

# **РАЗВЕРТЫВАНИЕ И НАСТРОЙКА СКАНЕРОВ УЯЗВИМОСТИ НА ОТЧЕСТВЕННЫХ ОС: CHKROOTKIT, LYNIS, RKHUNTER**

*Богданова Полина Александровна*

*Баев Ярослав Сергеевич*

*Студенты*

*Научный руководитель: Уймин Антон Григорьевич*

*Старший преподаватель*

*ФГАОУ ВО «РГУ НЕФТИ И ГАЗА ИМ. И. М. ГУБКИНА»*

*Аннотация:*

*В статье рассматривается процесс развертывания и установки сканеров уязвимости Chkrootkit, Lynis, Rkhunter на ОС таких как Alt Рабочая станция и Астра Linux. Рассматриваются специфические аспекты установки и конфигурации каждого из указанных инструментов. Приведены практические примеры использования сканеров в различных сценариях. Реализовано комбинирование и автоматизация сканеров Chkrootkit и Rkhunter.*

*Ключевые слова: Chkrootkit, Lynis, Rkhunter, сканер уязвимости, Alt Рабочая станция, Астра Linux.*

# **DEPLOYMENT AND CONFIGURATION OF VULNERABILITY SCANNERS ON DOMESTIC OPERATING SYSTEMS: CHKROOTKIT, LYNIS, RKHUNTER**

*Bogdanova Polina Aleksandrovna*

*Baev Yaroslav Sergeevich*

*Students*

*Supervisor: Uymin Anton Grigorievich*

*Senior Lecturer*

*Gubkin Russian State University of Oil and Gas*

*Annotation:*

*The article discusses the deployment and installation process of vulnerability scanners Chkrootkit, Lynis, and Rkhunter on operating systems such as Alt Workstation and Astra Linux. It examines the specific aspects of the installation and configuration of each of the mentioned tools. Practical examples of using the scanners in various scenarios are provided. The combination and automation of Chkrootkit and Rkhunter scanners have been implemented.*

*Keywords: Chkrootkit, Lynis, Rkhunter, vulnerability scanner, Alt Workstation, Astra Linux.*

## ВВЕДЕНИЕ

С ростом информационных технологий и увеличением количества цифровых угроз вопрос обеспечения безопасности операционных систем стал приоритетным для государственных и частных компаний. Операционные системы типа Alt Рабочая станция разработаны с учетом национальных стандартов безопасности и применяются в различных отраслях - от государственного сектора до критической инфраструктуры. Несмотря на встроенные меры защиты, операционные системы остаются уязвимыми перед киберугрозами вроде руткитов, вредоносного ПО и атак на уязвимости систем. Поэтому важно регулярно проверять безопасность и осуществлять анализ на предмет уязвимостей для выявления и устранения потенциальных опасностей вовремя.

В данной статье рассматривается процесс развертывания и настройки инструментов для поиска уязвимостей и руткитов: Chkrootkit, Lynis и Rkhunter на платформах Alt Рабочая станция и Астра Linux. В работе будут рассмотрены возможности каждого инструмента, а также проведено

тестирование их эффективности и совместимости с данной операционной системой.

**Chkrootkit** (Check Rootkit) — это удобный для использования инструмент командной строки, разработанный для обнаружения руткитов и связанных угроз. Его основная функция заключается в анализе системных файлов и процессов на предмет наличия признаков руткитов.

**Lynis** — это продвинутый инструмент аудита безопасности, ориентированный на обнаружение уязвимостей и конфигурационных ошибок, обеспечивающий комплексный анализ системы. Он проводит аудит безопасности и выявляет уязвимые конфигурации, ошибки и рекомендации для их исправления.

**Rootkit Hunter** (Rkhunter) — это сканер, предназначенный для обнаружения руткитов, бекдоров и других уязвимостей. Rkhunter фокусируется на целостности системы, обнаружении подозрительных файлов и анализе изменений в системных конфигурациях.

Сводная таблица основных характеристик:

Таблица 1 — Сводная таблица характеристик

Название	Разработчик	Цель разработки	Сфера применения	Лицензия	Использование в коммерческих продуктах
Chkrootkit	Pangeia Informatica	Обнаружение руткитов в UNIX/Linux	Серверы, рабочие станции, ОС на основе UNIX	GPLv2	Не входит в коммерческие продукты напрямую
Lynis	CISOfy	Аудит безопасности и соответствия стандартам	Организации, аудиторы ИБ	GPLv3	Входит в состав некоторых продуктов аудита
RKHunter	Michael	Обнаружение	Домашние	GPLv2	Не

	Boelen	руткитов, троянов и вредоносных программ	системы, серверы, критически важные системы		используется в коммерческих продуктах напрямую
--	--------	--	---	--	--

Для безопасного тестирования создадим виртуальную тестовую среду.

Установим Alt Рабочая станция и Астра Linux на виртуальной машине через VirtualBox с базовыми параметрами (2 CPU, 4 ГБ RAM, 20 ГБ диск) для проведения тестов.

Перейдем к описанию эксперимента.

## ПОРЯДОК ЭКСПЕРИМЕНТА

1. Прежде всего необходимо подготовить и настроить сканеры.

1.1. Подготовка к установке сканеров.

Убедимся, что все зависимости и дополнительные утилиты для корректной работы сканеров установлены. Выполним следующие команды:

Для Alt Рабочая станция:

*apt-get install gcc make glibc bash perl wget curl*

```

24: gcc10-10.3.1-alt2 ##### [ 55%]
25: gcc-10-alt1 ##### [ 57%]
26: i586-glibc-pthread-6:2.32-alt5.p10.3##### [ 59%]
27: curl-8.7.1-alt2 ##### [ 61%]
28: libnsl1-6:2.32-alt5.p10.3 ##### [ 64%]
29: wget-1.24.5-alt4 ##### [ 66%]
Очистка / удаление...
30: curl-8.5.0-alt1 ##### [ 68%]
31: libcurl-8.5.0-alt1 ##### [ 70%]
32: i586-glibc-pthread-6:2.32-alt5.p10.2##### [ 73%]
33: i586-glibc-core-6:2.32-alt5.p10.2 ##### [ 75%]
34: wget-1.21.3-alt1 ##### [ 77%]
35: glibc-nss-6:2.32-alt5.p10.2 ##### [ 80%]
36: glibc-timezones-6:2.32-alt5.p10.2 ##### [ 82%]
37: iconv-6:2.32-alt5.p10.2 ##### [ 84%]
38: glibc-gconv-modules-6:2.32-alt5.p10.2##### [ 86%]
39: glibc-utils-6:2.32-alt5.p10.2 ##### [ 89%]
40: glibc-locales-6:2.32-alt5.p10.2 ##### [ 91%]
41: glibc-pthread-6:2.32-alt5.p10.2 ##### [ 93%]
42: libnsl1-6:2.32-alt5.p10.2 ##### [ 95%]
43: glibc-core-6:2.32-alt5.p10.2 ##### [ 98%]
44: glibc-preinstall-6:2.32-alt5.p10.2 ##### [100%]
Завершено.
[root@vbox ~]#

```

Рисунок 1 — Установка зависимостей и утилит. Alt Рабочая станция

Для Астра Linux:

```
sudo apt install gcc make glibc bash perl wget curl
```

```
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
E: Не удалось найти пакет glibc
user@astra:~$ sudo apt install gcc make bash perl wget curl
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Уже установлен пакет bash самой новой версии (4.4-5.astra2).
Уже установлен пакет curl самой новой версии (7.52.1-5+deb9u16).
curl установлен вручную.
Уже установлен пакет gcc самой новой версии (4:6.3.0-4).
Уже установлен пакет make самой новой версии (4.1-9.1).
make установлен вручную.
Уже установлен пакет perl самой новой версии (5.24.1-3+deb9u7).
perl установлен вручную.
Уже установлен пакет wget самой новой версии (1.18-5+deb9u3).
обновлено 0, установлено 0 новых пакетов, для удаления отмечено 0 пакетов,
и 0 пакетов не обновлено.
user@astra:~$
```

Рисунок 2 — Установка зависимостей и утилит. Астра Linux

## 1.2. Выполним обновление для установки последних версий

пакетов, необходимых для стабильной работы сканеров:

Для Alt Рабочая станция:

```
apt-get update
```

```
[root@vbox ~]# apt-get update
Получено: 1 http://ftp.altlinux.org p10/branch/x86_64 release [4223B]
Получено: 2 http://ftp.altlinux.org p10/branch/x86_64-i586 release [1665B]
Получено: 3 http://ftp.altlinux.org p10/branch/noarch release [2844B]
Получено 8732B за 2s (3528B/s).
Найдено http://ftp.altlinux.org p10/branch/x86_64/classic pkglist
Найдено http://ftp.altlinux.org p10/branch/x86_64/classic release
Найдено http://ftp.altlinux.org p10/branch/x86_64-i586/classic pkglist
Найдено http://ftp.altlinux.org p10/branch/x86_64-i586/classic release
Найдено http://ftp.altlinux.org p10/branch/noarch/classic pkglist
Найдено http://ftp.altlinux.org p10/branch/noarch/classic release
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
[root@vbox ~]#
```

Рисунок 3 — Обновление операционной системы. Alt Рабочая станция

Для Астра Linux:

```
sudo apt update
```

```
user@astra:~$ sudo apt update

Мы полагаем, что Ваш системный администратор изложил Вам основы
безопасности. Как правило, всё сводится к трём следующим правилам:

    №1) Уважайте частную жизнь других.
    №2) Думайте, прежде что-то вводить.
    №3) С большой властью приходит большая ответственность.

[sudo] пароль для user:
Пол:1 https://dl.astralinux.ru/astra/stable/2.12_x86-64/repository ore1 In
Release [13,1 kB]
Пол:2 https://dl.astralinux.ru/astra/stable/2.12_x86-64/repository ore1/ma
in amd64 Packages [4 103 kB]
Пол:3 https://dl.astralinux.ru/astra/stable/2.12_x86-64/repository ore1/ma
in i386 Packages [508 kB]
Пол:4 https://dl.astralinux.ru/astra/stable/2.12_x86-64/repository ore1/co
ntrib amd64 Packages [4 458 B]
Пол:5 https://dl.astralinux.ru/astra/stable/2.12_x86-64/repository ore1/co
ntrib i386 Packages [1 174 B]
```

Рисунок 4 — Обновление операционной системы. Астра Linux

## 2. Установка и настройка сканеров.

### 2.1. Установка и настройка Chkrootkit. Выполним:

Для Alt Рабочая станция:

*apt-get update*

*apt-get install chkrootkit -y*

```
[root@vbox ~]# apt-get install chkrootkit -y
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие НОВЫЕ пакеты будут установлены:
  chkrootkit
0 будет обновлено, 1 новых установлено, 0 пакетов будет удалено и 386 не будет о
бновлено.
Необходимо получить 300кВ архивов.
После распаковки потребуется дополнительно 1095кВ дискового пространства.
Получено: 1 http://ftp.altlinux.org p10/branch/x86_64/classic chkrootkit 0.58b-a
lt1:p10+341767.100.1.1@1709302802 [300кВ]
Получено 300кВ за 6s (49,1кВ/s).
Совершаем изменения...
Подготовка... ##### [100%]
Обновление / установка...
1: chkrootkit-0.58b-alt1 ##### [100%]
Завершено.
```

Рисунок 5 — Установка Chkrootkit. Alt Рабочая станция

Для Астра Linux:

*sudo apt install chkrootkit*

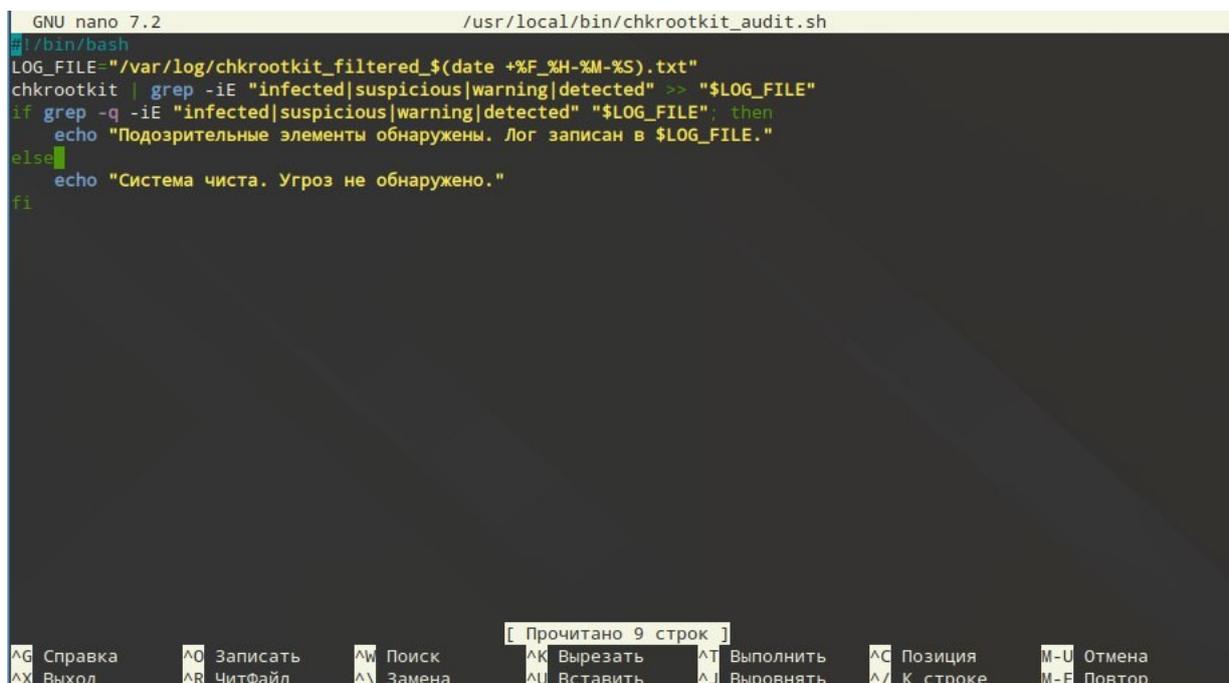
```
user@astra:~$ sudo apt install chkrootkit
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Уже установлен пакет chkrootkit самой новой версии (0.50-4+deb9u1),
обновлено 0, установлено 0 новых пакетов, для удаления отмечено 0 пакетов,
и 0 пакетов не обновлено.
```

Рисунок 6 — Установка Chkrootkit. Астра Linux

2.1.1. Запуск Chkrootkit и создание лог-файла. Для того чтобы автоматизированно сканировать систему и сохранять только подозрительные строки (например, содержащие INFECTED, warning или `suspicious`), используется следующий скрипт:

Для Alt Рабочая станция:

```
#!/bin/bash
LOG_FILE="/var/log/chkrootkit_filtered_$(date +%F_%H-%M-%S).txt"
chkrootkit | grep -iE "infected|suspicious|warning|detected" >> "$LOG_FILE"
if grep -q -iE "infected|suspicious|warning|detected" "$LOG_FILE"; then
    echo "Подозрительные элементы обнаружены. Лог записан в $LOG_FILE."
else
    echo "Система чиста. Угроз не обнаружено."
fi
```



```
GNU nano 7.2 /usr/local/bin/chkrootkit_audit.sh
#!/bin/bash
LOG_FILE="/var/log/chkrootkit_filtered_$(date +%F_%H-%M-%S).txt"
chkrootkit | grep -iE "infected|suspicious|warning|detected" >> "$LOG_FILE"
if grep -q -iE "infected|suspicious|warning|detected" "$LOG_FILE"; then
    echo "Подозрительные элементы обнаружены. Лог записан в $LOG_FILE."
else
    echo "Система чиста. Угроз не обнаружено."
fi
```

Рисунок 7 — Скрипт автоматического сканирования для Alt Рабочая станция

Для Астра Linux:

```
#!/bin/bash
LOG_FILE="/var/log/chkrootkit_filtered_$(date +%F_%H-%M-%S).txt"
sudo chkrootkit | grep -iE "infected|suspicious|warning|detected" | grep >>
"$LOG_FILE"
if grep -q -iE "infected|suspicious|warning|detected" "$LOG_FILE"; then
    echo "Подозрительные элементы обнаружены. Лог записан в $LOG_FILE."
else
    echo "Система чиста. Угроз не обнаружено."
fi
```

Рисунок 8 — Скрипт автоматического сканирования для Астра Linux

### 2.1.2. И сделаем скрипт исполняемым:

Для Alt Рабочая станция:

```
chmod +x /usr/local/bin/chkrootkit_audit.sh
```

Для Астра Linux:

```
sudo chmod +x /usr/local/bin/chkrootkit_audit.sh
```

## 2.2. Установка и настройка Lynis

Установим Lynis из стандартного репозитория для Альт Рабочая станция. Однако Лайнис для Астра Линукс нельзя так загрузить, поэтому

установим его с помощью официального репозитория Lynis на git.hub вручную:

Для Alt Рабочая станция:

*apt-get install lynis*

```
[root@vbox ~]# apt-get install lynis
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие НОВЫЕ пакеты будут установлены:
  lynis
0 будет обновлено, 1 новых установлено, 0 пакетов будет удалено и 386 не будет о
бновлено.
Необходимо получить 269кВ архивов.
После распаковки потребуется дополнительно 1741кВ дискового пространства.
Получено: 1 http://ftp.altlinux.org p10/branch/noarch/classic lynis 3.0.9-alt1:p
10+341796.100.1.1@1709312188 [269кВ]
Получено 269кВ за 5s (50,2кВ/s).
Совершаем изменения...
Подготовка... ##### [100%]
Обновление / установка...
1: lynis-3.0.9-alt1 ##### [100%]
```

Рисунок 9 — Установка Lynis для Alt Рабочая станция

Для Астра Linux:

*wget <https://github.com/CISOfy/lynis/archive/refs/tags/3.0.0.tar.gz>*

*tar -xvzf 3.0.0.tar.gz*

```
user@astra:~$ wget https://github.com/CISOfy/lynis/archive/refs/tags/3.0.0
.
.tar.gz
--2025-01-15 16:34:58-- https://github.com/CISOfy/lynis/archive/refs/tags
/3.0.0.tar.gz
Распознаётся github.com (github.com)... 140.82.121.4
Подключение к github.com (github.com)|140.82.121.4|:443... соединение уста
новлено.
HTTP-запрос отправлен. Ожидание ответа... 302 Found
Адрес: https://codeload.github.com/CISOfy/lynis/tar.gz/refs/tags/3.0.0 [ne
rehog]
--2025-01-15 16:35:06-- https://codeload.github.com/CISOfy/lynis/tar.gz/r
efs/tags/3.0.0
Распознаётся codeload.github.com (codeload.github.com)... 140.82.121.9
Подключение к codeload.github.com (codeload.github.com)|140.82.121.9|:443.
... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: нет данных [application/x-gzip]
Сохранение в: «3.0.0.tar.gz»
```

```
user@astra:~$ tar -xvzf 3.0.0.tar.gz
lynis-3.0.0/
lynis-3.0.0/.github/
lynis-3.0.0/.github/ISSUE_TEMPLATE/
lynis-3.0.0/.github/ISSUE_TEMPLATE/bug_report.md
lynis-3.0.0/.github/ISSUE_TEMPLATE/feature_request.md
lynis-3.0.0/.github/workflows/
```

Рисунок 10 — Установка Lynis для Астра Linux

Убедимся, что сканер успешно установлен:

Для Alt Рабочая станция: *lynis show version*

```
[root@vbox ~]# lynis show version
3.0.9
```

Рисунок 11 — Активная версия Lynis для Alt Рабочая станция

Для Астра Linux: *./lynis --version*

```
user@astra:~/lynis-3.0.0$ ./lynis --version
3.0.0
user@astra:~/lynis-3.0.0$
```

Рисунок 12 — Активная версия Lynis для Астра Linux

2.2.1. Настроим, чтобы сканер реагировал на слова «ПРЕДУПРЕЖДЕНИЕ|УЯЗВИМО|НЕБЕЗОПАСНО|ПРЕДЛОЖЕНИЕ»

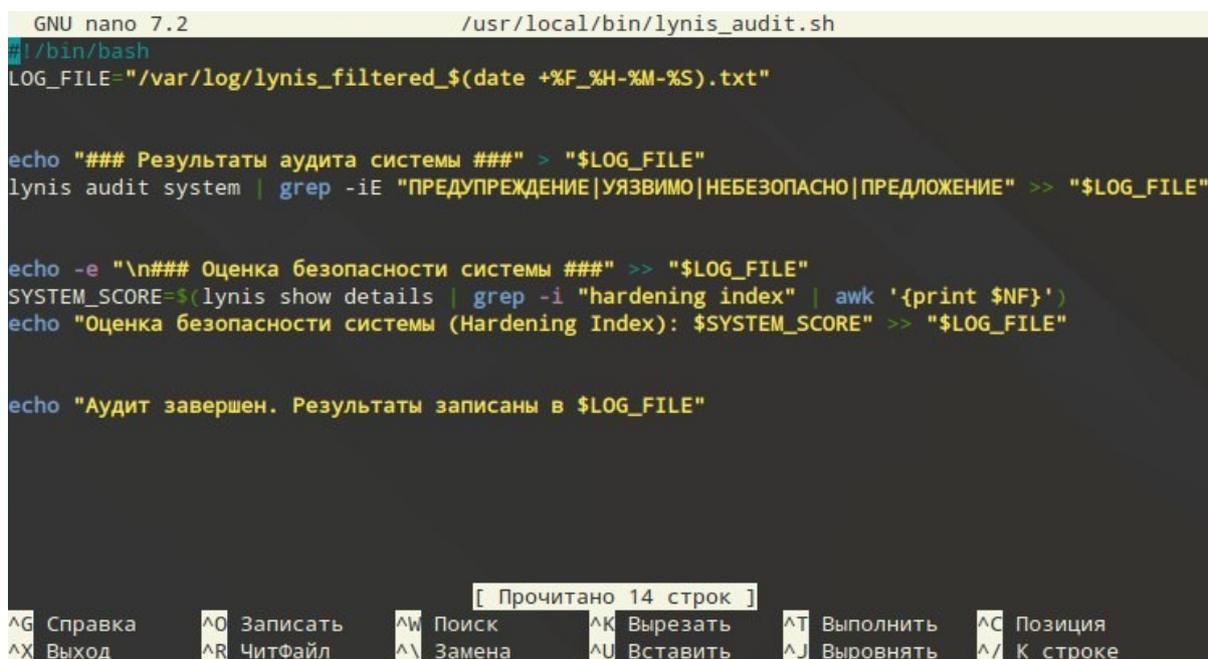
Для того чтобы сохранять только подозрительные, создадим bash-скрипт, который будет автоматически запускать сканирование и запись по ключевым словам и выводить оценку безопасности:

Для Альт Рабочая станция:

```
#!/bin/bash
LOG_FILE="/var/log/lynis_filtered_$(date +%F_%H-%M-%S).txt"
echo "### Результаты аудита системы ###" > "$LOG_FILE"
lynis audit system | grep -iE "ПРЕДУПРЕЖДЕНИЕ|УЯЗВИМОСТЬ|
НЕБЕЗОПАСНО|ПРЕДЛОЖЕНИЕ" >> "$LOG_FILE"
echo -e "\n### ###" >> "$LOG_FILE"
SYSTEM_SCORE=$(lynus show details | grep -I "hardening index" | awk '{print
$NF}')
```

```
echo "Оценка безопасности системы (Hardening Index): $SYSTEM_SCORE" >>
"$LOG_FILE"
```

```
echo "Аудит завершен. Подозрительные операции записаны в $LOG_FILE"
```



```
GNU nano 7.2 /usr/local/bin/lynis_audit.sh
#!/bin/bash
LOG_FILE="/var/log/lynis_filtered_$(date +%F_%H-%M-%S).txt"

echo "### Результаты аудита системы ###" > "$LOG_FILE"
lynis audit system | grep -iE "ПРЕДУПРЕЖДЕНИЕ|УЯЗВИМО|НЕБЕЗОПАСНО|ПРЕДЛОЖЕНИЕ" >> "$LOG_FILE"

echo -e "\n### Оценка безопасности системы ###" >> "$LOG_FILE"
SYSTEM_SCORE=$(lynis show details | grep -i "hardening index" | awk '{print $NF}')
echo "Оценка безопасности системы (Hardening Index): $SYSTEM_SCORE" >> "$LOG_FILE"

echo "Аудит завершен. Результаты записаны в $LOG_FILE"
```

[ Прочитано 14 строк ]

^G Справка	^O Записать	^W Поиск	^K Вырезать	^T Выполнить	^C Позиция
^X Выход	^R ЧитФайл	^L Замена	^U Вставить	^J Выровнять	^_ К строке

Рисунок 13 — bash-скрипт сканирования с фильтрацией для Alt Рабочая станция

Необходимо сделать файл исполняемым:

```
chmod +x /usr/local/bin/lynus_audit.sh
```

Для Астра Linux:

```
#!/bin/bash
```

```
LOG_FILE1="/var/log/lynis_filtered_$(date +%F_%H-%M-%S).txt"
```

```
./lynis audit system --quiet | grep -iE "warning|vulnerable|risk|unsafe" | sudo tee
"$LOG_FILE1" > dev/null
```

```
echo "Отчет Lynis сохранен в $LOG_FILE1"
```

```
GNU nano 2.7.4      Файл: /usr/local/bin/lynis_audit.sh
#!/bin/bash
# Определение выходного файла
LOG_FILE1="/var/log/lynis_filtered_$(date +%F_%H-%M-%S).txt"
# Запуск Lynis с фильтрацией
./lynis audit system --quiet | grep -iE "warning|vulnerable|risk|unsafe" $
# Уведомление о завершении
echo "Отчет Lynis сохранен в $LOG_FILE1"
[ File '/usr/local/bin/lynis_audit.sh' is unwritable ]
^G Помощь      ^O Записать    ^W Поиск      ^K Вырезать    ^J Вывернуть
^X Выход      ^R ЧитФайл    ^\ Замена     ^U Отмен. Выре  ^T Пров. синтак.
```

Рисунок 14 — bash-скрипт сканирования с фильтрацией для Астра Linux

### 2.3. Установка и настройка Rkhunter.

Для Альт Рабочая станция:

Установим сканер из стандартного репозитория и настроим скрипт для выполнения сканирования и вывода в лог-файл только предупреждающих строк:

```
nano /usr/local/bin/rkhunter_audit.sh
```

```
#!/bin/bash
```

```
LOG_FILE2="/var/log/rkhunter_filtered_$(date +%F_%H-%M-%S).txt"
```

```
rkhunter --check --skip-keypress | grep -iE "Invalid|Warning|Vulnerable|Suspicious|Alert" > "$LOG_FILE2"
```

```
if grep -q -iE "Invalid|Warning|Vulnerable|Suspicious|Alert" "$LOG_FILE2"; then
```

```
    echo "Подозрительные элементы обнаружены. Лог записан в $LOG_FILE2."
```

```
else
```

```
    echo "Система чиста. Угроз не обнаружено."
```

```
fi
```

```
GNU nano 7.2 /usr/local/bin/rkhunter_audit.sh
#!/bin/bash

LOG_FILE2="/var/log/rkhunter_filtered_$(date +%F_%H-%M-%S).txt"

rkhunter --check --skip-keypress | grep -iE "Invalid|Warning|Vulnerable|Suspicious|Alert" > "$LOG_FILE2"

if grep -q -iE "Invalid|Warning|Vulnerable|Suspicious|Alert" "$LOG_FILE2"; then
    echo "Подозрительные элементы обнаружены. Лог записан в $LOG_FILE2."
else
    echo "Система чиста. Угроз не обнаружено."
fi

[ Прочитано 14 строк ]
^G Справка      ^O Записать    ^W Поиск      ^K Вырезать   ^T Выполнить  ^C Позиция     M-U Отмена
^X Выход        ^R ЧитФайл    ^L Замена    ^U Вставить   ^J Выровнять  ^I К строке   M-E Повтор
```

Рисунок 15 — Скрипт фильтрации Rkhunter 1

И протестируем, запустив сканирование:

```
[root@vbox ~]# cat /var/log/rkhunter_filtered_2025-01-11_17-02-02.txt
Invalid XINETD_CONF_PATH configuration option: Non-existent pathname: /etc/xinetd.conf
[root@vbox ~]#
```

Рисунок 16 — Обнаружение некорректной работы

Для Астра Linux:

Для Астра же требуется установка из другого источника SourceForce, так как в стандартном репозитории его нет. Для этого требуется написать следующие команды:

*sudo wget*

<https://sourceforge.net/projects/rkhunter/files/rkhunter/1.4.6/rkhunter-1.4.6.tar.gz>

```

Installation complete
user@astra:~/lynis-3.0.0/rkhunter-1.4.6$ sudo wget https://sourceforge.net
/projects/rkhunter/files/rkhunter/1.4.6/rkhunter-1.4.6.tar.gz
--2025-01-15 18:17:44-- https://sourceforge.net/projects/rkhunter/files/r
khunter/1.4.6/rkhunter-1.4.6.tar.gz
Распознаётся sourceforge.net (sourceforge.net)... 104.18.12.149, 104.18.13.1
49, 2606:4700::6812:d95, ...
Подключение к sourceforge.net (sourceforge.net)|104.18.12.149|:443... соед
инение установлено.
HTTP-запрос отправлен. Ожидание ответа... 301 Moved Permanently
Адрес: https://sourceforge.net/projects/rkhunter/files/rkhunter/1.4.6/rkhu
nter-1.4.6.tar.gz/ [перехог]
--2025-01-15 18:17:46-- https://sourceforge.net/projects/rkhunter/files/r
khunter/1.4.6/rkhunter-1.4.6.tar.gz/
Повторное использование соединения с sourceforge.net:443.
HTTP-запрос отправлен. Ожидание ответа... 301 Moved Permanently
Адрес: https://sourceforge.net/projects/rkhunter/files/rkhunter/1.4.6/rkhu
nter-1.4.6.tar.gz/download [перехог]
--2025-01-15 18:17:47-- https://sourceforge.net/projects/rkhunter/files/r
khunter/1.4.6/rkhunter-1.4.6.tar.gz/download

```

Рисунок 17 — Установка с помощью стороннего ресурса

И распаковать архив. После чего необходимо запустить установщик:

*sudo ./installer.sh --layout /usr/local --install*

```

Directory /var/lib/rkhunter/db/i18n: exists and is writable.
Directory /var/lib/rkhunter/db/signatures: exists and is writable.
Installing check_modules.pl: OK
Installing filehashsha.pl: OK
Installing stat.pl: OK
Installing readlink.sh: OK
Installing backdoorports.dat: OK
Installing mirrors.dat: OK
Installing programs_bad.dat: OK
Installing suspscan.dat: OK
Installing rkhunter.8: OK
Installing ACKNOWLEDGMENTS: OK
Installing CHANGELOG: OK
Installing FAQ: OK
Installing LICENSE: OK
Installing README: OK
Installing language support files: OK
Installing ClamAV signatures: OK
Installing rkhunter: OK
Installing rkhunter.conf: OK

```

Рисунок 18 — Запуск установщика

Настроим фильтрацию строк:

```

GNU nano 2.7.4 Файл: /usr/local/bin/rkhunter_audit.sh

#!/bin/bash

LOG_FILE2="/var/log/rkhunter_filtered_$(date +%F_%H-%M-%S).txt"

sudo rkhunter --check --skip-keypress | grep -iE "Invalid|Warning|Vulnera$

if grep -q -iE "Invalid|Warning|Vulnerable|Suspicious|Alert" "$LOG_FILE2"$
    echo "Погозрительные элементы обнаружены. Лог записан в $LOG_FILE2."
else
    echo "Система чиста. Угроз не обнаружено."
fi

[ Прочитано 14 строк ]
^G Помощь      ^O Записать
^X Выход       ^R ЧитФайл
^W Поиск      ^K Вырезать
^N Замена    ^U Отмен. Выре
^J Выровнять
^T Пров. синтак.

```

Рисунок 19 — Скрипт фильтрации Rkhunter 2

### 3. Комбинирование и автоматизация сканеров.

Создадим скрипт, который будет выполнять поочередное выполнение сканеров и записи в единый лог-отчет:

```

#!/bin/bash

LOG_FILE="/var/log/chkrootkit_filtered_$(date +%F_%H-%M-%S).txt"
LOG_FILE2="/var/log/rkhunter_filtered_$(date +%F_%H-%M-%S).txt"
COMBINED_LOG="/var/log/combined_scan_$(date +%F_%H-%M-%S).txt"

DATE=$(date '+%Y-%m-%d %H:%M:%S')

chkrootkit | grep -iE "infected|suspicious|warning|detected" | grep -ivE "not infected|
nothing detected|Searching fo>

rkhunter --check --skip-keypress | grep -iE "Invalid|Warning|Vulnerable|Suspicious|
Alert" > "$LOG_FILE2"

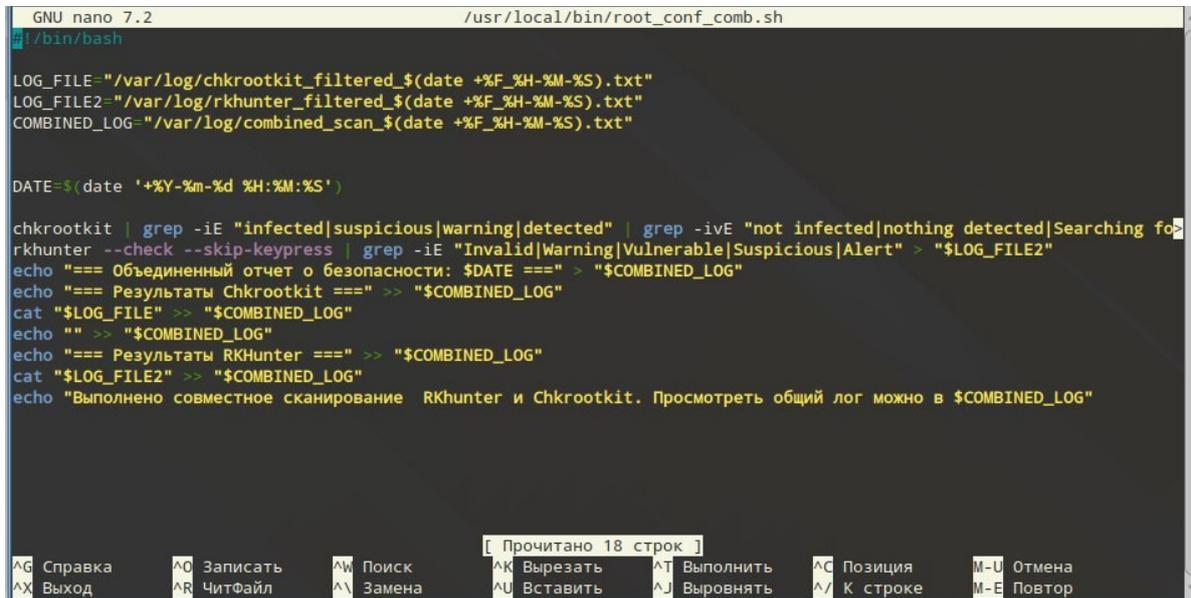
echo "=== Объединенный отчет о безопасности: $DATE ===" >
"$COMBINED_LOG"

echo "=== Результаты Chkrootkit ===" >> "$COMBINED_LOG"
cat "$LOG_FILE" >> "$COMBINED_LOG"
echo "" >> "$COMBINED_LOG"
echo "=== Результаты RKHunter ===" >> "$COMBINED_LOG"

```

```
cat "$LOG_FILE2" >> "$COMBINED_LOG"
```

*echo "Выполнено совместное сканирование RKHunter и Chkrootkit. Просмотреть  
общий лог можно в \$COMBINED\_LOG"*



```
GNU nano 7.2 /usr/local/bin/root_conf_comb.sh
! /bin/bash

LOG_FILE="/var/log/chkrootkit_filtered_$(date +%F_%H-%M-%S).txt"
LOG_FILE2="/var/log/rkhunter_filtered_$(date +%F_%H-%M-%S).txt"
COMBINED_LOG="/var/log/combined_scan_$(date +%F_%H-%M-%S).txt"

DATE=$(date '+%Y-%m-%d %H:%M:%S')

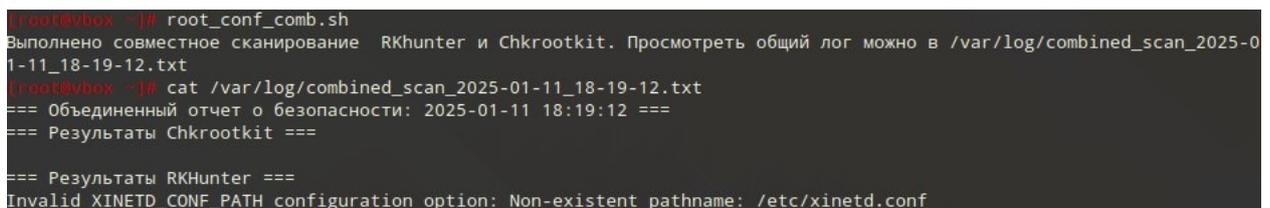
chkrootkit | grep -iE "infected|suspicious|warning|detected" | grep -ivE "not infected|nothing detected|Searching for"
rkhunter --check --skip-keypress | grep -iE "Invalid|Warning|Vulnerable|Suspicious|Alert" > "$LOG_FILE2"
echo "=== Объединенный отчет о безопасности: $DATE ===" > "$COMBINED_LOG"
echo "=== Результаты Chkrootkit ===" >> "$COMBINED_LOG"
cat "$LOG_FILE" >> "$COMBINED_LOG"
echo "" >> "$COMBINED_LOG"
echo "=== Результаты RKHunter ===" >> "$COMBINED_LOG"
cat "$LOG_FILE2" >> "$COMBINED_LOG"
echo "Выполнено совместное сканирование RKHunter и Chkrootkit. Просмотреть общий лог можно в $COMBINED_LOG"
```

Рисунок 20 — Скрипт объединения логов сканеров

Таким образом после того, как мы сделали файл выполняемым, можно проверить итоговый вывод:

```
root_conf_comb.sh
```

```
cat /var/log/combined_scan_2025-01-11_18-19-12.txt
```



```
[root@vbox ~]# root_conf_comb.sh
Выполнено совместное сканирование RKHunter и Chkrootkit. Просмотреть общий лог можно в /var/log/combined_scan_2025-01-11_18-19-12.txt
[root@vbox ~]# cat /var/log/combined_scan_2025-01-11_18-19-12.txt
=== Объединенный отчет о безопасности: 2025-01-11 18:19:12 ===
=== Результаты Chkrootkit ===

=== Результаты RKHunter ===
Invalid XINETD_CONF_PATH configuration option: Non-existent pathname: /etc/xinetd.conf
```

Рисунок 21 — Отчет сканирования

Объединенный сканер успешно отработал, поэтому можно создать Cron-задание для ежедневной исполняемой проверки операционной системы:

```
root@vbox ~|# crontab -e
crontab: installing new crontab
root@vbox ~|# crontab -l
30 2 * * * /usr/local/bin/root_conf_comb.sh

#minute (0-59),
#| hour (0-23),
#| | day of the month (1-31),
#| | | month of the year (1-12),
#| | | | day of the week (0-6 with 0=Sunday).
#| | | | | commands
root@vbox ~|#
```

Рисунок 22 — Cron автоматизация

Таким образом, мы автоматизировали работу системы мониторинга и уплотнили безопасность ОС, сканирование будет выполняться ежедневно в 2:30.

Для полного охвата безопасности операционной системы установим Cron задание на выполнение полного еженедельного сканирования с помощью Lynis:

```
GNU nano 7.2 /tmp/.private/root/crontab.oV0GUF Изменён
30 2 * * * /usr/local/bin/root_conf_comb.sh
0 2 * * 1 /usr/local/bin/lynis_audit.sh
#minute (0-59),
#| hour (0-23),
#| | day of the month (1-31),
#| | | month of the year (1-12),
#| | | | day of the week (0-6 with 0=Sunday).
#| | | | | commands

^G Справка      ^O Записать    ^W Поиск      ^K Вырезать   ^T Выполнить  ^C Позиция
^X Выход        ^R ЧитФайл   ^\ Замена     ^U Вставить   ^J Выровнять  ^/_ К строке
```

Рисунок 23 — Cron автоматизация Lynis

#### 4. Экспериментальная часть.

Создадим файл с содержимым теста EICAR, файл не представляет угрозы, он нужен для проверки работоспособности сканеров и антивирусов

```
echo 'X5O!P%@AP[4PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*' > /tmp/eicar.com
```

Убедимся, что файл создан:

```
cat /tmp/eicar.com
```

Запустим скрипт:

```
root_conf_comb.sh
```

RKHunter и Chkrootkit должны распознать eicar.com как потенциальную угрозу.

Однако сканеры вывели пустой результат, это ожидаемое поведение, поскольку эти утилиты не предназначены для обнаружения антивирусных тестовых файлов. Их основная задача — обнаружение руткитов, скрытых процессов, изменённых системных файлов и следов компрометации.

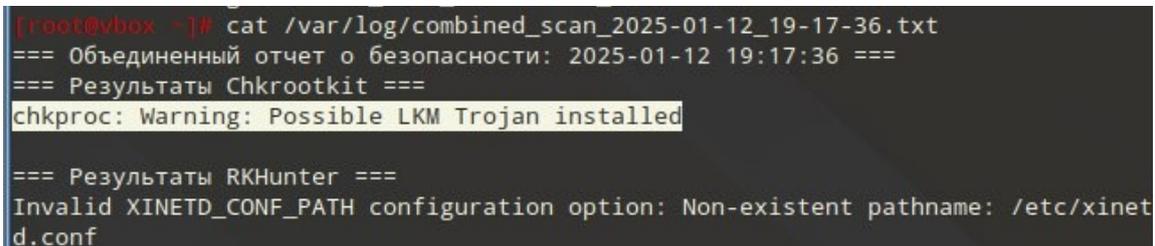
В таком случае попробуем создать скрытые процессы и имитировать руткит:

```
nohup sleep 9999 &
```

```
nohup bash -c "while :; do echo 'hidden'; done" &
```

И снова запустим скрипт объединения сканеров:

```
root_conf_comb.sh
```



```
[root@vbox ~]# cat /var/log/combined_scan_2025-01-12_19-17-36.txt
=== Объединенный отчет о безопасности: 2025-01-12 19:17:36 ===
=== Результаты Chkrootkit ===
chkproc: Warning: Possible LKM Trojan installed
=== Результаты RKHunter ===
Invalid XINETD_CONF_PATH configuration option: Non-existent pathname: /etc/xinetd.conf
```

Рисунок 24 — Обнаружение уязвимости

Это предупреждение говорит о том, что сканер Chkrootkit обнаружил аномалию, связанную с процессами, которая может быть признаком LKM-трояна.

Также из git-hub установим руткит, который часто применяется в лабораторных условиях для тестирования системы безопасности – adore-ng. Эксперимент выполняется на изолированной машине в Virtual Box и используется исключительно в научно-ознакомительных целях, что не противоречит законодательству РФ. С помощью него проведем контрольную проверку:

```
[root@vbox etc]# wget https://github.com/yaoyumeng/adore-ng/archive/refs/heads/master.zip
--2025-01-12 21:00:53-- https://github.com/yaoyumeng/adore-ng/archive/refs/heads/master.zip
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/yaoyumeng/adore-ng/zip/refs/heads/master [following]
--2025-01-12 21:00:54-- https://codeload.github.com/yaoyumeng/adore-ng/zip/refs/heads/master
Resolving codeload.github.com (codeload.github.com)... 140.82.121.10
Connecting to codeload.github.com (codeload.github.com)|140.82.121.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'master.zip'

master.zip          [ <=>          ] 21.00K  --.-KB/s   in 0.05s

2025-01-12 21:00:55 (433 KB/s) - 'master.zip' saved [21501]

[root@vbox etc]# unzip master.zip
Archive:  master.zip
522c80a2dc043c2d523256472becc88c90d66337
  creating:  adore-ng-master/
  inflating:  adore-ng-master/.gitignore
  inflating:  adore-ng-master/LICENSE
  inflating:  adore-ng-master/Makefile
  inflating:  adore-ng-master/README.md
  inflating:  adore-ng-master/adore-ng.c
  inflating:  adore-ng-master/adore-ng.h
  inflating:  adore-ng-master/ava.c
  inflating:  adore-ng-master/libinvisible.c
  inflating:  adore-ng-master/libinvisible.h
[root@vbox etc]# cd adore-ng-master
[root@vbox adore-ng-master]# dir
adore-ng.c  adore-ng.h  ava.c  libinvisible.c  libinvisible.h  LICENSE  Makefile  README.md
```

Рисунок 25 — Установка руткита

Руткит успешно установлен, поэтому запускаем скрипт:

```
[root@vbox rkhunter-1.4.6]# cat /var/log/combined_scan_2025-01-15_22-29-27.txt
=== Объединенный отчет о безопасности: 2025-01-15 22:29:27 ===
=== Результаты Chkrootkit ===
chkproc: Warning: Possible LKM Trojan installed

=== Результаты RKNHunter ===
      Adore Rootkit                                     [ Found ]
```

Рисунок 26 — Обнаружение руткита

Сканеры успешно обнаружили руткит и обнаружил аномалию.

#### 4.1 Тестирование Lypis

Сделаем проверку многоступенчатой, чтобы оценить многосторонность сканера и его комплексную эффективность.

#### 4.1.1 Проверка прав на файлы и каталоги

Цель: убедиться, что Lynis отслеживает неправильные права на чувствительные файлы

```
chmod 777 /etc/sudoers
```

Проверка прав на домашние директории:

```
chmod 777 /home/user
```

Lynis обнаружил неправильные права на чувствительные файлы, такие как /etc/sudoers:

Было обнаружено предупреждение о правах на файл /etc/sudoers.

Также были отмечены и домашние директории, которые могут привести к угрозам безопасности.

#### 4.1.2 Проверка sysctl и параметров ядра

Цель: проверить, как Lynis реагирует на изменения в параметрах ядра через sysctl.

Изменение параметров ядра:

```
sysctl -w kernel.randomize_va_space=0
```

```
sysctl -w net.ipv4.conf.all.rp_filter=0
```

```
echo "kernel.randomize_va_space=0" >> /etc/sysctl.conf
```

```
echo "net.ipv4.conf.all.rp_filter=0" >> /etc/sysctl.conf
```

Lynis обнаружил небезопасные параметры ядра:

Параметры `kernel.randomize_va_space=0` и `net.ipv4.conf.all.rp_filter=0`, были помечены как небезопасные, так как эти изменения могут ослабить защиту от атак, связанных с утечками памяти или сетевыми атаками.

#### 4.1.3 Проверка сетевых настроек

Цель: проверить, как Lynis оценивает настройки сети и брандмауэра.

Отключение брандмауэра:

```
systemctl stop firewalld
```

```
systemctl disable firewalld
```

Изменение конфигурации `hosts.allow` и `hosts.deny`:

```
echo "ALL: ALL" > /etc/hosts.allow
```

```
echo "ALL: ALL" > /etc/hosts.deny
```

Так как был отключен брандмауэр `firewalld` и неправильно настроены файлы `/etc/hosts.allow` и `/etc/hosts.deny`, Lynis корректно отметил это как проблему. Установка значений "ALL: ALL" в эти файлы — прямая угроза безопасности, так как открывает доступ ко всем системам без ограничений.

#### 4.1.4 Проверка учетных записей и паролей

Цель: проверить, как Lynis реагирует на слабые пароли и неправильные настройки для пользователей.

Создание или изменение учетных записей со слабыми паролями:

```
useradd testuser
```

```
echo "123" | passwd --stdin testuser
```

Изменение параметров в `/etc/login.defs`:

```
echo "PASS_MIN_LEN 3" >> /etc/login.defs
```

Были созданы пользователи с слабыми паролями - `testuser` с паролем "123", и были внесены изменения в файл `/etc/login.defs`, позволяющие использовать слабые пароли - `PASS_MIN_LEN 3`, и Lynis сообщил об этих слабых конфигурациях.

```

- accounts-daemon.service: [ НЕБЕЗОПАСНО ]
- ahttpd.service: [ НЕБЕЗОПАСНО ]
- alsa-state.service: [ НЕБЕЗОПАСНО ]
- alteratord.service: [ НЕБЕЗОПАСНО ]
- auditd.service: [ УЯЗВИМО ]
- avahi-daemon.service: [ НЕБЕЗОПАСНО ]
- colord.service: [ УЯЗВИМО ]
- crond.service: [ НЕБЕЗОПАСНО ]
- cups-browsed.service: [ НЕБЕЗОПАСНО ]
- cups.service: [ НЕБЕЗОПАСНО ]
- dbus.service: [ НЕБЕЗОПАСНО ]
- dm-event.service: [ НЕБЕЗОПАСНО ]
- emergency.service: [ НЕБЕЗОПАСНО ]
- getty@tty1.service: [ НЕБЕЗОПАСНО ]
- lightdm.service: [ НЕБЕЗОПАСНО ]
- lvm2-lvmpolld.service: [ НЕБЕЗОПАСНО ]
- network.service: [ НЕБЕЗОПАСНО ]
- nmb.service: [ НЕБЕЗОПАСНО ]
- pcscd.service: [ НЕБЕЗОПАСНО ]
- plymouth-start.service: [ НЕБЕЗОПАСНО ]
- polkit.service: [ НЕБЕЗОПАСНО ]
- rc-local.service: [ НЕБЕЗОПАСНО ]
- rescue.service: [ НЕБЕЗОПАСНО ]
- smb.service: [ НЕБЕЗОПАСНО ]
- systemd-ask-password-console.service: [ НЕБЕЗОПАСНО ]
- systemd-ask-password-plymouth.service: [ НЕБЕЗОПАСНО ]
- systemd-initctl.service: [ НЕБЕЗОПАСНО ]
- systemd-rfkill.service: [ НЕБЕЗОПАСНО ]
- udisks2.service: [ НЕБЕЗОПАСНО ]
- user@500.service: [ НЕБЕЗОПАСНО ]
- winbind.service: [ НЕБЕЗОПАСНО ]
- Administrator accounts [ ПРЕДУПРЕЖДЕНИЕ ]
- Unique UIDs [ ПРЕДУПРЕЖДЕНИЕ ]
- Permissions for: /etc/sudoers [ ПРЕДУПРЕЖДЕНИЕ ]
- Permissions for: /etc/sudoers.d/99-sudopw [ ПРЕДУПРЕЖДЕНИЕ ]
- umask (/etc/profile and /etc/profile.d) [ ПРЕДЛОЖЕНИЕ ]
- umask (/etc/login.defs) [ ПРЕДЛОЖЕНИЕ ]
- Checking /home mount point [ ПРЕДЛОЖЕНИЕ ]
- Checking /var mount point [ ПРЕДЛОЖЕНИЕ ]
- Minimal of 2 responsive nameservers [ ПРЕДУПРЕЖДЕНИЕ ]
- Checking for empty ruleset [ ПРЕДУПРЕЖДЕНИЕ ]
File: /etc/at.deny [ ПРЕДЛОЖЕНИЕ ]
File: /etc/cron.deny [ ПРЕДЛОЖЕНИЕ ]
Directory: /etc/cron.daily [ ПРЕДЛОЖЕНИЕ ]
Directory: /etc/cron.hourly [ ПРЕДЛОЖЕНИЕ ]
Directory: /etc/cron.weekly [ ПРЕДЛОЖЕНИЕ ]
Directory: /etc/cron.monthly [ ПРЕДЛОЖЕНИЕ ]
- Permissions of home directories [ ПРЕДУПРЕЖДЕНИЕ ]

```

Рисунок 27 — Обнаружение уязвимости Lynis

Таким образом, Lynis также успешно справился с тестом.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

### Учебные, научные и иные публикации

1. Образцов Е. С. Мониторинг как средство защиты информации // Перспективы развития информационных технологий. – 2012. – №9. – С. 129-131.

### Эмпирические материалы

2. Вызовы информационной безопасности для ТЭК в условиях существующих ограничений и трендов импортозамещения. – URL: <https://www.novostiitkanala.ru>.

3. Какие киберугрозы ждут нефтегазовых гигантов в 2024 году. – URL: <https://companies.rbc.ru>.

4. Крупнейшие утечки данных в 2024: как хакеры отнимают у нас приватность – URL: <https://gerwin.io>.

5. Утечки персональных данных в России: статистика первого полугодия 2024 – URL: <https://www.cibit.ru>.

6. Innostage: почти в 80% инцидентов ИБ напрямую виноват человеческий фактор. – URL: <https://companies.rbc.ru>.

7. Positive Technologies: число атак на промышленность выросло на 53%. – URL: <https://www.ptsecurity.com>.