INTERNATIONAL DEVELOPMENT PLANNING REVIEW
ISSN:1474-6743 | E-ISSN:1478-3401

ASSESSING THE VULNERABILITIES AND IMPACTS OF OPEN SSH PORTS ON CENTOS 9 VIRTUAL MACHINES HOSTED ON MAC ARM COMPUTERS CVE-2023-25136

Gurpreet Singh[1], Saurabh Singh[2*]

[1]Endicott College of International Studies, Woosong University, 171 Dongdaejeon-ro, Dong-gu, Daejeon, (34606), Republic of Korea

[2]Department of Ai and Big Data, Woosong University, 171 Dongdaejeon-ro, Dong-gu, Daejeon, (34606), Republic of Korea

[*]Corresponding author email: singh.saurabh@wsu.ac.kr

Abstract. In today's world where the technology is growing so fast and rapidly, there is a need of cybersecurity professionals in every field. The more data you have the more crucial it is to make it secure, in this research paper we dive into the vulnerability of SSH (Secure Shell Protocol) in CentOS 9 Stream which is by default open when installed on Parallels Desktop in Apple Mac Air M1 (Silicon chip & Intel), as on research we get to know that the companies like HCL Technologies, Wipro, Fujitsu and so many others are using Parallels Desktop now days for running Virtual Operating Systems, on testing different Virtual Operating system and finding basic vulnerabilities we came across an operating system which was behaving differently when installed on Windows Virtual Box and when installed on Parallels Desktop on Mac Air M1, on further testing we were able to find that on Windows Virtual Box the RHEL based Linux operating system CentOS when was installed was having SSH (Secure Shell) port closed but when installed on Parallels Desktop on Mac, the port was opened by default and was easily accessible with another machine by SSH command (Secure Shell). The research is based on understanding the ARM based operating system and why these operating system doesn't allow many Linux based operating system to get installed on mac. As mentioned, that big companies are using Parallels Desktop now a days, we understand and studied how ARM architecture works as for a beginner and would also discuss how the configuration file when downloaded on Parallels Desktop misconfigured by default as compared to when installed on Windows Virtual Box. In this paper we will discuss about the CVE-2023-25136 which was being public in February 2023 as well.

Keywords: SSH Vulnerabilities, CentOS 9, Mac M1, Virtual Machines, Cybersecurity, Network Security, Open SSH Ports, Vulnerability Assessment

Classification numbers: 4.6.1, 4.6.2, 4.6.3, 4.6.4

1. INTRODUCTION

In today's digital world, secure communication is important and plays crucial part when you are working on a large data scale (Big data), the study will be showing that the common Networking Protocol like **SSH (Secure Shell)** when misconfigured or unchecked can create potential threats to the companies. Companies like Wipro, HCL Technologies, Fujitsu and many more are working on Parallels Desktop which is a virtualization software only for the Mac operating systems, while there can be many reasons for the companies to use different operating system for different purposes like for Web server it is said that CentOS is used vastly. While CentOS performs differently when installed on Windows Virtual Box/ VMWare while performs different when installed on Parallels Desktop on Mac, while there is also a possibility that common protocols like SSH (Secure Shell), FTP (File Transfer Protocol) or SMTP (Simple Mail Transfer Protocol) are required for a smooth working of CentOS in Parallels Desktop on Mac OS.

CentOS is a well-known operating system used by small as well as big scale companies for services like Web Server while there are many other reasons as well for which CentOS is best known for like light weight and easy to understand, while CentOS is a part of RHEL (Red Hat Enterprise Linux) which has the paid services and licensed that means not every individual can work on that and here the CentOS comes offering people to understand about the services and learn about them. In summary **CentOS** is a community-supported Linux operating system that is compatible with **RHEL (Red Hat Enterprise Linux)** while **Red Hat Linux** is commercially oriented Linux Distribution.

Parallels Desktop whose initial release date is 15 June, 2006 is a well-known virtualization software known best for its performance in the Mac Industry, it's best known for providing smooth running and virtualization of operating systems like Windows 11, Kali Linux, Ubuntu and many more, Parallels Desktop only supports the operating system which are based on **ARM64** often called as **AARCH64,** operating systems which are not based on ARM64 cannot run on Parallels Desktop.

This research is all about the in depth understanding of ARM Based computers architecture how they perform and why installing any virtual operating system can create misconfigurations and if not checked can cause major threats.

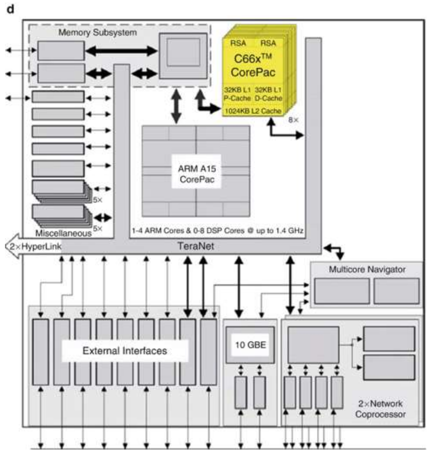
The objective of this study is not only to study SSH (Secure Shell) but also understanding this with in relation with CentOS 9 Stream, Parallels Desktop and with ARM Based computers' architectures. We will also be navigating through the demonstration. CVE-2023-25136 is also going to be the major talk.

2. IN-DEPTH INTRODUCTION TO MAC ARM, CENTOS 9, PARALLELS DESKTOP ARCHITECTURE AND SSH

This section describes the in-depth architecture information about the ARM based computer, CentOS and basic information about Parallels Desktop.

2.1. Mac M1 Architecture and Interaction with Parallels Desktop Virtualization

The Mac M1 silicon chip is known as to be revolutionary invention in the technology field. It consists of unique development by the company Apple and is based on a **system-on-chip (SoC) configuration [1]**. The M1 chip integrates central processing units (CPUs), graphics processing units (GPUs), memory, and other key components on a single piece of silicon. This brings a lot of benefits, including enhanced performance, energy efficiency, and an infinite integration of various other functions. The Mac M1's architecture are its ARM-based CPU cores. The chip consists of a heterogeneous multi-core design with four high-performance cores (Firestorm) and four high-efficiency cores (Icestorm). These are responsible and bring out computational power and helps in maintaining the background processes running smoothly. The integration of GPU cores within the M1 chip further exemplifies its versatility. Parallels Desktop, a virtualization platform designed to run multiple operating systems on a single Mac, has seamlessly adapted to the Mac M1 architecture. The latest stable versions of Parallels Desktop have been adopted in such a way to leverage the M1 chip's capabilities effectively. This growth guarantees that in essence machines (VMs) running on Parallels Producing publications with computer software can harness the complete potential of the Desktop computer M1's fittings.



System on Chip Configuration Design Example

Parallels Producing publications with computer software's unification with the M1 chip gives various benefits. First and foremost, it guarantees the effective utilization of Computer and GPU possessions inside virtual machines, developing in enhanced performance for tasks inside these

VMs. Virtualization admits for the concurrent operation of diversified operating structures on a single Desktop computer, a important feature for individuals and arrangements needing unity with a expansive array of operating system and platforms. Additionally, the M1's ARM-located design provides a instinctive floor for Parallels Desktop's support of ARM-located operating wholes. This support extends to ARM-located Linux distributions and additional podiums, facilitating different growth and testing atmospheres inside VMs. The collaborative interplay middle from two points the Desktop computer M1's architecture and Parallels Producing publications with computer software admits for the seamless unification of ARM-located and x86-located environments inside a sole system.

In this place elaborate interplay 'tween the Desktop computer M1 chip and Parallels Producing publications with computer software, users gain the benefit of flexible and high-act virtualization. The linked skills of the M1 chip and Parallels Desktop manage likely to accomplish diverse estimating needs inside a united environment. This structural collaboration provides consumers accompanying the forms they need to streamline growth, improve cross-program compatibility, and harness the adequate potential of the Desktop computer M1's innovative design. As we progress in our investigation of SSH exposures within this environment, understanding the latent structural intricacies and the interplays middle from two points elements is fundamental. The Mac M1 semiconductor crystal, Parallels Personal computer, and the CentOS 9 distribution together form a complex still very adaptable atmosphere at which point SSH configurations play a important role.

2.2. UNDERSTAING ARM BASED LINUX ARCHITECTURE

ARM, that originally signified Acorn RISC Motor, traces its inceptions back to 1983. ARM Holdings generally licenses allure intellectual property (IP) cores to generate microcontrollers (MCUs) and main processing parts (CPUs).

Key Features of ARM Architecture

1. **RISC (Reduced Instruction Set):** ARM uses simple and efficient instruction sets in a small package for high-performance, low-power operation.

2. **Load/store architecture:** Data processing is done strictly on registers, increasing performance.

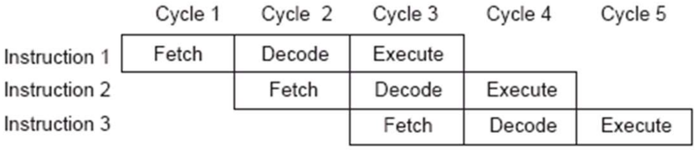
3. **Pipelining**: Instructions are executed in stages, so that the next instruction is initiated within the previous instruction.

4. **Uniform instruction length:** All ARM instructions are fixed in length, making the instructions easier to pick up and handle.

5. **Processor Modes:** ARM supports seven processor modes, each with unique attributes and privileges, such as User, Supervisor, Fast Interrupt, and others

Pipelining in ARM

Pipelining is a key concept in ARM. Enhances instruction processing by allowing the next instruction to be fetched and decoding to begin before the execution of the current instruction is completed. This makes for efficient and convenient scheduling.



	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
Instruction 1	Fetch	Decode	Execute		
Instruction 2		Fetch	Decode	Execute	
Instruction 3			Fetch	Decode	Execute

Understanding Pipelining in ARM

Encrypted Processor Modes

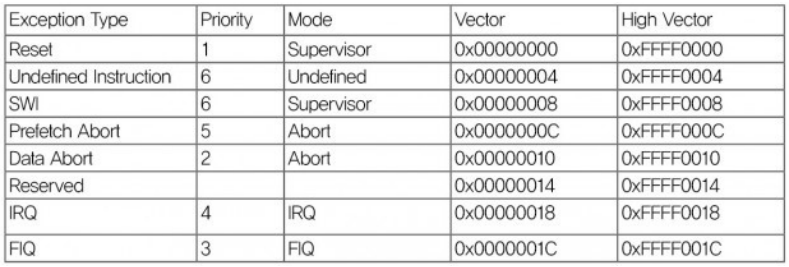
Most applications run in USER mode, without access to protected system resources. Exception conditions trigger changes to the privileged modes of accessing system resources. Each mode has specific registers to maintain state integrity.

Understanding ARM Registers

The ARM has a set of 37 32-bit registers, including 30 as general-purpose registers. Notably, R0 through R7 are not banked and maintain the same address in all modes, while R8 through R14 are banked and have separate replicas in different modes. The main registers include R13 (stack pointer) and R14 (link register), which store the return codes.

Exception handling in ARM

If an exception occurs, ARM ensures that the current instruction has completed. The processor then stores the program counter (PC) in the link register (LR) and stores the current process status in the saved process status register (SPSR). It changes the mode, enables the lower-priority exception, and sends the new mode instruction to the PC, directing it to the exception handler.

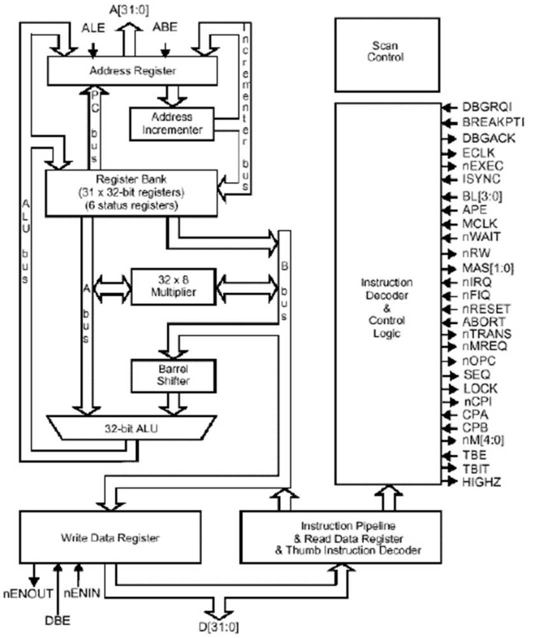


Exception Type	Priority	Mode	Vector	High Vector
Reset	1	Supervisor	0x00000000	0xFFFF0000
Undefined Instruction	6	Undefined	0x00000004	0xFFFF0004
SWI	6	Supervisor	0x00000008	0xFFFF0008
Prefetch Abort	5	Abort	0x0000000C	0xFFFF000C
Data Abort	2	Abort	0x00000010	0xFFFF0010
Reserved			0x00000014	0xFFFF0014
IRQ	4	IRQ	0x00000018	0xFFFF0018
FIQ	3	FIQ	0x0000001C	0xFFFF001C

Exception Handling in ARM

ARM Core Specifications

The ARM core consists of two main parts: the data path and the decoder. It has two read ports for registering banks, a barrel shifter for bit shifting, an ALU for arithmetic logic operations, and an address register for maintaining operands The ARM system follows the von Neumann design with a 32-. a single bit data bus carrying both data and instructions.



ARM Core Structure

Including ARM Architecture in this Review

As you explore the ARM system, consider how specific features of the ARM7TDMI, such as pipelining, processor modes, and exception handling might affect your system's performance Understanding these basic concepts will be critical to the performance of your ARM-based and testing and improving security measures.

2.3. UNDERSTANDING PARALLELS DESKTOP FOR MAC

Parallels Desktop for Mac is a powerful desktop virtualization tool that allows Mac users to run operating systems on virtual machines. Uses Mac system resources to create and manage these virtual environments, allowing for a variety of operating systems such as Windows and Linux

Virtualization approach

Parallels Desktop for Mac uses a different type of desktop virtualization called "hardware-assisted full hardware virtualization." This method uses Intel VT-x technology that enables efficient simulation of all hardware and software components of the entire computer. In effect, this approach ensures that virtual machines created with Parallels Desktop closely mimic the performance of stand-alone machines.

Customizability

One of the main advantages of using Parallels Desktop is that it offers a high degree of flexibility. Mac users can use the Parallels Toolbox to configure and control hardware-oriented options such as CD/DVD drives, USB devices, sound settings, and video capture. This change is important for your analysis, as it allows you to fine-tune the virtual environment for your CentOS 9 virtual machine.

Windows integration

Since your research involves running a CentOS 9 virtual machine on a Mac M1 computer, it is important to note that Parallels Desktop is designed to seamlessly integrate Windows applications with the Mac environment This is through technology such as Virtual Trusted Platform Module (TPM) over and comes, with Windows security features, Mac Keychain It also provides access.

Secure Boot

CentOS 9 also supports Parallels Desktop Secure Boot for virtual machines. This feature ensures the security and integrity of the virtual machines and their communication with the Mac system.

Modeling hardware

Parallels Desktop for Mac acts as hardware emulation virtualization software, using hypervisor technology to map the hardware resources of the host computer directly to those of the virtual machine this ensures that each virtual machine acts like a standalone computer and around.

Portable objects

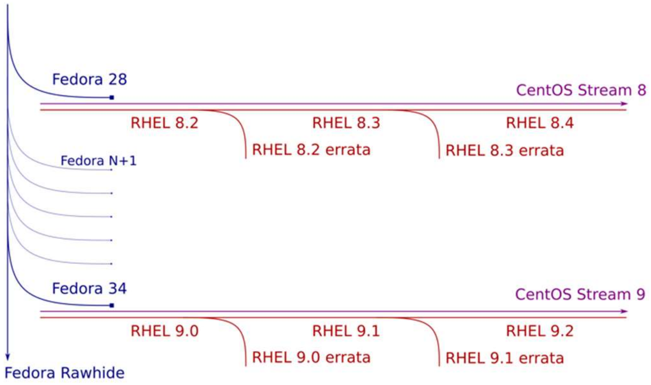
Virtual machines created with Parallels Desktop are highly portable between computers. This means that you can easily stop a running virtual machine, copy it to another physical computer, and start again. This feature is useful for your research, allowing you to work on Mac M1 computers while maintaining the same virtual environment.

Virtual hardware

Parallels Desktop for Mac virtualizes all standard PC hardware components, such as virtualized CPU, ACPI-compliant system, generic motherboard, memory, video RAM, video adapters with

OpenGL and DirectX support, storage devices, optical drives, network cards, . USB devices, sound cards, accessories, and tools.

2.4 UNDERSTANDING ABOUT CENTOS 9 STREAM



CentOS and Virtualization on ARM64/AARCH64

Known for its robustness and open-source nature, CentOS has become a popular choice for server deployments. When it comes to virtualization on the ARM64/AARCH64 architecture, CentOS offers a robust solution. Understanding how CentOS interacts with this configuration is important for your analysis using CentOS 9 virtual machines on a Mac M1 computer using Parallels Desktop for Mac.

ARM64/AARCH64 Architecture

ARM64, also known as AARCH64, represents the 64-bit version of the ARM system. It is known for its efficiency, energy efficiency and widespread use in mobile devices. However, it has gained popularity in server environments due to its efficiency and energy savings.

CentOS on ARM64

CentOS has optimized the ARM64 framework, offering a 64-bit version designed to utilize the full capabilities of ARM-based servers. It provides a robust and secure platform for various applications, including server virtualization.

Virtualization Technologies

CentOS on ARM64 supports various virtualization technologies, such as KVM (Kernel-based Virtual Machine), built into the Linux kernel. KVM allows you to create and manage virtual machines using hardware virtualization extensions to the CentOS ARM64 architecture. This ensures efficient and virtually native operations for virtualized operations.

Parallels Desktop for Mac Compatibility

Now, let’s explore how CentOS on ARM64 architecture works easily with Parallels Desktop for Mac.

Parallels Desktop for Mac is designed to run virtual machines on Intel-based Macs. However, the introduction of Mac M1 computers with the ARM64 process brought new challenges and opportunities. Parallels recognized this change and created an optimized version of Parallels Desktop for the Mac M1, which allowed users to run both ARM64 and x86_64 virtual machines on their Mac M1 computers.

ARM64 Virtual Machines on Mac M1

The compatibility of Parallels Desktop for Mac with CentOS and Mac M1 computers on ARM64 makes it easy to create and use ARM64-based virtual machines. This opens up new possibilities for testing, development and analysis, as you can now directly examine the state of CentOS 9 on Mac M1 hardware.

Integration and Performance

Parallels Desktop optimizes the addition of ARM64-based virtual machines to the Mac M1 environment. This integration ensures that a virtualized CentOS 9 environment can efficiently use underlying hardware resources to deliver high performance.

Security and Compatibility

Parallels Desktop for Mac provides a secure environment for running virtual machines, including CentOS on ARM64. You can confidently search for and assess the vulnerability and impact of open SSH ports in this controlled virtual environment, without the security of your Mac M1 host system.

Understanding how CentOS works on an ARM64 system combined with Parallels Desktop for Mac not only enables efficient virtualization but also enhances your analytics capabilities, ensuring a flexible environment and safe for your testing and research.

2.5 HOW SSH WORKS AND IT’S UNDERLYING TECHNOLOGY

SSH, or Secure Shell, is an important network security protocol that ensures secure access and file transfer. Unlike traditional methods like Telnet and FTP, which send data in plain text, SSH enhances security with an encryption and authentication mechanism. This allows for secure web services in potentially secure environments. Let’s take a closer look at how SSH works and the technology behind it.

Why SSH is Needed

In traditional Internet communication, data is transmitted in plain text, and interception is much easier. SSH solves this problem by following a client-server model, which authenticates both parties and encrypts data between them. This provides a secure messaging channel for network operations on an insecure network. SSH is mainly used for remote access and file transfer.In traditional Internet communication, data is transmitted in plain text, and interception is much easier. SSH solves this problem by following a client-server model, which authenticates both parties and encrypts data between them. This provides a secure messaging channel for network operations on an insecure network. SSH is mainly used for remote access and file transfer.

SSH Working Process

SSH utilizes the client-server model and the following phases:

1. Connection settings: A specific port is used for SSH connection. The SSH server listens for connection requests on this port. When a client sends a request, a TCP connection is established, enabling communication between the client and the server on this port. Uses SSH port 22 by default.

2. Distribution Connection: There are two versions of SSH, SSH1.X and SSH2.0. The server sends its supported version of SSH to the client, which then decides which version to use. This agreement ensures that the selected SSH versions are compatible.

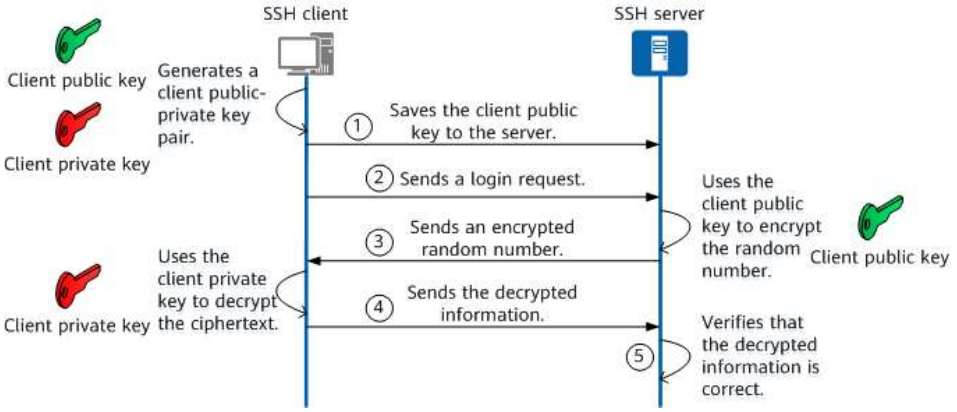
3. Algorithm Communication: SSH uses different algorithms for encryption, data integrity, and authentication. Both client and server return their supported algorithms, ready to negotiate the most appropriate one.

4. Key exchange: The SSH server and client use a key exchange algorithm to create shared session keys and session IDs. This key is used to encrypt data during transmission. The basic adjustments are made smoothly without moving the key in an unsafe manner.

5. User authentication: After the key is exchanged, the SSH client sends an authentication request to the server, which authenticates the client. SSH supports various authentication methods including password and public key authentication.

6. Session Request: After success, the client requests to establish a session with the server. This begins the conference interaction.

7. Session communication: In this phase, client and server exchange encrypted data over the session, where the data is decrypted using a shared session key.1. Connection settings: A specific port is used for SSH connections. The SSH server listens for connection requests on this port. When a client sends a request, a TCP connection is established, allowing the client to and.


SSH client
SSH server
Client public key
Client private key
Generates a client public-private key pair.
① Saves the client public key to the server.
② Sends a login request.
Uses the client public key to encrypt the random number.
Client public key
③ Sends an encrypted random number.
Uses the client private key to decrypt the ciphertext.
Client private key
④ Sends the decrypted information.
⑤
Verifies that the decrypted information is correct.

Working of SSH Protocol

Security Mechanisms

- Password authentication: The client uses the server’s public key to encrypt the password, and the server uses its private key to verify the password.
- Key authentication: This more secure method uses client public keys, encryption, and decryption to verify authentication. The server requests permission using the client's public key.

CVE-2023-25136:

A bug was found in the OpenSSH server (sshd), which caused a non-duplicate vulnerability when processing options.kex_algorithms. an unauthorized attacker can trigger a double-free in the default setting.

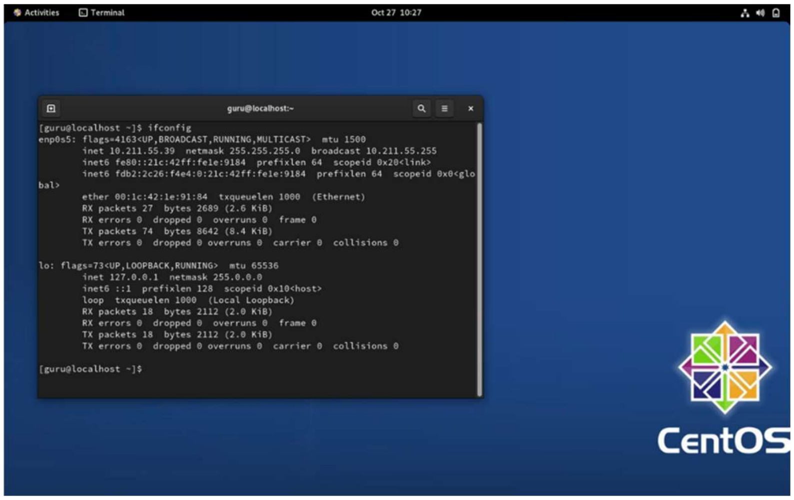
Common SSH Connection Tools

The two most common tools for SSH connections are PuTTY and OpenSSH. PuTTY is used on Windows systems for remote access, while OpenSSH is an open-source implementation for Unix systems. Windows 10 includes OpenSSH client and server software.

(1)

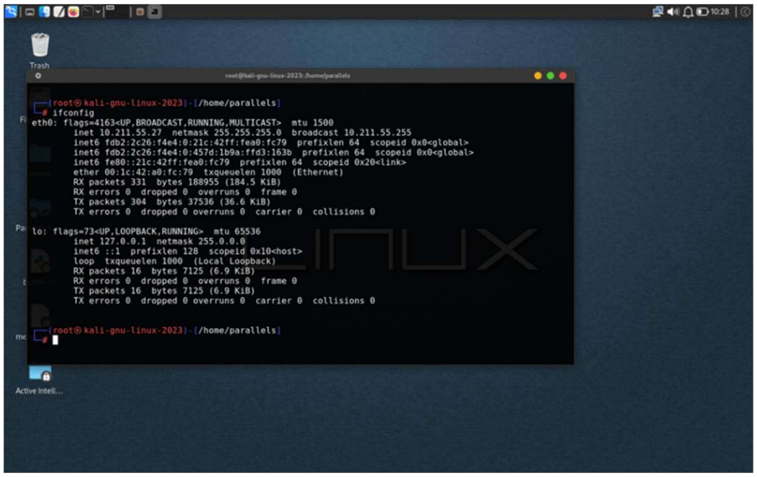
3. DEMONSTRATION

In this demonstration, I present a unique and dynamic program that incorporates a multi-operating system environment within the confines of a Mac system. Using Parallels Desktop, I have easily paired CentOS 9 as my hunting machine with Kali Linux as my designated hacking machine. These operating systems exist synchronously within the virtual realm of my Mac, allowing for a useful web-based testing environment. What makes this demonstration particularly interesting is the fact that both CentOS 9 and Kali Linux are connected to the same network, fostering an environment of thorough security scanning and penetration testing Notably, the CentOS machine has An IP address of 10.211.55.39, when its Kali Linux-1. The peer is assigned an IP address of 10.211.55.27. This demonstration highlights the versatility of virtualization technology and its critical role in creating a secure yet powerful network in a controlled and isolated.



```
[guru@localhost ~]$ ifconfig
enp0s5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.211.55.39  netmask 255.255.255.0  broadcast 10.211.55.255
        inet6 fe80::21c:42ff:fe1e:9184  prefixlen 64  scopeid 0x20<link>
        inet6 fdb2:2c26:f4e4:0:21c:42ff:fe1e:9184  prefixlen 64  scopeid 0x0<glo
bal>
        ether 00:1c:42:1e:91:84  txqueuelen 1000  (Ethernet)
        RX packets 27  bytes 2689 (2.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 74  bytes 8642 (8.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 18  bytes 2112 (2.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18  bytes 2112 (2.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[guru@localhost ~]$
```

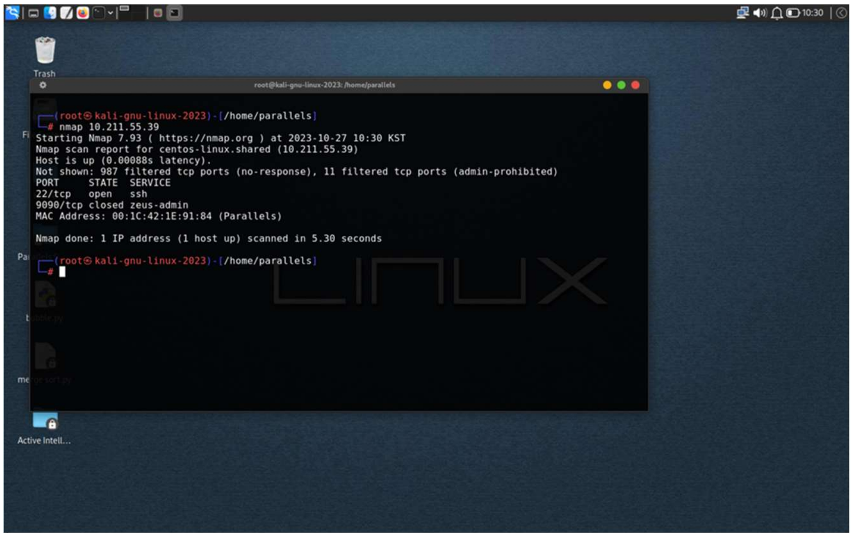
CentOS 9 Stream (Victim Machine)



```
┌──(root㉿kali-gnu-linux-2023)-[/home/parallels]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.211.55.27  netmask 255.255.255.0  broadcast 10.211.55.255
        inet6 fdb2:2c26:f4e4:0:21c:42ff:fea0:fc79  prefixlen 64  scopeid 0x0<global>
        inet6 fdb2:2c26:f4e4:0:457d:1b9a:ffd3:163b  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::21c:42ff:fea0:fc79  prefixlen 64  scopeid 0x20<link>
        ether 00:1c:42:a0:fc:79  txqueuelen 1000  (Ethernet)
        RX packets 331  bytes 188955 (184.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 304  bytes 37536 (36.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 16  bytes 7125 (6.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 7125 (6.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(root㉿kali-gnu-linux-2023)-[/home/parallels]
└─#
```

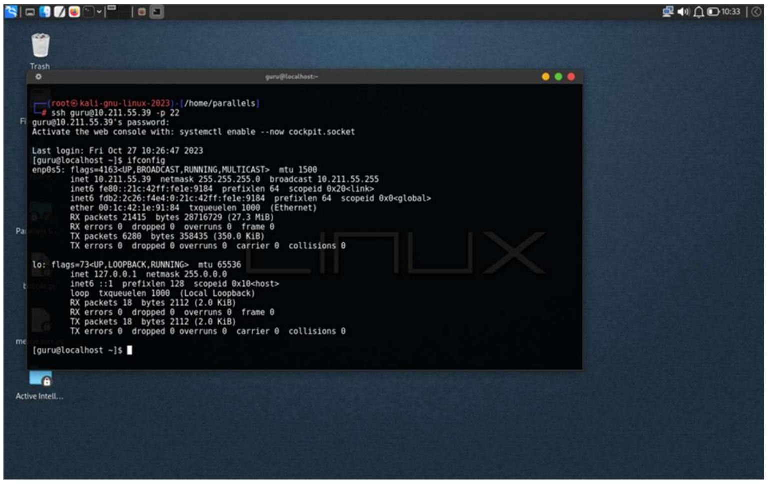
Kali Linux machine for testing purpose with IP of 10.211.55.27

We are going to scan for the open ports on our victim machine as you can see, we are on the same network from our Kali Linux machine.



```
┌──(root㉿kali-gnu-linux-2023)-[/home/parallels]
└─# nmap 10.211.55.39
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-27 10:30 KST
Nmap scan report for centos-linux.shared (10.211.55.39)
Host is up (0.00088s latency).
Not shown: 987 filtered tcp ports (no-response), 11 filtered tcp ports (admin-prohibited)
PORT     STATE  SERVICE
22/tcp   open   ssh
9090/tcp closed zeus-admin
MAC Address: 00:1C:42:1E:91:84 (Parallels)

Nmap done: 1 IP address (1 host up) scanned in 5.30 seconds

┌──(root㉿kali-gnu-linux-2023)-[/home/parallels]
└─# 
```

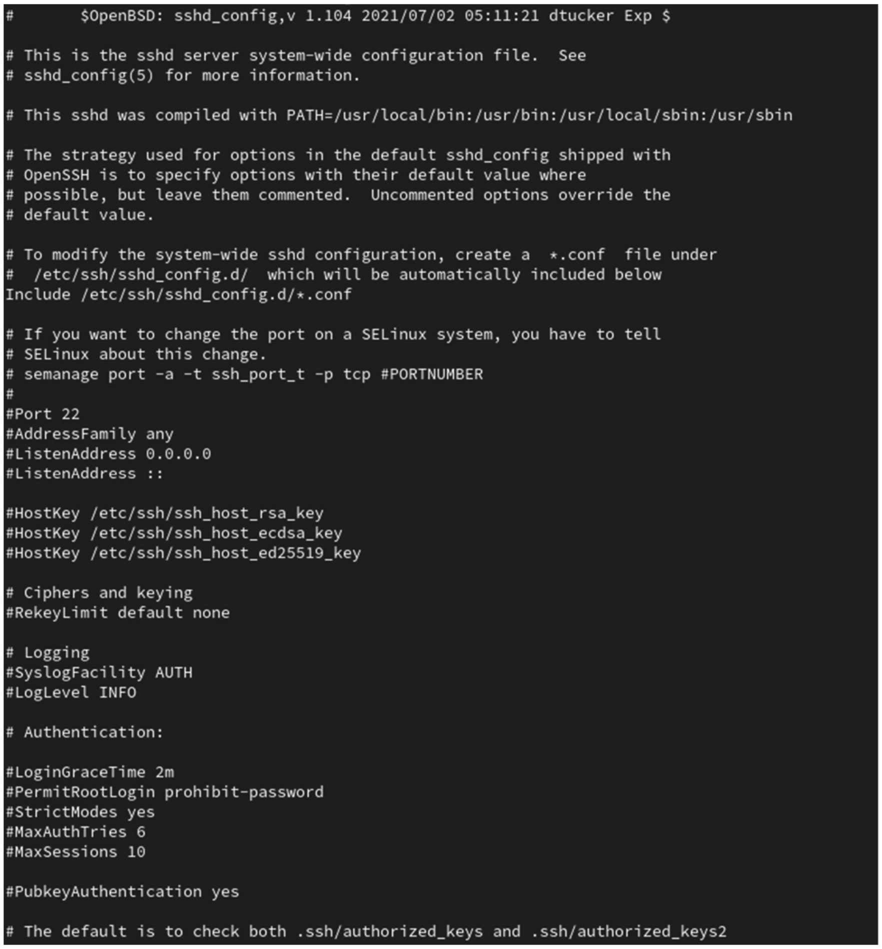
As you can see that the centos 9 has default port number ssh open which is a major security flaw, not let’s try to connect via ssh to our CentOS 9 machine by using ssh command.



```
┌──(root㉿kali-gnu-linux-2023)-[/home/parallels]
└─# ssh guru@10.211.55.39 -p 22
guru@10.211.55.39's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Oct 27 10:26:47 2023
[guru@localhost ~]$ ifconfig
enp0s5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.211.55.39  netmask 255.255.255.0  broadcast 10.211.55.255
        inet6 fe80::21c:42ff:fe1e:9184  prefixlen 64  scopeid 0x20<link>
        inet6 fdb2:2c26:f4e4:0:21c:42ff:fe1e:9184  prefixlen 64  scopeid 0x0<global>
        ether 00:1c:42:1e:91:84  txqueuelen 1000  (Ethernet)
        RX packets 21415  bytes 28716729 (27.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6280  bytes 358435 (350.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 18  bytes 2112 (2.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18  bytes 2112 (2.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[guru@localhost ~]$ 
```

As there was no password being applied on the Centos machine, we get an easy access to the centos machine without entering any password.

3.1. Checking for any misconfiguration in ssh file in CentOS /etc/ssh/sshd_config



```
#       $OpenBSD: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

# To modify the system-wide sshd configuration, create a  *.conf  file under
#  /etc/ssh/sshd_config.d/  which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
```

1. Port number: The default SSH port (Port 22) is noted. To use a different port for SSH, clear the Port line and specify the desired port number. Switching ports can increase security.

2. Root Login: Root Login (PermitRootLogin) is not set explicitly, but you must specify it if you want to disable or allow root access. Permitting direct root access is generally discouraged for security reasons.

3. Password authentication: PasswordAuthentication is set to ‘yes’, allowing password-based authentication. For increased security, consider changing this to ‘no’ and relying on public key authentication. However, be sure to configure two SSH keys for authorized users.

4. Key authentication: PubkeyAuthentication is commented out and defaults to 'yes', which is generally secure. This means that public key authentication is enabled. Ensure that the authorized keys for the user account are correctly configured.Authorized Keys File: AuthorizedKeysFile specifies .ssh/authorized_keys. Ensure that user's authorized keys are correctly located in their home directory in the .ssh folder.

5. LoginGraceTime: LoginGraceTime is commented out, which means it uses the default value (2 minutes). You may want to specify when you want a grace period.

6. Log Level: The LogLevel is set to 'INFO'. You can configure the logging level based on your needs for debugging or security auditing purposes.

7. IgnoreRhosts: IgnoreRhosts is set to 'yes', which is a good security practice because it prevents the user from reading .rhosts and .shosts files.

8. PermitEmptyPasswords: PermitEmptyPasswords is set to ‘no’, which is the recommended setting to prevent empty passwords.

9. UsePAM: UsePAM is set to 'no', which may not be necessary if you need PAM-based authentication or account/session authentication. Depending on your configuration, you may want to enable and configure PAM.

10. Subsystem SFTP: The SFTP subsystem is configured and points to /usr/libexec/openssh/sftp-server, which is the default configuration. You can change this if needed.

Understanding how it’s related with ARM architecture and why it’s important:

1. • SSH configuration on CentOS (Virtual Machine): The SSH configuration file you provide is associated with your CentOS virtual machine. It shows how the SSH service is configured and secured in a CentOS environment. This is important for remote access and security when accessing your CentOS virtual machine from a Mac.

2. • Parallels Desktop on Mac (Apple Silicon): Parallels Desktop allows you to run virtual machines on your Mac, including machines with different operating systems, such as CentOS. For your Mac with an M1 chip (Apple Silicon), it is important to use a virtualization solution such as Parallels Desktop that is compatible with ARM64 architectures, ensuring proper CentOS performance.

3. • ARM architecture and CentOS: Improperly configured SSH settings on your CentOS virtual machine can affect the overall security of your system. A proper SSH configuration is required for a secure connection between your Mac and the virtual machine. Significance of SSH Configuration:

Setting up SSH (Secure Shell) on a CentOS virtual machine hosted by Parallels Desktop on your Mac with an M1 chip is key for many reasons SSH is your virtual environment's primary means of secure communication with remote access.

- Remote access: SSH allows you to access and manage your CentOS virtual machine remotely from your Mac. This is particularly valuable for system management, troubleshooting, and software development purposes.

- Security: A proper SSH setting is required to ensure that only authorized users can access your virtual machine. This includes aspects such as password management, key-based authentication, and access control, all of which directly affect the security of your CentOS system.

- Data Transfer: SSH makes it easy to securely transfer data between your Mac and the virtual machine. This is important for services such as file sharing and data encryption, which protects your information from being intercepted or compromised.

4. Compatibility and Optimization of Parallels Desktop for ARM Architectures:

Running CentOS with the M1 chip on your Mac requires a virtualization solution that is compatible with the ARM64 architecture. Parallels Desktop is optimized for ARM systems and ensures a smooth virtualization experience. Here’s why this is important:

- Performance efficiency: Parallels Desktop makes your ARM-based Mac work more efficiently, using its resources more efficiently. This optimization results in a seamless experience when running CentOS on your M1-powered machines.

- Compatibility: Parallels Desktop is designed to run a variety of operating systems, including CentOS, on ARM architecture. It provides essential virtualization features that ensure compatibility and performance and differentiate between different architectures.

5. Emphasizing SSH Misconfigurations and Security Implications:

Misconfiguration of your CentOS virtual machine's SSH settings can have significant security implications, which can be magnified at runtime on ARM architectures. This is because ARM systems may have different security considerations. Security specifications include:

- Unauthorized access: If SSH is not properly configured, this can allow unauthorized access to your virtual machine. This can cause data corruption, system vulnerabilities, and possible malicious intent on your CentOS system.

- Data disclosure: Improper configuration may result in the disclosure of sensitive data, jeopardizing the confidentiality and integrity of the information stored on the virtual machine.

- Malicious attacks: An insecure SSH system can open the door to a variety of attacks, including brute-force attacks and man-in-the-middle (MITM) attacks ARM systems may not have the same security mechanisms as x86 systems, making them more vulnerable to an attack.

6. Challenges of Running CentOS on an ARM-Based System and SSH Configuration:

Running CentOS on an ARM-based system like your Mac M1 can present unique challenges due to architectural differences. To overcome these challenges, a proper SSH configuration is essential in a case like this:

- Architecture compatibility: ARM architecture is different from traditional x86 architectures. Make sure the SSH configuration is enabled for this setting is important for synchronization and performance.
- Security Considerations: ARM-based systems may have specific security considerations. SSH configuration must be optimized for these considerations to maintain a secure environment.
- Resource consumption: A proper SSH protocol helps to maintain resources, considering that ARM systems have different resource consumption policies compared to x86 systems

3.2. PREVENTION OF UNAUTHORIZED ACCESS

The configuration of the SSH file should be the first step to prevent unauthorized access by individuals or anyone, this is important because large enterprises now a days use parallels desktop and parallels support mac operating systems allowing virtualization in mac environment only in the 1990s.

Analyzing network packets using tools like Wireshark is a valuable exercise for network administrators and security professionals. It provides insight into the traffic flowing through the network and helps identify potential problems, anomalies, or security risks. When using Wireshark to analyze packets from Kali Linux machines, it is important to follow best practices to gain meaningful insights and ensure network security and integrity.

Solution:

To make the most of Wireshark's packet analysis capabilities, consider the following steps:

- Filtering: Wireshark captures more data. Use display filters to highlight specific packets of interest. For example, you can parse packets by a specific source or destination IP, protocol, or keyword. This makes the dataset smaller and more manageable.

- Deep packet inspection: Perform packet inspection at a granular level. Look for patterns, anomalies, and signs of a negative person. Wireshark's disconnects can provide detailed information about the protocol, helping you identify issues.
- Statistics: Use Wireshark's statistical tools to gain insights into network performance. Analyze trends, packet loss, delay, and bandwidth usage. This data is important for the optimal performance of the network.
- Security analytics: Wireshark is a valuable tool for identifying security breaches or vulnerabilities. Watch out for suspicious traffic, unexpected port scans, unauthorized login attempts, or any deviation from expected network behavior
- Good packet capture management: Ensure packet capture is done responsibly. Avoid capturing sensitive data such as passwords and be mindful of privacy and compliance rules.
- Documentation: Keep detailed records of your research. Document your findings, including packet capture, display filter settings, and any preventive actions.
- Collaboration: If you identify potential security risks or anomalies, collaborate with your security team to investigate in detail and take appropriate action.

- Regular monitoring: Use packet analysis regularly to continuously monitor the health and security of your network. This proactive approach can help you identify and address problems before they escalate.

4. CONCLUSIONS

In conclusion, this paper investigated the dynamic interplay of virtualization, ARM architecture, SSH configuration, and network packet analysis in the case of running CentOS on a Mac with an M1 chip, using Parallels Desktop This research this multi-faceted series shed light on several important features in virtualized ARM systems based while maintaining strong network security is important for users looking to customize their experience.

First of all, Parallels Desktop's compatibility and optimization for ARM architecture, as demonstrated on the Mac M1, demonstrates the critical role virtualization technology plays This efficient and versatile software for users can run operating systems with ease, extending the advantages of ARM-based devices. Such developments are particularly noteworthy given the evolving computing landscape, where ARM processing is gaining popularity in terms of energy efficiency and performance.

In this case, the SSH configuration of the CentOS virtual machine emerges as an important feature. It allows secure remote access and file transfer, ensuring that users can securely interact with their virtual environment. However, incorrect configuration in the SSH configuration can lead to vulnerabilities, resulting in security implications. These issues are especially important in an ARM framework, where security considerations differ from traditional frameworks. Incorrect SSH configuration can increase the security challenges inherent in running CentOS on ARM-based systems. Therefore, it's important to invest time and effort in setting up a well-built SSH protocol, strengthening network security, and downgrading potential threats.

In this study, the importance of packet analysis using tools such as Wireshark became apparent. Analyzing network packets provides a deeper understanding of network performance, potential security threats, and network anomalies. Responsible use of this powerful tool is essential to protect privacy and comply with the law when capturing and analyzing packets.

In conclusion, the proper combination of virtualization, ARM architecture, SSH configuration, and packet analysis opens up exciting possibilities in modern computing It enables users to leverage the power of ARM-based systems , and tightly protect their networks and data through careful SSH configuration and network packet inspection Layered nature: modern deep and complex computing, where easy integration and security measures lie presence is key to a successful and rich user experience As technology continues to evolve, the flexibility of users and researchers, creativity and hard work will continue to drive growth and innovation in that field in this dynamic.

Acknowledgements. I would like to express my heartfelt gratitude to all those who contributed to the successful completion of this research. I want to acknowledge the unwavering support of my advisor Dr. Saurabh Singh whose guidance and expertise were invaluable throughout this journey.

I am also thankful for the resources provided by Woosong University, which made this research possible. Furthermore, I extend my appreciation to my colleagues and peers who offered valuable insights and encouragement during this research endeavor. In compliance with transparency and ethical standards, I declare that there are no conflicts of interest associated with this research. This work was carried out without specific funding or sponsorship.

Acknowledgment: The work was supported by the Woosong University Academic Research Fund in 2024.

CRediT authorship contribution statement. Author 1: Methodology, Investigation, Funding acquisition. Author 2: Formal analysis.

Declaration of competing interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

Reference to a journal publication

1. Chakravarthi, V. S. (2019). System on Chip (SOC) design. In Springer eBooks. https://doi.org/10.1007/978-3-030-23049-4_2

Reference to a website:

2. Sengar, S. (2015, December 28). ARM Architecture Basics. Linux Kernel for Newbies. https://saurabhsengarblog.wordpress.com/2015/12/05/arm-architecture-basics/
3. Arjunan, A. N. (2023, February 23). What is Parallels Desktop for Mac and how does it work? MUO. https://www.makeuseof.com/tag/parallels-desktop-for-mac/
4. Introducing CentOS Stream 9 – Blog.CentOS.org. (n.d.). https://blog.centos.org/2021/12/introducing-centos-stream-9/
5. Yuanyuan, G. P. F. (2023, May 12). What is SSH? How does SSH work? - Huawei. Huawei. https://info.support.huawei.com/info-finder/encyclopedia/en/SSH.html
6. Yuanyuan, G. P. F. (2023b, May 12). What is SSH? How does SSH work? - Huawei. Huawei. https://info.support.huawei.com/info-finder/encyclopedia/en/SSH.html
7. Parallels Desktop commands 0.05% market share in Virtualization Platforms. (n.d.). https://enlyft.com/tech/products/parallels-desktop