Network Protection for Energy Control Systems

CSP004_C_E

PRESENTATION BY:

- **CRISTINA ALCARAZ**
  UNIVERSITY OF MALAGA, SPAIN

Acknowledgement

- *Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.*
- *Project Agreement no. 101083594*

Topic-3: Essential Protection for Energy Control Networks

Overview

- Overview of the main TCP/IP security protocols
- Endpoint protection, such as HMIs and servers
- Final remarks
- References and sources

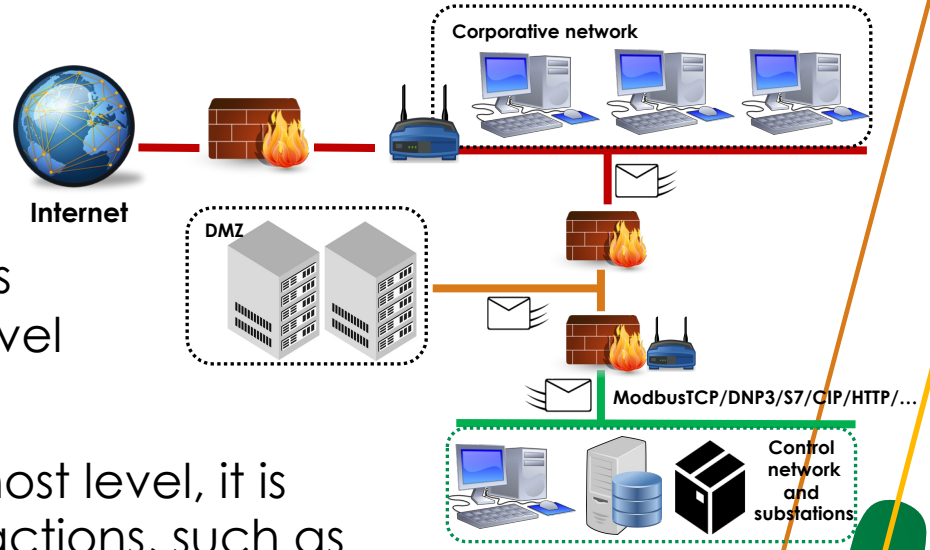
Topic-3: Essential Protection for Energy Control Networks

Overview

- Overview of the main TCP/IP security protocols
- **Endpoint protection, such as HMIs and servers**
- Final remarks
- References and sources

Domain and host level protection measures

- As illustrated in the figure, a SCADA system is usually composed by a set of networks and systems that require preventive measures at both network and host level



- Therefore, at domain and host level, it is essential to ensure a set of actions, such as
  - **Isolation and segregation of IT-OT domains**
  - **Inspection of systems in terms of vulnerabilities and services**
  - User account management and policy review
  - Activation of anti-malware systems
  - Regular updating of systems and application of patches
  - Remote attestation

In this module, we will only explore these two security measures and their implications for energy control networks and their hosts

ENDPOINT PROTECTION

Domain and host level protection measures – Isolation/segregation

- Isolation or segregation of IT-OT networks can be established by applying:
  - Diode communication - unidirectional communication channels
    - Synonymous: "data diode" = "unidirectional comm. gateway"
  - Firewalls
  - Virtual Local Area Networks (VLANs)
  - VPNs

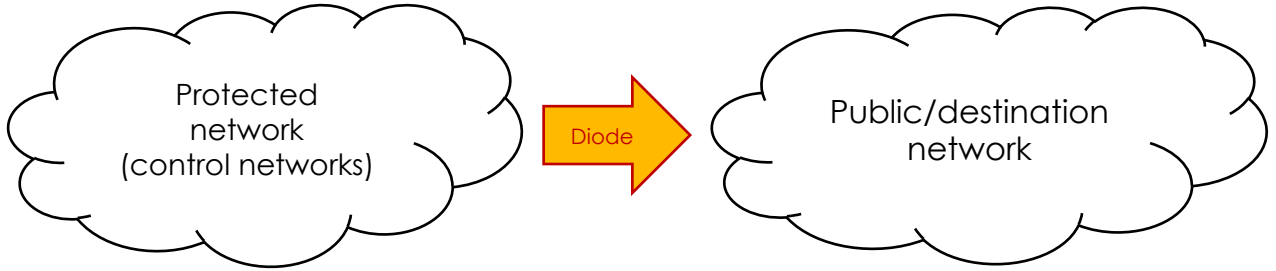
Diode | Firewalls | VLANs

Domain and host level protection measures – Isolation/segregation

- Isolation or segregation of IT-OT networks can be established by applying:
  - Diode communication - unidirectional communication channels
    - Synonymous: "data diode" = "unidirectional comm. gateway"
  - Firewalls
  - Virtual Local Area Networks (VLANs)
  - VPNs

Domain and host level protection measures – Isolation/segregation

- **Diode communication** is the way to enable the transfer of critical information in a single direction and under controlled actions

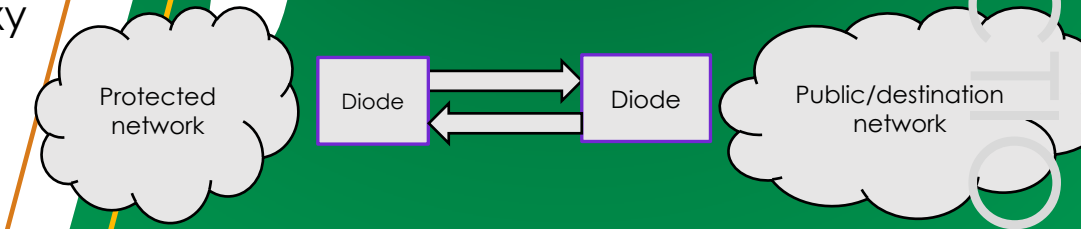


  - This type of communication is based on a specific HW device composed of two separate electronic circuits, such that:
    - (1) one of the circuits only receives packets from the protected network and, (2) the other only sends packets to the destination network

Domain and host level protection measures – Isolation/segregation

- However, diode communication are NOT effective for:
  - IT-OT networks where the source and destination network are in the same domain

  - Protection against cyber-attack vectors that flow in the sense of communication
- When the communication needs to be bidirectional:
  - It is necessary to configure the two data diodes with two proxy interfaces to process the incoming data on both sides of the network
  - The aim is to build separate data channels to help avoid the creation of **covert channels attacks**

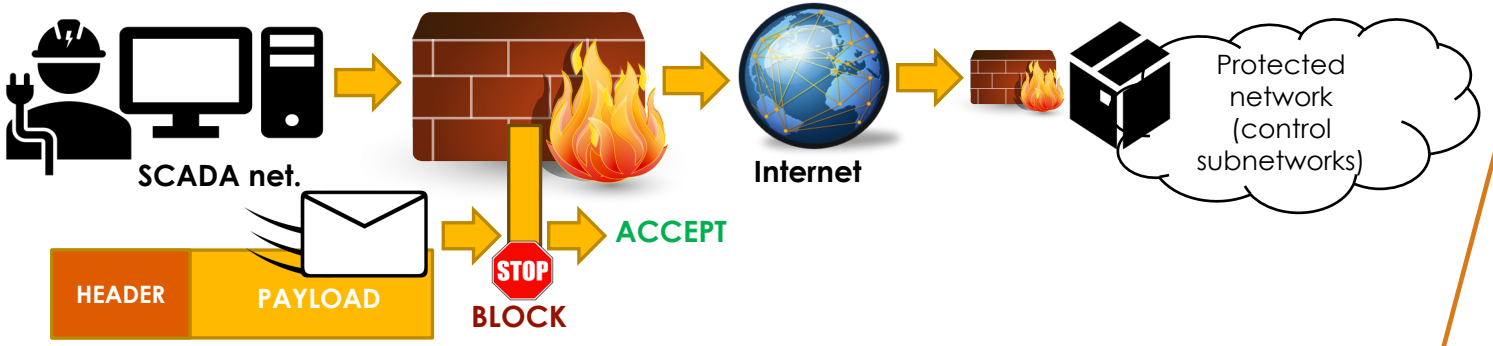


Domain and host level protection measures – Isolation/segregation

- Isolation or segregation of IT-OT networks can be established by applying:
  - Diode communication - unidirectional communication channels
    - Synonymous: "data diode" = "unidirectional comm. gateway"
  - Firewalls
  - Virtual Local Area Networks (VLANs)
  - VPNs

Domain and host level protection measures – Isolation/segregation

- **Firewalls** in industrial ecosystems have some support for interpreting multiple types of IIoT/CPS and industrial protocols, with appropriate levels of physical and electronic protection to be able to operate in industrial environments
  - Note that:
    - This level of interpretation of industry protocols remains limited ☹
    - This level of protection does not contemplate access control and VPNs ☹
    - Firewalls are developed in multiple devices such as routers, hosts, servers… ☺
- The purpose of a firewall is basically to accept or deny incoming or outgoing traffic according to established security policies

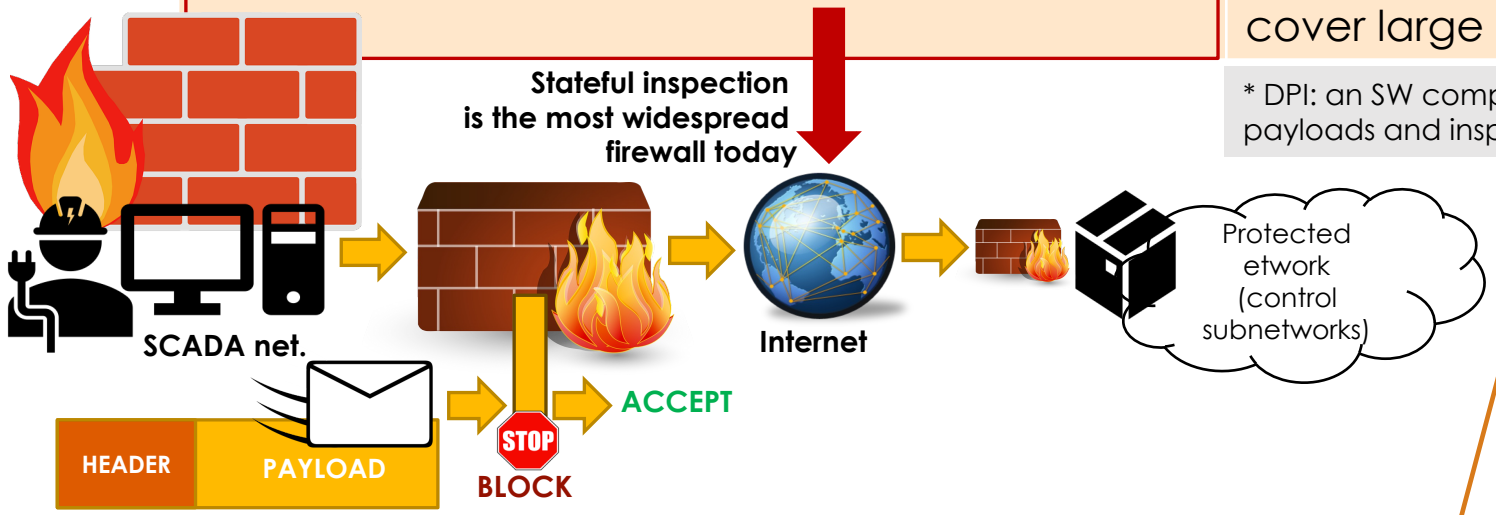


Domain and host level protection measures – Isolation/segregation

- There are diverse types of firewalls with:

Traditional firewalls with basic functions	Advanced firewalls with advanced functions
• ***Packet filtering***: to filter incoming or outcoming packets according to IPs, MAC, ports, protocols, interfaces, etc. • ***Stateful inspection***: similar to packet filtering but with the capability to recall sessions and statuses	• ***Unified Threat Management (UTM):*** to adapt advanced functions such as Deep Packet Inspection (DPI), antimalware or VPNs • ***Next-Generation FireWalls (NGFW)***: similar to UTM, but firewalls have more capacity to cover large environments

* DPI: an SW component capable of checking datagram payloads and inspecting the attack signatures in the payload



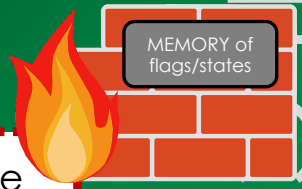
ENDPOINT PROTECTION

Domain and host level protection measures – Isolation/segregation

- **Packet filtering / stateless packet firewall** typically works at the network and transport layers
  - Filtering decisions are based on IP addresses, protocols, port numbers and interfaces predefined within the packet header
  - There are based on two filtering approaches (policies):
    - What is not expressly allowed is prohibited
    - What is not expressly prohibited is allowed

    → Typical Actions: ACCEPT, DROP (rejects but no notification), REJECT (rejects but notifies)
  - Its use may avoid specific attacks such as spoofing, TCP SYN attacks, Smurf attacks, flooding, etc.
- **Stateful inspection firewall** is capable of filtering packets like the packet filtering firewall but with the additional capability of controlling states
  - A packet could be part of a:
    - New connection: SYN (e.g. in the TCP 3-way handshake)
    - Part of an existing connection: SYN/ACK / ACK
    - Related with a connection: SYN/ACK / ACK

    → The stateful packet inspection model allows the firewall to maintain a record of the state of all conversations occurring "through" the firewall



ENDPOINT PROTECTION

Domain and host level protection measures – Isolation/segregation

- All firewalls are based on "**firewall policies**", and some recommendations for their specification can be found in NIST SP 800-41-rev1
  - Firewall policies should be prohibitive rather than permissive
  - Firewall policies should contemplate a set of protection conditions to ACCEPT / DROP traffic, such as: **IP addresses, types of protocols, network activity or type of firewall and its application**
- **IP address (IPv4 and IPv6)**: firewalls must block all outgoing traffic to a network
  - However, deciding which addresses should be blocked may be one of the most complex tasks for an IT/OT administrator
  - At the same time, it is also a critical task, as a bad decision may lead to multiple security risks
  - NIST SP 800-41-rev1 provides a number of recommendations in this respect

ACTION	RECOMMENDATIONS - NIST SP 800-41-rev1
ACCEPT (and LOG)	Only those valid and known incoming/outgoing IP addresses, as well as valid networks
DROP (and LOG)	Any traffic that has not been expressly permitted by the firewall policy
DROP (and LOG)	Only those invalid IP addresses relative to: • the localhost address as 127.0.0.0 (to 127.255.255.255.255) • the broadcast address to all networks as 0.0.0.0.0 • link-local addressed (e.g., 180.254.0.0)
DROP (and LOG)	Any invalid external address applying a pre-defined private network by RFC1918 "Address Allocation for Private Internets" There are private networks that should not be applicable for external addresses: • 10.0.0.0 to 10.255.255.255 (10.0.0.0/8) • 172.16.0.0 to 172.31.255.255 (172.16.0.0/12) • 192.168.0.0 to 192.168.255.255 (192.168.0.0/16)
DROP (and LOG)	Any external net. traffic containing "broadcast/multicast" addresses and directed to inside the net. This can cause many DoS
ACCEPT (and LOG)	Any ICMPv6 should be allowed – RFC 4890

Source: K. Scarfone and P. Hoffman, "Guidelines on Firewalls and Firewall Policy", NIST SP 800-41-rev1, NIST, Sept. 2009

Domain and host level protection measures – Isolation/segregation

- **TCP and UDP**

ACTION - PROTOCOLS	RECOMMENDATIONS - NIST SP 800-41-rev1
ACCEPT (and LOG)	Any TCP/UDP traffic that includes valid and required protocol ports
DROP (and LOG)	Any traffic toward an insecure or unnecessary TCP/UDP port for the internal network; or in other words, default policies (e.g., get access to the 80 port / 23 port) should be closed by the OS or rejected by the firewall
DROP (and LOG)	Any abuse or malformed UDP and TCP traffic should be blocked, but also reported as various ports can be scanned to conduct subsequent attacks

- **ICMP**

ACTION - DIAGNOSIS	RECOMMENDATIONS - NIST SP 800-41-rev1
ACCEPT (and LOG)	Any ICMP traffic that includes valid and required protocol ports
DROP (and LOG)	Any traffic with ICMP protocol but only if the the organization allows it since ICMP packets are very useful for diagnostics. Another solution would be to allow some ICMP traffic with specific codes for diagnostics (e.g. ping command for heartbeat - code 8)

- **VPN**

ACTION - PROTECTION	RECOMMENDATIONS - NIST SP 800-41-rev1
ACCEPT (and LOG)	Any VPN traffic should be allowed if the organization permits it, and only for the peers involved in the VPN communication

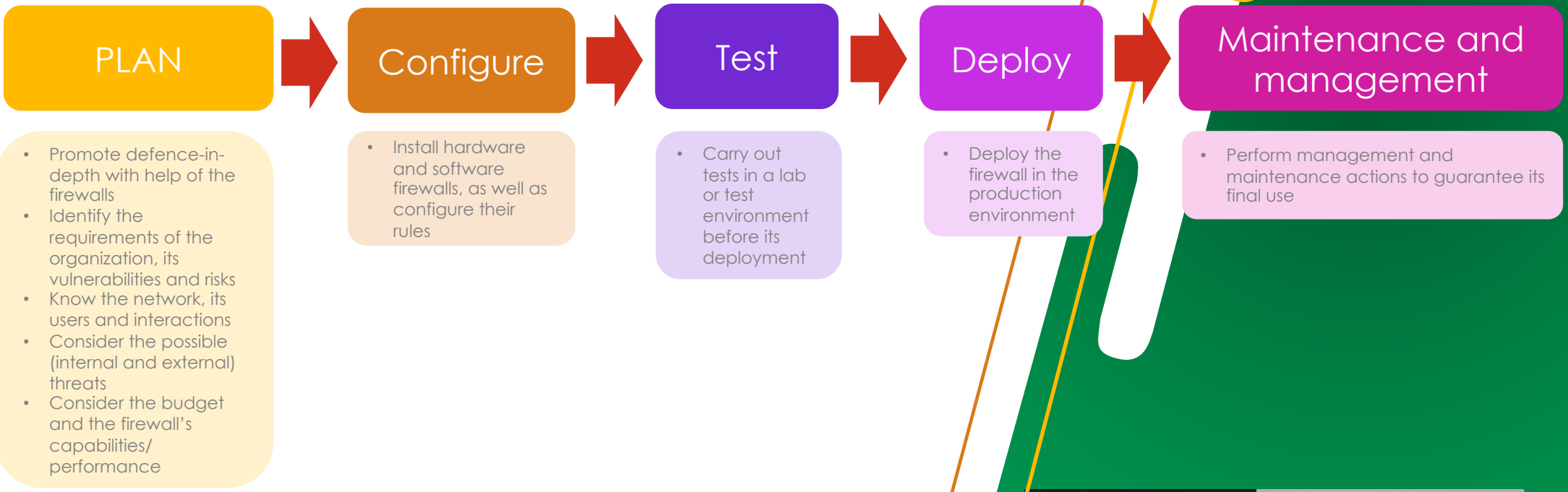
- **IT-OT Administrators** may accept or block traffic depending on the type of activity taking place on the network (e.g. access attempts, multi-port scanning)
  - This allows them to control and prevent potential threats

ACTION - ACTIVITY	DESCRIPTION
ACCEPT (and LOG)	Any traffic that is received within a pre-defined and reasoned timeframe. For example, if a connection is established and there is no subsequent traffic for an acceptable amount of time, the connection should be blocked
DROP (and LOG)	Any anomalous activity (e.g., as above) in a particular node, but also counting the number of times performing this anomalous activity
DROP (and LOG)	Any anomalous activity that exceeds the normal packet rate. For example, high incoming ICMP rates may mean a DoS

Source: K. Scarfone and P. Hoffman, "Guidelines on Firewalls and Firewall Policy", NIST SP 800-41-rev1, NIST, Sept. 2009

Domain and host level protection measures – Isolation/segregation

- Cyclical procedures must be taken to support the "good" security of a firewall system, especially when deployed in IT-OT ecosystems
- **Firewall lifecycle** includes:



Source: K. Scarfone and P. Hoffman, "Guidelines on Firewalls and Firewall Policy", NIST SP 800-41-rev1, NIST, Sept. 2009

Domain and host level protection measures – Isolation/segregation

- In the following, we will explore three relevant Linux firewalls

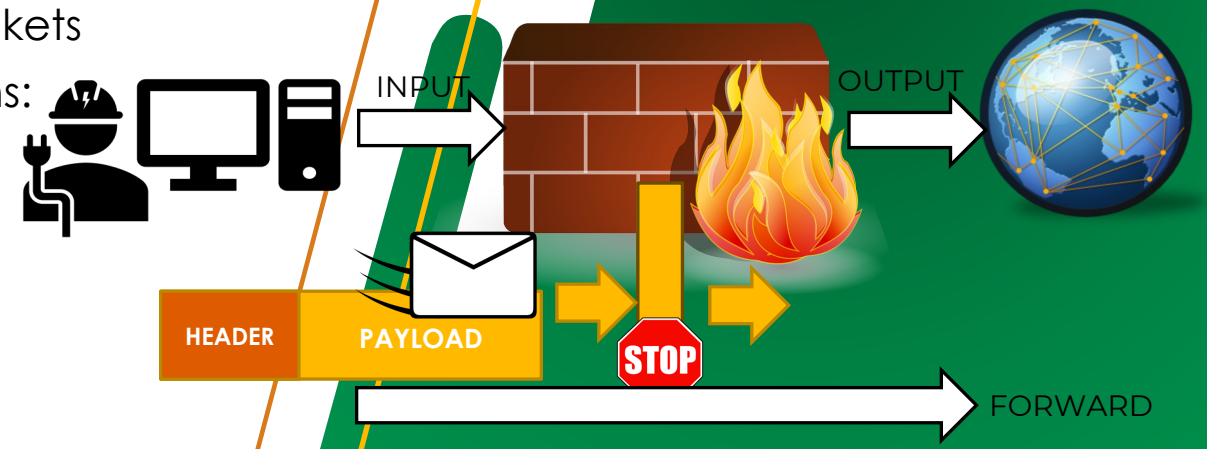
IPTables | Nftables / nft | ufw

Domain and host level protection measures – Isolation/segregation

- In the following, we will explore three relevant Linux firewalls

IPTables | Nftables / nft | ufw

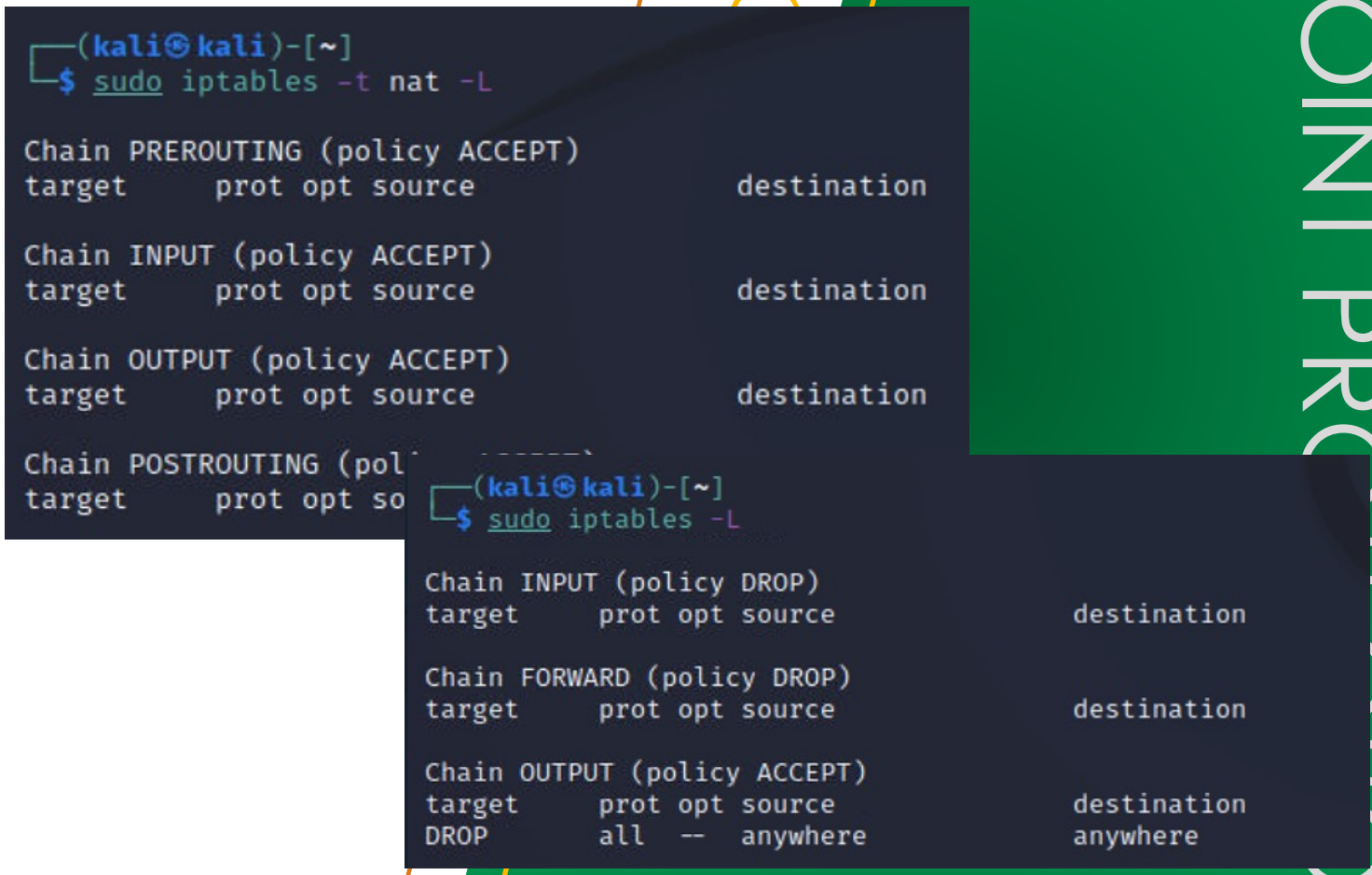
- **IPTables** is the kernel firewall of the Linux Operating System
  - It was implemented as part of the Netfilter project, which was defined to intercept and manipulate network packets
  - It is based on simple firewall rules for specific chains:
    - *INPUT*: incoming network traffic entering the firewall
    - *OUTPUT*: outgoing network traffic leaving the firewall
    - *FORWARD*: network traffic that is forwarded to other nodes



Source: Archlinux, iptables, 2024
URL: https://wiki.archlinux.org/title/iptables

Domain and host level protection measures – Isolation/segregation

- To list the basic rules:
  - **$ iptables –L**
- To list the rules associated with NAT:
  - **$ iptables -t nat -L**
- To delete all rules:
  - **$ iptables -t nat -F INPUT**
  - **$ iptables -t nat -F FORWARD**
  - **$ iptables -t nat -F OUTPUT**
  - **$ iptables -F**
  - **$ iptables -Z**
  - **$ iptables -X**
  - **$ iptables -P INPUT DROP**
  - **$ iptables -P OUTPUT DROP**
  - **$ iptables -P FORWARD DROP**
  - **$ iptables -t nat -P PREROUTING ACCEPT**
  - **$ iptables -t nat -P POSTROUTING ACCEPT**



Domain and host level protection measures – Isolation/segregation

- The IPTables rules follow the following structure:
  - ***$ iptable <operations> <string> <conditions> <action> <action>***

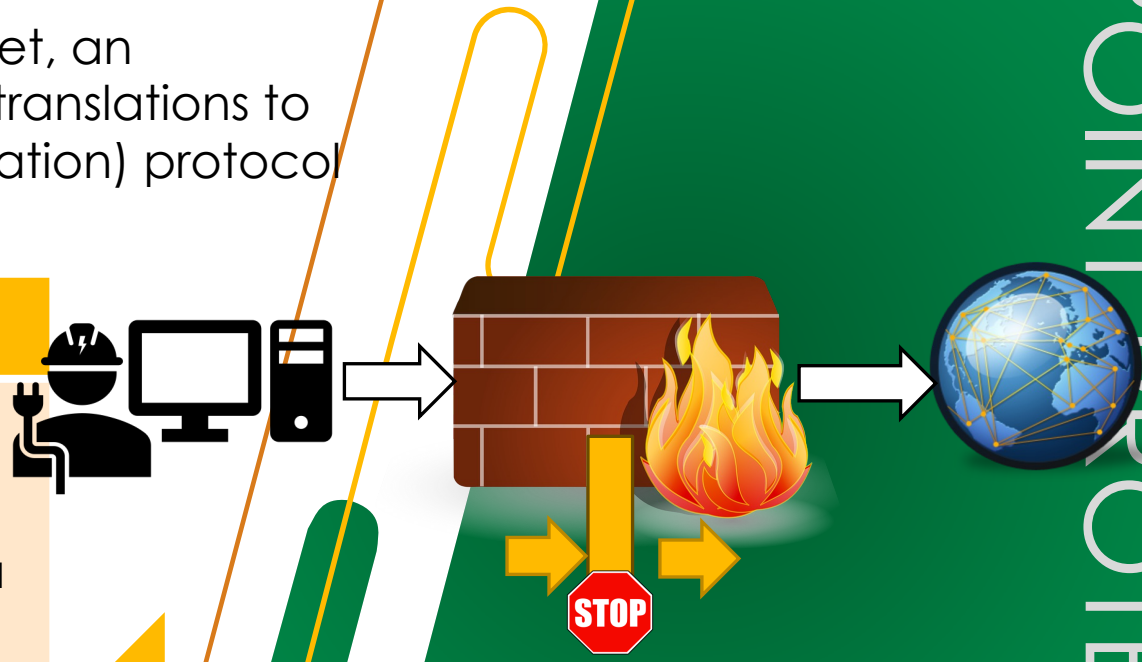
Main operations	Chain	Conditions	Actions
-A, --append: add the rule -I, --insert: insert at a specific position -D, --delete: delete the rule	INPUT OUTPUT FORWARD	-s, --source: the source IP address -d, --destination: the destination IP address -p, --protocol: the protocol to scan (tcp, udp, icmp, etc.) -sport, -dport: the source and destination ports -i, --in-interface: the incoming network interface (eth0, eth1,...) -o, --out-interface: the outgoing network interface (eth0, eth1,...)	ACCEPT: accept the network traffic DROP: drop the network traffic without notifying the sender of the deletion REJECT: drop the network traffic, notifying with ICMP the sender of the deletion

- Examples:
  - Deny incoming traffic and for a range of ports:
    $iptables -A INPUT -p tcp --dport 1: 689 DROP
  - Accept traffic from a specific IP:
    $iptables -A INPUT -s 12.11.112.11 -j ACCEPT

Domain and host level protection measures – Isolation/segregation

- When the packet comes from or goes to the Internet, an encapsulation process is required to allow network translations to be performed with the NAT (Network Address Translation) protocol of the firewall, so that:

INTERNET ➔ PREROUTING ➔ INPUT / FORWARD	OUTPUT / FORWARD ➔ POSTROUTING ➔ INTERNET
$iptables -t nat -A PREROUTING -p tcp --dport PORT -i eth0 -j DNAT --to IP-local • -j DNAT --to IP" change the destination IP address (with global IP) to IP (local IP address)	Option 1: $iptables -t nat -A POSTROUTING -j MASQUERADE Option 2: $iptables -t nat -A POSTROUTING -j SNAT --to IP-global • -j SNAT –to" allows to change the source address (local IP) to a Global IP address (equivalent to doing MASQUERADE)
E.g. $iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth0 -j DNAT --to 192.168.120.12	E.g. Option 1: $iptables -t nat -A POSTROUTING -j MASQUERADE



ENDPOINT PROTECTION

Domain and host level protection measures – Isolation/segregation

- In the following, we will explore three relevant Linux firewalls

IPTables | **Nftables / nft** | ufw

- **Nftables** was developed by Netfilter for Debian 10 (and subsequent versions) to solve many of the problems of IPTables
  - Some of these problem are for example:
    - Support rules for IPv4 and IPv6
    - Allows the creation of simple rules, in which firewall tables can be customised - although it is compatible with IPTable rules
  - Moreover, nft is an evolution of IPTables with support for debugging processes and monitoring of rules and states – thus, it is compatible with IPTables rules

nft is not enabled by default as IPTables, and would have to be installed: **$ sudo apt install nftables**, and the IPv4/v6 rules are normally stored in /etc/nftables.conf

Source: Archlinux, nftables, 2024
URL: https://wiki.archlinux.org/title/nftables

Domain and host level protection measures – Isolation/segregation

- To save IPv4/IPv6 rules in memory:
  - **$ sudo nft list ruleset > /etc/nftables.conf**
  - **$ sudo systemctl restart nftables.service**
- Empty a table:
  - **$ nft flush table [family] [name_table]**
- Create or delete a rule:
  - **$ nft add/delete table [family] [table_name],** where family corresponds to ip (corresponds to IPTables) ; ip6 (Ip6tables) ; inet (Iptables and ip6tables at the same time) ; arp (Arptables) ; bridge (ebtables)
- Add / insert a chain:
  - **$ nft add chain [family] [table] [chain]**
  - **$ nft insert rule [family] [table] [chain] handle [identifier] [declaration]**
- Delete a chain:
  - **$ nft flush chain [family] [table] [chain]**
  - **$ nft delete chain [family] [table] [chain]**
- Consult the rules and their tables:
  - **$ nft list ruleset**
  - **$ nft list table [family] [table] [chain]**

- Set up the postrouting table :
  - **$ nft add table nat**
  - **$ nft add chain nat prerouting { type nat hook prerouting priority 0 ; }**
  - **$ nft add chain nat postrouting { type nat hook postrouting priority 100 ; }**
  - Empty a table:
  - **$ nft flush table [family] [name_table]**
- Set up postrouting and prerouting tables, and masquerading:
  - **$ nft add table nat**
  - **$ nft add chain nat prerouting { type nat hook prerouting priority 0 ; }**
  - **$ nft add chain nat postrouting { type nat hook postrouting priority 100 ; }**
  - **$ nft add rule nat postrouting masquerade**
  - **$ nft add rule nat prerouting iif eth0 tcp dport { 80, 443, X, Y … } dnat IP-local**

- Traffic control (from the Internet to prerouting to INPUT/FORWARD), is controlled by the neftfilter's HOOK system
- This hook system consists of a connection tracking system integrated in the Neftfilter, capable of monitoring NAT conditions, queue management, traffic tracking (where it goes or where it comes from), etc.

Source: Archlinux, nftables, 2024
URL: https://wiki.archlinux.org/title/nftables

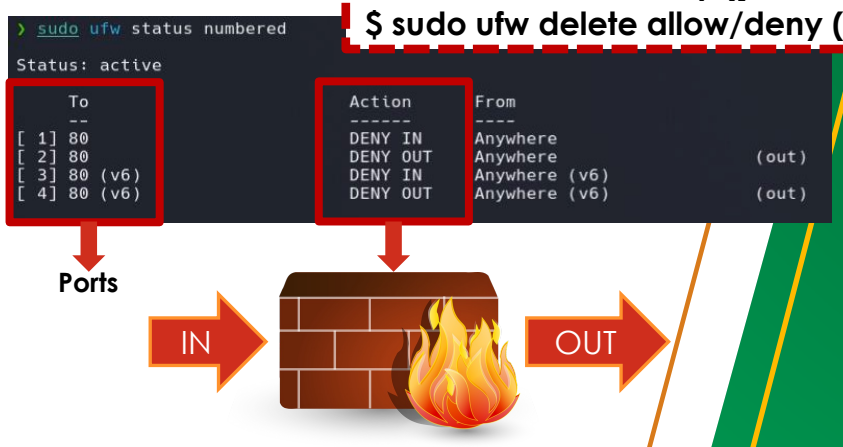
Domain and host level protection measures – Isolation/segregation

- In the following, we will explore three relevant Linux firewalls

IPTables | Nftables / nft | **ufw**

- **ufw** (uncomplicated firewall) is another Linux firewall
  - Install and enable the ufw packages:
    - **$ sudo apt install -y ufw**
    - **$ sudo systemctl enable ufw**
    - **$ sudo systemctl restart ufw**
    - **$ yes | sudo ufw enable**
  - Display the list of rules:
    - **$ sudo ufw status**
  - Reload and reset firewall rules:
    - **$ sudo ufw reload**
    - **$ sudo ufw reset**

Add or delete rules
$ sudo ufw allow/deny ([<PORT>/<PROTOCOL>][<SERVICES>][app <apps>])
$ sudo ufw delete allow/deny ([<PORT>/<PROTOCOL>][<SERVICES>][app <apps>])



Source: Ubuntu, UFW, Ubuntu documentation, 2024
URL: https://help.ubuntu.com/community/UFW

Homework: IPTable, nftable and ufw (for endpoints)

- **Task**: considering one of the virtual machines running on Linux, perform the following actions on **IPTable, nftable and ufw** (apply tool by tool):
  1. Disable ports 80 and 443, and check that it is not possible to make queries from the Internet browser
  2. Disable ports 23 and 20/21, and check that it is not possible to connect to a telnet and FTP server
  3. Deny any access from a local IP on the home LAN, as well as access to a specific web domain
  4. Accept ping requests from a specific IP on the home LAN, and deny access for all other IPs
  5. Restore the rules to their original state to regain connectivity

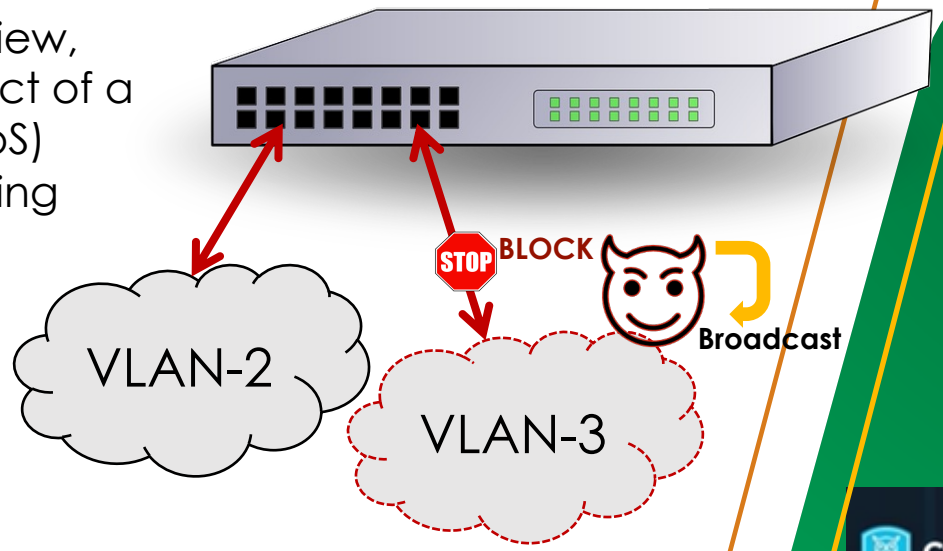
OPTIONAL PRACTICAL TASKS

Domain and host level protection measures – Isolation/segregation

- Isolation or segregation of IT-OT networks can be established by applying:
  - Diode communication - unidirectional communication channels
    - Synonymous: "data diode" = "unidirectional comm. gateway"
  - Firewalls
  - Virtual Local Area Networks (VLANs)
  - VPNs

Domain and host level protection measures – Isolation/segregation

- A **VLAN** aims to create independent virtual LANs within the same physical network
  - This requires configuring switches so that each port of the switches can host a virtual LAN
  - In other words, each port of the switch is reserved to host a virtual LAN, assigned to each VLAN with a unique identifier (ID)
  - The number of VLANs depends on the model of the equipment, but the maximum is 4096
  - From a security point of view, the VLAN delimits the effect of a possible broadcast ((D)DoS) to a VLAN without extending an entire network



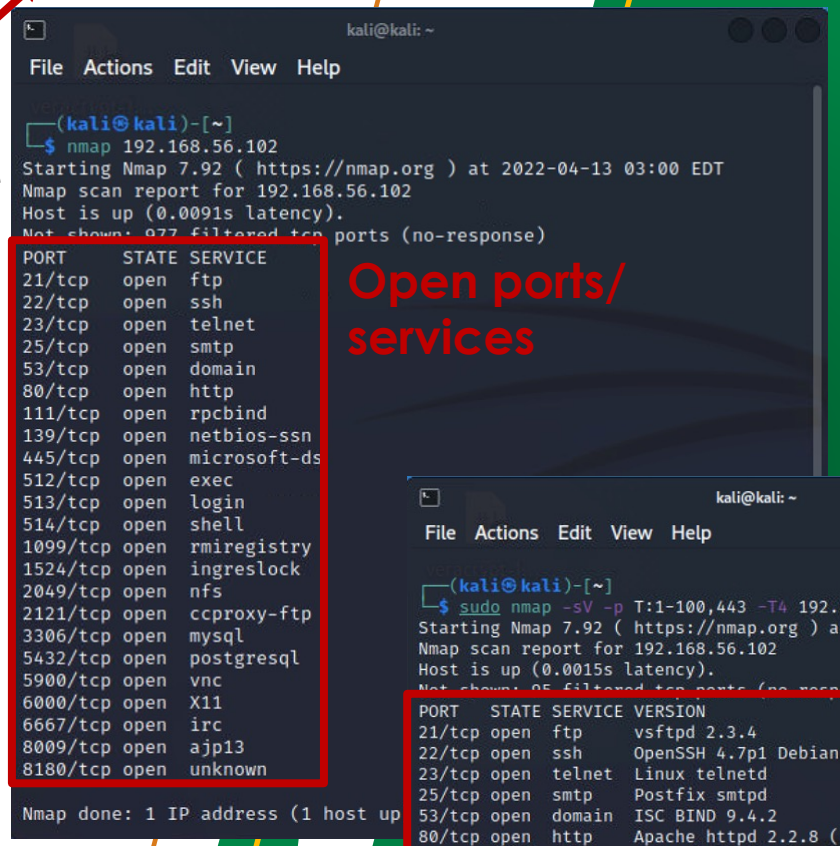
Domain and host level protection measures – Inspection

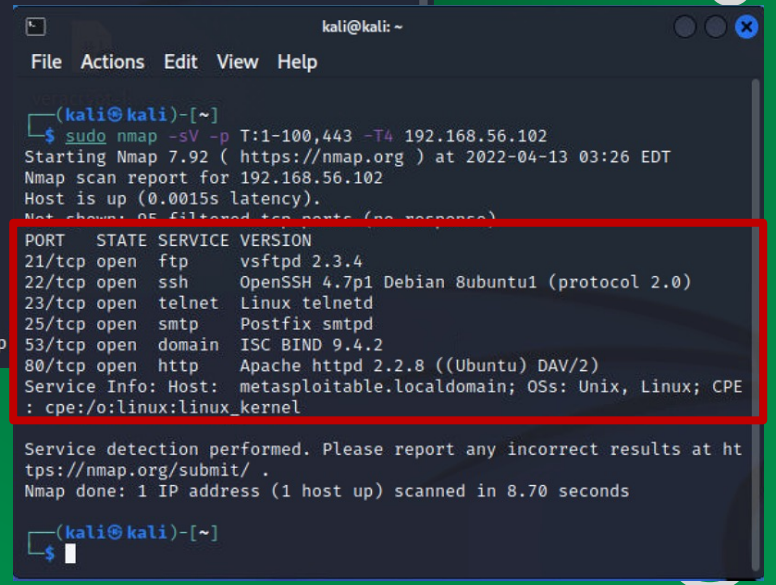
- Inspection of insecure or unnecessary ports and services, vulnerabilities (e.g. CVEs or zero-day vulnerabilities) and the state of network connections is a primary condition to prevent possible penetrations
- There are several tools that inspect the state of the ports and the existence of possible vulnerabilities, such as: **nmap, zenmap, OpenVAS or Nessus**
- **Nmap** (https://nmap.org) scans ports to determine which are open to penetration
  - It is compatible with Windows, Linux, MacOS and Solaris

Commands

$ nmap IP/localhost: open ports and services
$ nmap -sP IP/localhost: perform a simple ping
$ nmap -p T:1-65535 -T4 IP/localhost: ports in aggressive mode (-T4)
$ nmap -sS -T4 IP/localhost: TCP/SYN ports
$ nmap -sTU -T4 IP/localhost: TCP and UDP ports
$ nmap -p T:X-Y,Z -T4 IP/localhost: particular ports
$ nmap --top-ports X -T4 IP/localhost: scan the X most common ports
$ nmap -sV -T4 IP/localhost: ports, their services and versions
…

The main objective of an APT



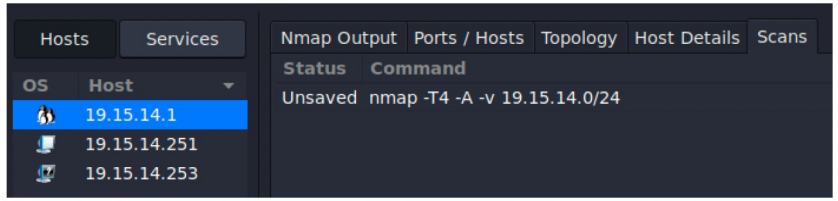


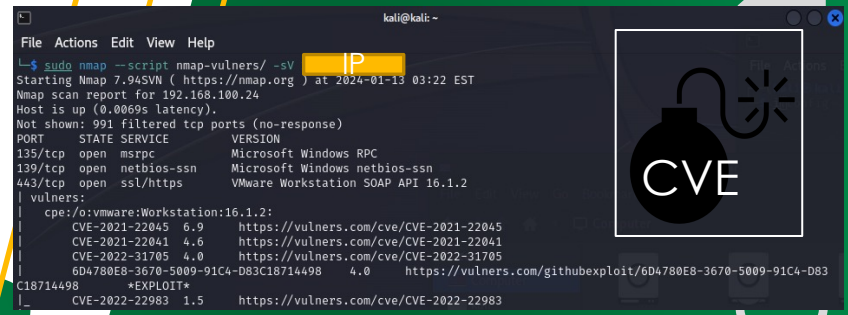
Domain and host level protection measures – Inspection

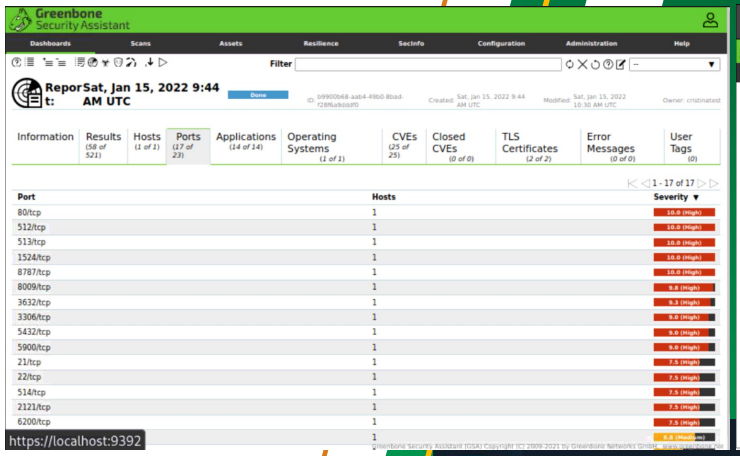
- **Zenmap** (https://nmap.org/zenmap/) is the version of nmap, but with an integrated GUI - it depicts the deployment of nodes, their ports and services within a specific network

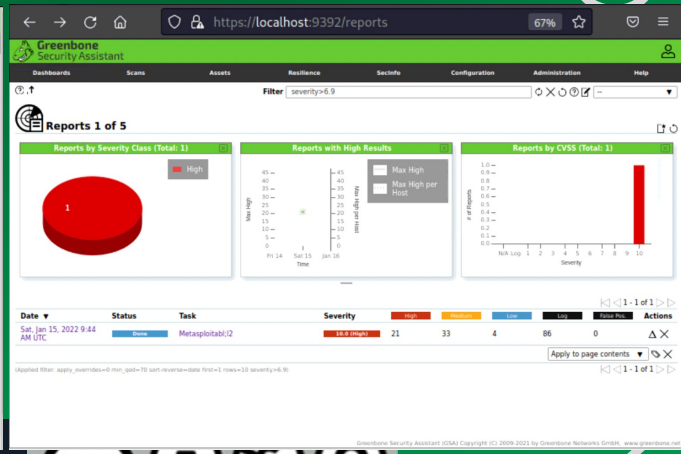
- **Nmap-vulners** (https://nmap.org/nsedoc/scripts/vulners.html) is a specific version of nmap that allows tracing vulnerabilities within a network or a node by querying locally stored CVEs to verify if they correspond to known vulnerabilities of type CVE-YYYY-NNNNN

- **OpenVAS** (https://www.openvas.org) is a network security scanner, based on a server with a set of network vulnerability tests to detect vulnerabilities in remote systems and applications







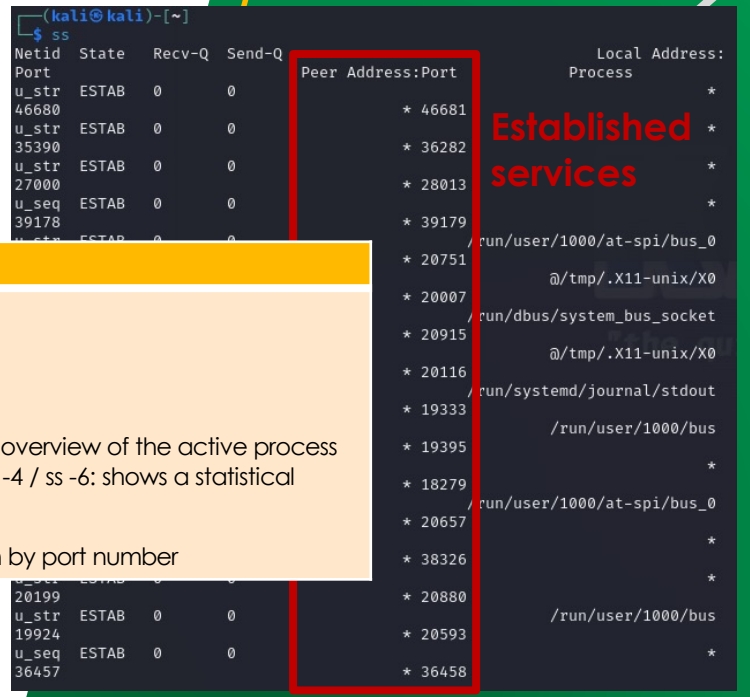


Domain and host level protection measures – Inspection

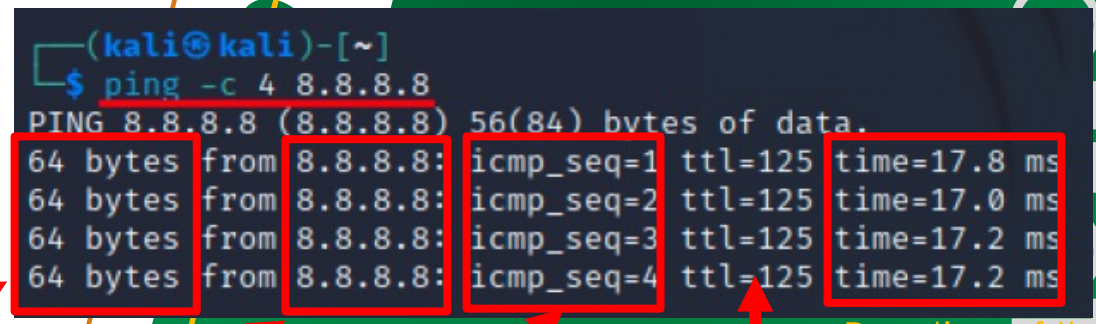
- **SS** (https://man7.org/linux/man-pages/man8/ss.8.html) replaces the traditional Linux netstat, providing information about the status of active connections within the network (e.g. ports or sockets)

Commands

$ ss: show all active connections
$ ss -a: show all ports
$ ss -l: show all active sockets
$ ss -t: show all TCP connections
$ ss -ua: show all UDP connections
$ ss -p: show the PID of the active process $ ss -s: show an overview of the active process
$ ss -s: shows a statistical summary of the connections $ ss -4 / ss -6: shows a statistical summary of connections
$ ss -4 / ss -6: shows IPv4 or IPv6 connections
$ ss -at (dport =: 21 or sport =: 21): displays the connection by port number



- **Ping** also focuses on verifying the liveness level of a node within the network, but also obtaining information about the time it takes for the packet to reach the target system and return, as well as reporting traffic losses
  - Basically, "ping" is an ICMP packet that sends an "echo request" to an endpoint and waits for an "echo reply"
  - **Fping** (https://fping.org) is an alternative tool to ping in charge of automatically tracing the nodes of a network - **$fping –a –g IP-IP > hosts.txt** (-a to display hosts, and -g to track a range of IPs)



Homework: Nmap, SS and Ping and Fping (for endpoints)

- **Task 1**. Considering the use of two VMs running on Linux and on GNS3 or home LAN:
  1. Install **Nmap** and scan all ports on the host and the other VM – take advantage of the option verbose **#nmap -O -v <target>**
  2. With **SS** check the active ports and compare the results with the data obtained from the previous scan
  3. Finally with **ping and fping**, check the status of other nodes in the network and practise using ping, taking into account the different options of the command

- **Task 2**. Explain why ping should not be disabled when this command is often used for DoS/DDoS attacks

OPTIONAL PRACTICAL TASKS

Homework: Nmap-vulners and CVSS 3.0/3.1 (for endpoints)

- **Task 1**. Considering the use of two VMs running on Linux and on GNS3 or home LAN:
  1. Open 3 insecure ports on one of the VMs using ufw/IPTables
  2. Run the **nmap-vulners** tool from another VM and trace all possible vulnerabilities of type CVE-YYYYY-NNNNN
  3. For each CVE found, investigate the characteristics of the vulnerability but taking into account the specific CVE repositories, and indicate the severity level according to CVSS 3.0/3.1 - https://nvd.nist.gov/vuln-metrics/cvss - https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator

CVSS (Common Vulnerability Scoring System) 3.0/3.1 is an online calculator that establishes metrics to compute the characteristics, impact and severity of vulnerabilities found, for example, in IT/OT devices

OPTIONAL PRACTICAL TASKS

Homework: OpenVAS (for endpoints)

- **Task 1**: considering the use of one of the Linux VMs running on GNS3 or home LAN:
  1. Install **OpenVAS** and the vulnerable VM **Metasploitable2** (https://github.com/rapid7/)
  2. Identify at least 1 network service (ports) with the highest risk and 1 service (port) with medium risk
- **Task 2**: from the vulnerabilities found:
  1. Extract and report (with your own words) their corresponding CVEs considering the existing trusted repositories, such as: https://www.cvedetails.com, https://cve.mitre.org, https://nvd.nist.gov
  2. Once network ports/services (of Task 1) have been identified, try to identify at least 2 recommendations to prevent the attack or vulnerability. To do so, it is necessary to consider the following recommendations:
     - NIST CSF 2.0
     - NIST IR 7628 Rev. 1
     - ENISA, "Appropriate security measures for smart grids. Guidelines to assess the sophistication of security measures implementation"

Maximum care with this MV, which is vulnerable per nature and should NOT be applied for personal use

OPTIONAL PRACTICAL TASKS

Source: NIST, "Cybersecurity Framework (CSF) 2.0 Reference Tool", 2024.
URL: https://csrc.nist.gov/Projects/Cybersecurity-Framework/Filters#/csf/filters
Source: NIST, "Guidelines for Smart Grid Cybersecurity", NIST IR 7628 Rev. 1, 2024.
URL: https://csrc.nist.gov/pubs/ir/7628/r1/final
Source: ENISA, "Appropriate security measures for smart grids. Guidelines to assess the sophistication of security measures implementation", 2012
URL: https://www.enisa.europa.eu/publications/appropriate-security-measures-for-smart-grids

Connect with CyberSecPro:
How to register and other practical information

1. Website: www.cybersecpro-project.eu
2. X (Twitter): https://twitter.com/CyberSecPro_eu
3. LinkedIn: https://www.linkedin.com/company/cybersecpro-euproject/

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.

Project Agreement no. 101083594

Thank you

If you have any questions, please do not hesitate to contact:

- Cristina Alcaraz
  Associate Professor
  University of Malaga
  alcaraz@uma.es