Khaled:

Introduction:

Today, the presence of the internet or any other type of network in our lives is indisputable. These networks allow us to communicate between distant places with a negligible delay.

It is as a result of this technology that a new computing model known as Cloud Computing emerges. However, this technology predates the use of the internet and the idea of distributed computing arose in the 1960s.

when John McCarthy proposed the idea of shared system computing to sell the resources of a computer as if it were another service. However, due to the limitations of technology in terms of networks, the project was put on hold.

It was in the 90s when this idea reappeared, since the internet of the time already had enough bandwidth to support cloud computing. From then on, more and more companies began to develop this technology, including Compaq Computer or Amazon, which, in view of the amount of idle resources on their servers, launched Amazon Web Services.

Khaled:

History:

Although it may seem that the cloud is a very recent environment, it comes from the 1960s, when access to a computer was reserved for universities and large companies.

As technology advanced, the creation of networks of computers connected to each other was proposed, which sold their use as a more resource for a company, giving rise to what was known as ARPANET, this being the forerunner of the Internet.

To this all this was added the emergence of virtualization technology, making it possible in the 70s to run more than one operating system in isolation on the same machine. The orchestrator of this technology was IBM who launched the VMOS (Virtual Machine Operating 6 System).

This concept evolved along with the Internet until virtual private network , now leading to Cloud Computing in the 1990s.

Later, at the beginning of the century, AWS emerged, launching Elastic Compute Cloud (EC2) shortly after allowing companies and individuals to rent virtual machines through which they could use their own programs and applications.

At the same time, Google also launched its Google Docs service.

Shortly after, IBM, Google, and several universities collaborated to develop a server farm in which to jointly investigate. It also coincided in the same year, in 2007, that Netflix launched its streaming service using the cloud to transmit movies and audiovisual content to its customers.

After all this, the cloud has not stopped growing until today.

Hamada:

03.Why do we need Security ?

- Increase in security breaches over time.

- Cyber attacks.

- Possible data loss.

-The increasing transfer of data and applications to the cloud.

- Leaks and stolen data.

3.1.Security requirements Data:

- Access control.

- Viewing information according to privileges.

- Monitoring and recording of operations.

- Redundancy and backups.

- Encryption.

- Blocking access to files.

3.2.Security requirements Servers:

Necessary dependencies.

Minimum privilege.

Strong passwords.

Double factor authentication.

Intrusion detection, attacks and monitoring.

Intrusion and configuration tests.

Isolated MVs.

Access control, monitoring and audits of resources.

3.3.**Security requirements Networks:**

- Segmentation.

- End-to-end encryption.

- Monitoring, detection and action against DDos attacks.

- Necessary access points.

- Restrict Internet access.

- Standard and secure communication protocols: VPN, TLS, HTTPS.

Hamada:

04.Security flaws

the most common risks that can occur when we do not implement any of the necessary security measures :

unauthorized access to both servers and data : if they access the machine managed by the provider, the responsibility lies with the provider, but the services hosted and the data used by them are the responsibility of the developer.

- Zero Trust Architecture: is based on the principle of not trusting any entity it allows you to limit their access and block operations to both identified and non-identified users.

Although we are more aware of cloud security today, many companies still include unreliable security measures that can be circumvented by attackers or malware to gain access to servers and data.

here are the most Attackers:

-Brute force.

-Account hacking.

-Phishing.

-Social engineering.

-DDoS.

-Malware injection.

Khaled:

05. Examples of attacks:

Tesla:

Unprotected AWS Management Console.

Confidential information.

Crypto jacking.

Crypto mining programs to a server.

Low CPU usage.

SSL communication encryption.

Dropbox:

68 million user credentials on the dark web.

Dropbox employee used the same password as on Linkedin when this platform was also attacked.

Solution: new passwords.

Onedrive:

Fake Microsoft mail to verify files.

Fake website to authenticate.

Steal credentials.

Continue the attack.

Attacks against user contacts.

Dyn:

The company did not expect a DDos attack of this magnitude.

Drop in service for several companies in Europe and the US:

Twitter,Github,Spotify, PayPal.

Hamada:

06.Security tools: we have a lot off kind with tools security I chse fort :

CloudGuard Dome9:

- Configuration, good practices in legislation and security.

- Protection against theft of credentials and data.

- Contextual intelligence.

- Threat information.

- Improved detection.

- Payment tool.

- AWS, Azure and Google Cloud

Panther:

- Open source.

- AWS CloudFormation.

- Threat detection with deterministic rules.

- Log analysis to detect unauthorized access and suspicious behavior.

- Automatic correction of vulnerabilities in the configuration.

ZeeK :

- Open source.

- Intrusion detection and prevention.

- Network monitoring.

- Adaptation to the context of the network.

- Customize their behavior through scripts.

- Forensic services.

Okta :

- Payment tool.

- Network protection.

- Monitoring and control of user and employee access.

- Multi-device.

- Management of particular privileges and authentication.

- Configurable login.

Khaled:

7.Infection monkey:

Out of all the existing tools, we have decided to give Infection Monkey a try. It is an open source software developed by the Guardicore cloud security company, aimed at testing a cloud system carried out various attacks to verify its resistance. It can be applied to both public and private clouds.

Its origin lies in an investigation carried out by the company in 2017 about micro-segmentation applied to networks. With this methodology, a network can be divided into several subnets dedicated to a single entity, such as processes, applications, users, among others.

However, this technique requires that the company have enough information about its network to be able to divide it, which is often an extremely complicated task to be carried out by one person.

Therefore, this company developed a tool called **Guardicore**, capable of obtaining information about its infrastructure through a series of software agents and records of its information flow.

It helps to define micro-segmentation policies through a simple interface based on the information collected, suggesting measures to improve network fragmentation.

This tool is valid for hybrid environments, as well as for virtual machines, containers and instances in providers such as Amazon Web Services, Microsoft Azure and Google Cloud.

After the success of the previous tool, the Guardicore company decided to develop Infection Monkey, a new software with which to test the infrastructure of a system.

Unlike the previous one, they decided that this one would be open source, so that it could be modified according to particular needs.

It is multiplatform since it supports operating systems such as Windows and Linux, as well as those of cloud systems such as OpenStack.

Among the various characteristics it has, three in particular have caught our attention. First of all, Infection Monkey is designed so that its evaluations do not affect the performance of the system, so that they can be performed without slowing down the performance of your tasks.

Likewise, at the end of the tests, it generates various reports with different levels of detail, ranging from an overview of the results to a detailed record of the operations and their results. In this way, security experts can use this information to gain an in-depth understanding of which system components have security problems.

Third relevant quality, we highlight that for each security incident it suggests at least one solution to resolve it.

Hamada:

Among the different attacks that it can carry out, a set of them is based on replicating some of the most popular exploits, such as the following.

● Sambacry. It allows executing code by accessing in write mode one of the shared resources located on a Linux server. To do this, the tool uses brute force to test the connections to these resources in order to infiltrate a system.

● Shellshock. Allows attackers to execute code by appending it to the end of environment variable values.

● Elastic Groovy. The tool examines the default ports that servers are serving queries on and tries to perform the same operations as above.

● Struts2. This vulnerability is associated with certain versions of the framework of the same name, which allows an attacker to insert commands into the system payload to perform malicious actions.

● WebLogic. It is a security flaw in Oracle's WebLogic servers, specialized in the development of business Java applications, which allows machines to be infected by sending malicious packets.

● Theft of credentials. If it is a Linux system, it generates a multitude of SSH keys and then infiltrates the system.

● Hadoop. It is a framework that allows the execution of remote tasks in distributed systems. If the tool finds such a server, it creates a task to spread the infection to all nodes in the infrastructure.

● Brute force intrusion. Infection Monkey uses the credentials stolen in previous steps to enter systems through the following protocols:

○ SSH. If it manages to enter the system through SSH it collects Infrastructure and kernel information on the machine.

○ SMB (Server Message Block). It is a network protocol that allows resources to be shared,, between several Windows machines. If the tool manages to sneak in, it creates a new server to run an instance of it on the machine.

○ WMI (Windows Management Instrumentation). It is an administration server to manage distributed systems remotely. When the tool infiltrates, it copies itself to other computers and spreads the infection.

Infection Monkey also includes analysis tools to prevent some of the most common attacks:

● Credential analysis. It applies different techniques such as analyzing the caches to obtain the keys or their hash.

● Network segmentation. If the tool detects that the network is fragmented, it tries to infect the systems of a subnet and then tries to spread the infection to the rest through the channels that communicate them.

● Tunneling. Analyze segmentation rules to check the strength of communications between subnets. If the systems do not verify the origin of the messages received, they can attend requests from malicious machines with which they establish communication to start the attacks.

Khaled:

Practice :

Once we have detailed the qualities of Infection Monkey, we are going to test it experimentally. To do this, we first register on your platform to receive the download address. In addition to some personal data, we must specify the environment in which we want to install it. In our case, we are going to apply it on Azure, since we have a virtual machine in which some microservices deployed for the Cloud Computing subject of the previous semester are stored.

Once we obtain the link of the download by mail, when choosing Azure as a cloud provider, it redirects us directly to the creation of a new virtual machine with the Infection Monkey image, in which we can decide its characteristics, as can be seen in the following two figures. In our we have opted for the basic configuration.

we can observe the specification of some fields such as the assigned resource group, the name of the machine, the region in which it is located and the base image that is that of the tool itself. To then be able to connect to it, we generate a pair of SSH keys, specifying the public one. For this we have followed the steps of the official documentation for the Ubuntu operating system.

A general panel appears in which, as a previous step, the purpose of the same is explained. Then with Run Monkey, you can specify specific values for the previously explained attacks and vulnerabilities.

As a first test we have run the tool inside the machine that contains Infection Monkey. In this case we have left the default configuration. Once the analysis is finished, it shows the general results graphically, indicating the detected vulnerabilities in red.

Those in gray are those that could not be tested due, in our case, to the fact that we do not have tasks running on the machine or data transfers or use.

Among the different flaws it has found is the absence of antivirus or firewalls and the possibility of creating a new user with which to connect to the Internet.

Likewise, it makes a record of the operations carried out and their respective results, as we explained above, in such a way that it generates a kind of detailed logs of how it has managed to carry out the attacks. It also suggests

some solutions such as those shown, which in our case advise us to install some security software in addition to restricting the network access policy for users.

Hamada:

Conclusions

As a conclusion to this work, it has been seen that security is an indispensable feature in cloud environments. These incidents can come from various sources and be of various kinds. In addition, despite relying on this type of paradigm, it must be backed up with strong security measures.

That is why supplier companies must implement as many measures as possible. For this, we can include techniques for collecting and analyzing detected threats as well as the infrastructure, to generate a knowledge base that allows improving the detection capacity of security tools.

Another important aspect to take into account is that they must be careful with their data.

Infection Monkey is a powerful tool for cloud environments, providing us with valuable information regarding our environment. It will be in our power to solve the problems that said tool detects in addition to using other tools such as those also commented to obtain additional information on aspects not covered by the software used.
