

- Filing Income Tax Returns (ITR)
- Submitting GST returns
- Digital signing for Aadhaar eSign
- Approving official communications and tenders

Banking and Financial Sector

- Online banking authentication
- Loan applications and approval workflows
- Secure fund transfers
- Digital KYC processes

Legal and Corporate Sector

- Signing contracts, agreements, MoUs
- Board resolutions and compliance documents
- Preventing document tampering in legal cases

Education Sector

- Issuing digital certificates and transcripts
- Online verification of academic records
- Digital signing on institutional documents

Healthcare Sector

- Secure sharing of patient records
- E-prescriptions
- Medical insurance claims

Business and Industry

- Vendor onboarding
- Purchase orders, invoices, and quotation approvals
- Employee onboarding and HR processes

IT and Cybersecurity

- Code signing for software applications
- Securing email communication (S/MIME)
- Authentication for cloud services

Benefits of Digital Signatures

Benefit	**Description**
Security	Uses strong cryptographic algorithms.
Authentication	Verifies signer's identity.
Integrity	Prevents tampering of documents.
Non-repudiation	Signer cannot deny the signature.
Time and Cost Saving	Paperless, fast, and efficient.
Legal Validity	Recognized under IT laws in many countries.

Procedure (for Practical Demonstration)

Steps to use a digital signature tool (Example using any digital signature platform):

1. Create or open an account in a digital-signature application.
2. Upload the document (PDF/Word).
3. Choose "Add Signature" option.
4. Select type of signature:
    - Draw

    - Type
    - Upload certificate-based signature
5. Apply the signature at required location.
6. Save/export the signed document.
7. Verify the signature details:
    - Signature validity
    - Certificate information
    - Timestamp

Observation Points

- Was the digital signature created successfully?
- Check whether the verification shows “Valid Signature.”
- Check for certificate details (issuer, public key, validity date).
- Note changes in document security properties.
- Verify whether tampering breaks signature validity.

Result

After performing the practical, students will understand:

- How digital signatures work.
- How to sign and verify documents electronically.
- Real-life applications in government, business, finance, and security.
- The importance of digital signatures in protecting data and ensuring authenticity.

Conclusion

In this practical, we explored various applications of digital signatures. Digital signatures ensure document authenticity, integrity, and legal validity. They play a major role in secure transactions across government, banking, legal, and business sectors. Mastering digital signature usage helps in working efficiently in a paperless and secure digital environment.