# Cyber Security Lesson 1 23/08/21

## Introduction

Cyber Security: The ability to protect or defend the use of Cyber-attacks,

* The prevention of damage, protection and restoration of communication and electrical services.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Availability** | **Integrity** | **Authentication** | **Confidentiality** | **Non-Repudiation** |
| Authorized users that can freely access the system, network and data needed to perform their daily task | To Ensure that a system, software, data has not been modified by unauthorized users or unauthorized means | Assurance and confirmation of a user’s identify | Ensures that sensitive information are accessed only by authorised person and kept away from unauthorized users | Ensures the inability to refute responsibility |

## Cyber Security Cube

* A model framework for establishing information security

|  |  |  |
| --- | --- | --- |
| **Desired Goals** | **Data States** | **Safeguards** |
| Confidentiality | Transmission | Human Factor |
| Integrity | Storage | Policy & Practice |
| Availability | Processing | Technology |

**Asset** – An Asset is what we are trying to protect (Data, Hardware, Software etc.)

**Vulnerability**: A weakness in our protection efforts

**Exploit**: How hackers take advantage of vulnerability’s

**Threat**: Potential negative action that results in a unwanted impact to a computer system or network.

**Risk**: Probability of exposure, loss of data or assets as a result of a cyber-attack

## Causes of Vulnerabilities

* Design & Development Errors
* Poor System configuration
* Human Error (Human Factors)
* Connectivity (using unsecure networks)
* Complexity (Adding too many layers of security = harder to maintain)
* Passwords
* User input (User input validation)
* Management
* Lack of Training Staff
* Communication

|  |  |
| --- | --- |
| **Cyber Threat Actors** | **Motivation** |
| Nation-states | Geopolitical |
| Cyber Criminals | Profit |
| Hacktivist | Ideological |
| Terrorist Groups | Ideological Violence |
| Thrill-Seeker/ Script kiddie | Satisfaction. |
| Insider Threat | Motivated by discounted. |

## Cyber Threat

|  |  |
| --- | --- |
| **Malware** | Software that does malicious tasks on a device or network. |
| **Spyware** | Is a form of malware that hides on a deceive providing real-time information. |
| **Phishing Attacks** | Is when a cyber-criminal attempts to lure individual into providing sensitive data. |
| **Ransomware** | Denies Access to a computer system or data until paid. |
| **Advanced Persistence Threat** | Unauthorised users have access and remains undetected inside the network. |
| **Trojan** | Misleads user of its true intent. |
| **Backdoor** | A way for unauthorised users to gain access to a system or network. |
| **Rogue Software** | Type of malware that misleads users into believing there is a virus on their system and aim to convince them to pay for a fake malware removal tool |
| **Data Destruction** | Is when a cyber-attacker attempts to delete data |
| **Drive-by-down Intellectual Property load** | Installing malware through a website visit |
| **Intellectual Property Theft** | Stealing the Intellectual Property of a organisation |
| **Unpatched Software** | Software that has weak security |
| **Zero-day attack** | Unknown vulnerabilities in a system |
| **Natural Disasters** | Earthquakes, Floods, Tsunami, Hurricanes |

### Cyber Threat Surface

* All the available endpoints that a threat actor may attempt to exploit a computer systems or data (Looking through windows, over the shoulder)
* Process that produces, deliver and reply on information systems connected to the internet are also a potential threat vector
* Services, Devices & Data can all be targeted
* System that connect physical entities with the internet

### Cyber Kill Chain

#### Steps to attacking the Target

Reconnaissance – Delivery – Weaponisation – Exploitation = Installation – Command & Control – Actions on Objectives

# Cyber Security Lesson 2 24/08/21

## Virtualisation

* Process of separating software form Hardware
* Hypervisor works as a interfaces between VM’s & the host
* Virtual Server exists digitally not physically
* When using multiple VM’s they are usually separate and can’t communicate with each other
* Host OS & Virtual OS can be different Operating systems

### Traditional Architecture

Application 🡪 Operating System 🡪 Physical Node

### Virtual Architecture

OS OS OS 🡪 Virtual Layer 🡪 Physical Node

#### Virtualisation Type

|  |  |
| --- | --- |
| Type 1 Hypervisor | Type 2 Hypervisor |
| * Hypervisor running straight from the hardware/Bare-metal | * Hypervisor running form and existing host operating system |

### Vulnerability with Type 1 and Type 2

|  |  |
| --- | --- |
| **Type 1** | * Better Security (Less Layers) * Performance (Due to no OS) |
| **Type 2** | * Better simplicity (Easy to set up) |

## Linux Architecture

**Inner Layer – *Hardware*** = Physical electrical device,

**Outer Layer 1 - *Kernel*** = The centre of a computer operating system, it’s the core that provides basic services

**Outer Layer 2 – *Shell*** – The interface between the user and the kernel, allowing the user to execute programs and commands

**Outer Layer 3 *– Application & Utilities*** = Software that performs specific tasks for the end user

## Linux Distributions

|  |  |  |
| --- | --- | --- |
|  | Linux |  |
| Debian | **Arch Linux** | **Fedora** |
| Ubuntu & Kali |  | **Red Hat & Centos** |

* Each distribution focuses on a different purpose, for example Ubuntu is commonly for general use, while Kali is designed for networking or cyber-security users.

## Linux File system

* A file system is the system that control how to store & retrieve data
* **Microsoft** == FAT, FAT32 & NTFS while **Linux** == EXT$, EXT£, BTRFS, XFS
* Windows FS uses letters for partitions, while Linus works on concepts that everything starts at the root, (A Tree Concept)

# Cyber Security Lesson 3 25/08/21

## Bash Scripting

|  |  |
| --- | --- |
| Bash Commands |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Cyber Security Lesson 4 26/08/21

## Introduction

Deployment Environment = Is an environment that is where your developing the applications. Where developers will be using their IDE, editors, programming languages to build to application and test it.

Production Environment = Is where the application will be providing the services, and sent out to the customer to use. For example on a server. New update versions of the software will be push to the production environment.

Each Virtual Machine, might have different: operating systems, updates, plugins etc., which can cause issues. To solve these issue:

* Vagrant Tool (Usually used in the development Environments)
* Docker (Usually used in the production Environment)

## Vagrant

* Is a tool used to manage and configure multiple Virtual Machines at once, using vagrant files to, in a single work flow.
  + This lowers deployment environment setup time, as multiple Virtual Machine can be setup and configured to the developers liking quickly
* This will later allow the Production Environment to mirror the developers machine.
* Developer will still be able to use their favourite editors, IDE & browsers
* It can also create a virtual Environment for each Production Environment
* Vagrant isn’t a Hypervisor. However, is requires and depends on Hypervisor to create the Virtual Machine using virtual Machines such as; Virtual Box, VM Ware or Hyper –V

## Containers

* Containerization is the process of encapsulating and or packaging up software code and all its dependencies, so that it can run uniformly and consistently on any infrastructure.
* A single container is abstracted away from the host operation system, making it stand alone and portable – able to run across any or multiple platforms or cloud

Virtual Machines will use a guest operating systems, whereas containers don’t. container, will share from the Host operating systems kernel.

Virtual Machine can produce extra overhead due to Guest Operating system memory and storage,

### Benefits

|  |  |
| --- | --- |
| **Benefits** | **Description** |
| **Portability** | Due to a container being abstract, it can be sent or run across any platform or cloud, with consistence |
| **Speed/Lightweight** | Containers share the host operating systems kernel, but are not bogged down with its extra overhead, also using less recourses |
| **Ease of Management** | Container orchestration platform automates the installation, scaling and management of containerized workless and services. |

### Concerns

|  |  |
| --- | --- |
| Concerns | Description |
| Security | If the user can get access to Host OS, they can get access to containers.   * This is because containers share the same kernel. |
| Poor Configuration | Poor config, can leave vulnerabilities in container |

## Docker

Docker is an open platform for developing, shipping and running applications. Using containers, Docker can separate application forms infrastructure and deliver software quicker.

Docker creates, pulls and pushes images from/to the Docker Daemon API

* Docker Daemon, will search for existing image locally, with not locally then from the Docker hub

## Cloud Computing

Cloud Computing is On Demand delivery of computer resources, such as:

* Software
* Storage
* Databases
* Network

***Benefits of cloud computing,***

Flexibility/Saleability – Cloud computing allows for scalability, easily upscaling or downscaling IT requirements as required.

Cost/Efficiency – Allows for quick remote access and decrease in electricity consumption via datacentres running in the cloud. As well as limits cost in purchasing hardware needed for demanding software.

***Concerns of Cloud Computing***

Security – Part of security responsibly is with the cloud provider, so if there is a vulnerability with their security, it might affect the customers of the service.

Data Portability - Some Cloud providers, don’t allowing for data to be move to other cloud services until end of contracts, holding the data hostage.

Cloud computer – Deployment Type

|  |  |
| --- | --- |
| **Deployment Type** | **Description** |
| **Public Cloud** | Cloud services use and operating by members of the public, usually free, data can be accessed, stored, retrieved from anyway. (i.e Google Cloud, iCloud) |
| **Private Cloud** | Organisation pay for a private section of the cloud, public are unable to access private cloud data, only though with the organisation have access to their data with the cloud. |
| **Hybrid Cloud** | When 2 or more organisation share a private cloud, both sharing the private cloud together, splitting the money  A combination of a public and private cloud, allowing data and applications to be shared between them |
|  |  |

Cloud Services

On-Premise = Services of application developed, deployed, maintained on hardware with the organisation

IaaS (Infrastructure as a services) = Allows for clients to have greater control of hardware resources, software and scalability. Services customers have responsibility of their own data, software, security, networking, confirmation and maintenance.

PaaS (Platform as a Service) = Allows for client to build software on a pre-configured platform within the cloud. The have very little control over resource scalability and control.

SaaS (Software as a Service) = Clients can only use software through a subscription or purchase scheme, the client only has control over their data.