UNIVERSITY OF ŽILINA
Faculty of Management Science and Informatics

Prednáška 2
Prostriedky Linuxu ako podpora analýz v kybernetickej bezpečnosti

Riešenie bezpečnostných incidentov
(CyberOps Associate v1.02)

Mgr. Jana Uramová, PhD.
Katedra informačných sietí
Fakulta riadenia a informatiky, ŽU

Vytvorené v rámci projektu KEGA 026TUKE-4/2021.

Ktorý výsledok pokrýva táto prednáška

Výsledky vzdelávania

Študent po absolvovaní predmetu získa vedomosti a zručnosti potrebné na úspešné zvládnutie úloh, povinností a zodpovedností bezpečnostného analytika v operačnom centre bezpečnosti.
Študent po absolvovaní predmetu bude vedieť:

- Vysvetliť rolu analytika v rámci kybernetickej bezpečnosti
- Vysvetliť prostriedky operačného systému Windows a Linux a charakteristiky pre podporu analýzy v rámci kybernetickej bezpečnosti
- Analyzovať operácie v rámci sieťových protokolov a služieb
- Vysvetliť operácie sieťovej infraštruktúry
- Klasifikovať rôzne typy sieťových útokov
- Použiť sieťové monitorovacie nástroje na identifikáciu útokov proti sieťovým protokolom a službám
- Použiť rôzne metódy na prevenciu škodlivého prístupu do počítačových sietí, k používateľom a k dátam
- Vysvetliť vplyvy kryptografie v rámci monitorovania bezpečnostných sietí
- Vysvetliť, ako skúmať zraniteľnosti a útoky koncových zariadení
- Identifikovať hlásenia v rámci sieťovej bezpečnosti
- Analyzovať sieťovú prevádzku na overenie potencionálneho zneužitia siete
- Aplikovať reakčné modely na incident, a získať prostriedky na manažovanie sieťových bezpečnostných incidentov

- Prerekvizity:
  - Princípy IKS, Počítačové siete 1, Úvod do OS

Preliminary version of topics for lectures

Planning

Week	CyberOps Modules in lectures	Test from:
1	Chapter 1 The Danger Chapter 2 Fighters in the War Against Cybercrime Chapter 3: The Windows Operating System	none
2	Chapter 4: Linux Overview Chapter 5 Network Protocols Chapter 6 Ethernet and Internet Protocol (IP) Chapter 7 Connectivity Verification Chapter 8 Address Resolution Protocol Chapter 10 Network Services Chapter 11 Network Communication Devices	Lecture 1
3	Chapter 9 The Transport Layer (+nmap) Chapter 12 Network Security Infrastructure	Lecture 2
4	Chapter 13 Attackers and Their Tools Chapter 14 Common Threats and Attacks	Lecture 3
5	Chapter 15 Network Monitoring and Tools *(SIEM, SOAR)* Chapter 16 Attacking the Foundation *(L2, L3 protocols vulnerabilities and attacks)* Chapter 17 Attacking What We Do *(L7 vulnerabilities and attacks)*	Lecture 4
6	Chapter 18 Understanding Defense *(security management)* Chapter 19 Access Control *(AAA)* Chapter 20 Threat Intelligence *(commercials, CVE database)*	Lecture 5
7	Chapter 21 Cryptography Chapter 22 Endpoint Protection	Lecture 6
8	Chapter 23 Endpoint Vulnerability Assessment Chapter 24 Technologies and Protocols	Lecture 7
9	Chapter 25 Network Security Data Chapter 26 Evaualting Alerts (in Security Onion)	Lecture 8
10	Chapter 27 Working with Network Security Data (Security Onion and ELK) Chapter 28 Digital Forensics and Incident Analysis and Response	Lecture 9
11	Expert talk (invited lecture)	Lect. 10

Hodnotenie

Gradebook

Ongoing assessment during semester

10 Moodle test
3 points/test

+

1 bonus point
for invited lecture (L11)

=

30 points
for all tests (+1)

LABs 1-8:
24 points
3 points/LAB

+

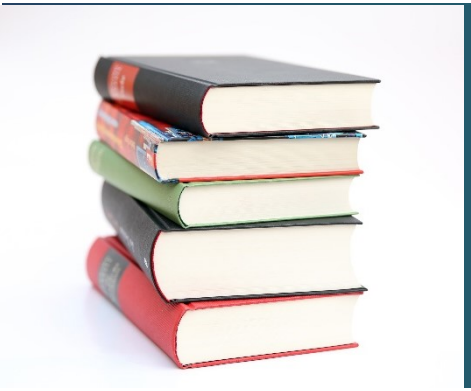
LABs 9-12
16 points
4 points/LAB

=

40 points
for all LABs

Overal grades

Formy a metódy hodnotenia	Váha %
Priebežné písomné testy	30%
Vyriešenie zadaných problémov v rámci praktických zadaní	40%
Záverečný písomný test	10%
Vyriešenie komplexného problému v rámci praktického zadania alebo projektu	20%



Content of this lecture

- **Chapter 4: Linux Overview**
  - Linux pre SOC analytika, opakovanie predmetu ÚdOS, a doplníme tému – Hardening Linuxu
- Opakovanie predmetu PIKS, len na domáce prečítanie z Netacadu:
  - **Chapter 5 Network Protocols**
  - **Chapter 6 Ethernet and Internet Protocol (IP)**
  - **Chapter 7 Connectivity Verification**
  - **Chapter 8 Address Resolution Protocol**
  - **Chapter 9 The Transport Layer (+nmap)**
  - **Chapter 10 Network Services**
  - **Chapter 11 Network Communication Devices**

Modul 4
Linux overview

Module Objective: Implement basic Linux security.

Topic Title	Topic Objective
Linux Basics	**Explain why Linux skills are essential for network security monitoring and investigation.**
Working in the Linux Shell	Use the Linux shell to manipulate text files.
Linux Servers and Clients	Explain how client-server networks function.
Basic Server Administration	**Explain how a Linux administrator locates and manipulates security log files.**
The Linux File System	Manage the Linux file system and permissions.
Working in the Linux GUI	Explain the basic components of the Linux GUI.
Working on a Linux Host	**Use tools to detect malware on a Linux host.**

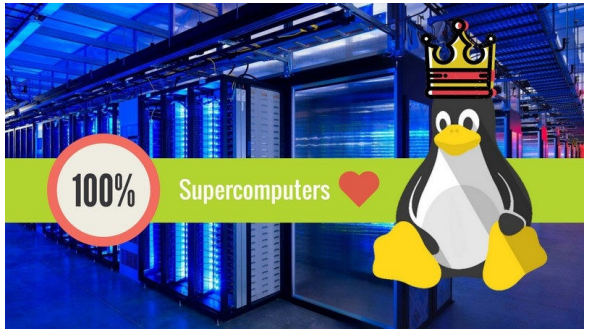
4.1 Linux Basics

What is Linux?

Linux smartwatch Olio Model One

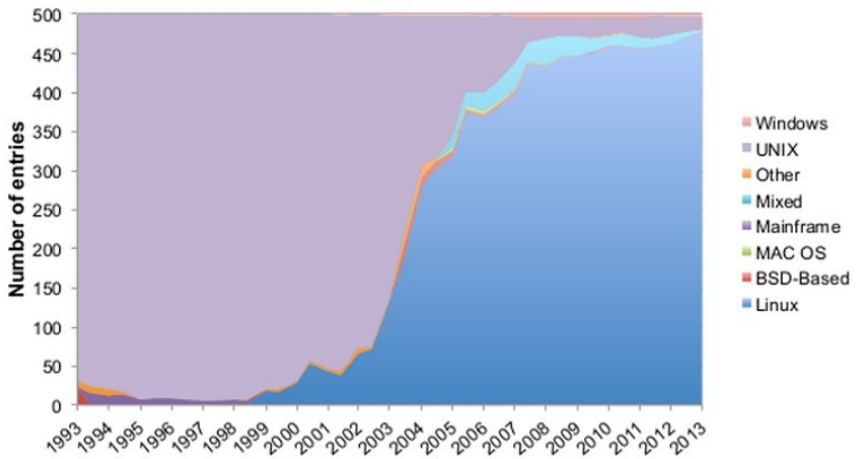
Ubuntu Watch Face for the Samsung Galaxy Watch ☺

- Linux is an operating system that was created in 1991.
- Linux is open source, fast, reliable, and small. It requires **very little hardware resources** to run and is **highly customizable**.
- Linux is **part of several platforms** and can be found on devices anywhere from **wristwatches** to **supercomputers**.
- Linux is designed to be connected to the network, which makes it much simpler to write and use network-based applications.
- A **Linux distribution** is the term used to describe packages created by different organizations and include the Linux kernel with customized tools and software packages.



Linux Runs on All of the Top 500 Supercomputers, Again!

- In 2012: 94%
- In 2013: 95%
- In 2014: 97%
- In 2015: 97.2%
- In 2016: 99.6%
- In 2017: 99.6%
- In 2018: 100%
- In 2019: 100%
- **In 2020: 100%**



Distribution of the 500 most powerful supercomputers worldwide from 2017 to 2022, by operating system

The Value of Linux

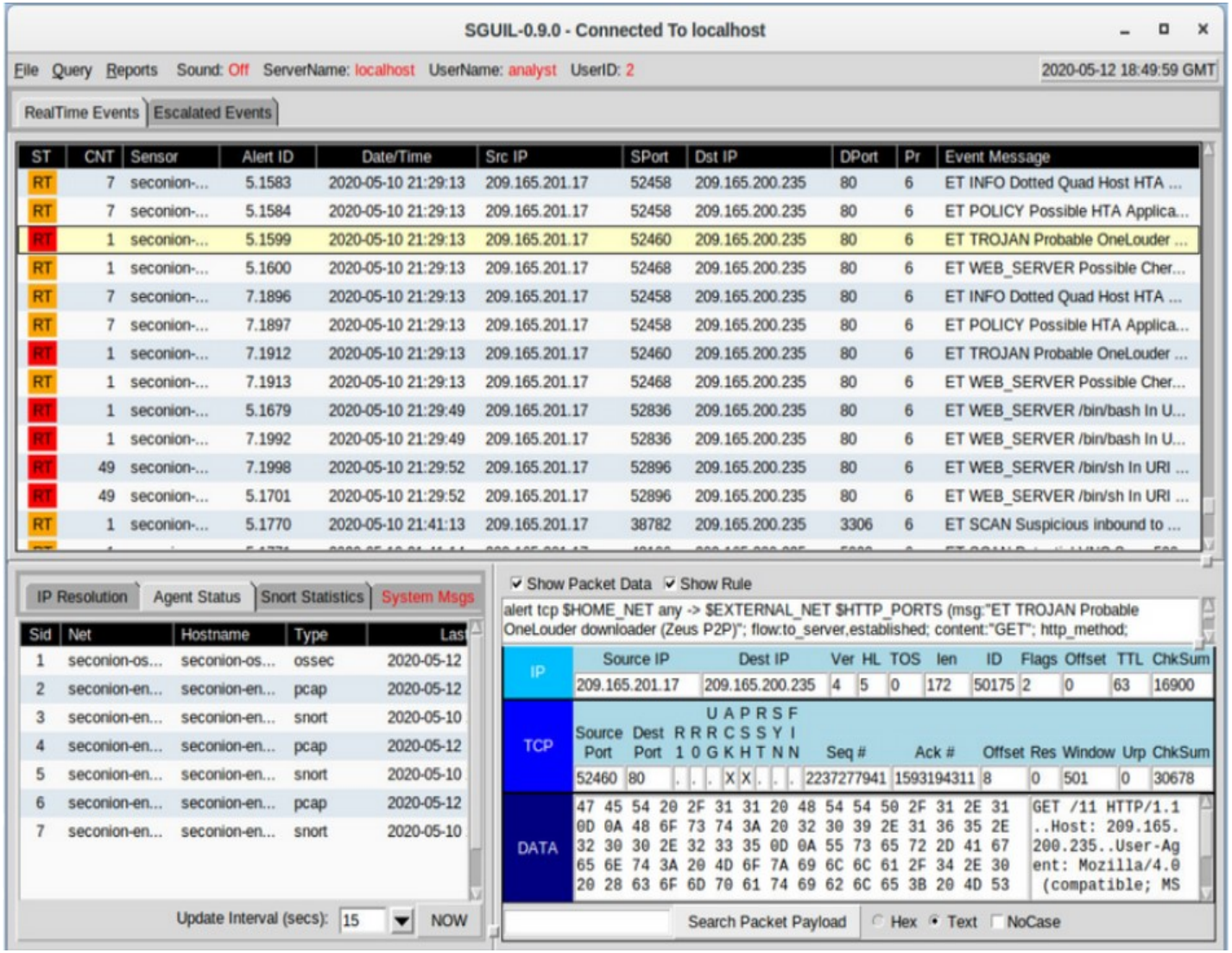
Linux is often the operating system of **choice in the SOC**

These are some of the reasons to choose Linux:

- **Linux is open source -** Any person can acquire Linux at no charge and modify it to fit specific needs.
- **The Linux CLI is very powerful -** The Linux Command Line Interface (CLI) is extremely powerful and enables analysts to perform tasks not only directly on a terminal, but also remotely.
- **The user has more control over the OS -** The administrator user in Linux, known as the root user, or superuser, can modify any aspect of the computer with a few keystrokes.
- **It allows for better network communication control -** Control is an inherent part of Linux.

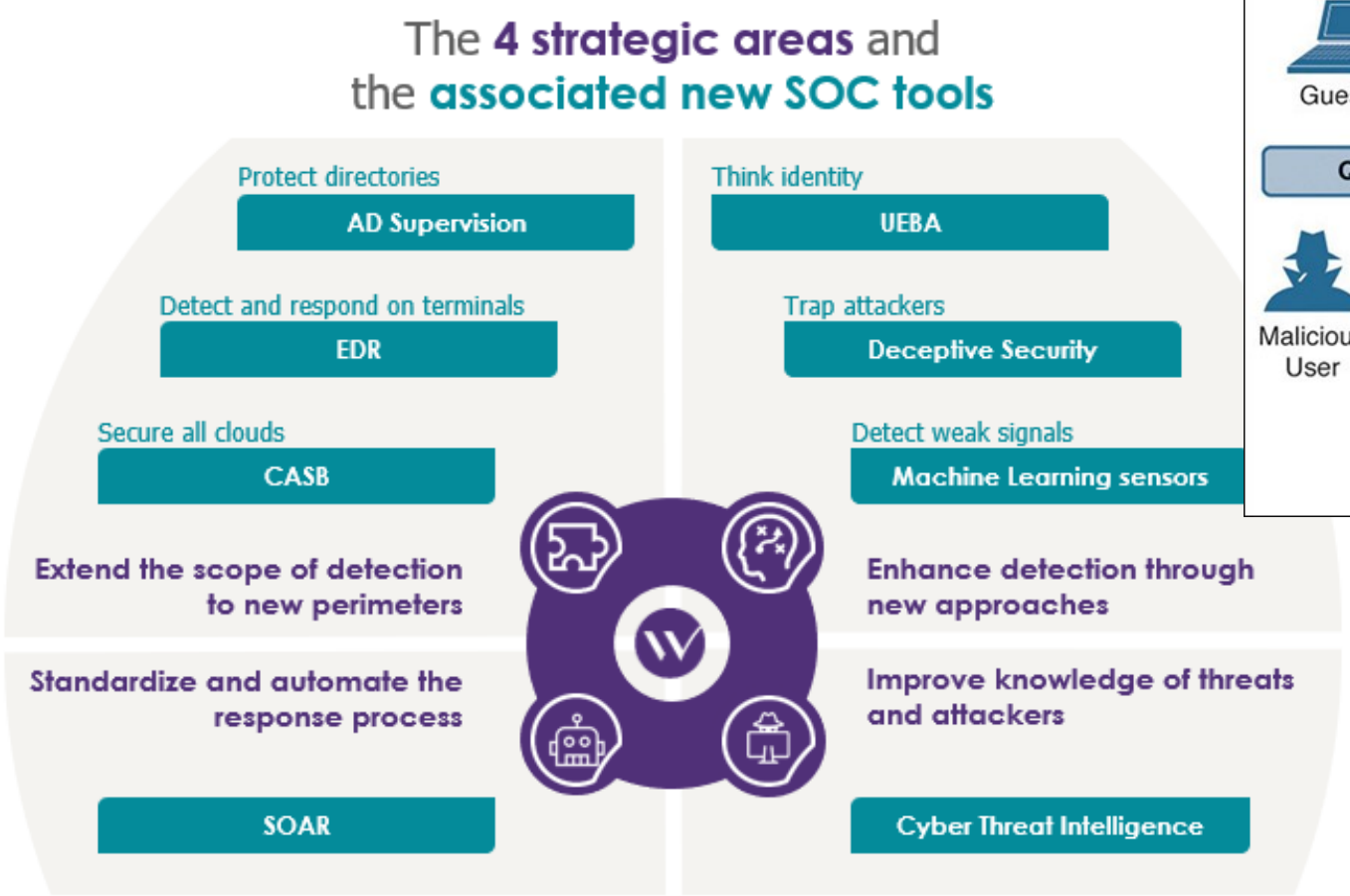
Linux in the SOC

- The **flexibility** provided by Linux is a great feature for the SOC.
- The entire OS can be **tailored** (*na mieru*) to become the perfect security analysis platform.
- **Sguil** is the cybersecurity analyst console **in a special version of Linux** called **Security Onion**.
- Security Onion is an **open source suite of tools** that work together for network security analysis.

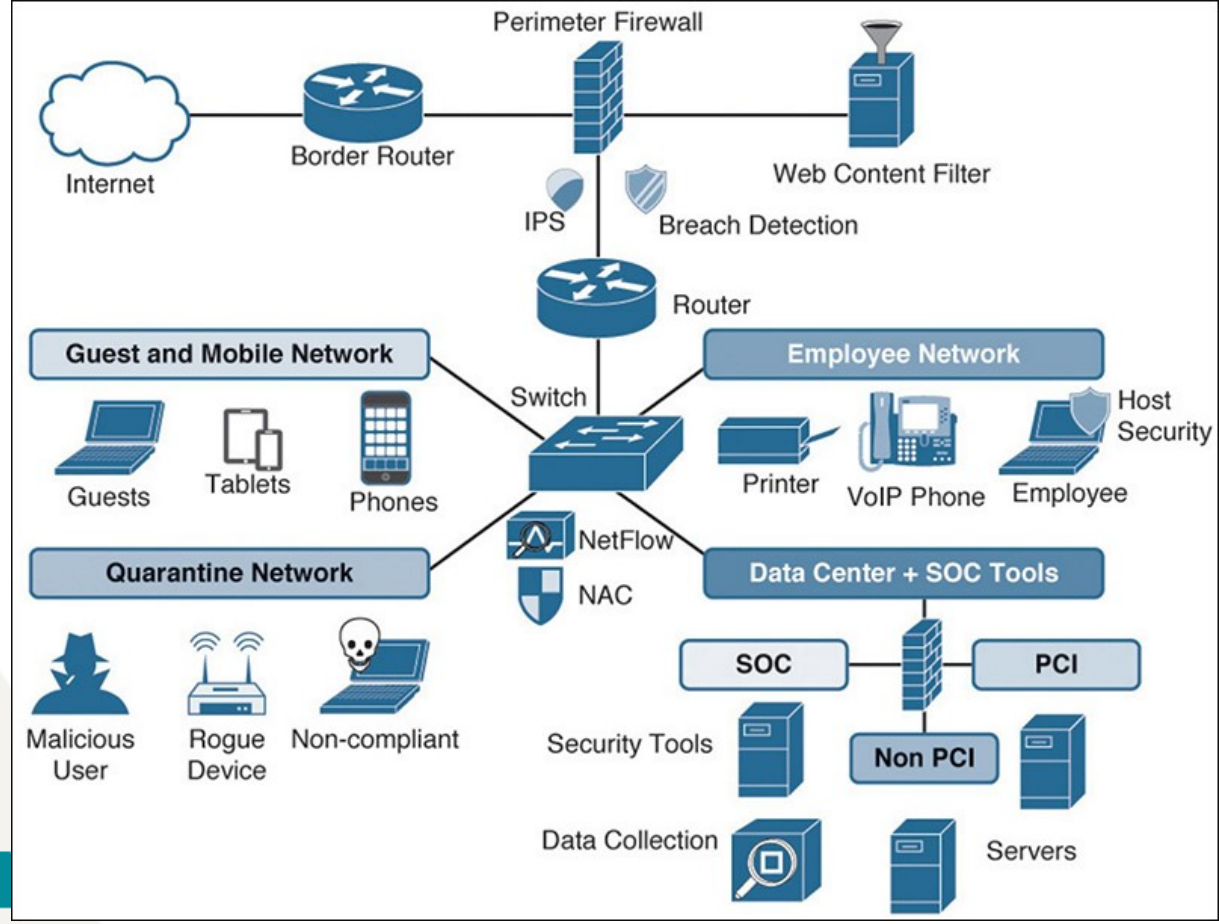


SOC tools

- Evolution...



2019, https://www.riskinsight-wavestone.com/en/2019/04/new-tools-soc-33/

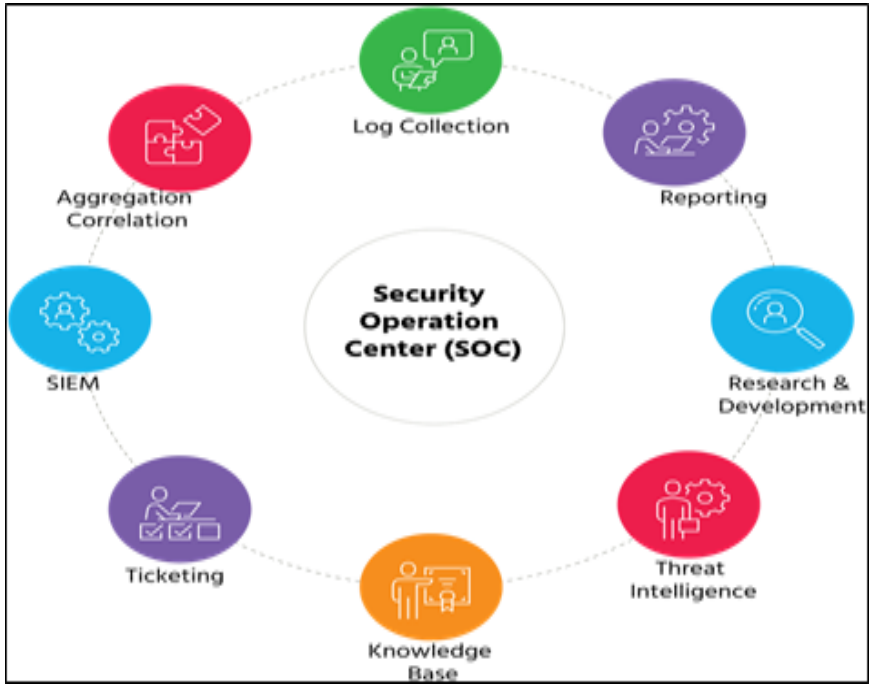


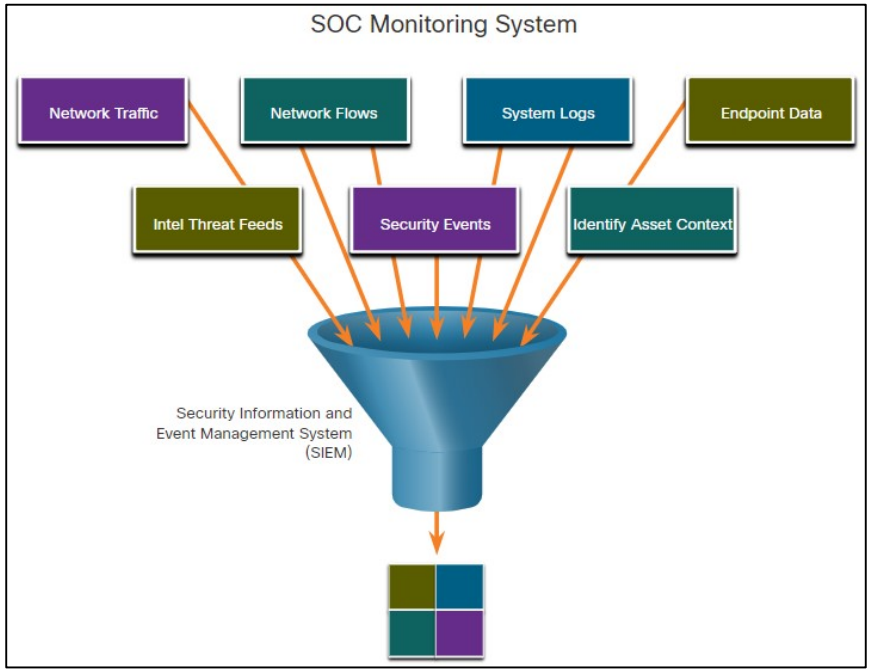
2015, Ciscopress.com

Linux in the SOC (Contd.)

The following table lists a few tools that are often found in a SOC:

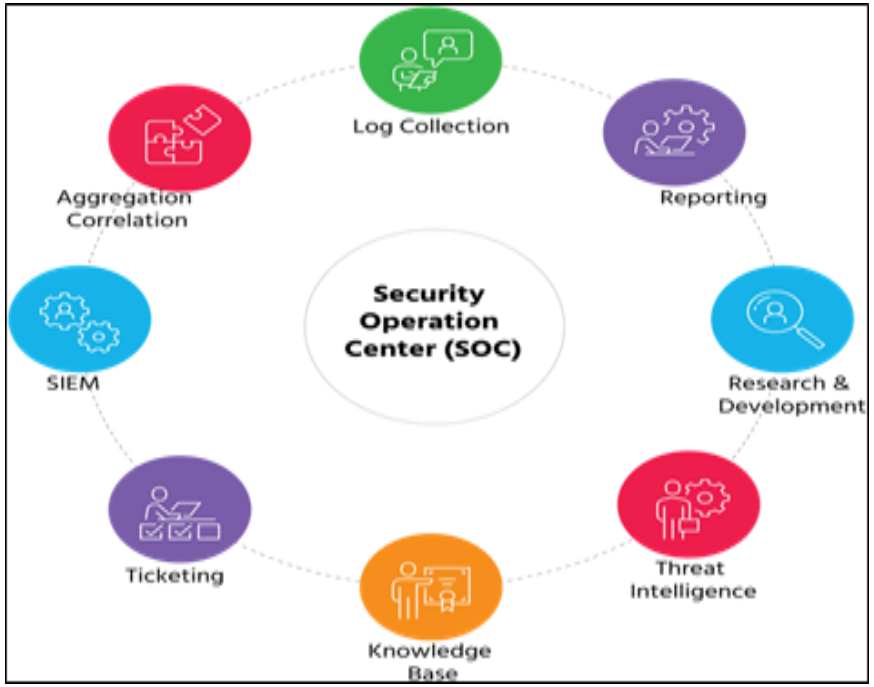
SOC Tool	Description
Network packet capture software	• A crucial tool for a SOC analyst as it makes it possible to observe and understand **every detail** of a network transaction. • Wireshark is a popular packet capture tool.
Log managers	• Log files are used to **record events**. • Because a network can generate a very large number of log entries, log manager software is employed to facilitate **log monitoring**.
Intrusion detection systems (IDSs)	• These tools are used for real-time traffic monitoring and inspection. • If any aspect of the currently flowing traffic matches any of the established rules, a pre-defined action is taken.

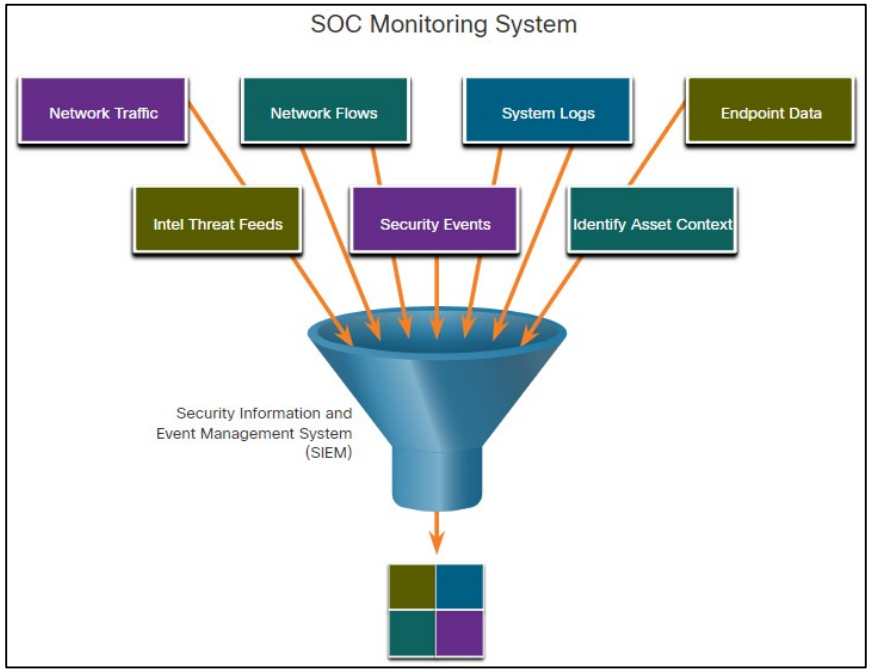




Linux in the SOC (Contd.)

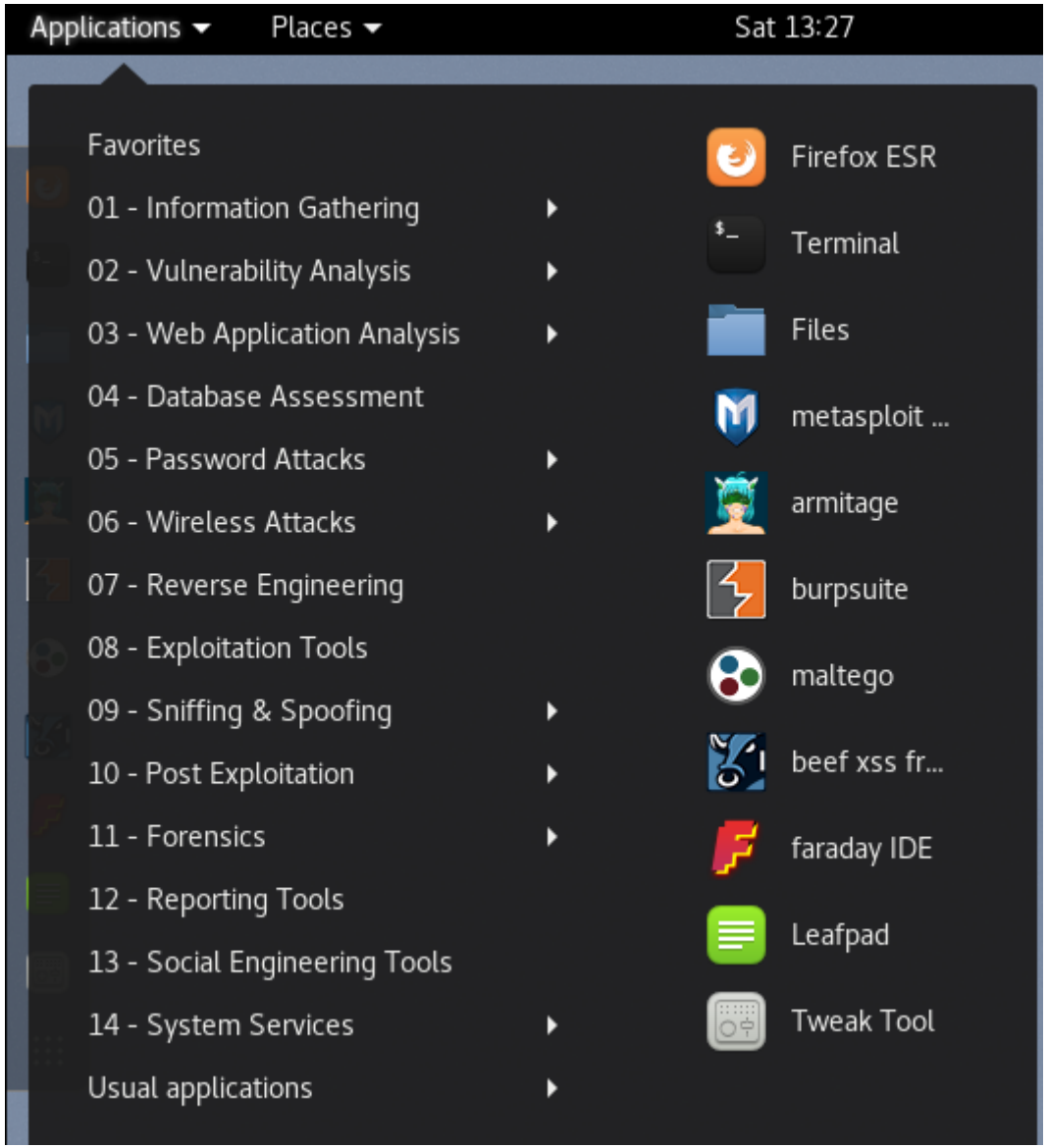
SOC Tool	Description
Firewalls	• This software is used to specify, based on pre-defined rules, whether traffic is allowed to enter or leave a network or device.
Security information and event management (SIEM)	• SIEMs provide **real-time analysis** of **alerts** and **log entries** generated by network appliances such as IDSs and firewalls.
Malware analysis tools	• These tools allow analysts to safely **run and observe malware execution** without the risk of compromising the underlying system.
Ticketing systems	• Task ticket assignment, editing, and recording is done through a ticket management system. Security alerts are often assigned to analysts through a ticketing system.





Linux Tools used in SOC

- Linux computers that are used in the SOC often contain **penetration testing tools**.
- A penetration test, also known as PenTesting, is the process of **looking for vulnerabilities** in a network or computer by attacking it.
- **Packet generators**, **port scanners**, and **proof-of-concept exploits** are examples of PenTesting tools.
- **Kali Linux** is a Linux distribution which contains many **penetration tools** together in a single Linux distribution.
- Notice all the major categories of penetration testing tools of Kali Linux.

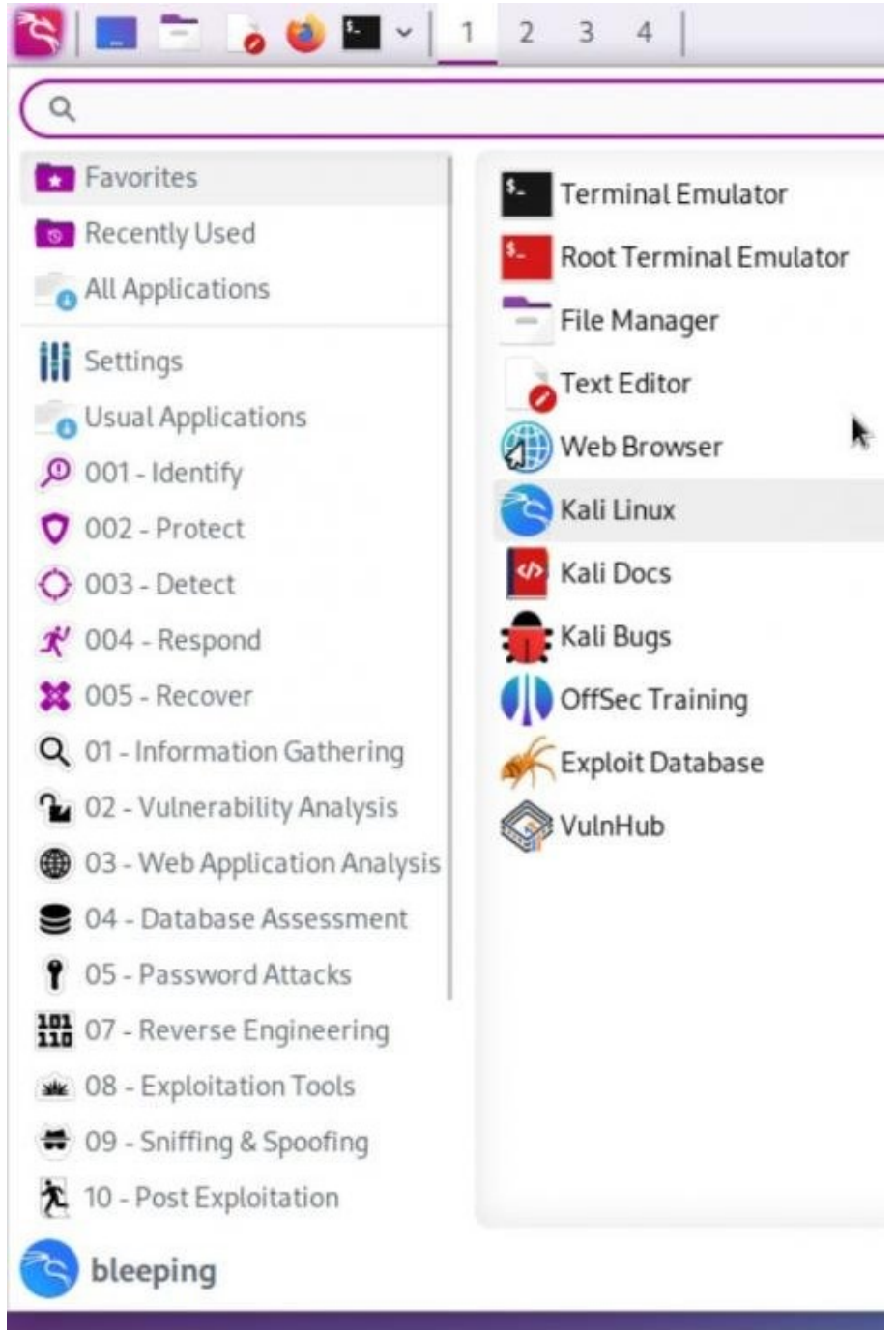


Linux Overview

Linux Tools used in SOC

With Kali Purple

- It is able to perform many of the **same labs** and testing scenarios as you can with Kali Linux.
- However, the focus of Kali Purple is on making it easier for you to **get started** and to **focus** on the most important aspects of **security testing** and **assessment**.



4.2 Working in the Linux Shell

Známe veci z predmetu UdOS (len opakovanie na doma)

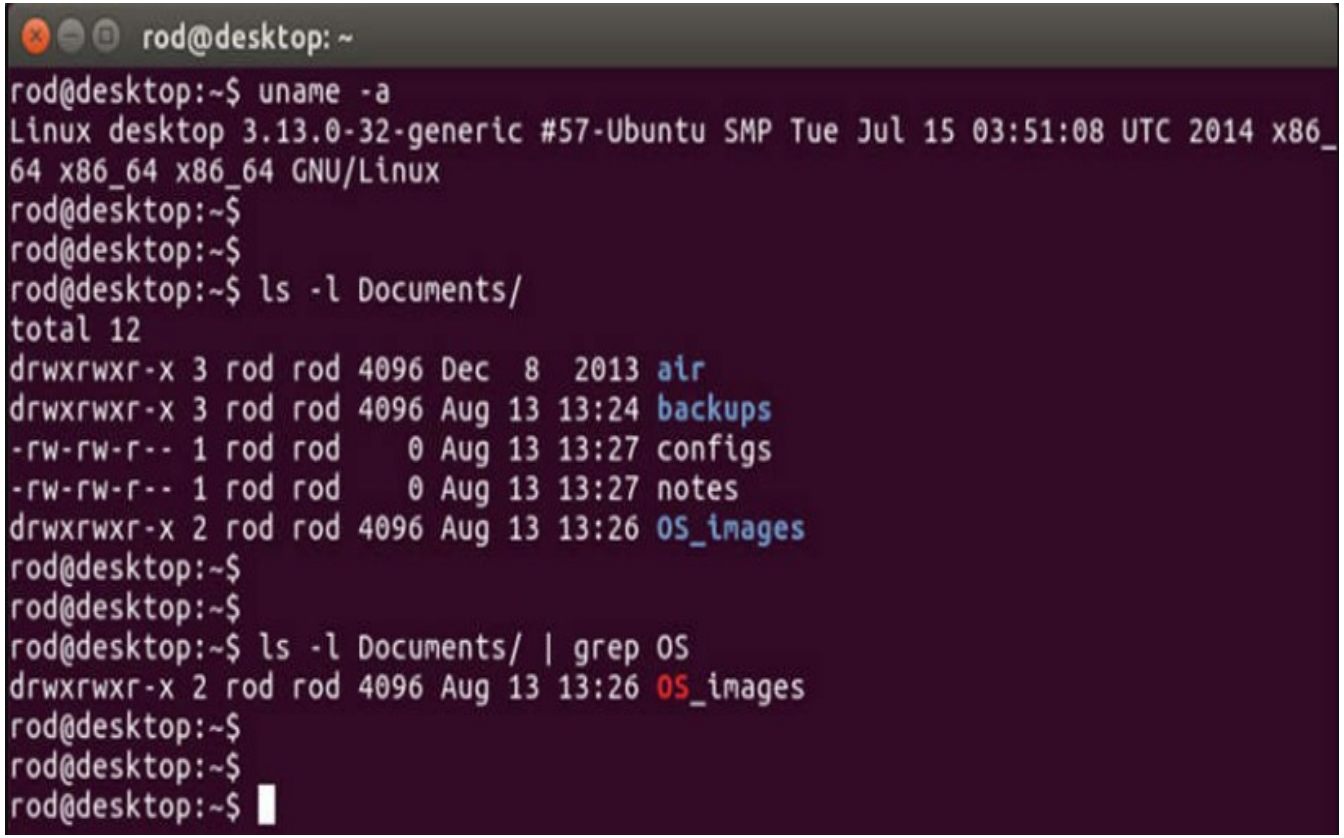
The Linux Shell

- In Linux, the user communicates with the OS by using the CLI or the GUI.
- Linux often starts in the GUI by default. This hides the CLI from the user.
- One way to access the CLI from the GUI is through a terminal emulator application. These applications provide user access to the CLI and are named as some variation of the word terminal.
- In Linux, popular terminal emulators are **Terminator, eterm, xterm, konsole,** and **gnome-terminal**.
- Fabrice Bellard has created JSLinux which allows an emulated version of Linux to run in a browser.

Note: ***The terms shell, console, console window, CLI terminal, and terminal window are often used interchangeably.***

The Linux Shell (Contd.)

The figure shows gnome-terminal, a popular Linux terminal emulator.



Basic Commands

- Linux commands are **programs** created to perform a specific task.
- As the commands are programs stored on the disk, when a user types a command, the **shell must find it on the disk** before it can be executed.
- The following table lists basic Linux commands and their functions:

Command	Description
mv	Moves or renames files and directories.
chmod	Modifies file permissions.
chown	Changes the ownership of a file.
dd	Copies data from an input to an output.
pwd	Displays the name of the current directory.
ps	Lists the processes that are currently running in the system.
su	Simulates a login as another user or to become a superuser.

Basic Commands (Contd.)

Command	Description
sudo	Runs a command as a super user, by default, or another named user.
grep	Used to search for specific strings of characters within a file or other command outputs.
ifconfig	Used to display or configure network card related information.
apt-get	Used to install, configure and remove packages on Debian and its derivatives.
iwconfig	Used to display or configure wireless network card related information.
shutdown	Shuts down the system and performs shut down related tasks including restart, halt, put to sleep or kick out all currently connected users.
passwd	Used to change the password.
cat	Used to list the contents of a file and expects the file name as the parameter.
man	Used to display the documentation for a specific command.

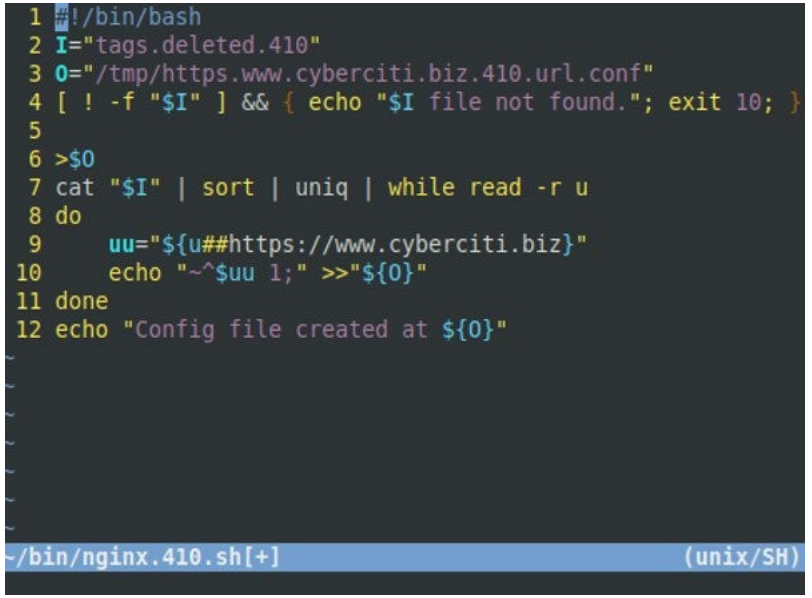
File and Directory Commands

Many command line tools are included in Linux by default. The following table lists a few of the most common **commands related to files and directories**:

Command	Description
ls	Displays the files inside a directory.
cd	Changes the current directory.
mkdir	Creates a directory under the current directory.
cp	Copies files from source to destination.
mv	Moves files to a different directory.
rm	Removes files.
grep	Searches for specific strings of characters within a file or other commands outputs.
cat	Lists the contents of a file and expects the file name as the parameter.

Working with Text Files

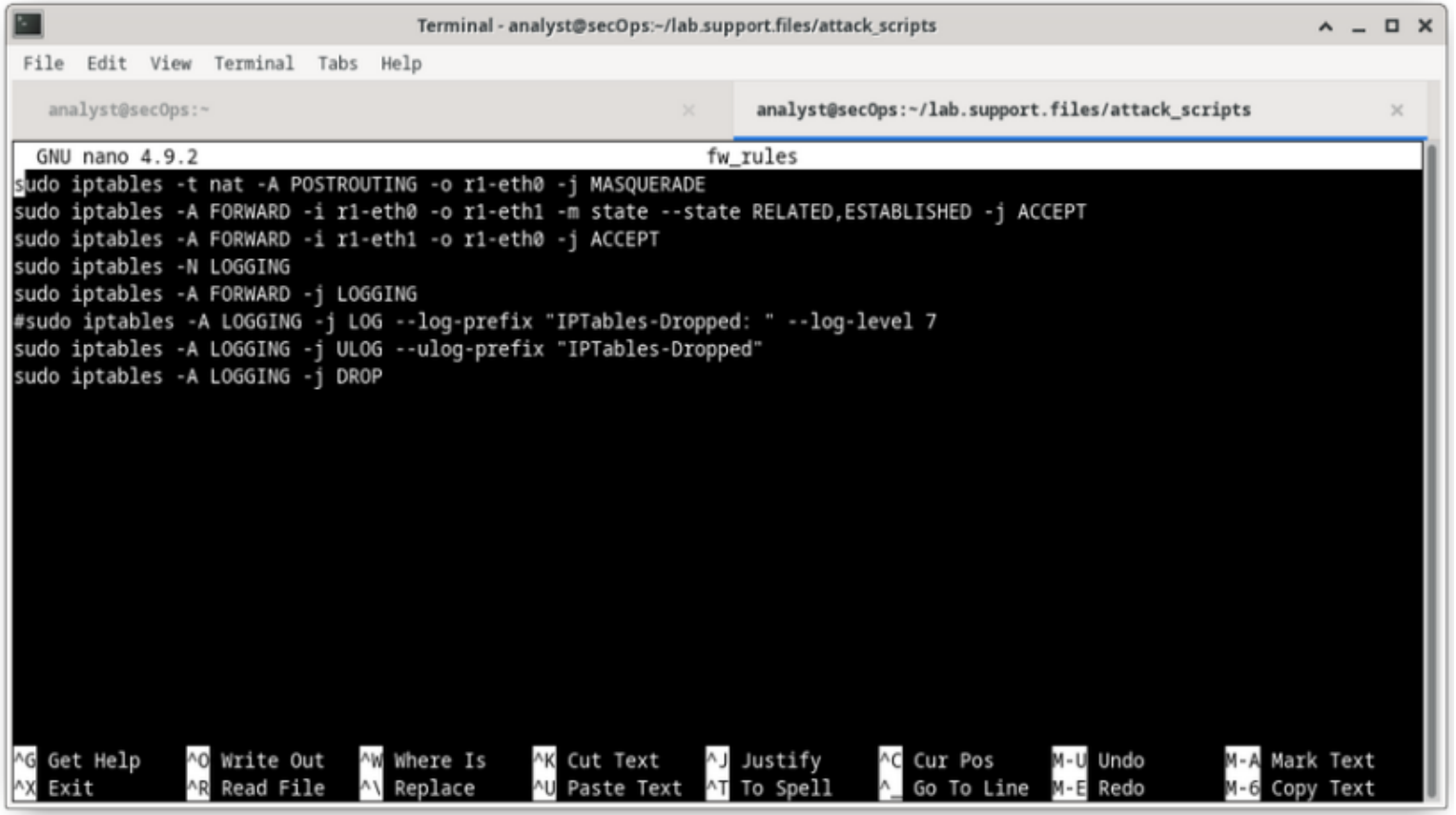
- Linux has many different **text editors**, with various features and functions.
- Some text editors include **graphical interfaces** while others are **command-line only** tools. Each text editor includes a feature set designed to support a specific type of task.
- Some text editors focus on the programmer and include features such as **syntax highlighting**, **parenthesis check**, and other programming-focused features.
- While graphical text editors are convenient and easy to use, command line-based text editors are very important for Linux users. The **main benefit** of **command-line-based text editors** is that they **allow for text file editing from a remote computer**.



Using a vim color scheme

Working with Text Files (Contd.)

- The figure shows **nano**, a popular **command-line text editor**.
- The administrator is editing firewall rules. Text editors are often used for **system configuration** and **maintenance** in Linux.
- Due to the lack of graphical support, nano (or GNU nano) **can only be controlled with the keyboard**.



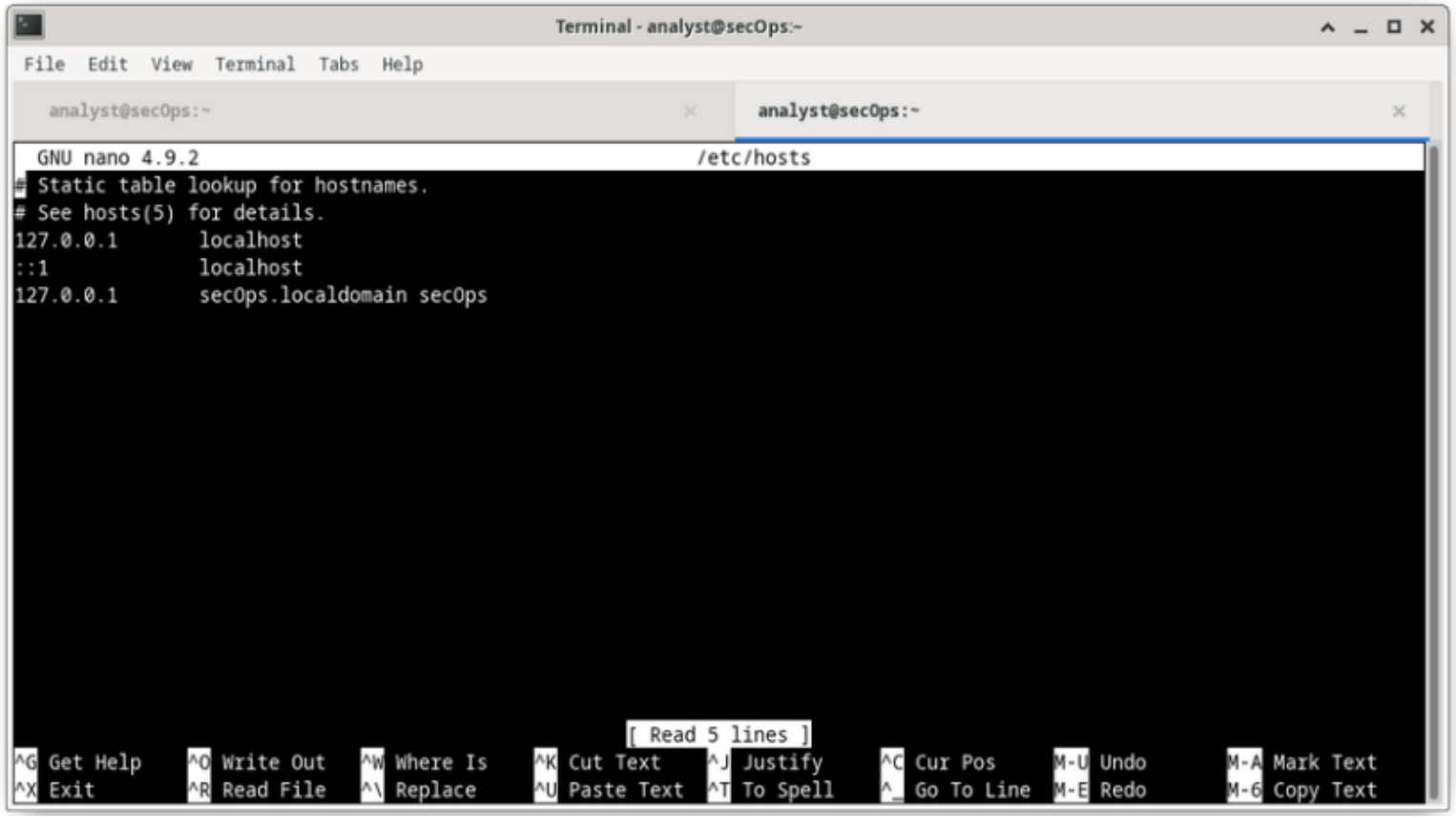
The Importance of Text Files in Linux

- In Linux, **everything is treated as a file**. This includes the memory, the disks, the monitor, and the directories.
- Configuration files are text files which are used **to store adjustments** and **settings** for specific **applications** or **services**.
- Users with **proper permission levels** can use text editors to change the contents of configuration files.
- After the changes are made, the file is saved and can be used by the related service or application. Users are able to specify exactly how they want any given application or service to behave. When launched, services and applications check the contents of specific configuration files to adjust their behavior accordingly.

Note: ***The administrator used the command sudo nano /etc/hosts to open the file. The command sudo (short for "superuser do") invokes the superuser privilege to use the nano text editor to open the host file.***

The Importance of Text Files in Linux (cont.)

- In the figure, the administrator opened the host configuration file in **nano** for editing.
- The host file contains static mappings of host IP addresses to names.
- The **names** serve as **shortcuts** that allow **connecting to other devices** by using a name instead of an IP address. **Only the superuser can change the host file**.



4.3 Linux Servers and Clients

Známe veci z predmetu PIKS (len opakovanie na doma)

4.4 Basic Server Administration

Známe veci z predmetu UdOS
(len opakovanie na doma)

Service Configuration Files

- In Linux, **services** are managed using configuration files.
- Common **options** in configuration files are **port number**, **location of the hosted resources**, and **client authorization details**.
- When the service starts, it looks for its configuration files, loads them into **memory**, and adjusts itself according to the settings in the files.
- The command output shows a portion of the **configuration file** for **Nginx**, which is a **lightweight web server for Linux**.

```
[analyst@secOps ~]$ cat /etc/nginx/nginx.conf
#user html;
worker_processes  1;
#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;
#pid        logs/nginx.pid;
events {
    worker_connections  1024;
}
http {
    include       mime.types;
    default_type  application/octet-stream;
    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';
    #access_log  logs/access.log  main;
```

Service Configuration Files (Contd.)

The command output shows the configuration file for the network time protocol (NTP).

```
[analyst@secOps ~]$ cat /etc/ntp.conf
# Please consider joining the pool:
#
#       http://www.pool.ntp.org/join.html
#
# For additional information see:
# - https://wiki.archlinux.org/index.php/Network_Time_Protocol_daemon
# - http://support.ntp.org/bin/view/Support/GettingStarted
# - the ntp.conf man page
# Associate to Arch's NTP pool
server 0.arch.pool.ntp.org
server 1.arch.pool.ntp.org
server 2.arch.pool.ntp.org
server 3.arch.pool.ntp.org
# By default, the server allows:
# - all queries from the local host
# - only time queries from remote hosts, protected by rate limiting and kod
restrict default kod limited nomodify nopeer noquery notrap
restrict 127.0.0.1
restrict ::1
# Location of drift file
[analyst@secOps ~]$
```

Service Configuration Files (Contd.)

- The command output shows the configuration file for Snort, a Linux-based intrusion detection system (IDS).
- There is **no rule for a configuration file format**. It is the choice of the service's developer. However, the **option = value** format is often used.

```
[analyst@secOps ~]$ cat /etc/snort/snort.conf
#--------------------------------------------------
#   VRT Rule Packages Snort.conf
#
#   For more information visit us at:
#       http://www.snort.org                    Snort Website
#       http://vrt-blog.snort.org/      Sourcefire VRT Blog
#
#       Mailing list Contact:   snort-sigs@lists.sourceforge.net
#       False Positive reports: fp@sourcefire.com
#       Snort bugs:             bugs@snort.org
#
#       Compatible with Snort Versions:
#       VERSIONS : 2.9.9.0
#
#       Snort build options:
#       OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofiling --
enable-zlib --enable-active-response --enable-normalizer --enable-reload --enable-react --enable-
flexresp3
<output omitted>
##################################################
# Step #1: Set the network variables.  For more information, see README.variables
##################################################
# Setup the network addresses you are protecting
###ipvar HOME_NET any
###ipvar HOME_NET [192.168.0.0/24,192.168.1.0/24]
ipvar HOME_NET [209.165.200.224/27]
# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
```

Hardening Devices

- Device hardening involves implementing proven methods of securing the device and protecting its administrative access.
- Some of these methods involve **maintaining passwords**, **configuring enhanced remote login features**, and implementing **secure login with SSH**.
- Depending on the Linux distribution, many services are enabled by default. **Stopping such services** and ensuring they **do not automatically start at boot time** is another device hardening technique.
- OS **updates** are extremely important to maintaining a hardened device. OS developers create and issue **fixes** and **patches** regularly.

Hardening Devices (Contd.)

The following are basic best practices for device hardening:

- Ensure physical security
- Minimize installed packages
- Disable unused services
- Use SSH and disable the root account login over SSH
- Keep the system updated
- Disable USB auto-detection
- Enforce strong passwords
- Force periodic password changes
- Keep users from re-using old passwords

Monitoring Service Logs

- Log files are the records that a computer stores to keep track of important events. **Kernel**, **services**, and **application** events are all recorded in **log files**.
- By monitoring Linux log files, an administrator gains a clear picture of the computer's performance, security status, and any underlying issues.
- In Linux, log files can be categorized as:
  - Application logs
  - Event logs
  - Service logs
  - System logs
- Some logs contain information about **daemons** that are running in Linux. A daemon is a background process that runs without the need for user interaction.

Monitoring Service Logs (Contd.)

The following table lists a few popular Linux log files and their functions:

Linux Log File	Description/Function
/var/log/messages	• This directory contains generic computer activity logs. • It is mainly used to store informational and non-critical system messages.
/var/log/auth.log	• This file stores all authentication-related events in Debian and Ubuntu computers. • Anything involving the user authorization mechanism can be found in this file.
/var/log/secure	• This directory is used by RedHat and CentOS computers. • It also tracks sudo logins, SSH logins, and other errors logged by SSSD.
/var/log/boot.log	• This file stores boot-related information and messages logged during the computer startup process.

Monitoring Service Logs (Contd.)

Linux Log File	Description
/var/log/dmesg	• This directory contains **kernel ring buffer messages**. • Information related to **hardware** devices and their **drivers** is recorded here. • It is very important because, due to their low-level nature, logging systems such as syslog are not running when these events take place and are unavailable to the administrator in real-time.
/var/log/kern.log	• This file contains information logged by the kernel.
/var/log/cron	• Cron is a service used to **schedule automated tasks in Linux** and this directory stores its events. • Whenever a scheduled task (or cron job) runs, all its relevant information including execution status and error messages are stored here.
/var/log/mysqld.log or **/var/log/mysql.log**	• This is the **MySQL** log file. • All debug, failure and success messages related to the mysqld process and mysqld_safe daemon are logged here.

Monitoring Service Logs (Contd.)

- The command output shows a portion of **/var/log/messages** log file.
- Each line represents a logged event.
- The timestamps at the beginning of the lines mark the moment the event took place.

```
[analyst@secOps ~]$ sudo cat /var/log/messages
Mar 20 15:28:45 secOps kernel: Linux version 4.15.10-1-ARCH (builduser@heftig-18961) (gcc version 7.3.1
20180312 (GCC)) #1 SMP PREEMPT Thu Mar 15 12:24:34 UTC 2018
Mar 20 15:28:45 secOps kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-linux root=UUID=07c6b457-3f39-
4ddf-bfd8-c169e8a877b2 rw quiet
Mar 20 15:28:45 secOps kernel: KERNEL supported cpus:
Mar 20 15:28:45 secOps kernel:   Intel GenuineIntel
Mar 20 15:28:45 secOps kernel:   AMD AuthenticAMD
Mar 20 15:28:45 secOps kernel:   Centaur CentaurHauls
Mar 20 15:28:45 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
Mar 20 15:28:45 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
Mar 20 15:28:45 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
Mar 20 15:28:45 secOps kernel: x86/fpu: xstate_offset[2]:  576, xstate_sizes[2]:  256
Mar 20 15:28:45 secOps kernel: x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using
'standard' format.
Mar 20 15:28:45 secOps kernel: e820: BIOS-provided physical RAM map:
Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x0000000000100000-0x00000003ffefffff] usable
Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x00000003fff0000-0x00000003ffffffff] ACPI data
Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
Mar 20 15:28:45 secOps kernel: NX (Execute Disable) protection: active
Mar 20 15:28:45 secOps kernel: random: fast init done
Mar 20 15:28:45 secOps kernel: SMBIOS 2.5 present.
Mar 20 15:28:45 secOps kernel: DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
Mar 20 15:28:45 secOps kernel: Hypervisor detected: KVM
Mar 20 15:28:45 secOps kernel: e820: last_pfn = 0x3fff0 max_arch_pfn = 0x400000000
Mar 20 15:28:45 secOps kernel: MTRR: Disabled
Mar 20 15:28:45 secOps kernel: x86/PAT: MTRRs disabled, skipping PAT initialization too.
Mar 20 15:28:45 secOps kernel: CPU MTRRs all blank - virtualized system.
```

4.5 The Linux File System

Známe veci z predmetu UdOS (len opakovanie na doma)

The File System Types in Linux

- There are many different kinds of file systems, varying in properties of speed, flexibility, security, size, structure, logic and more.
- The administrator decides the file system type which is suitable for the operating system.
- The following table lists a few file system types commonly found and supported by Linux.

Linux File System	Description
ext2 (second extended file system)	• ext2 was the default file system in several major Linux distributions until supplanted by ext3. • ext2 is still the file system of choice for flash-based storage media, as its lack of a journal, increases performance and minimizes the number of writes. • As flash memory devices have a limited number of write operations, minimizing write operations increases the device's lifetime.

The File System Types in Linux (Contd.)

Linux File System	Description
ext3 (third extended file system)	• ext3 is a journaled file system designed to improve the existing ext2 file system. • A journal or log, the main feature added to ext3, is a technique used to minimize the risk of file system corruption in the event of sudden power loss. • The file systems keeps a log of all the changes to be made. • If the computer crashes before the change is complete, the journal can be used to restore or correct any issues created by the crash. • The maximum file size in ext3 file systems is 32 TB.
ext4 (fourth extended file system)	• ext4 was created based on a series of extensions to ext3. • While the extensions improve the performance of ext3 and increase supported file sizes, developers were concerned about stability issues and were opposed to adding the extensions to the stable ext3. • The ext3 project was split in two; one kept as ext3 and its normal development and the other, named ext4, incorporated the mentioned extensions.

The File System Types in Linux (Contd.)

Linux File System	Description
NFS (Network File System)	• NFS is a network-based file system, allowing file access over the network. • From the user standpoint, there is no difference between accessing a file stored locally or on another computer on the network. • NFS is an open standard which allows anyone to implement it.
CDFS (Compact Disc File System)	• CDFS was created specifically for optical disk media.
Swap File System	• The swap file system is used by Linux when it runs out of RAM. • When this happens, the kernel moves inactive RAM content to the swap partition on the disk. • While swap partitions can be useful to Linux computers with a limited amount of memory, they should not be considered as a primary solution. • Swap partition is stored on disk which has much lower access speeds than RAM.

The File System Types in Linux (Contd.)

Linux File System	Description
HFS Plus or HFS+ (Hierarchical File System Plus)	• A file system used by Apple in its Macintosh computers. • The Linux kernel includes a module for mounting HFS+ for read-write operations.
APFS (Apple File System)	• An updated file system that is used by Apple devices. • It provides strong encryption and is optimized for flash and solid-state drives.
Master Boot Record (MBR)	• Located in the first sector of a partitioned computer, the MBR stores all the information about the way in which the file system is organized. • The MBR quickly hands over control to a loading function, which loads the OS.

The File System Types in Linux (Contd.)

- Mounting is the term used for the process of assigning a directory to a partition.
- After a successful mount operation, the file system contained on the partition is accessible through the specified directory.
- The command output shows the output of the **mount** command issued in the Cisco CyberOPS VM.

```
[analyst@secOps ~]$ mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sys on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
dev on /dev type devtmpfs (rw,nosuid,relatime,size=494944k,nr_inodes=123736,mode=755)
run on /run type tmpfs (rw,nosuid,nodev,relatime,mode=755)
/dev/sda1 on / type ext4 (rw,relatime)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup2 on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=11792)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev)
```

Linux Roles and File Permissions

- Linux uses file permissions in order to organize the system and enforce boundaries within the computer.
- Every file in Linux carries its file permissions, which define the actions that the owner, the group, and others can perform with the file.
- The possible permission rights are Read, Write, and Execute.
- The **ls** command with the **-l** parameter lists additional information about the file.

Linux Roles and File Permissions (Contd.)

The output of the **ls -l** command provides a lot of information about the file **space.txt**:

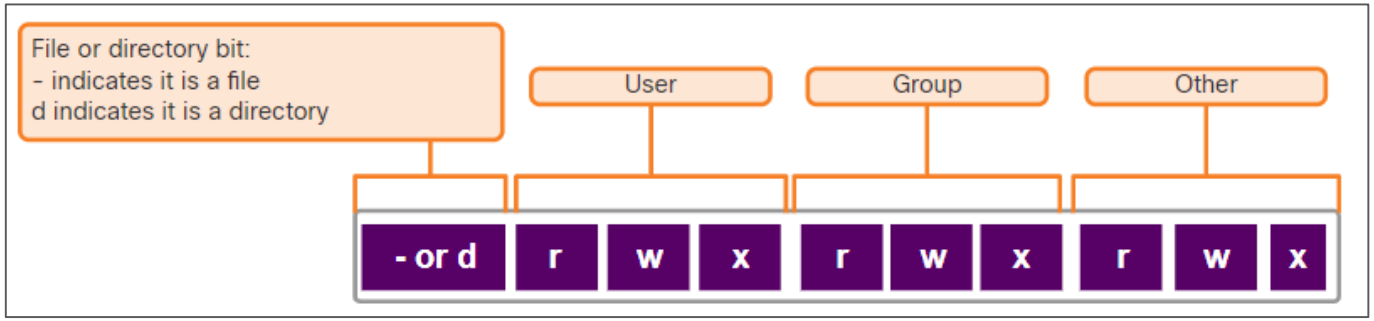
- The first field displays the permissions with **space.txt** (**-rwxrw-r--**).
- The second field defines the number of hard links to the file (number **1** after the permissions).
- The third and fourth field display the user (**analyst**) and group (**staff**) who own the file, respectively.
- The fifth field displays the file size in bytes. The **space.txt** file has 253 bytes.
- The sixth field displays the date and time of the last modification.
- The seventh field displays the file name.

```
[analyst@secOps ~]$ ls -l space.txt
-rwxrw-r-- 1 analyst staff 253 May 20 12:49 space.txt
 (1)(2)(3)(4)(5)(6)(7)
[analyst@secOps ~]$
```

Linux Roles and File Permissions (Contd.)

The figure here shows a breakdown of file permissions in Linux. The file **space.txt** has the following permissions:

- The dash (-) means that this is a file.
- The first set of characters (**rwx**) is for user permission. The user (**analyst**) who owns the file can **R**ead, **W**rite and e**X**ecute the file.
- The second set of characters is for group permissions (**rw-**). The group (**staff**) who owns the file can **R**ead and **W**rite to the file.
- The third set of characters is for any other user or group permissions (**r--**) who can only **R**ead the file.



Linux Roles and File Permissions (Contd.)

- Octal values are used to define permissions.
- File permissions are a fundamental part of Linux and cannot be broken.
- The only user that can override file permission on a Linux computer is the root user.

Binary	Octal	Permission	Description
000	0	---	No access
001	1	--x	Execute only
010	2	-w-	Write only
011	3	-wx	Write and Execute
100	4	r--	Read only
101	5	r-x	Read and Execute
110	6	rw-	Read and Write
111	7	rwx	Read, Write and Execute

Hard Links and Symbolic Links

- A hard link is another file that points to the same location as the original file.
- Use the command **ln** to create a hard link.
- The first argument is the existing file and the second argument is the new file.
- As shown in the command output, the file **space.txt** is linked to **space.hard.txt** and the link field now shows 2.
- Both files point to the same location in the file system. If you change one file, the other is changed, as well.
- The **echo** command is used to add some text to **space.txt**.

```
[analyst@secOps ~]$ ln space.txt space.hard.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ ls -l space*
-rw-r--r-- 2 analyst analyst 239 May  7 18:18 space.hard.txt
-rw-r--r-- 2 analyst analyst 239 May  7 18:18 space.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ echo "Testing hard link" >> space.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ ls -l space*
-rw-r--r-- 2 analyst analyst 257 May  7 18:19 space.hard.txt
-rw-r--r-- 2 analyst analyst 257 May  7 18:19 space.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ rm space.hard.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ more space.txt
Space... The final frontier…
These are the voyages of the Starship Enterprise. Its continuing mission:
- To explore strange new worlds…
- To seek out new life; new civilizations…
- To boldly go where no one has gone before!
Testing hard link
[analyst@secOps ~]$
```

Hard Links and Symbolic Links (Contd.)

- A symbolic link, also called a symlink or soft link, is similar to a hard link in that applying changes to the symbolic link will also change the original file.
- As shown in the command output, use the **ln** command option **-s** to create a symbolic link.
- Notice that adding a line of text to **test.txt** also adds the line to **mytest.txt**.

```
[analyst@secOps ~]$ echo "Hello World!" > test.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ ln -s test.txt mytest.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ echo "It's a lovely day!" >> mytest.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ more test.txt
Hello World!
It's a lovely day!
[analyst@secOps ~]$
[analyst@secOps ~]$ rm test.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ more mytest.txt
more: stat of mytest.txt failed: No such file or directory
[analyst@secOps ~]$
[analyst@secOps ~]$ ls -l mytest.txt
lrwxrwxrwx 1 analyst analyst 8 May  7 20:17 mytest.txt -> test.txt
[analyst@secOps ~]$
```

Hard Links and Symbolic Links (Contd.)

The following table shows several benefits of symbolic links over hard links:

Hard Links	Soft Links
Locating hard links is difficult.	Symbolic links show the location of the original file in the **ls -l** command.
Hard links are limited to the file system in which they are created.	Symbolic links can link to a file in another file system.
Hard links cannot link to a directory as the system itself uses hard links to define the hierarchy of the directory structure.	Symbolic links can link to directories.

4.6 Working with the Linux GUI

Známe veci z predmetu UdOS
(len opakovanie na doma)

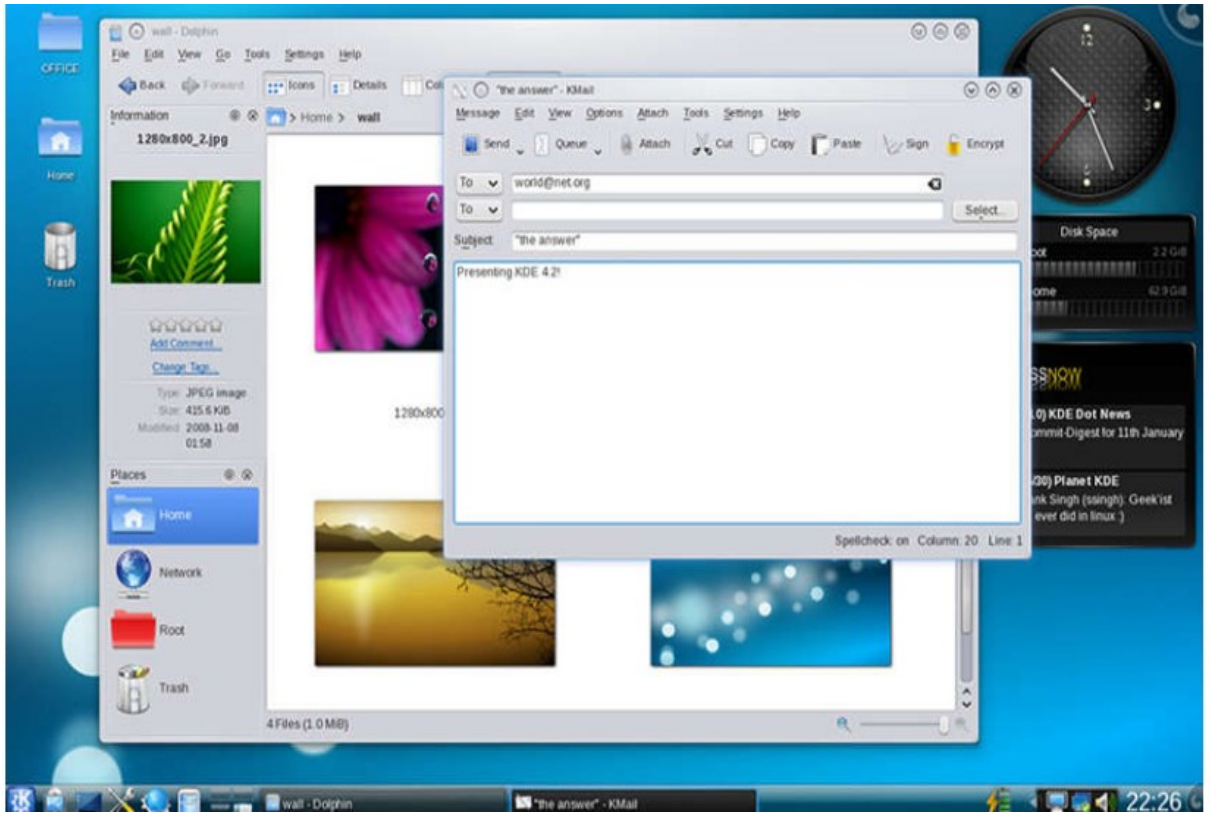
X Window System

- The graphical interface present in most Linux computers is based on the X Window System.
- X Window, also known as X or X11, is a windowing system designed to provide the basic framework for a GUI.
- X includes functions for drawing and moving windows on the display device and interacting with a mouse and keyboard.
- X works as a server, which allows a remote user to use the network to connect, start a graphical application, and have the graphical window open on the remote terminal.
- X does not specify the user interface, leaving it to other programs, such as window managers, to define all the graphical components.

X Window System (Contd.)

Examples of window managers are Gnome and KDE.

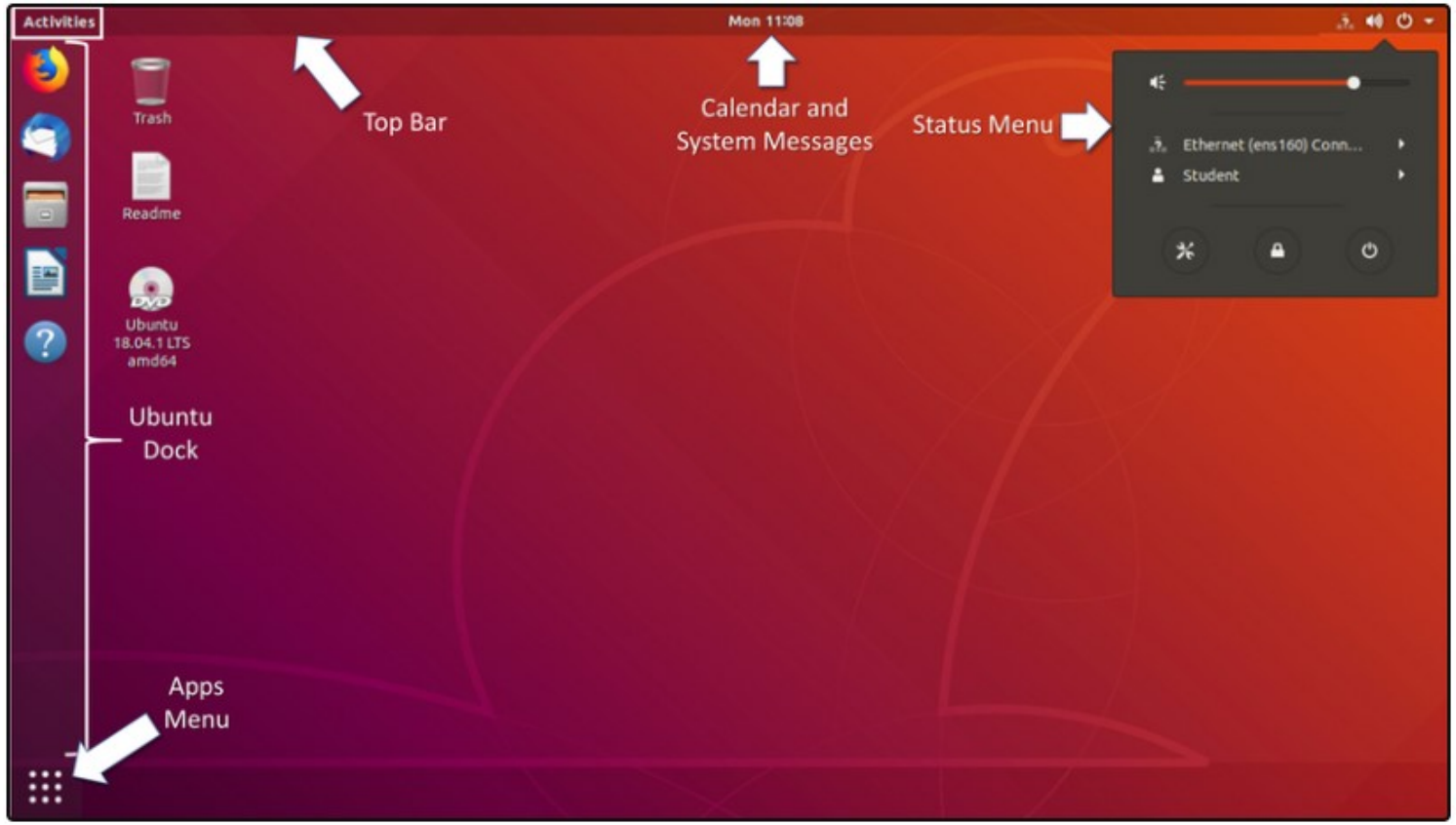
The Gnome Window Manager



The KDE Window Manager

The Linux GUI

- While an operating system does not require a GUI to function, GUIs are considered more user-friendly than the CLI. The Linux GUI as a whole can be easily replaced by the user.
- Ubuntu is a very popular and user-friendly Linux distribution.
- Ubuntu Linux uses Gnome 3 as its default GUI.
- The figure shows the location of some of the features of the Ubuntu Gnome 3 Desktop.



The Linux GUI (Contd.)

The following table lists the main UI components of Unity:

UI Component	Description
Apps Menu	• The Apps Menu shows icons for the apps that are installed on the system. • A right-click menu provides shortcuts that allow starting or configuring the apps. • The system search box is available from Activities View.
Ubuntu Dock	• A dock on the left side of the screen that serves as an application launcher and switcher for app favorites. • Click to launch an application and when the application is running, click again to switch between running applications. • If more than one application is running, launcher will display all instances. • Right-click any application on the launcher to see details about that the application.
Top Bar	• This menu bar contains a menu for the application that currently has the focus. • It displays the current time and indicates whether there are new system messages. • It also provides access to the Activity desktop view and the system Status Menu.

The Linux GUI (Contd.)

UI Component	Description
Calendar and System Message Tray	• Click the day and time to see the full appointment calendar and any current system messages. • Access the appointment calendar from here to create new appointments.
Activities	• Switch to application view to switch to or close running applications. • A powerful search tool is available here that will find apps, files, and values within files. • Allows switching between workspaces.
Status Menu	• Allows configuration of the network adaptor and other running devices. • The current user can logoff or change their settings. • System configuration changes can be made here. • The workstation can be locked or shutdown from here.

4.7 Working on a Linux Host

Známe veci z predmetu UdOS (len opakovanie na doma)

Installing and Running Applications on a Linux Host

- Many end-user applications are complex programs written in compiled languages.
- To aid in the installation process, Linux includes programs called package managers.
- By using a package manager to install a package, all the necessary files are placed in the correct file system location.
- A package is the term used to refer to a program and all its supporting files.
- The command output shows the output of a few **apt-get** commands used in Debian distributions.

```
analyst@cuckoo:~$ sudo apt-get update
[sudo] password for analyst:
Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Get:3 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [534 kB]
<output omitted>
Fetched 4,613 kB in 4s (1,003 kB/s)
Reading package lists... Done
analyst@cuckoo:~$
analyst@cuckoo:~$ sudo apt-get upgrade
Reading package lists   Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
linux-generic-hwe-16.04 linux-headers-generic-hwe-16.04
linux-image-generic-hwe-16.04
The following packages will be upgraded:
firefox firefox-locale-en girl.2-javascriptcoregtk-4.0 girl.2-webkit2-4.0 libjavascriptcoregtk-4.0-18
libwebkit2gtk-4.0-37 libwebkit2gtk-4.0-37-gtk2 libxen-4.6 libxenstore3.0 linux-libc-dev logrotate
openssh-client
qemu-block-extra qerau-kvm qemu-system-common qemu-system-x86 qemu-utils
```

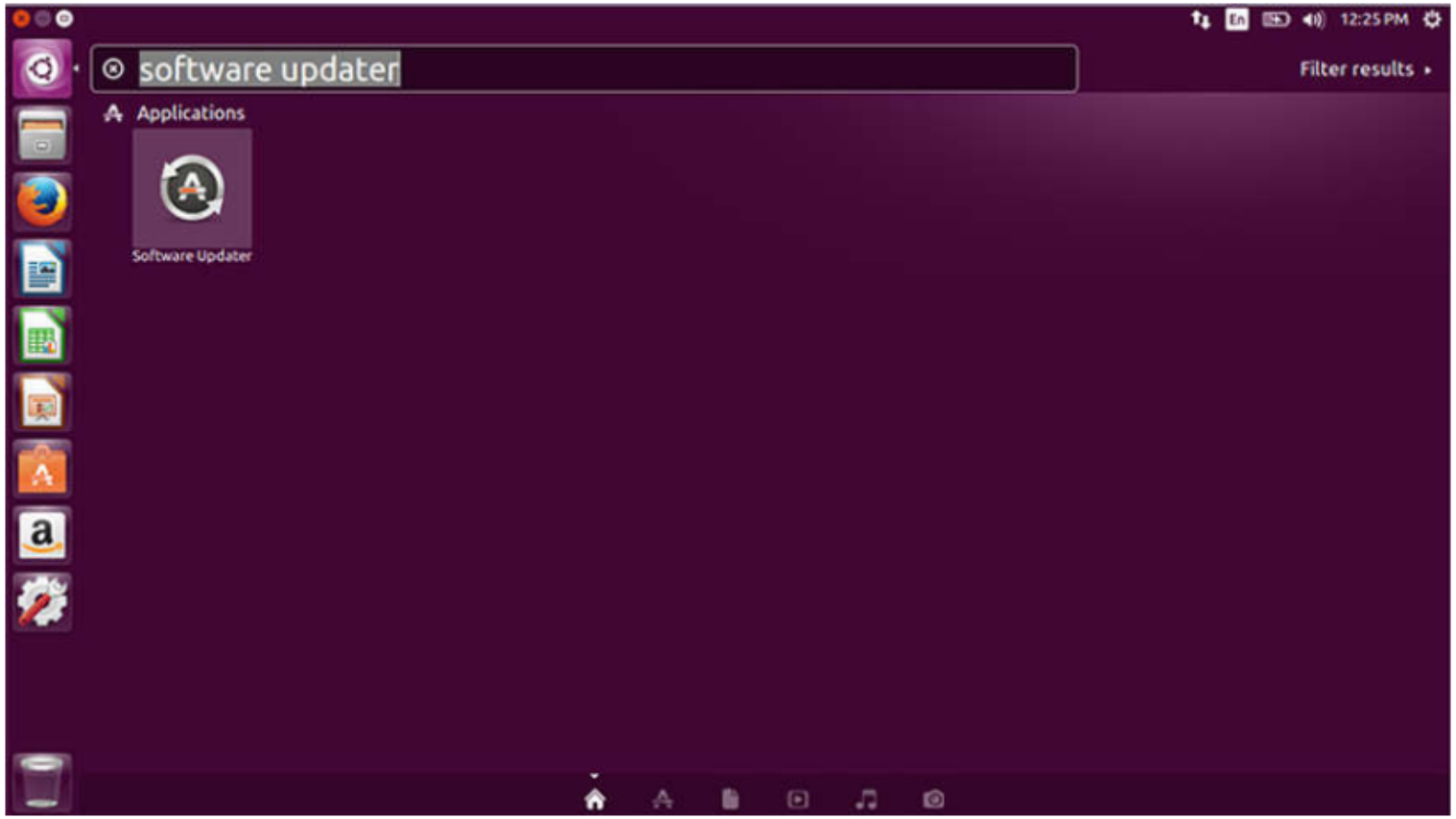
Keeping the System Up to Date

- OS updates, also known as patches, are released periodically by OS companies to address any known vulnerabilities in their operating systems.
- Modern operating systems will alert the user when updates are available for download and installation, but the user can check for updates at any time.
- The following table compares Arch Linux and Debian/Ubuntu Linux distribution commands to perform package system basic operations.

Task	Arch	Debian/Ubuntu
Install a package by name	**pacman -S**	**apt install**
Remove a package by name	**pacman -Rs**	**apt remove**
Update a local package	**pacman -Syy**	**apt-get update**
Upgrade all currently installed packages	**pacman -Syu**	**apt-get upgrade**

Keeping the System Up to Date (Contd.)

- A Linux GUI can also be used to manually check and install updates.
- In Ubuntu for example, to install updates you would click **Dash Search Box,** type **software updater**, and then click the **Software Updater** icon.



Processes and Forks

- A process is a running instance of a computer program.
- Forking is a method that the kernel uses to allow a process to create a copy of itself.
- Processes need a way to create new processes in multitasking operating systems. The fork operation is the only way of doing so in Linux.
- When a process calls a fork, the caller process becomes the parent process and the newly created process becomes its child.
- After the fork, the processes are, to some extent, independent processes. They have different process IDs but run the same program code.

Processes and Forks (Contd.)

The following table lists three commands that are used to manage processes.

Command	Description
ps	• Used to list the processes running on the computer at the time it is invoked. • It can be instructed to display running processes that belong to the current user or other users.
top	• Used to list running processes, but unlike **ps**, **top** keeps displaying running processes dynamically. • Press **q** to exit top.
kill	• Used to modify the behavior of a specific process. • Depending on the parameters, **kill** will remove, restart, or pause a process. • In many cases, the user will run **ps** or **top** before running kill. • This is done so the user can learn the PID of a process before running kill.

Processes and Forks (Contd.)

The command output shows the output of the **top** command on a Linux computer.

```
[analyst@secOps ~]$ top
top - 11:29:16 up 0 min,  1 user,  load average: 1.09, 0.31, 0.11
Tasks: 119 total,   1 running, 118 sleeping,   0 stopped,   0 zombie
%Cpu(s):  5.4 us,  2.0 sy,  0.0 ni, 87.4 id,  2.7 wa,  1.4 hi,  1.0 si,  0.0 st
MiB Mem :       982.8 total,    67.9 free,      765.8 used,     149.1 buff/cache
MiB Swap:       0.0 total,      0.0 free,       0.0 used.       39.3 avail Mem
        PID USER        PR  NI  VIRT    RES     SHR S  %CPU  %MEM       TIME+ COMMAND
        729 analyst   20   0 2652376 284472  61076 S   2.7  28.3   0:06.75 Web Con+
        570 analyst   20   0 2691388 215728  62404 S   2.0  21.4   0:06.99 firefox
        357 root        20   0  267972  91960  18468 S   1.3   9.1   0:01.63 Xorg
        461 analyst   20   0  322208  21000   7480 S   1.3   2.1   0:00.67 xfce4-p+
        121 root        20   0          0       0       0 S   0.7   0.0   0:00.43 kswapd0
        1 root          20   0  174376   4196   1688 S   0.3   0.4   0:00.66 systemd
        294 root        20   0  245036  11876   868 S   0.3   1.2   0:00.34 python2+
        539 analyst   20   0  150824    660     0 S   0.3   0.1   0:00.02 VBoxCli+
        800 analyst   20   0  477768  18968   9800 S   0.3   1.9   0:00.30 xfce4-t+
        2 root          20   0          0       0       0 S   0.0   0.0   0:00.00 kthreadd
        3 root          0 -20          0       0       0 I   0.0   0.0   0:00.00 rcu_gp
        4 root          0 -20          0       0       0 I   0.0   0.0   0:00.00 rcu_par+
        5 root          20   0          0       0       0 I   0.0   0.0   0:00.00 kworker+
        6 root          0 -20          0       0       0 I   0.0   0.0   0:00.00 kworker+
        7 root          20   0          0       0       0 I   0.0   0.0   0:00.00 kworker+
        8 root          0 -20          0       0       0 I   0.0   0.0   0:00.00 mm_perc+
        9 root          20   0          0       0       0 S   0.0   0.0   0:00.02 ksoftir+
[analyst@secOps ~]$
```

Malware on a Linux Host

- Linux malware includes viruses, Trojan horses, worms, and other types of malware that can affect the operating system.
- A common Linux attack vector is its services and processes.
- The command output shows an attacker using the Telnet command to probe the nature and version of a web server (port 80).
- The attacker has learned that the server is running nginx version 1.12.0. The next step would be to research known vulnerabilities in the nginx 1.12.0 code.

```
analyst@secOps ~]$ telnet 209.165.200.224 80
Trying 209.165.200.224...
Connected to 209.165.200.224.
Escape character is '^]'.
<type anything to force an HTTP error response>
HTTP/1.1 400 Bad Request
Server: nginx/1.12.0
Date: Wed, 17 May 2017 14:27:30 GMT
Content-Type: text/html
Content-Length: 173
Connection: close
<html>
<head><title>400 Bad Request</title></head>
<body bgcolor="white">
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx/1.12.0</center>
</body>
</html >
Connection closed by foreign host.
analyst@secOps ~]$
```

Rootkit Check

- A rootkit is a type of malware designed to increase an unauthorized user's privileges or grant access to portions of the software that should not normally be allowed.
- A rootkit is destructive as it changes kernel code and its modules, changing the most fundamental operations of the OS itself.
- Rootkit detection methods include booting the computer from a trusted media.
- Rootkit removal can be complicated. Re-installation of the operating system is the only real solution to the problem.
- **chkrootkit** is a popular Linux-based program designed to check the computer for known rootkits.
- The command output shows the output of **chkrootkit** on an Ubuntu Linux.

```
analyst@cuckoo:~$ sudo ./chkrootkit
[sudo] password for analyst:
ROOTDIR is '/'
Checking 'amd'... not found
Checking 'basename'... not infected
Checking 'biff'... not found
Checking 'chfn'... not infected
Checking 'chsh'... not infected
Checking 'cron'... not infected
Checking 'crontab'... not infected
Checking 'date'... not infected
Checking 'du'... not infected
Checking 'dirname'... not infected
Checking 'echo'... not infected
Checking 'egrep'... not infected
Checking 'env'— not infected
Checking 'find'... not infected
Checking 'fingerd'... not found
Checking 'gpm'... not found
Checking 'grep'... not infected
Checking 'hdparm'... not infected
Checking 'su'... not infected
Checking 'ifconfig'... not infected
Checking 'inetd'... not tested
Checking 'inetdconf'... not found
```

Piping Commands

- Although command line tools are usually designed to perform a specific, well-defined task, many commands can be combined to perform more complex tasks by a technique known as piping.
- Piping consists of chaining commands together, feeding the output of one command into the input of another.
- The two commands, **ls** and **grep**, can be piped together to filter out the output of **ls**. This is shown in the output of the **ls -l | grep host** command and the **ls -l | grep file** command.

```
[analyst@secOps ~]$ ls -l
total 40
drwxr-xr-x 2 analyst analyst   4096 Mar  22  2018 Desktop
drwxr-xr-x 3 analyst analyst   4096 April 2 14:44 Downloads
-rw-r--r-- 1 analyst analyst      9 May 20 10:51 hostfile1.txt
-rw-r--r-- 1 analyst analyst      9 May 20 10:51 hostfile2.txt
-rw-r--r-- 1 analyst analyst      9 May 20 10:52 hostfile3.txt
drwxr-xr-x 9 analyst analyst   4096 Jul 19  2018 lab.support.files
-rw-r--r-- 1 analyst analyst     19 May 20 10:53 mytest.com
-rw-r--r-- 1 analyst analyst 228844 May 20 10:54 rkhunter-1.4.6-1-any.pkg.tar.xz
drwxr-xr-x 2 analyst analyst   4096 Mar 21  2018 second_drive
-rw-r--r-- 1 analyst analyst    257 May 20 10:52 space.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ ls -l | grep host
-rw-r--r-- 1 analyst analyst      9 May 20 10:51 hostfile1.txt
-rw-r--r-- 1 analyst analyst      9 May 20 10:51 hostfile2.txt
-rw-r--r-- 1 analyst analyst      9 May 20 10:52 hostfile3.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ ls -l | grep file
-rw-r--r-- 1 analyst analyst      9 May 20 10:51 hostfile1.txt
-rw-r--r-- 1 analyst analyst      9 May 20 10:51 hostfile2.txt
-rw-r--r-- 1 analyst analyst      9 May 20 10:52 hostfile3.txt
drwxr-xr-x 9 analyst analyst   4096 Jul 19  2018 lab.support.files
[analyst@secOps ~]$
```

4.8 Linux Basics Summary

What Did I Learn in this Module?

- Linux is a fast, reliable, and small open-source operating system.
- In Linux, the user communicates with the operating system through a GUI or a command-line interface (CLI), or shell.
- Servers are computers that have software installed that enables them to provide services to client computers across the network.
- In Linux, servers are managed by using configuration files. Various settings can be modified and saved in configuration files.
- Linux supports a number of different file systems that vary by speed, flexibility, security, size, structure, logic, and more. Some of the file systems that are supported by Linux are ext2, ext3, ext4, NFS, and CDFS.
- The X Windows, or X11, system is a basic software framework that includes functions for creating, controlling, and configuring a windows GUI in a point-and-click interface.
- To install applications on Linux hosts, programs called package managers are used. Packages are software applications and all of their supporting files.

Linux hardening

Čo je hardening

- súbor techník, osvedčených postupov a nástrojov za účelom **zníženia zraniteľnosti** v aplikáciách, infraštruktúre, firmvéri a iných oblastiach
- Podľa NIST:
  - „Hardening je proces **eliminácie prostriedkov útoku** opravením zraniteľností a vypnutím nepodstatných služieb.“
- Súčasťou je
  - zakázanie oprávnení, portov, vymazanie nepotrebných aplikácií, používateľských účtov a ďalších funkcií,
  - čo má za dôsledok **zosilnenie systému**, takže útočníci majú menšiu šancu získať prístup k citlivým informáciám počítačového systému.

Treba vôbec riešiť hardening Linuxu?

- Určite áno
- Najčastejšie hrozby:
  - Softvér na ťaženie kryptomien
  - Botnet malware
  - Ransomware
- Príklad - známy prípad z 2018 – malvér „XBash“
  - kombinácia vydieračského vírusu, botnetu, červa plus poskytoval možnosť ťažiť virtuálnu menu
  - v linuxovom prostredí aktivoval modul ransomware
    - ten v rámci napadnutého servera vyhľadal a odstránil všetky existujúce databázy typu MySQL, MongoDB a PostgreSQL
    - následne sa vytvorila nová databáza s názvom „PLEASE_READ_ME_XYZ“
      - obsahovala základné informácie o údajnom zašifrovaní súborov a návod ako zaplatiť výkupné
      - Plus výstraha: nezaplatenie = zverejnenie údajov

https://unit42.paloaltonetworks.com/unit42-xbash-combines-botnet-ransomware-coinmining-worm-targets-linux-windows/

Výhody hardeningu

- Vyššia miera bezpečnosti
  - zníženie rizika, že náš systém sa stane obeťou kybernetických útokov
- Väčšia efektivita systému
  - zvýšenie výkonu nášho zariadenie alebo infraštruktúry a menšie riziko prevádzkových problémov
- Dlhodobé úspory
  - ušetríme financie, ktoré by boli nevyhnuté na
    - obnovu po havárií
    - strata reputácie
    - pokuty... GDPR, a iné

Typy hardeningu

Týka sa všetkého, čo útočník môže využiť na preniknutie do systému

A. Hardening serveru
B. Hardening operačného systému
C. Hardening aplikácií
D. Hardening databáz
E. Hardening siete

A. Hardening servera

- poskytuje zabezpečenie komponentov, portov, údajov, funkcií servera
- pomocou bezpečnostných opatrení ochraňuje server na SW aj HW úrovni

Všeobecné odporúčania:

- Fyzické zabezpečenie systému
- Vypnutie **USB portov** počas zavádzania systému
- Využitie **viacfaktorovej** autentifikácie pre prihlásenie
- Pravidelná **aktualizácia OS** servera
- Pravidelná **aktualizácia programov tretích strán**, ktoré sú potrebné pre beh systému
- **Odstránenie** aplikácií tretích strán, ktoré **nespĺňajú** stanovené **štandardy** kybernetickej bezpečnosti
- Použitie **silnejších hesiel** a nastavenie striktných pravidiel pre vytvorenie hesla pre používateľské účty
- Zamknutie používateľa na určitý čas po 3 **neúspešných pokusoch** o prihlásenie sa do systému
- **Zašifrovanie** pevného disku
- Použitie **firewallu, antivírusového** programu a ďalších bezpečnostných balíčkov pre systém
- Zálohovanie dôležitých zdrojov informácií pomocou pravidla 3-2-1
  - 3 kópie zálohy na 2 typy médií a 1 kópia je uložená mimo lokality

B. Hardening operačného systému

- niele aktualizáciou
- ale aj vhodnou implementáciou bezpečnostných opatrení
  - ... OS dáva povolenia iným aplikáciám na vykonanie určitých zmien v systéme

Odporúčané kroky:

- Konfigurácia a spustenie Secure Boot
- Šifrovanie disku, na ktorom sa nachádza operačný systém
- Odinštalovanie nepotrebných aplikácií prípadne ovládačov (zakázanie/vypnutie)
- Riadenie prístupu
  - Nastavenie vlastných rolí pre používateľov a použite silných hesiel
  - Nastavenie oprávnení podľa potreby
  - Vymazanie neaktívnych používateľov
  - Nepoužívať účet root, ktorý má silné oprávnenia
  - Obmedzenie počtu členov v skupinách správcov

Vsuvka – bezpečnosť v distribúcií Ubuntu

- ochrany jadra
  - kolekcia mnohých technológií
    - dôkladne oddelenie kernel a userspace pamäte (kernel nespustí kód, ktorý sa nachádza v userspace)
  - **Kernel lockdown:** umožňuje v jadre pri zavádzaní systému zamknúť jadro tak, že niektoré časti jadra nie sú prístupné ani pre superpoužívateľa(root)
  - Obsahuje 2 režimy:
    - Integrity: blokuje možnosť konfigurácie bežiaceho jadra
    - Confidentiality: blokovanie čítania citlivých údajov z bežiaceho jadra
- univerzálna viacfaktorová a bezheslová autentifikácia na zmiernenie sociálneho inžinierstva
  - **Fast ID Online multi-factor authentication(FIDO):** bezpečné viacfaktorové on-line overenie, ktorá redukuje nutnosť používať heslá a namiesto hesiel poskytuje použite hardwarových bezpečnostných kľúčov, rozpoznanie tváre, hlasu alebo odtlačkov prstov
- ochrana pred rootkits
- **WireGuard VPN**
  - VPN je charakteristická s nízkym počtom riadkov v zdrojovom kóde, z tohto dôvodu je výrazne jednoduchšie odhaľovanie a ladenie prípadných bezpečnostných chýb

C. Hardening softvérových aplikácií

- sústreďuje už na konkrétne aplikácie, bežiace v danom operačnom systéme

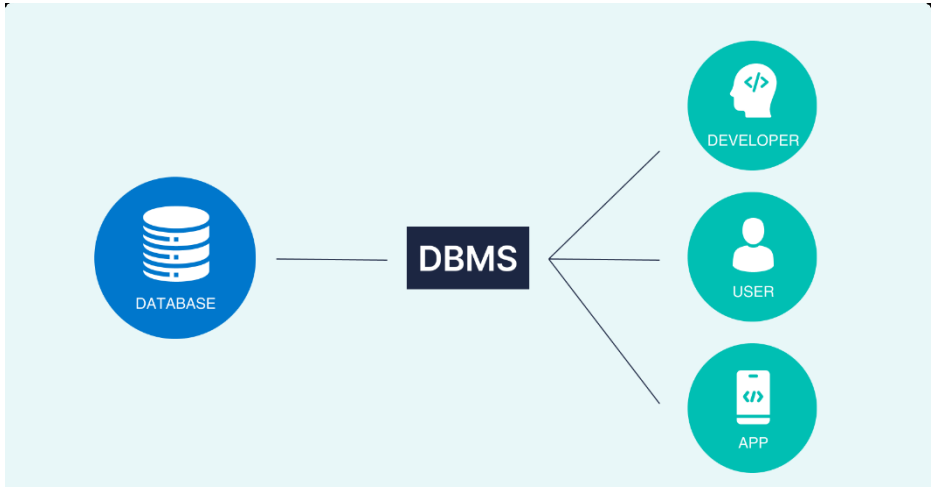
Mal by zahŕňať :

- Aktualizácia a automatická oprava aplikácií, vrátane aplikácií tretích strán
- Používanie firewall a ďalších programov zameraných na ochranu voči malwaru, spywaru, ...
- Zašifrovanie údajov
- Používanie aplikácii na vytvorenie bezpečných hesiel a uloženie hesiel
- Používanie procesorov, ktoré využívajú Intel Software Guard Extensions (SGX)
  - pomáhajú
    - chrániť používané údaje prostredníctvom technológie izolácie aplikácií
    - chrániť vybraný kód a dáta pred modifikáciami
    - vývojári môžu rozdeliť svoje aplikácie do „hardened enclaves“ alebo dôveryhodných vykonávacích modulov, aby pomohli zvýšiť bezpečnosť aplikácií
- Podpora funkcií IDS a IPS

https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html

D. Hardening databázy

- dôležité informácie organizácie
- na prístup k databáze – zväčša vzdialený prístup, DBMS
- zabezpečiť svoju databázu aj systém správy databáz (DBMS)
- ale aj vhodnou implementáciou bezpečnostných opatrení
  - ... OS dáva povolenia iným aplikáciám na vykonanie určitých zmien v systéme



Hardening zahŕňa tieto procesy:

- Použitie komplexných hesiel na prístup
- Ak sa zistí podozrivá aktivita pri prihlasovaní, tak je potrebné uzamknúť účet
- Šifrovanie údajov
- Vypnutie nepotrebných funkcií a služieb
- Pravidelná aktualizácia DBMS

E. Hardening siete

- monitorovať a hlásiť podozrivé aktivity v danej sieti
  - čo pomáha administrátorom zabrániť neoprávnenému prístupu do siete

Techniky zahŕňajú:

- Použitie VPN alebo reverzného proxy na pripojenie
- Správne nastavenie sieťových firewallov, IDS/IPS
- Kontrola privilégií prístupu k sieti a pravidiel v sieti
- Zakázanie nepotrebných alebo nepoužívaných sieťových portov
- Zašifrovanie sieťovej prevádzky
- Zakázanie sieťových služieb a zariadení, ktoré nie sú využívané v sieti
  - Zakázať nezabezpečené protokoly, ako sú SMBv1, Telnet a http

Benchmark pre systém hardening

- stanoviť si základnú líniu
  - zabezpečený stav systému, ktorý by sme sa mali snažiť dosiahnuť
  - Potom už len sledovať odchýlky
  - Zväčša pomocou benchmarku
    - súbor najlepších bezpečnostných postupov poskytovaných expertmi v oblasti KB
      - NIST (National Institute of Standards and Technology)
      - CIS (Center of Internet Security, https://learn.cisecurity.org/benchmarks)
      - Výrobcovia...
    - akceptované vládou, obchodom, priemyslom a akademickou obcou
    - konkrétne:
      - najskôr vykonať hodnotenie systému, pre ktorý chceme aplikovať systém hardening
        - ako sa súčasná konfigurácia zhoduje s relevantným CIS benchmarkom
        - často sú dostupné nástroje na automatické testovanie systému
      - potom nakonfigurovať systém, ktorý bude spĺňať bezpečnostné od-porúčania

Niektoré nástroje pre skenovanie systému

- Antivírusové programy:
  - Bitdefender, Endpoint Security Tools, ClamAV
- Skenovanie možných exploitov v systéme
  - RootKit hunter
- Kontrola vzoriek online
  - VirusTotal
- Správa softvérových záplat, skenovanie škodlivého softvéru a detekciu zraniteľností
  - Lynis
  - Lunar

Antivírus: ClamAV

- open source riešenie na hľadanie trojských koňov, malwaru a vírusov
- Dostupný pre Linux, ale aj Win a Mac OS
- poskytuje skenovanie systému z príkazového riadka, ale aj GUI
- Ištalácia:
  - `$ sudo apt install clamav clamav-daemon -y`
  - `$ sudo systemctl stop clamav-freshclam`
  - `$ sudo freshclam`
  - `$ sudo systemctl enable clamav-freshclam`
  - `$ sudo systemctl start clamav-daemon`
- zobrazenie infikovaných položiek a ich odstránenie v systéme
  - `$ sudo clamscan -i -r --remove /`
- získanie verzie s grafickým rozhraním:
  - `$ sudo apt-get install clamtk`

RootKit hunter

- dostupný vo väčšine linuxových distribúciách
- na hľadanie možných exploitov v systéme
- skenuje aj nesprávne nastavené oprávnenia pre binárne súbory, skryté súbory, podozrivé reťazce v moduloch jadra
- inštalácia:
  - `$ sudo apt install rkhunter`
  - `$ sudo rkhunter --propupd`
- skenovanie systému:
  - `$ sudo rkhunter --check`

VirusTotal

- bezplatná webová služba
- na vyhľadávanie škodlivého softvéru typu trojský kôň, červ a pod.
  - v súboroch akéhokoľvek typu
  - Ich nahratím na webovú stránku VirusTotal (max veľkosť 550MB)
- online nástroj pomocou 70+ rôznych antivírusových programov napr. Kaspersky, McAfee, ... zistí, či v našom súbore sa nachádza škodlivý obsah alebo nie
- https://www.virustotal.com/gui/home/upload

Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community.

Skenery: Lynis, Lunar

- Lynis
  - jeden z najdôveryhodnejších nástrojov automatického auditu na správu softvérových záplat, skenovanie škodlivého softvéru a detekciu zraniteľností
  - určený pre systémy Linux, macOS a UNIX
  - na skenovanie systému, či neobsahuje
    - základné bezpečnostné zraniteľnosti
    - chyby v konfigurácii
    - používateľské účty bez hesla
    - neoprávnené povolenia k súborom
    - audit brány firewall atď.
  - inštaláca:
    - `$ sudo apt update`
    - `$ wget -O - https://packages.cisofy.com/keys/cisofy-software-public.key | sudo apt-key add -`
    - `$ echo "deb https://packages.cisofy.com/community/lynis/deb/ stable main" | sudo tee /etc/apt/sources.list.d/cisofy-lynis.list`
    - `$ sudo apt update`
    - `$ sudo apt install lynis`
  - vykonanie auditu:
    - `$ sudo lynis audit system`

Automatický sprievodcovia pre zabezpečenie systému

Ubuntu security guide

- Zabezpečenie systému a audit podľa CIS benchmarkov – príklad pre Ubuntu
- Potrebné kroky:
  - Inštalovanie príslušného balíka:
    - `$sudo apt install ubuntu-advantage-tools`
  - Registrácia na stránke www.ubuntu.com , pre získanie tokenu. Keď získame token, použijeme príkaz:
    - `$sudo ua attach <token>`
  - Povoliť a nainštalovať USG:
    - `$sudo ua enable usg`
    - `$sudo apt install usg`
  - Vykonať bezpečnostný audit, na výber máme z týchto profilov:
    - **CIS level1 OR 2, workstation OR server**
    - `$ sudo usg audit cis_level1_workstation`
  - Po vykonaní si môžeme pozrieť, aký bezpečný je náš systém. Môžeme pritom využiť aj webový prehľadávač.
  - Nastaviť systém, tak aby vyhovoval štandardom CIS:
    - `$ sudo usg fix cis_level1_workstation`

UNIVERSITY OF ŽILINA
Faculty of Management Science and Informatics

Ďakujem za pozornosť

Obsahom bol modul 4 – Linux pre SOC analytika + (opakovanie UdOS) + pridali sme si Linux hardening.

Moduly 5-11 len v rámci opakovania PIKS (CCNA1) na samostatné prečítanie z Netacadu, kto si potrebuje pripomenúť, prejde si v Netacad kurze CyberOps.

Vyjadrite spätnú väzbu na prednášku a/alebo cvičenie v anonymnej ankete cez google form: link

Vytvorené v rámci projektu KEGA 026TUKE-4/2021.