UNIVERSIDAD EAFIT

Technical report

Operating Systems ST0257

March 01 of 2022

UNIVERSIDAD EAFIT

Contents

1 General Research

The first activity is simple, the idea is just research about the history of the operating systems.

1.1 Assignment

- Make a conceptual map about operating systems.
- Make a timeline that reflects the history and generation of operating systems.

1.2 Conceptual Map

This is the conceptual map using **draw.io**:

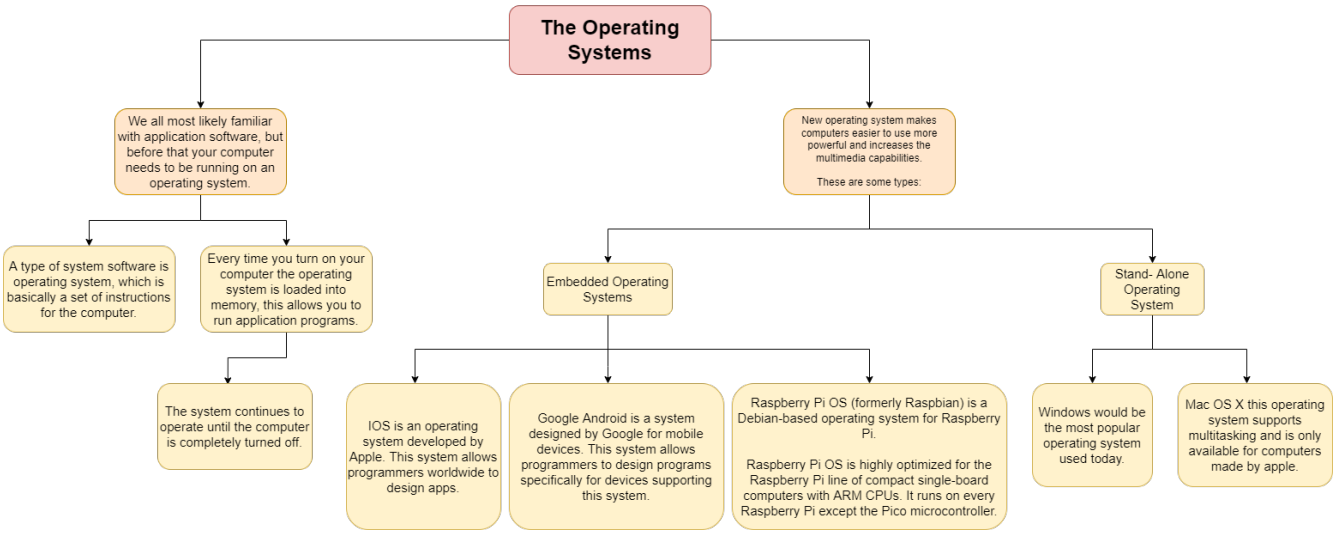


Figure 1: Operating Systems Conceptual Map

URL

ConceptualMap.png GitHub file

1.3 Timeline

This is the Operating System Timeline using **draw.io**:

URL

OperatingSystemsTimeLine.png GitHub file

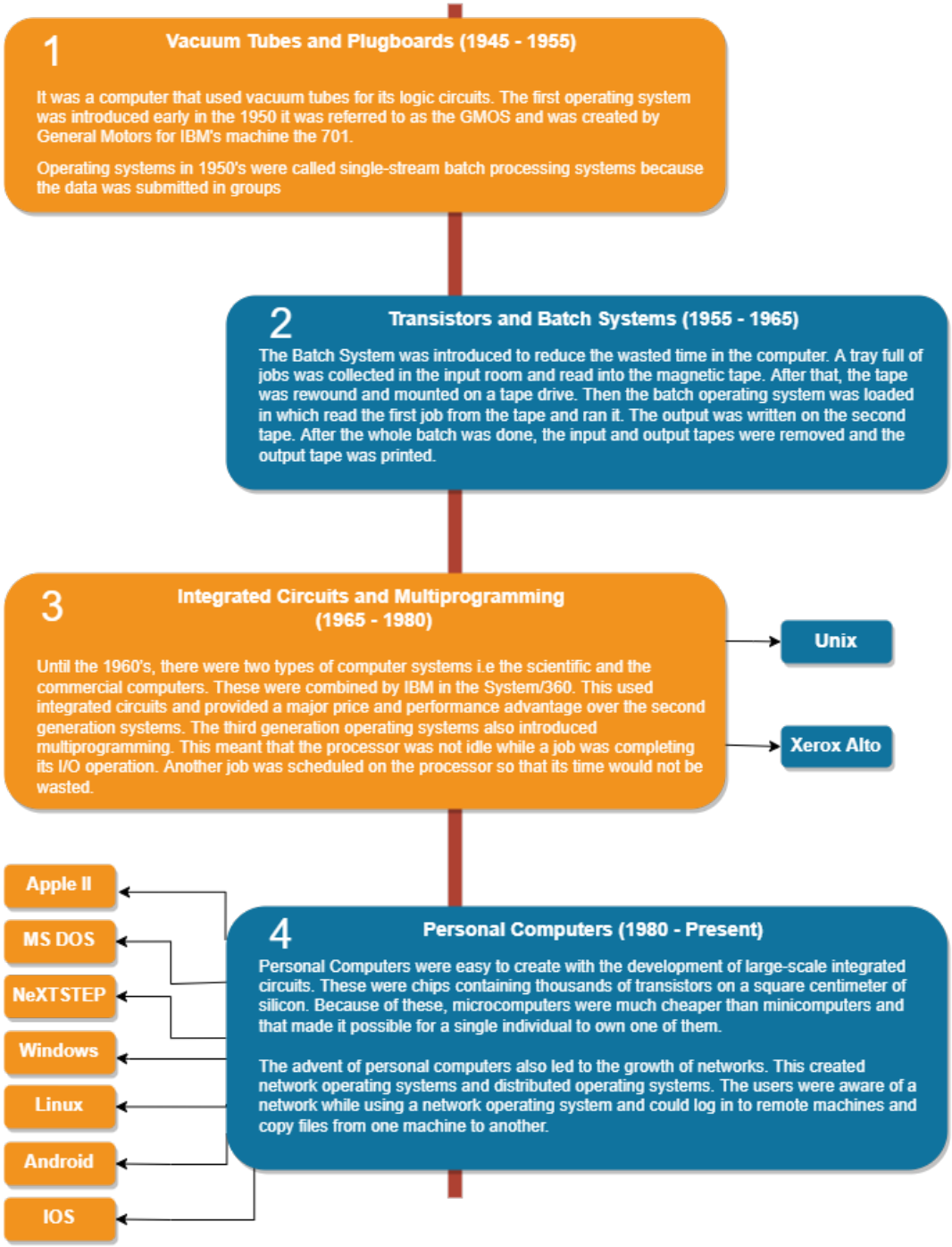


Figure 2: Operating Systems Timeline

2 Operating Systems Interruptions

The idea is just research and implementation of some of the operating system interruptions.

2.1 What is and O.S. Interruption?

An interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention. It alerts the processor to a high-priority process requiring interruption of the current working process.

2.2 Buffer overflow

A buffer overflow occurs when a program or process attempts to write more data to a fixed-length block of memory, or buffer, than the buffer is allocated to hold. Buffers contain a defined amount of data; any extra data will overwrite data values in memory addresses adjacent to the destination buffer.

Figure 3: Buffer overflow diagram, image taken from imperva.com

```c
# https://github.com/Youngermaster/ST0257-Operating-Systems/blob/main/Challenges/Challenge_1/BufferOverflow/buffer_overflow.c

#include <stdio.h>

int main(int argc, char const *argv[]) {
    char *s = "hello world";
    char c = s[20];
    printf("%p - %p -> %d\n", &c, __builtin_frame_address(0), c);
    printf("%p - %p -> %s\n", &s, __builtin_frame_address(0), s);
    printf("%p - %p -> %d\n", &s[20], __builtin_frame_address(0), s[20]);
    return 0;
}
```

Code 1: This script allow us to extract nmap generated info

3 Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflects the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. **EAFIT** prioritized the assessment to identify the weakest security controls an attacker would exploit. **EAFIT** recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

3.1 Confidentiality Statement

This document is the exclusive property of **EAFIT**. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of **EAFIT**.

3.2 Contact info

Contact info.			
Name	Title	Email	Contact
Juan	Cybersecurity Lead	juan@test	(99) 9999 9999
Manuel	Junior Pentester	manuel@test	(99) 9999 9999
Young	Junior Pentester	young@test	(99) 9999 9999

3.3 Assesment overview

From *Date 1* to *Date 2*, **EAFIT** engaged Penetration tests to evaluate the security posture of its infrastructure compared to current industry best practices. All testing performed is based on the NIST SP 800-115 Technical Guide to Information Security Testing and Assessment, OWASP Testing Guide (v4), and customized testing frameworks.

Phases conducted for penetration testing are the following:

- Planning and preparation.
- Reconnaissance / Discovery.
- Vulnerability Enumeration / Analysis.
- Initial Exploitation.
- Expanding Foothold / Deeper Penetration.
- Cleanup.
- Report Generation.

4 Antecedents

The following document takes all the processes and results given by the audit made to the machine **Operating Systems ST0257** from the platform **HackTheBox**.

Figure 4: Details of the machine

URL

https://app.hackthebox.com/machines/98

4.1 considerations

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur molestie nulla nec hendrerit blandit. Aenean euismod tincidunt sapien at lacinia. Praesent at tortor nec nunc sodales posuere in eu mi. Fusce ac fringilla velit. Nam libero leo, consequat sit amet est varius, imperdiet iaculis massa. Suspendisse scelerisque eu erat eu cursus. Vivamus ac euismod ex. Integer euismod sem et euismod mattis. Integer vel rutrum velit, eu dictum nisi. Pellentesque venenatis et nisi a sodales. Integer pretium ut nisi at fringilla. Quisque nunc dui, consequat id turpis non, dignissim consectetur nunc.

4.2 examples

Morbi id turpis bibendum, ultrices turpis eu, molestie nunc. Duis aliquet aliquam turpis, vitae ultricies turpis. Ut tristique elementum nunc ac euismod. Proin viverra ultrices enim, et bibendum sem dignissim venenatis. Pellentesque non lectus nec erat congue viverra feugiat a urna. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Nam vel ex sit amet nulla mollis lacinia. Nulla vehicula orci sit amet fermentum egestas. Mauris blandit ultricies sem, id convallis nunc malesuada non. Morbi venenatis ultricies leo, vel ullamcorper ligula mattis a.

5 Objectives

The idea is to check the machine state of the machine **Operating Systems ST0257**.

5.1 considerations

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur molestie nulla nec hendrerit blandit. Aenean euismod tincidunt sapien at lacinia. Praesent at tortor nec nunc sodales posuere in eu mi. Fusce ac fringilla velit. Nam libero leo, consequat sit amet est varius, imperdiet iaculis massa. Suspendisse scelerisque eu erat eu cursus. Vivamus ac euismod ex. Integer euismod sem et euismod mattis. Integer vel rutrum velit, eu dictum nisi. Pellentesque venenatis et nisi a sodales. Integer pretium ut nisi at fringilla. Quisque nunc dui, consequat id turpis non, dignissim consectetur nunc.

5.2 results

Morbi id turpis bibendum, ultrices turpis eu, molestie nunc. Duis aliquet aliquam turpis, vitae ultricies turpis. Ut tristique elementum nunc ac euismod. Proin viverra ultrices enim, et bibendum sem dignissim venenatis. Pellentesque non lectus nec erat congue viverra feugiat a urna. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Nam vel ex sit amet nulla mollis lacinia. Nulla vehicula orci sit amet fermentum egestas. Mauris blandit ultricies sem, id convallis nunc malesuada non. Morbi venenatis ultricies leo, vel ullamcorper ligula mattis a.

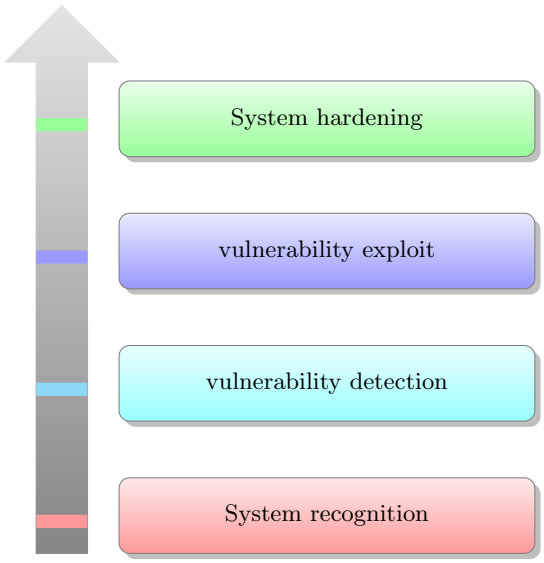


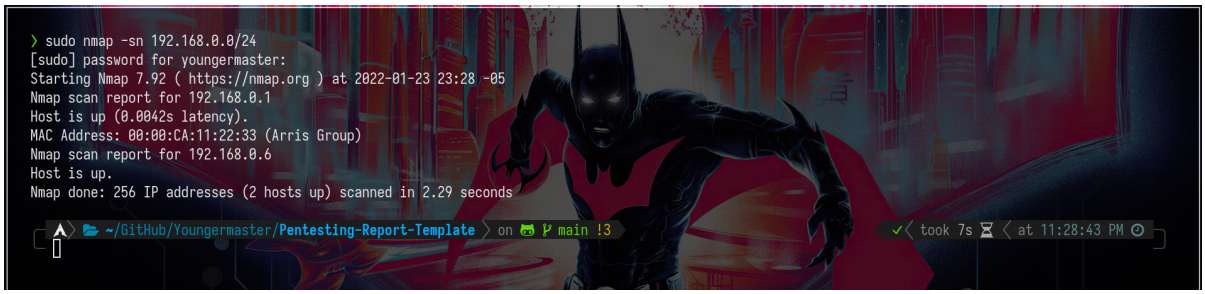
Figure 5: Workflow

6 Vulnerability analysis

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur molestie nulla nec hendrerit blandit. Aenean euismod tincidunt sapien at lacinia. Praesent at tortor nec nunc sodales posuere in eu mi.

6.1 Initial Recognition

Fusce ac fringilla velit. Nam libero leo, consequat sit amet est varius, imperdiet iaculis massa. Suspendisse scelerisque eu erat eu cursus. Vivamus ac euismod ex. Integer euismod sem et euismod mattis. Integer vel rutrum velit, eu dictum nisi. Pellentesque venenatis et nisi a sodales. Integer pretium ut nisi at fringilla. Quisque nunc dui, consequat id turpis non, dignissim consectetur nunc.



6.2 Improvement

Morbi id turpis bibendum, ultrices turpis eu, molestie nunc. Duis aliquet aliquam turpis, vitae ultricies turpis. Ut tristique elementum nunc ac euismod. Proin viverra ultrices enim, et bibendum sem dignissim venenatis. Pellentesque non lectus nec erat congue viverra feugiat a urna. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Nam vel ex sit amet nulla mollis lacinia. Nulla vehicula orci sit amet fermentum egestas. Mauris blandit ultricies sem, id convallis nunc malesuada non. Morbi venenatis ultricies leo, vel ullamcorper ligula mattis a.

```
# This function is taken from S4vitar's blog.
# https://s4vitar.github.io/bspwm-configuration-files/

ports="$(cat $1 | grep -oP '\d{1,5}/open' | awk '{print $1}' FS='/' | xargs | tr ' ' ',')"
ip_address="$(cat $1 | grep -oP '\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}' | sort -u | head -n 1)"
echo -e "\n[*] Extracting information...\n" > extractPorts.tmp
echo -e "\t[*] IP Address: $ip_address"  >> extractPorts.tmp
echo -e "\t[*] Open ports: $ports\n"  >> extractPorts.tmp
echo $ports | tr -d '\n' | xclip -sel clip
echo -e "[*] Ports copied to clipboard\n"  >> extractPorts.tmp
cat extractPorts.tmp; rm extractPorts.tmp
```

Code 2: This script allow us to extract nmap generated info

Donec ut tincidunt dolor. Curabitur sit amet porttitor magna, nec consectetur mi. Praesent quis congue tellus, a tincidunt mauris. Aenean sed luctus enim. Donec ut maximus nisi, sed malesuada erat. Aliquam sollicitudin ullamcorper sem vitae ultrices. Sed iaculis enim egestas, suscipit arcu ac, lacinia risus. Proin scelerisque mi eu feugiat euismod:

TCP

Ports
593, 1337

Vivamus vitae elit porta, tempor justo tincidunt, accumsan ligula. Proin nec magna sit amet leo dignissim sollicitudin sit amet ut quam:

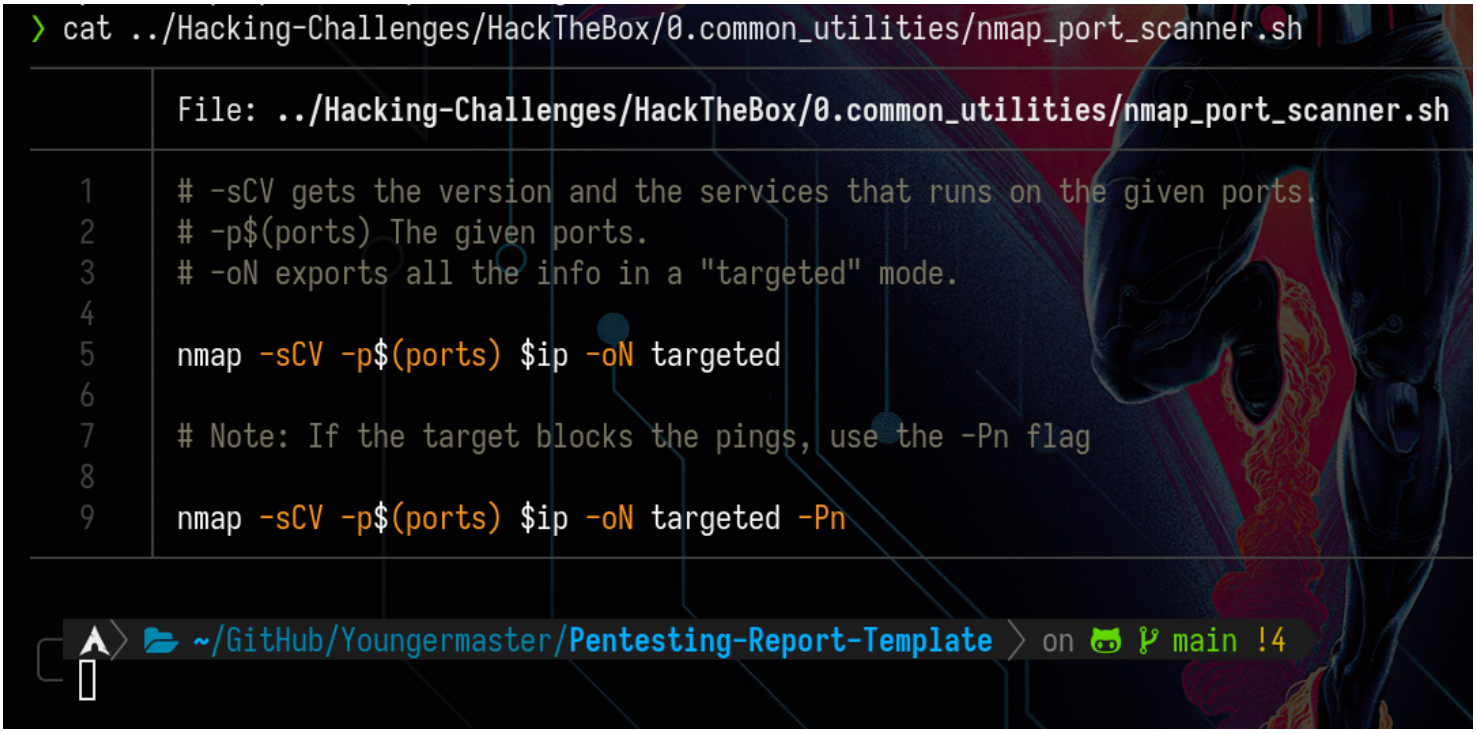


Figure 6: These are the results of lorem ipsum

Ut vulputate fermentum scelerisque 6 and 9. Interdum et malesuada fames ac ante ipsum primis in faucibus.