

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**  
«Операционные системы»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7**  
«Обнаружение работы в виртуальной машине»

**Выполнили:**

Бардышев Артём Антонович,  
студент группы N3246

---

(подпись)

Суханкулиев Мухаммет,  
студент группы N3246



---

(подпись)

**Проверил:**

Савков Сергей Витальевич,  
инженер

---

(отметка о выполнении)

---

(подпись)

Санкт-Петербург  
2024 г.

## СОДЕРЖАНИЕ

Введение .....	3
1      Способы обнаружения работы в виртуальной машине .....	4
1.1    dmesg .....	4
1.2    lscpu .....	4
1.3    BIOS .....	4
1.4    drivers .....	5
1.5    cpuid .....	5
1.6    cpuinfo .....	6
1.7    Сканирование устройств в /sys и /proc .....	6
1.8    virt-what .....	6
1.9    systemd-detect-virt .....	7
1.10   Hardware Lister .....	7
1.11   Проверка сетевых параметров .....	7
1.12   hostnamectl .....	8
1.13   Проверка системных характеристик с использованием WMI (Windows Management Instrumentation) .....	8
1.14   systeminfo .....	9
1.15   msinfo32 .....	9
2      На ассемблере .....	10
2.1    Анализ временных задержек .....	10
2.2    Использование инструкций CPUID .....	10
3      Способ выхода из виртуальной машины .....	11
Заключение .....	12
Список использованных источников .....	13

## **ВВЕДЕНИЕ**

**Цель работы** – перечислить все известные способы обнаружения работы в виртуальной машине. ( $\geq 5$ )

Сложный вариант (или):

- Привести способ выхода из виртуальной машины;
- Выполнить на ассемблере.

# 1 СПОСОБЫ ОБНАРУЖЕНИЯ РАБОТЫ В ВИРТУАЛЬНОЙ МАШИНЕ

## 1.1 dmesg

Команда для вывода буфера сообщений ядра в стандартный поток вывода, может содержать сообщения о виртуализации, если ядро распознаёт гипервизор.

```
(kali@kali)-[~/Desktop/lab7]
$ dmesg | grep -i virtual
[ 0.000000] DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[ 0.345440] Booting paravirtualized kernel on KVM
[ 0.898832] Performance Events: PMU not available due to virtualization, using software events only.
[ 7.755444] usb 1-1: Manufacturer: VirtualBox
[ 7.828390] input: VirtualBox USB Tablet as /devices/pci0000:00/0000:00:06.0/usb1/1-1/1-1:1.0/0003:80EE:0021.0001/input/input6
[ 7.854568] hid-generic 0003:80EE:0021.0001: input,hidraw0: USB HID v1.10 Mouse [VirtualBox USB Tablet] on usb-0000:00:06.0-1/input0
[ 15.354312] systemd[1]: Detected virtualization oracle.
[ 18.800164] input: VirtualBox mouse integration as /devices/pci0000:00/0000:00:04.0/input/input8
[ 166.243877] 05:23:15.888496 main Service: VirtualBox host version check
```

Рисунок 1 – dmesg

## 1.2 lscpu

Команда lscpu покажет сведения о процессоре, включая строки, характерные для виртуализации.

```
(kali@kali)-[~/Desktop/lab7]
$ lscpu | grep -i hypervisor
Flags:                                fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov p
at pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt rdtscp lm constant_tsc rep_good nopl
nonstop_tsc cpuid extd_apicid tsc_known_freq pni pclmulqdq ssse3 cx16 sse4_1 sse4_2 movbe popcnt aes
rdrand hypervisor lahf_lm cmp_legacy cr8_legacy abm sse4a misalignsse 3dnowprefetch vmmcall fsgsbase
bmi1 bmi2 invpcid rdseed clflushopt arat
Hypervisor vendor:                  KVM
```

Рисунок 2 – lscpu

## 1.3 BIOS

Информация о BIOS может содержать строки, характерные для виртуальных машин.

```
(root@kali)-[/home/kali/Desktop/lab7]
# dmidecode -s bios-vendor
dmidecode -s system-product-name

innotek GmbH
VirtualBox
```

Рисунок 3 – dmidecode

## 1.4 drivers

Виртуальные машины часто устанавливают свои драйверы. Эти драйверы можно найти с помощью команды `lsmod`.

```
(root@kali)-[/home/kali/Desktop/lab7]
# lsmod | grep -iE 'vmw|vbox|kvm'

vboxsf                45056      1
vboxguest              53248      6 vboxsf
vmw_vsock_virtio_transport_common  61440      1 vsock_loopback
vmw_vsock_vmci_transport  45056      0
vsock                  61440      5 vmw_vsock_virtio_transport_common,vsock_loopback,vmw_vsock_vmci_trans
port
vmw_vmci               110592      1 vmw_vsock_vmci_transport
vmwgfx                 466944      3
drm_ttm_helper         16384      2 vmwgfx
ttm                    102400      2 vmwgfx,drm_ttm_helper
drm_kms_helper         249856      2 vmwgfx,drm_ttm_helper
drm                    765952      8 vmwgfx,drm_kms_helper,drm_ttm_helper,ttm
```

Рисунок 4 – `lsmod`

## 1.5 cpuid

Утилита `cpuid` позволяет напрямую узнать о наличии гипервизора.

```
(root@kali)-[/home/kali/Desktop/lab7]
# cpuid | grep -i hypervisor

hypervisor guest status = true
hypervisor_id (0x40000000) = "KVMKVMKVM\0\0\0"
hypervisor features (0x40000001/eax):
hypervisor features (0x40000001/edx):
hypervisor_id (0x40000100) = "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
hypervisor guest status = true
hypervisor_id (0x40000000) = "KVMKVMKVM\0\0\0"
hypervisor features (0x40000001/eax):
hypervisor features (0x40000001/edx):
hypervisor_id (0x40000100) = "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
hypervisor guest status = true
hypervisor_id (0x40000000) = "KVMKVMKVM\0\0\0"
hypervisor features (0x40000001/eax):
hypervisor features (0x40000001/edx):
hypervisor_id (0x40000100) = "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
hypervisor guest status = true
hypervisor_id (0x40000000) = "KVMKVMKVM\0\0\0"
hypervisor features (0x40000001/eax):
hypervisor features (0x40000001/edx):
hypervisor_id (0x40000100) = "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
hypervisor guest status = true
hypervisor_id (0x40000000) = "KVMKVMKVM\0\0\0"
hypervisor features (0x40000001/eax):
hypervisor features (0x40000001/edx):
hypervisor_id (0x40000100) = "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
```

Рисунок 5 – `cpuid`

## 1.6 cpuinfo

```
(root@kali)-[/home/kali/Desktop/lab7]
# grep -i hypervisor /proc/cpuinfo

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx
fxsr sse sse2 ht syscall nx mmxext fxsr_opt rdtscp lm constant_tsc rep_good nopl nonstop_tsc cpuid e
xts apicid tsc_known_freq pni pclmulqdq ssse3 cx16 sse4_1 sse4_2 movbe popcnt aes rdrand hypervisor l
ahf_lm cmp_legacy cr8_legacy abm sse4a misalignsse 3dnowprefetch vmmcall fsgsbase bmi1 bmi2 invpcid r
dseed clflushopt arat
```

Рисунок 6 – grep с cpuinfo

## 1.7 Сканирование устройств в /sys и /proc

Файлы и директории в /sys и /proc могут содержать указания на виртуализацию. Например:

```
(root@kali)-[/home/kali/Desktop/lab7]
# ls /proc/sys | grep -i vm
vm
```

Рисунок 7 – /proc/sys

Или проверить информацию о диске:

```
(root@kali)-[/home/kali/Desktop/lab7]
# ls /dev/disk/by-id | grep -i vbox
ata-VBOX_CD-ROM_VB2-01700376
ata-VBOX_HARDDISK_VB704661db-9b980c98
ata-VBOX_HARDDISK_VB704661db-9b980c98-part1
```

Рисунок 8 – /dev/disk

## 1.8 virt-what

Утилита virt-what специально разработана для определения гипервизора.

```
(root@kali)-[/home/kali/Desktop/lab7]
# virt-what
virtualbox
kvm
```

Рисунок 9 – virt-what

## 1.9 systemd-detect-virt

Утилита `systemd-detect-virt` входит в состав большинства современных Linux-дистрибутивов. Она выдаёт название гипервизора, если система виртуализована.

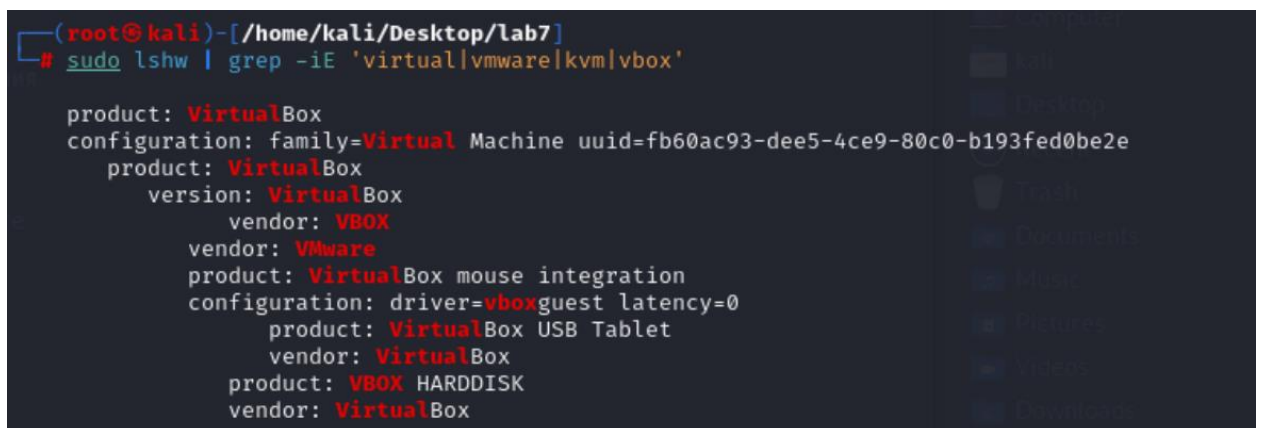


```
(root@kali)-[/home/kali/Desktop/lab7]
# systemd-detect-virt
oracle
```

Рисунок 10 – `systemd-detect-virt`

## 1.10 Hardware Lister

Утилита `lshw` показывает подробную информацию о системе, включая признаки виртуализации.



```
(root@kali)-[/home/kali/Desktop/lab7]
# sudo lshw | grep -iE 'virtual|vmware|kvm|vbox'
product: VirtualBox
configuration: family=Virtual Machine uuid=fb60ac93-dee5-4ce9-80c0-b193fed0be2e
product: VirtualBox
version: VirtualBox
vendor: VBOX
vendor: VMware
product: VirtualBox mouse integration
configuration: driver=vboxguest latency=0
product: VirtualBox USB Tablet
vendor: VirtualBox
product: VBOX HARDDISK
vendor: VirtualBox
```

Рисунок 11 – `lshw`

## 1.11 Проверка сетевых параметров

Многие гипервизоры назначают MAC-адреса с определёнными префиксами. Проверить MAC-адрес можно командой:



```
(root@kali)-[/home/kali/Desktop/lab7]
# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
   link/ether 08:00:27:ad:25:87 brd ff:ff:ff:ff:ff:ff
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT group default qlen 1000
   link/ether 02:42:66:0a:58:dc brd ff:ff:ff:ff:ff:ff
```

Рисунок 12 – `ip`



Примеры известных префиксов MAC-адресов:

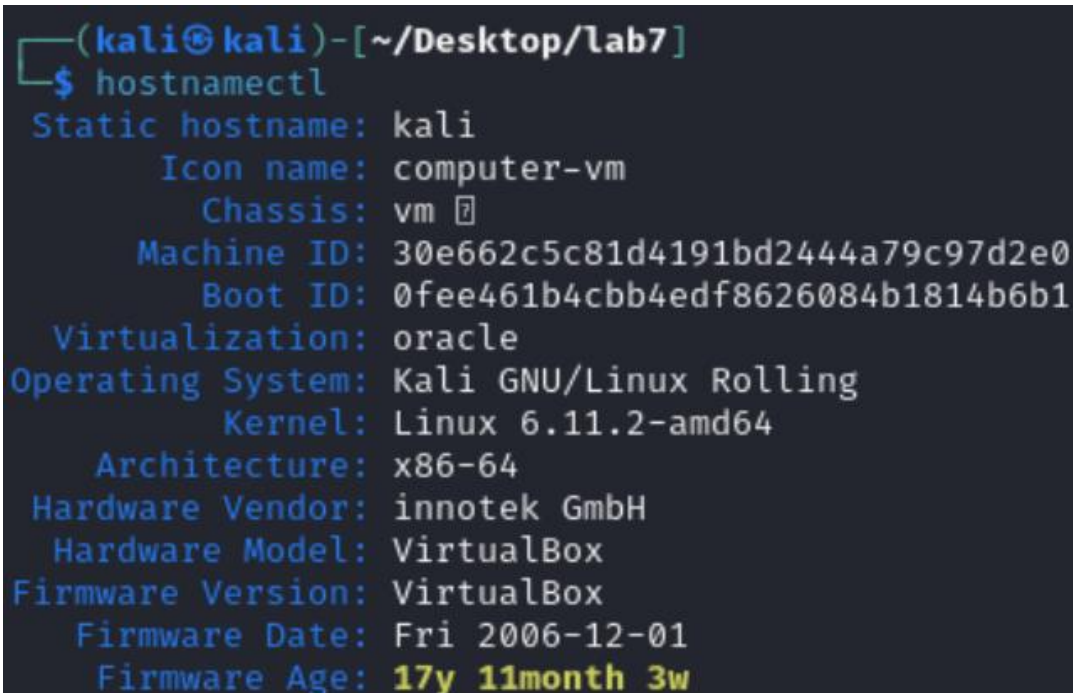
00:50:56 — VMware.

08:00:27 — VirtualBox.

52:54:00 — QEMU/KVM.

## 1.12 hostnamectl

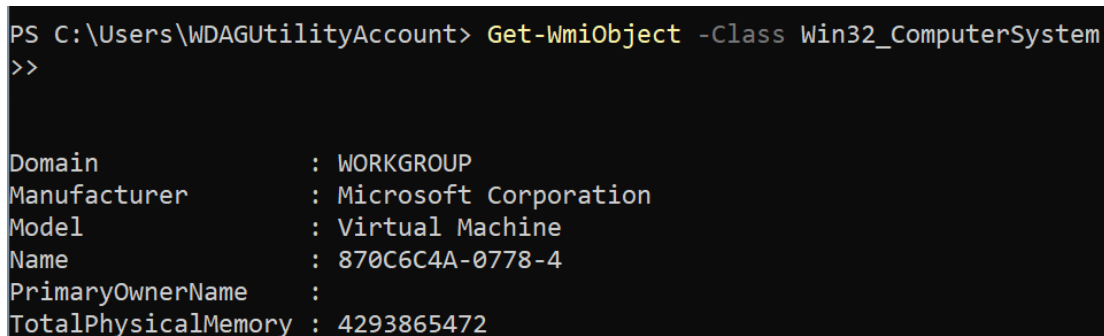
hostnamectl – управляет именем компьютера.



```
(kali@kali)-[~/Desktop/lab7]
$ hostnamectl
Static hostname: kali
Icon name: computer-vm
Chassis: vm
Machine ID: 30e662c5c81d4191bd2444a79c97d2e0
Boot ID: 0fee461b4cbb4edf8626084b1814b6b1
Virtualization: oracle
Operating System: Kali GNU/Linux Rolling
Kernel: Linux 6.11.2-amd64
Architecture: x86-64
Hardware Vendor: innotek GmbH
Hardware Model: VirtualBox
Firmware Version: VirtualBox
Firmware Date: Fri 2006-12-01
Firmware Age: 17y 11month 3w
```

Рисунок 13 – hostnamectl

## 1.13 Проверка системных характеристик с использованием WMI (Windows Management Instrumentation)



```
PS C:\Users\WDAGUtilityAccount> Get-WmiObject -Class Win32_ComputerSystem
>>

Domain                : WORKGROUP
Manufacturer          : Microsoft Corporation
Model                 : Virtual Machine
Name                  : 870C6C4A-0778-4
PrimaryOwnerName      :
TotalPhysicalMemory    : 4293865472
```

Рисунок 14 – WMI



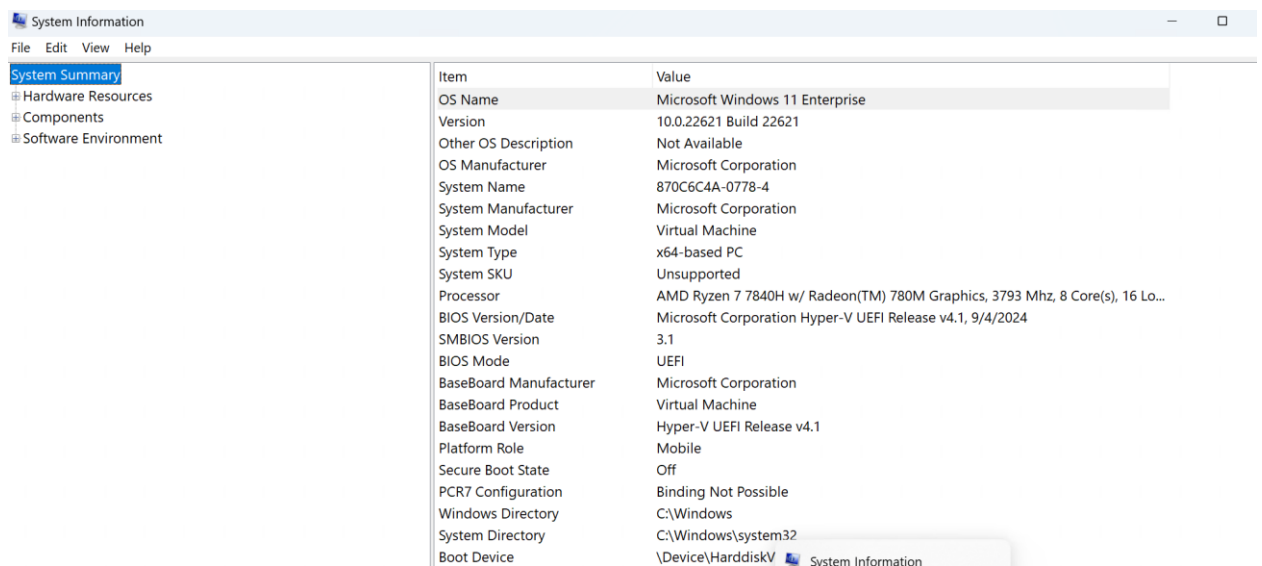
## 1.14 systeminfo

```
PS C:\Users\WDAGUtilityAccount> systeminfo

Host Name:                               870C6C4A-0778-4
OS Name:                                  Майкрософт Windows 11 Корпоративная
OS Version:                              10.0.22621 N/A Build 22621
OS Manufacturer:                         Microsoft Corporation
OS Configuration:                        Standalone Workstation
OS Build Type:                             Multiprocessor Free
Registered Owner:                          N/A
Registered Organization:                   N/A
Product ID:                               00328-90000-00000-AAOEM
Original Install Date:                     1/1/1970, 3:00:00 AM
System Boot Time:                          11/21/2024, 9:36:00 AM
System Manufacturer:                       Microsoft Corporation
System Model:                               Virtual Machine
System Type:                               x64-based PC
Processor(s):                              1 Processor(s) Installed.
[01]: AMD64 Family 25 Model 116 Stepping 1 AuthenticAMD ~3793 Mhz
BIOS Version:                              Microsoft Corporation Hyper-V UEFI Release v4.1, 9/4/2024
Windows Directory:                         C:\Windows
System Directory:                          C:\Windows\system32
Boot Device:                               \Device\HarddiskVolume2
System Locale:                              ru;Russian
Input Locale:                              en-us;English (United States)
Time Zone:                                  N/A
Total Physical Memory:                      4,095 MB
Available Physical Memory:                  2,781 MB
Virtual Memory: Max Size:                   5,823 MB
Virtual Memory: Available:                  4,355 MB
Virtual Memory: In Use:                     1,468 MB
Page File Location(s):                      C:\pagefile.sys
Domain:                                     WORKGROUP
Logon Server:                               \\870C6C4A-0778-4
Hotfix(s):                                  N/A
Network Card(s):                           1 NIC(s) Installed.
[01]: Microsoft Hyper-V Network Adapter
      Connection Name: Ethernet
      DHCP Enabled:      Yes
      DHCP Server:       172.27.240.1
      IP address(es)
      [01]: 172.27.242.17
      [02]: fe80::da13:ca1:9c8d:a52e
Hyper-V Requirements:                       A hypervisor has been detected. Features required for Hyper-V will not be displayed.
```

Рисунок 15 – systeminfo

## 1.15 msinfo32



Item	Value
OS Name	Microsoft Windows 11 Enterprise
Version	10.0.22621 Build 22621
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	870C6C4A-0778-4
System Manufacturer	Microsoft Corporation
System Model	Virtual Machine
System Type	x64-based PC
System SKU	Unsupported
Processor	AMD Ryzen 7 7840H w/ Radeon(TM) 780M Graphics, 3793 Mhz, 8 Core(s), 16 Lo...
BIOS Version/Date	Microsoft Corporation Hyper-V UEFI Release v4.1, 9/4/2024
SMBIOS Version	3.1
BIOS Mode	UEFI
BaseBoard Manufacturer	Microsoft Corporation
BaseBoard Product	Virtual Machine
BaseBoard Version	Hyper-V UEFI Release v4.1
Platform Role	Mobile
Secure Boot State	Off
PCR7 Configuration	Binding Not Possible
Windows Directory	C:\Windows
System Directory	C:\Windows\system32
Boot Device	\Device\HarddiskV

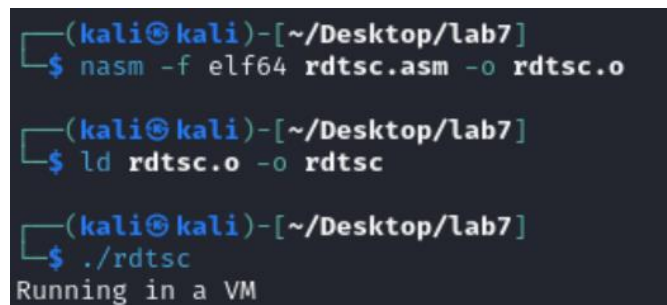
Рисунок 16 – msinfo32

## 2 НА АССЕМБЛЕРЕ

### 2.1 Анализ временных задержек

Виртуальные машины часто эмулируют процессор и другие компоненты, что может замедлить выполнение операций. Алгоритм ассемлера:

Использование инструкции `rdtsc` (чтение таймера процессора) для измерения времени до и после выполнения цикла с нагрузкой. Разница во времени сравнивается с порогом (1000000 тактов). Если разница больше порога, это указывает на возможное выполнение в виртуальной машине, так как виртуализация может замедлять выполнение инструкций. Если разница меньше порога, выводится сообщение о работе на реальном железе.



```
(kali@kali)-[~/Desktop/lab7]
$ nasm -f elf64 rdtsc.asm -o rdtsc.o

(kali@kali)-[~/Desktop/lab7]
$ ld rdtsc.o -o rdtsc

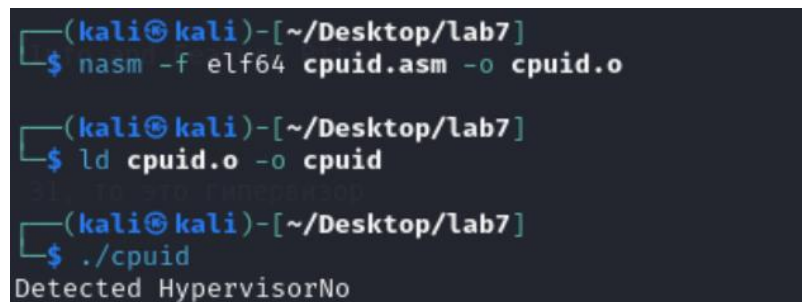
(kali@kali)-[~/Desktop/lab7]
$ ./rdtsc
Running in a VM
```

Рисунок 17 – rdtsc.asm

### 2.2 Использование инструкций `CPUID`

Команда `CPUID` позволяет определить возможности процессора. Если использовать её с определёнными параметрами, можно получить признаки виртуализации:

Используем инструкцию `CPUID` с параметром `EAX = 1`, чтобы получить информацию о процессоре, включая бит "Hypervisor present" в регистре `ECX` (бит 31). Если этот бит установлен, значит, на машине работает гипервизор, и программа выводит сообщение "Detected Hypervisor". В противном случае выводится "No Hypervisor Detected".



```
(kali@kali)-[~/Desktop/lab7]
$ nasm -f elf64 cpuid.asm -o cpuid.o

(kali@kali)-[~/Desktop/lab7]
$ ld cpuid.o -o cpuid

(kali@kali)-[~/Desktop/lab7]
$ ./cpuid
Detected HypervisorNo
```

Рисунок 18 – cpuid.asm

### 3 СПОСОБ ВЫХОДА ИЗ ВИРТУАЛЬНОЙ МАШИНЫ

Если цель — просто завершить работу VM, можно использовать стандартные команды, например:

```
sudo shutdown now
shutdown -s -t 0
```

Это приведёт к выключению виртуальной машины.

#### Эксплуатация уязвимостей гипервизора.

VirtualBox и его компоненты, такие как Guest Additions, подвержены уязвимостям, особенно в драйверах, которые взаимодействуют с хост-системой. Например, одна из известных уязвимостей:

CVE-2018-2844 — переполнение буфера в обработке I/O-запросов в драйверах VirtualBox.

Пример кода для вызова функции I/O в Guest Additions (на C):

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>

int main() {
    int fd = open("/dev/vboxguest", O_RDWR);
    if (fd < 0) {
        perror("Failed to open /dev/vboxguest");
        return 1;
    }
    char buffer[64];
    memset(buffer, 'A', 128);
    write(fd, buffer, 128);
    close(fd);
    return 0;
}
```

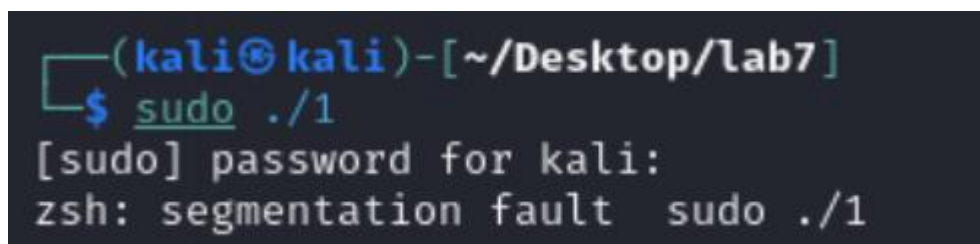


Рисунок 19 – Переполнение буфера

## **ЗАКЛЮЧЕНИЕ**

В ходе лабораторной работы были рассмотрены различные методы обнаружения работы в виртуальной машине. Были использованы как стандартные системные команды, так и более сложные методы, такие как анализ временных задержек с помощью инструкций RDTSC и использование CPUID для определения гипервизора. Эти подходы позволяют эффективно выявить виртуализацию как на уровне операционной системы, так и на уровне аппаратных средств.

Также была рассмотрена эксплуатация уязвимостей гипервизора, таких как переполнение буфера в драйверах VirtualBox, что демонстрирует важность защиты виртуализированных сред.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. [Oracle VirtualBox](#)
2. [dmesg\(1\) - Linux manual page](#)
3. [CVE Record | CVE](#)
4. [9 команд для проверки информации о CPU в Linux / Хабр](#)
5. [Команда lshw | Linux FAQ](#)
6. [virtualization - What range of MAC addresses can I safely use for my virtual machines? - Server Fault](#)
7. [Настройка пулов MAC-адресов в сетевой структуре VMM | Microsoft Learn](#)
8. [What is Prefix-Based MAC Address Allocation](#)
9. [CPUID — Википедия](#)
10. [Rdtsc — Википедия](#)