Week 5

Day 21

Social Engineering

Social Engineering is a type of cyber attack that exploits human psychology rather than technical vulnerabilities to gain unauthorized access to systems, networks, or sensitive information. Instead of hacking systems directly, attackers manipulate users into revealing confidential data such as passwords, personal details, or login credentials.

Studying social engineering helps in understanding how human behavior can be the weakest link in cybersecurity and why user awareness and training are critical for security.

Tool Used

SET Toolkit (Social-Engineer Toolkit)

SET Toolkit is an open-source penetration testing framework designed specifically for social engineering attacks.

Practical Work Performed Using SET Toolkit

i. Studied different social engineering attack vectors.

ii. Simulated phishing attacks in a controlled lab environment.

iii. Created fake login pages to understand credential harvesting techniques.

iv. Analyzed how users can be tricked into submitting sensitive information.

v. Learned how social engineering attacks are launched and how they can be detected.

SET Toolkit provided hands-on experience with real-world social engineering techniques while maintaining ethical standards.

Types of Social Engineering Attacks Studied

- Phishing
- Spear Phishing
- Pretexting
- Baiting
- Tailgating
- Credential Harvesting

Work Description

The topic of Social Engineering was studied in detail as per the cybersecurity syllabus. All practical activities were conducted on authorized systems within a controlled lab environment, ensuring ethical and legal compliance. The exercises enhanced understanding of human-based attack techniques and preventive security awareness measures.

Prevention Techniques Studied

- Security awareness training
- Email filtering and phishing detection
- Strong authentication mechanisms
- User verification policies
- Incident reporting procedures

Conclusion

Social engineering attacks pose a serious threat because they target human trust. Through practical exposure using the SET Toolkit, a deeper understanding of social engineering techniques and their impact was gained. This study emphasized the importance of user education and security awareness in preventing such attacks.

Day 22

Wireless Security

Wireless Security focuses on protecting wireless networks (Wi-Fi) from unauthorized access, data interception, and network-based attacks. Since wireless communication uses radio signals, it is more vulnerable than wired networks if not properly secured. Studying wireless security helps in understanding how attackers exploit weak configurations and how to implement strong protective measures.

Tools Used

1. Aircrack-ng

Aircrack-ng is a popular wireless network security auditing tool used to test the strength of Wi-Fi encryption.

Practical Work Performed Using Aircrack-ng

i. Monitored wireless networks in range.

ii. Captured wireless packets and handshakes in a controlled setup.

iii. Analyzed Wi-Fi security protocols such as WEP, WPA, and WPA2.

iv. Studied how weak passwords and misconfigurations can compromise wireless security.

This practical work demonstrated how attackers exploit weak wireless encryption and why strong passwords are essential.

2. Wireshark

Wireshark is a network packet analyzer used to capture and analyze network traffic.

Practical Work Performed Using Wireshark

i. Captured wireless network traffic.
ii. Analyzed packet flows and communication protocols.
iii. Identified unencrypted data transmitted over wireless networks.
iv. Studied authentication and association processes in Wi-Fi communication.

Wireshark helped in understanding how data travels over wireless networks and how insecure configurations can lead to data leakage.

Types of Wireless Attacks Studied

- Unauthorized access to Wi-Fi networks
- Packet sniffing
- Man-in-the-Middle (MITM) attacks
- Weak encryption exploitation
- Rogue access points

Work Description

The topic of Wireless Security was studied in detail as per the cybersecurity syllabus. All practical activities were performed only on authorized systems within a controlled lab environment, ensuring ethical and legal compliance. The exercises enhanced understanding of wireless threats and security testing techniques.

Prevention Techniques Studied

- Use of strong encryption (WPA2/WPA3)
- Strong Wi-Fi passwords

- Disabling WPS
- Network monitoring and access control
- Secure configuration of wireless routers

Conclusion

Wireless networks are highly vulnerable if not properly secured. Through hands-on practice using Aircrack-ng and Wireshark, practical knowledge of wireless attacks and defense mechanisms was gained. This study emphasized the importance of strong encryption, proper configuration, and continuous monitoring to ensure wireless network security.

Day 23

DoS & DDoS Attacks

Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are cyber-attacks aimed at disrupting the normal functioning of a system, server, or network by overwhelming it with excessive traffic or resource requests. The main objective of these attacks is to make services unavailable to legitimate users.

Understanding DoS and DDoS attacks is important to identify network weaknesses and implement effective defensive mechanisms.

Types of Attacks Studied

- **DoSAttack:**

  A single system sends a large number of requests to exhaust the target’s resources.

- **DDoSAttack:**

  Multiple compromised systems (botnets) simultaneously flood the target, making detection and mitigation more difficult.

Tools Used

1. hping3

hping3 is a command-line network testing tool used to generate custom TCP/IP packets.

Practical Work Performed Using hping3

i. Generated TCP, UDP, and ICMP traffic.

ii. Simulated SYN flood and ICMP flood attacks in a lab setup.

iii. Analyzed server response under heavy traffic.

iv. Studied how abnormal traffic patterns affect system availability.

This helped in understanding how attackers flood networks and services.

2. Metasploit Framework

Metasploit is a penetration testing framework that includes modules for simulating DoS attacks.

Practical Work Performed Using Metasploit

i. Studied DoS-related auxiliary modules.

ii. Simulated controlled DoS scenarios against test systems.

iii. Observed system behavior during service disruption.

iv. Analyzed vulnerabilities that could lead to denial of service.

Metasploit provided a structured environment for testing DoS vulnerabilities.

Work Description

The topic of DoS & DDoS Attacks was studied in detail as per the cybersecurity syllabus. All practical activities were performed only on authorized systems within a controlled lab environment, ensuring ethical and legal compliance. These exercises enhanced understanding of attack mechanisms and system resilience.

Impact of DoS & DDoS Attacks

- Service downtime
- Financial losses
- Reputation damage
- Network congestion

Prevention Techniques Studied

- Traffic filtering and rate limiting
- Firewalls and IDS/IPS
- Load balancing
- DDoS mitigation services
- Network monitoring

Conclusion

DoS and DDoS attacks pose serious threats to network availability. Through hands-on practice using hping3 and Metasploit, practical knowledge of traffic flooding attacks and mitigation strategies was gained. This study emphasized the importance of network monitoring and defensive security measures.

Day 24

Mobile Security

Mobile Security focuses on protecting mobile devices and mobile applications from security threats such as data leakage, malware, unauthorized access, and insecure communication. With the widespread use of smartphones and mobile applications, ensuring mobile security has become a critical aspect of cybersecurity.

Studying mobile security helps in understanding how vulnerabilities in mobile apps and device configurations can be exploited and how to implement effective security controls.

Tools Used

1. Android Emulator

The Android Emulator provides a virtual Android device environment for testing mobile applications without using a physical device.

Practical Work Performed Using Android Emulator

i. Installed and executed Android applications in a virtual environment.

ii. Observed application behavior and permissions.

iii. Tested application functionality in a safe and isolated setup.

iv. Simulated different Android versions and device configurations.

This allowed secure testing of applications without risk to real devices.

2. MobSF (Mobile Security Framework)

MobSF is an automated mobile application security testing framework used for analyzing Android applications.

Practical Work Performed Using MobSF

i. Performed static analysis of Android APK files.

ii. Identified insecure permissions and hardcoded secrets.

iii. Analyzed application components such as activities, services, and broadcast receivers.

iv. Checked for insecure data storage and weak cryptographic practices.

v. Reviewed security reports generated by MobSF.

MobSF provided insights into common mobile application vulnerabilities and security misconfigurations.

Types of Mobile Security Issues Studied

- Insecure data storage
- Weak authentication mechanisms
- Excessive app permissions
- Insecure network communication
- Hardcoded credentials

Work Description

The topic of Mobile Security was studied in detail as per the cybersecurity syllabus. All practical testing was conducted on authorized applications within a controlled lab environment, ensuring ethical and legal compliance. The exercises enhanced understanding of mobile application threats and secure development practices.

Conclusion

Mobile devices are increasingly targeted by attackers. Through hands-on practice using the Android Emulator and MobSF, practical knowledge of mobile application vulnerabilities and security testing techniques was gained. This study emphasized the importance of secure coding, proper permission management, and thorough security testing for mobile applications.

Day 25

Android Pentesting

Android Pentesting refers to the process of testing Android applications and devices to identify security vulnerabilities that could be exploited by attackers. It involves analyzing application code, permissions, storage mechanisms, and communication channels to ensure that sensitive user data and system resources are properly protected.

Studying Android pentesting helps in understanding common Android vulnerabilities and secure mobile application development practices.

Tools Used

1. ADB (Android Debug Bridge)

ADB is a command-line tool that allows communication with Android devices or emulators.

Practical Work Performed Using ADB

i. Connected Android emulator/devices using ADB.

ii. Installed and managed applications on the device.

iii. Accessed application directories and logs.

iv. Analyzed application behavior and permissions.

v. Tested file system access and data storage locations.

ADB helped in understanding how attackers interact with Android devices at the system level.

2. MobSF (Mobile Security Framework)

MobSF is an automated security testing framework used for mobile application analysis.

Practical Work Performed Using MobSF

i. Performed static analysis of Android APK files.

ii. Identified insecure permissions and exposed components.

iii. Detected hardcoded credentials and API keys.

iv. Analyzed encryption and data storage practices.

v. Reviewed detailed security reports generated by MobSF.

MobSF provided insights into common Android application vulnerabilities.

Types of Android Vulnerabilities Studied

- Insecure data storage
- Weak authentication mechanisms
- Excessive application permissions
- Insecure inter-process communication
- Hardcoded secrets

Work Description

The topic of Android Pentesting was studied in detail as per the cybersecurity syllabus. All practical tasks were performed on authorized applications within a **controlled lab environment**, ensuring ethical and legal compliance. The exercises enhanced understanding of Android security testing methodologies and defensive measures.

Conclusion

Android pentesting is essential for securing mobile applications. Through hands-on practice using ADB and MobSF, practical knowledge of Android vulnerabilities and penetration testing techniques was gained. This study emphasized the importance of secure coding, permission management, and regular security testing for Android applications.