

## **ГЛАВА 3 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ К СРЕДСТВАМ ЗАЩИТЫ ИНФОРМАЦИИ**

### **3.1 Проект СЗИ для автоматизированной системы (АС)**

Система защиты информации (СЗИ) представляет собой комплекс организационных и технических мер, направленных на обеспечение конфиденциальности, целостности и доступности данных в автоматизированной системе. В рамках проекта для Управления «К» СЗИ включает:

- Программно-аппаратные средства (Wazuh, Suricata, UFW, ClamAV).
- Нормативно-правовые акты (ФЗ-152, ФЗ-432, внутренние регламенты).
- Организационные меры (пропускной режим, видеонаблюдение, контроль доступа).

Основные цели внедрения СЗИ:

- Минимизация рисков утечки, модификации или уничтожения конфиденциальных данных.
- Обеспечение соответствия законодательным требованиям в области информационной безопасности.
- Повышение устойчивости системы к кибератакам.

Основные характеристики СЗИ:

- Многоуровневая защита
- Контроль доступа (двухфакторная аутентификация, разграничение прав)
- Шифрование данных при передаче и хранении
- Регулярное обновление ПО и сигнатур угроз
- Мониторинг и реагирование
- Системы обнаружения вторжений (IDS/IPS)
- Анализ журналов событий (SIEM)
- Отказоустойчивость

- Резервное копирование данных
- Защита от DDoS-атак

Топология защищенности сети

Для Управления «К» предложена следующая структура:

- Внешний периметр:
- Межсетевой экран (UFW) с фильтрацией трафика
- Система защиты от DDoS-атак (Suricata)
- Внутренняя сеть:
- Изолированные VLAN для работы с конфиденциальными данными
- Серверная зона с контролем доступа и мониторингом («Wazuh»)

Внедрение средств защиты

Таблица 11 – Средства защиты и их предназначение

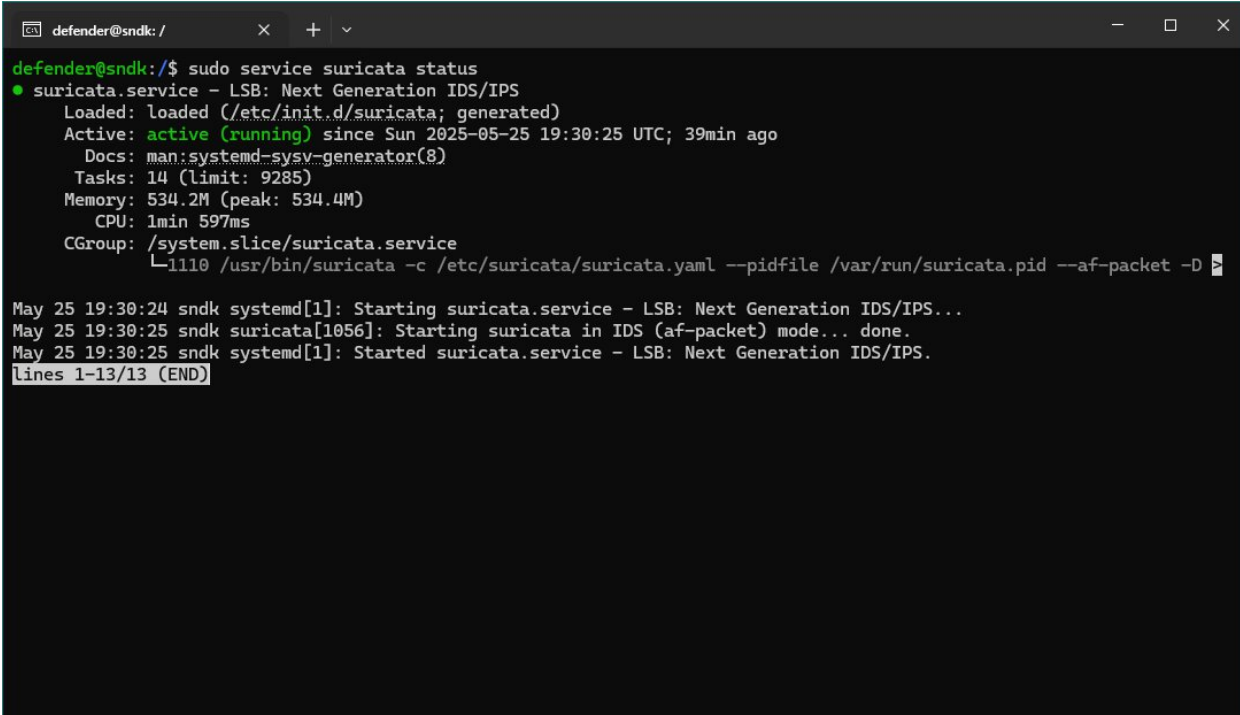
Средство защиты	Назначение
Wazuh (SIEM)	Контроль целостности ПО, Аудит системы
ClamAV	Антивирусная защита, контроль запускаемых процессов
UFW	Фильтрация трафика
Suricata (IDS)	Мониторинг трафика, выявление аномалий

Используемое ПО:

1. Suricata — это высокопроизводительная система обнаружения и предотвращения вторжений с открытым исходным кодом, предназначенная для мониторинга сетевого трафика в реальном времени. Её основная цель заключается в обнаружении и блокировке кибератак, анализе угроз и обеспечении безопасности сети. Suricata способна выявлять атаки на основе сигнатур и аномалий, предотвращать вторжения в режиме IPS, а также поддерживает современные протоколы, такие как HTTP, TLS, DNS и SSH.

Она легко интегрируется с системами SIEM, включая Wazuh, что позволяет централизованно собирать и анализировать события безопасности.

Ключевые преимущества Suricata включают её открытый исходный код, что делает её бесплатной и гибкой в настройке. Благодаря поддержке многопоточности и аппаратного ускорения (например, через Intel Hyperscan), система демонстрирует высокую производительность даже под большой нагрузкой. Suricata отличается масштабируемостью, работая как в небольших сетях, так и в крупных корпоративных средах. Она поддерживает анализ зашифрованного трафика (TLS) и современных протоколов, таких как HTTP/2 и DNS-over-HTTPS, что особенно важно в условиях роста сложных киберугроз.

A terminal window with a dark background and light-colored text. The window title is 'defender@sndk: /'. The user has entered the command 'sudo service suricata status'. The output shows that the 'suricata.service' is an LSB: Next Generation IDS/IPS service, loaded from '/etc/init.d/suricata', and is currently 'active (running)' since Sun 2025-05-25 19:30:25 UTC, 39 minutes ago. It lists various details like docs, tasks, memory usage (534.2M), CPU usage (1min 597ms), and the CGroup. At the bottom, there are log messages from systemd and suricata showing the service starting successfully in IDS (af-packet) mode. The terminal also shows a prompt for the user to press a key to continue.

```
defender@sndk: /
defender@sndk:/$ sudo service suricata status
● suricata.service - LSB: Next Generation IDS/IPS
   Loaded: loaded (/etc/init.d/suricata; generated)
   Active: active (running) since Sun 2025-05-25 19:30:25 UTC; 39min ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 14 (limit: 9285)
  Memory: 534.2M (peak: 534.4M)
     CPU: 1min 597ms
   CGroup: /system.slice/suricata.service
           └─1110 /usr/bin/suricata -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid --af-packet -D

May 25 19:30:24 sndk systemd[1]: Starting suricata.service - LSB: Next Generation IDS/IPS...
May 25 19:30:25 sndk suricata[1056]: Starting suricata in IDS (af-packet) mode... done.
May 25 19:30:25 sndk systemd[1]: Started suricata.service - LSB: Next Generation IDS/IPS.
lines 1-13/13 (END)
```

Рисунок 3 - Проверка статуса программы Suricata (IDS)

```
defender@sndk: /
CPU: 1min 597ms
CGroup: /system.slice/suricata.service
└─1110 /usr/bin/suricata -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid --af-packet -D

May 25 19:30:24 sndk systemd[1]: Starting suricata.service - LSB: Next Generation IDS/IPS...
May 25 19:30:25 sndk suricata[1056]: Starting suricata in IDS (af-packet) mode... done.
May 25 19:30:25 sndk systemd[1]: Started suricata.service - LSB: Next Generation IDS/IPS.

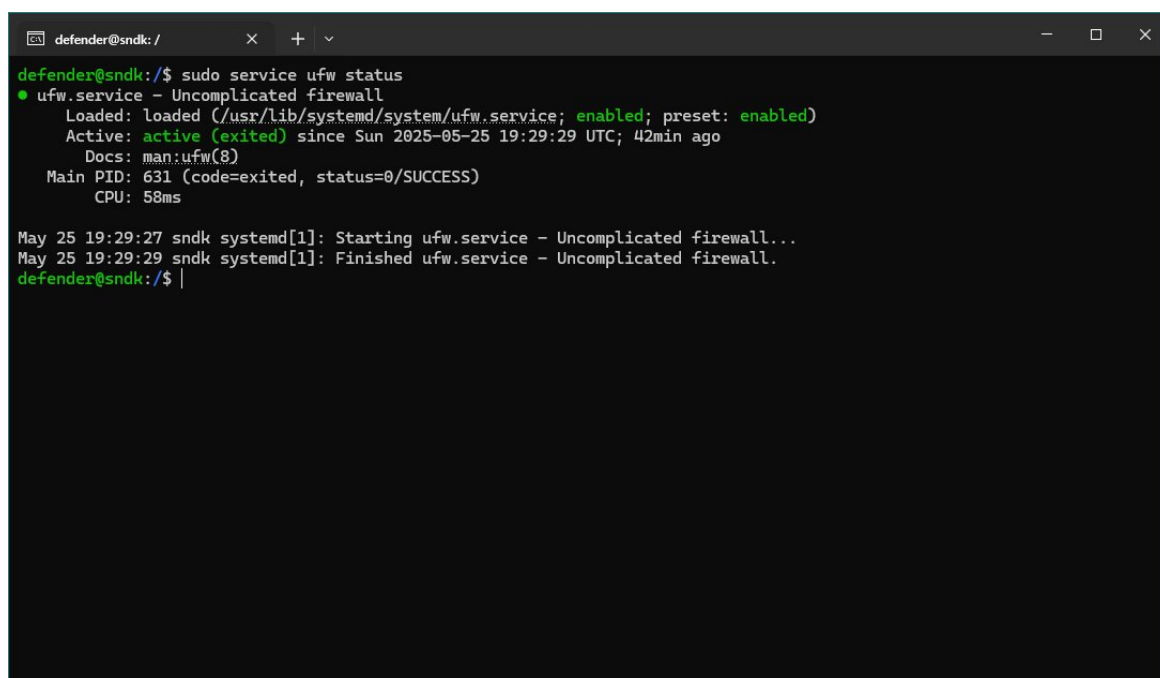
defender@sndk:/$ sudo tail -f /var/log/suricata/fast.log
05/25/2025-19:56:26.567909  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package ma
nagement [**] [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.1.213:39758 -> 91.189.91.81:80
05/25/2025-19:56:26.607534  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package ma
nagement [**] [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.1.213:57464 -> 185.125.190.83:80
05/25/2025-19:56:38.032493  [**] [1:2210059:1] SURICATA STREAM pkt seen on wrong thread [**] [Classification: (null)]
[Priority: 3] {TCP} 192.168.1.213:57070 -> 185.125.190.81:80
05/25/2025-19:56:38.152361  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package ma
nagement [**] [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.1.213:57070 -> 185.125.190.81:80
05/25/2025-19:56:38.389380  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package ma
nagement [**] [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.1.213:57070 -> 185.125.190.81:80
05/25/2025-19:56:38.389380  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package ma
nagement [**] [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.1.213:57070 -> 185.125.190.81:80
05/25/2025-19:56:38.389380  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package ma
nagement [**] [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.1.213:57070 -> 185.125.190.81:80
05/25/2025-19:56:38.389380  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package ma
nagement [**] [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.1.213:57070 -> 185.125.190.81:80
05/25/2025-19:56:45.150919  [**] [1:2210059:1] SURICATA STREAM pkt seen on wrong thread [**] [Classification: (null)]
[Priority: 3] {TCP} 192.168.1.213:52708 -> 104.17.196.15:443
05/25/2025-19:56:47.905580  [**] [1:2210059:1] SURICATA STREAM pkt seen on wrong thread [**] [Classification: (null)]
[Priority: 3] {TCP} 192.168.1.213:52314 -> 54.247.62.1:443
```

Рисунок 4 – Проверка «Alert» в логах программы

2. UFW – Программа Uncomplicated Firewall (UFW) представляет собой фронтенд для управления межсетевым экраном iptables, разработанный для упрощения настройки и администрирования брандмауэра в Linux-системах. Основная цель UFW — предоставить пользователям интуитивно понятный интерфейс для настройки правил фильтрации сетевого трафика без необходимости глубокого изучения сложного синтаксиса iptables.

Ключевые преимущества UFW включают простоту использования благодаря понятному синтаксису команд, позволяющему разрешать или запрещать трафик одной командой (например, `ufw allow 22/tcp`). Программа хорошо интегрирована с системой, входит в стандартные репозитории популярных дистрибутивов Linux и совместима с различными сетевыми сервисами. Несмотря на кажущуюся простоту, UFW поддерживает гибкие настройки, включая сложные правила фильтрации по IP-адресам, портам и протоколам, а также функции NAT и перенаправления портов. Дополнительными преимуществами являются встроенные механизмы логирования сетевой активности и возможность автоматизации через скрипты. Важной особенностью является политика безопасности по умолчанию, при

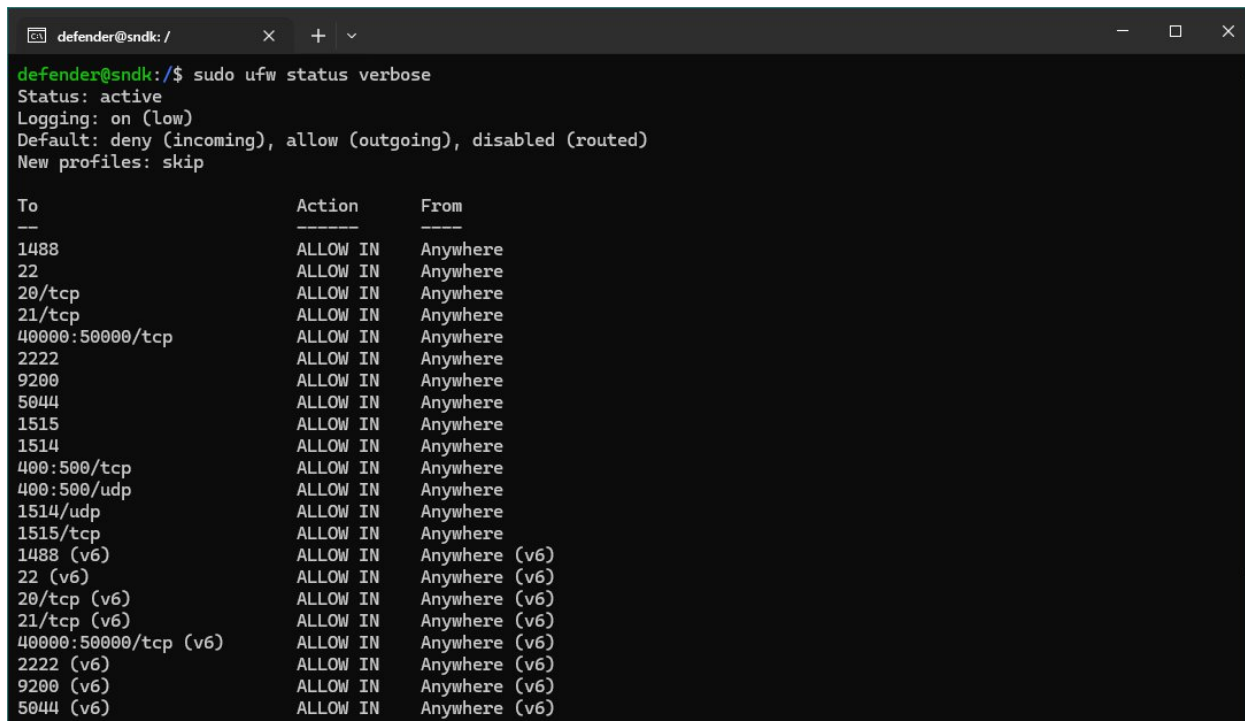
которой блокируется весь входящий трафик, что соответствует принципу минимальных привилегий и существенно повышает уровень базовой защиты системы.



```
defender@sndk: /
defender@sndk:/$ sudo service ufw status
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/usr/lib/systemd/system/ufw.service; enabled; preset: enabled)
   Active: active (exited) since Sun 2025-05-25 19:29:29 UTC; 42min ago
     Docs: man:ufw(8)
   Main PID: 631 (code=exited, status=0/SUCCESS)
      CPU: 58ms

May 25 19:29:27 sndk systemd[1]: Starting ufw.service - Uncomplicated firewall...
May 25 19:29:29 sndk systemd[1]: Finished ufw.service - Uncomplicated firewall.
defender@sndk:/$
```

Рисунок 5 – Проверка статуса программы UFW



```
defender@sndk: /
defender@sndk:/$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To Action From
--
1488 ALLOW IN Anywhere
22 ALLOW IN Anywhere
20/tcp ALLOW IN Anywhere
21/tcp ALLOW IN Anywhere
40000:50000/tcp ALLOW IN Anywhere
2222 ALLOW IN Anywhere
9200 ALLOW IN Anywhere
5044 ALLOW IN Anywhere
1515 ALLOW IN Anywhere
1514 ALLOW IN Anywhere
400:500/tcp ALLOW IN Anywhere
400:500/udp ALLOW IN Anywhere
1514/udp ALLOW IN Anywhere
1515/tcp ALLOW IN Anywhere
1488 (v6) ALLOW IN Anywhere (v6)
22 (v6) ALLOW IN Anywhere (v6)
20/tcp (v6) ALLOW IN Anywhere (v6)
21/tcp (v6) ALLOW IN Anywhere (v6)
40000:50000/tcp (v6) ALLOW IN Anywhere (v6)
2222 (v6) ALLOW IN Anywhere (v6)
9200 (v6) ALLOW IN Anywhere (v6)
5044 (v6) ALLOW IN Anywhere (v6)
```

Рисунок 6 – Конфигурация политик UFW

3. Wazuh — это платформа с открытым исходным кодом, предназначенная для обеспечения безопасности информационных систем,

включая обнаружение вторжений, мониторинг целостности файлов, анализ уязвимостей и соответствия требованиям, а также реагирование на инциденты. Основная цель Wazuh — предоставить комплексное решение для защиты инфраструктуры организаций, объединяя возможности SIEM (Security Information and Event Management) и XDR (Extended Detection and Response). Платформа работает на основе агентов, которые устанавливаются на конечные устройства, и централизованного сервера управления, что позволяет собирать, анализировать и коррелировать данные о безопасности в режиме реального времени.

Благодаря модульной архитектуре Wazuh поддерживает интеграцию с другими инструментами безопасности, такими как Suricata и ClamAV, что расширяет его функциональность. Платформа обеспечивает детектирование аномалий и угроз на основе сигнатур и поведенческого анализа, а также позволяет автоматизировать реакции на инциденты через predefined правила. Кроме того, Wazuh поддерживает соответствие различным стандартам безопасности, включая PCI DSS, GDPR и HIPAA, что упрощает процесс аудита и compliance-отчётности.

Ещё одним важным преимуществом Wazuh является его способность работать в гетерогенных средах, поддерживая Windows, Linux, macOS, а также облачные платформы, такие как AWS, Azure и Google Cloud. Это делает его универсальным решением для гибридных инфраструктур. Механизм мониторинга целостности файлов (FIM) позволяет отслеживать изменения критически важных файлов и конфигураций, а система сканирования уязвимостей помогает выявлять потенциальные точки компрометации до их эксплуатации злоумышленниками. Благодаря активному сообществу и постоянным обновлениям Wazuh остаётся актуальным инструментом в условиях быстро меняющегося ландшафта киберугроз.

```
defender@snrk: /  
● wazuh-manager.service - Wazuh manager  
   Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; enabled; preset: enabled)  
   Active: active (running) since Sun 2025-05-25 19:31:04 UTC; 41min ago  
     Tasks: 141 (limit: 9285)  
    Memory: 823.0M (peak: 824.0M)  
       CPU: 1min 15.849s  
    CGroup: /system.slice/wazuh-manager.service  
            └─1355 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py  
              └─1394 /var/ossec/bin/wazuh-authd  
                └─1403 /var/ossec/bin/wazuh-db  
                  └─1410 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py  
                    └─1413 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py  
                      └─1416 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py  
                        └─1441 /var/ossec/bin/wazuh-execd  
                          └─1455 /var/ossec/bin/wazuh-analysisd  
                            └─1464 /var/ossec/bin/wazuh-syscheckd  
                              └─1482 /var/ossec/bin/wazuh-remoted  
                                └─1493 /var/ossec/bin/wazuh-logcollector  
                                  └─1502 /var/ossec/bin/wazuh-monitord  
                                    └─1511 /var/ossec/bin/wazuh-modulesd  
  
May 25 19:30:58 snrk env[1059]: Started wazuh-db...  
May 25 19:30:59 snrk env[1059]: Started wazuh-execd...  
May 25 19:30:59 snrk env[1059]: Started wazuh-analysisd...  
May 25 19:31:00 snrk env[1059]: Started wazuh-syscheckd...  
May 25 19:31:01 snrk env[1059]: Started wazuh-remoted...  
May 25 19:31:01 snrk env[1059]: Started wazuh-logcollector...  
May 25 19:31:01 snrk env[1059]: Started wazuh-monitord...  
May 25 19:31:02 snrk env[1059]: Started wazuh-modulesd...  
lines 1-29
```

Рисунок 7 - Проверка статуса программы Wazuh

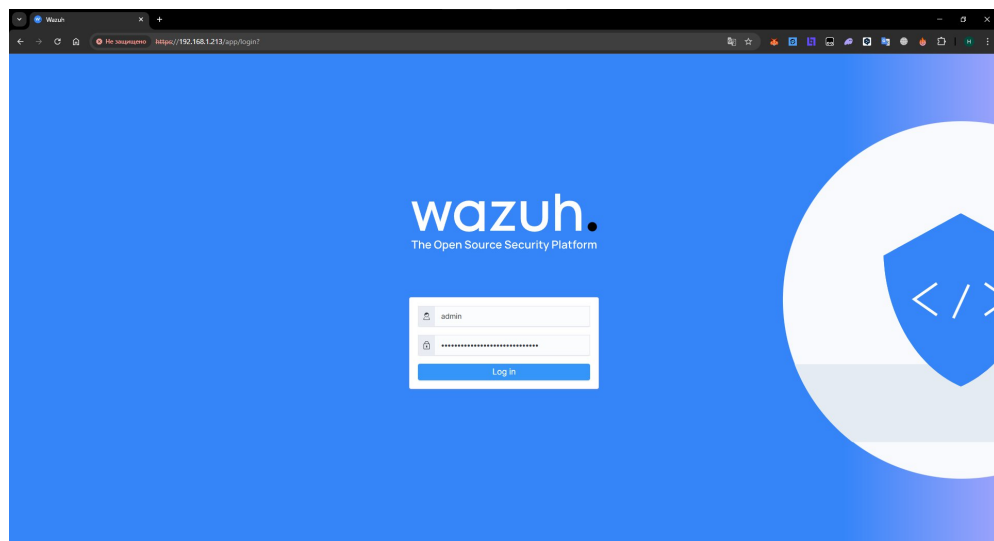


Рисунок 8 – Переходим в браузер и попадаем в Dashboard Wazuh



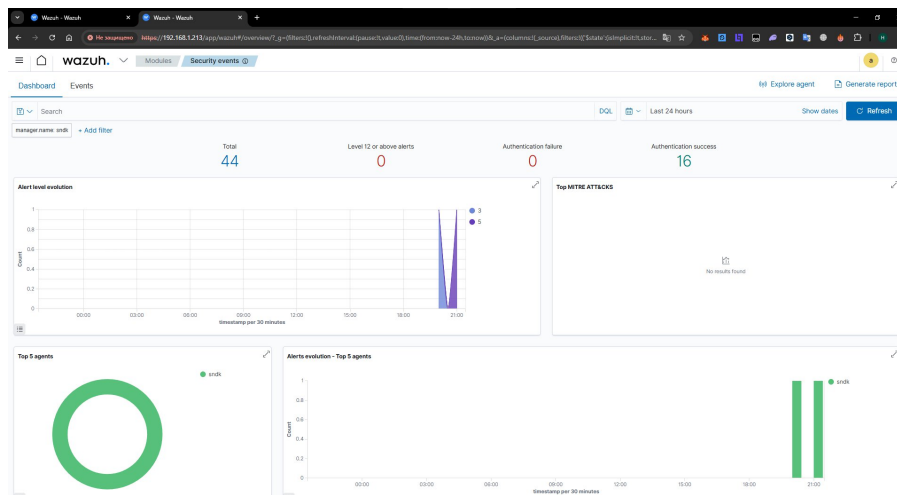


Рисунок 9 – Dashboard программы Wazuh

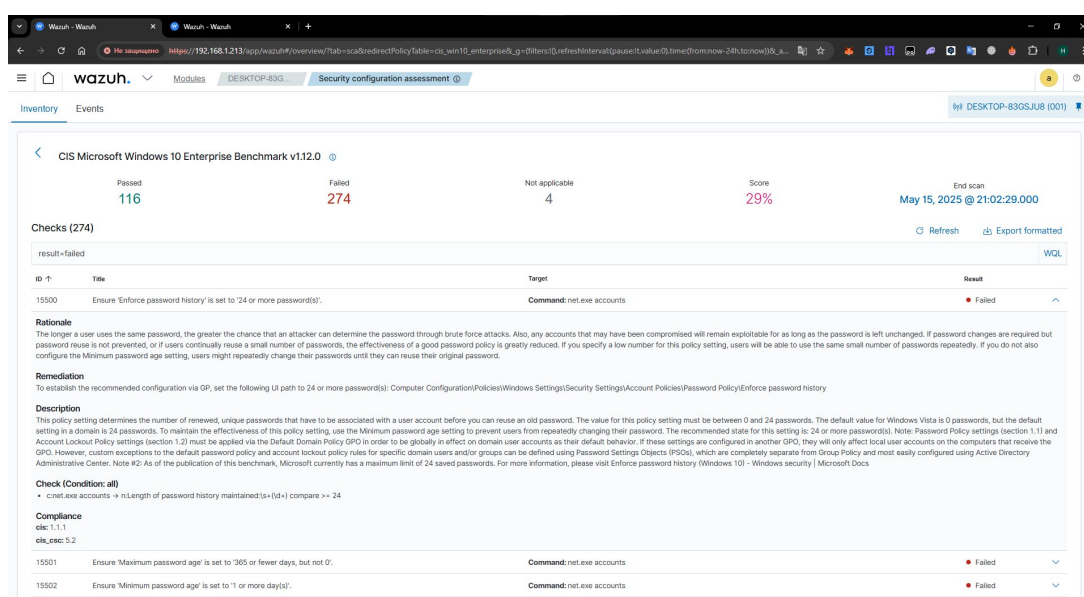


Рисунок 10 – Обнаруженные уязвимости безопасности на «Агенте»

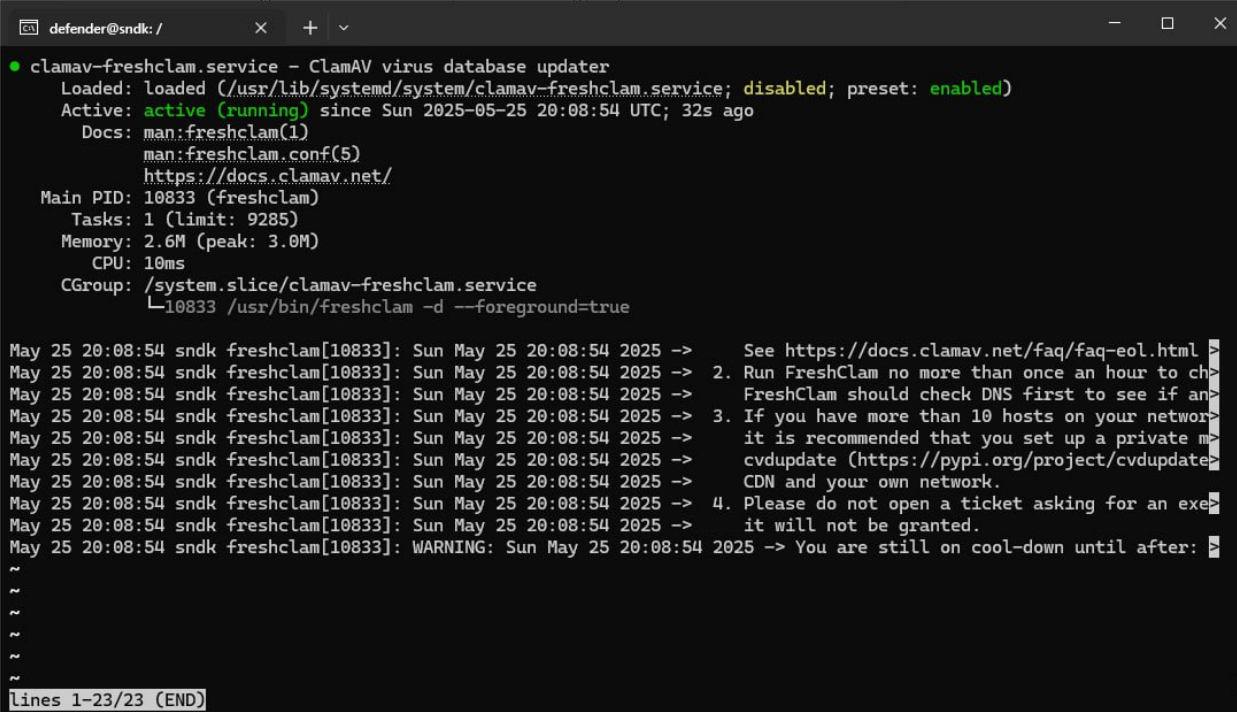
3. ClamAV — это высокопроизводительный антивирусный сканер с открытым исходным кодом, предназначенный для обнаружения и нейтрализации широкого спектра вредоносных программ, включая вирусы, трояны, черви, руткиты и шпионское ПО. Система сочетает в себе несколько методов анализа: сигнатурное сканирование на основе регулярно обновляемых вирусных баз, эвристический анализ для выявления новых и неизвестных угроз, а также детектирование обфусцированного и упакованного вредоносного кода.

Ключевым преимуществом ClamAV является его кроссплатформенность — антивирус работает на Linux, Windows, macOS и



различных Unix-системах, что делает его универсальным решением для защиты разнородной ИТ-инфраструктуры. Благодаря модульной архитектуре, ClamAV легко интегрируется с платформами SIEM для централизованного мониторинга угроз. Для крупных организаций особенно ценна поддержка распределенного сканирования через Clamd — демон, позволяющий проверять файлы на нескольких узлах одновременно.

ClamAV отличается высокой производительностью даже при обработке больших объемов данных, что достигается за счет оптимизированных алгоритмов сканирования и поддержки многопоточности. Система также включает дополнительные защитные механизмы, такие как анализ файлов в карантинной зоне (sandbox) и проверка цифровых подписей. Благодаря этим особенностям ClamAV остается одним из наиболее востребованных антивирусных решений для организаций, ищущих надежную, гибкую и экономически эффективную защиту от киберугроз.



```
defender@sndk: /
● clamav-freshclam.service - ClamAV virus database updater
   Loaded: loaded (/usr/lib/systemd/system/clamav-freshclam.service; disabled; preset: enabled)
   Active: active (running) since Sun 2025-05-25 20:08:54 UTC; 32s ago
     Docs: man:freshclam(1)
           man:freshclam.conf(5)
           https://docs.clamav.net/
   Main PID: 10833 (freshclam)
     Tasks: 1 (limit: 9285)
    Memory: 2.6M (peak: 3.0M)
       CPU: 10ms
    CGroup: /system.slice/clamav-freshclam.service
            └─10833 /usr/bin/freshclam -d --foreground=true

May 25 20:08:54 sndk freshclam[10833]: Sun May 25 20:08:54 2025 -> See https://docs.clamav.net/faq/faq-eol.html
May 25 20:08:54 sndk freshclam[10833]: Sun May 25 20:08:54 2025 -> 2. Run FreshClam no more than once an hour to ch
May 25 20:08:54 sndk freshclam[10833]: Sun May 25 20:08:54 2025 -> FreshClam should check DNS first to see if an
May 25 20:08:54 sndk freshclam[10833]: Sun May 25 20:08:54 2025 -> 3. If you have more than 10 hosts on your network
May 25 20:08:54 sndk freshclam[10833]: Sun May 25 20:08:54 2025 -> it is recommended that you set up a private
May 25 20:08:54 sndk freshclam[10833]: Sun May 25 20:08:54 2025 -> cvdupdate (https://pypi.org/project/cvdupdate
May 25 20:08:54 sndk freshclam[10833]: Sun May 25 20:08:54 2025 -> CDN and your own network.
May 25 20:08:54 sndk freshclam[10833]: Sun May 25 20:08:54 2025 -> 4. Please do not open a ticket asking for an exe
May 25 20:08:54 sndk freshclam[10833]: Sun May 25 20:08:54 2025 -> it will not be granted.
May 25 20:08:54 sndk freshclam[10833]: WARNING: Sun May 25 20:08:54 2025 -> You are still on cool-down until after:
~
~
~
~
lines 1-23/23 (END)
```

Рисунок 11 – Запустили ClamAV на активное сканирование