Второв Константин

ДОМАШНИЙ SOC

КНИГА АДМИНИСТРАТОРА И АНАЛИТИКА SOC LAB

ИЗДАНИЕ 5.0

ПЕРЕРАБОТАННОЕ И ДОПОЛНЕННОЕ

WAZUH SURICATA IRIS VELICIRAPTOR MISP TELEGRAM

РУКОВОДСТВО ПО ВНЕДРЕНИЮ SOC LAB
НА БАЗЕ ОТКРЫТЫХ РЕШЕНИЙ

Константин Второв

Домашний SOC. Книга для администратора и аналитика

https://litres.ru/74222614
SelfPub; 2026

Аннотация

«Домашний SOC» — практическое руководство по созданию и эксплуатации собственного центра мониторинга информационной безопасности на базе открытых решений. Книга ведёт читателя от подготовки Debian 12 и Docker до построения рабочего контура обнаружения, анализа и расследования инцидентов.

Подробно рассматриваются Wazuh, Suricata IDS, DFIR-IRIS, MISP, Velociraptor, Greenbone/OpenVAS, локальный AI-контур на базе Ollama и Qwen, FinCERT IOC Pipeline, WireGuard и Telegram-уведомления. Отдельное внимание уделено FIM, SCA, Rootcheck, управлению уязвимостями, работе с IOC, фильтрации ложных срабатываний, приоритизации, дедупликации и playbookам реагирования.

Издание содержит команды, конфигурации, проверки, чек-листы, схемы архитектуры и реальные примеры расследований. Книга будет полезна администраторам, аналитикам SOC,

специалистам по информационной безопасности, студентам и всем, кто хочет самостоятельно собрать лабораторный SOC.

Содержание

Константин Второв
Домашний SOC. Книга для администратора и аналитика

О редакции 5

Издание 5 создается на базе полного исходного комплекта редакции 4. Оно не сокращает установочные и эксплуатационные главы, а добавляет фактические изменения стенда, накопленные после практической эксплуатации.

Основные изменения

• DFIR-IRIS закреплен как основная система incident response вместо TheHive/Cortex;

• Wazuh актуализирован до фактической версии 4.14.5;

• добавлена безопасная архитектура custom-iris: production filter, contextual assessment, classification, severity, priority, Telegram policy и fail-open dedup;

• SCA отделен от malware/exploitation и остается в Wazuh;

• Rootcheck обрабатывается как host-anomaly, добавлены canonical path и dedup merged-/usr;

• подробно описаны проверки passwd, chfn, chsh через inode, dpkg и контрольные суммы;

• добавлен удаленный VPS 10.77.77.40 и Wazuh Agent vps-telegram-gateway;

• расширены WireGuard, JumpServer и timezone-aware расследования;

• добавлен Greenbone/OpenVAS как автономный внешний vulnerability scanner; внешние клиентские отчеты не смешиваются с SOC, а проверенные находки по собственному контуру переносятся в IRIS вручную;

• включены реальные ошибки патчей, bind mount, отсутствующий jq, restart loop и диагностика по слоям;

• добавлены playbook-и SCA/Rootcheck, чек-лист Greenbone и приемка редакции.

• добавлена архитектурная схема на белом фоне с разделением автоматических, ручных и автономных потоков.

Три режима работы

1. **операторский**: контроль здоровья, очередь IRIS, первичная проверка и документирование;

2. **аналитический**: Wazuh, Suricata, MISP, Velociraptor, Greenbone и AI для проверки гипотез;

3. **административный**: Docker, systemd, сети, сертификаты, права, backup, обновление и rollback.

Маршрут эксплуатации

Health/NOC ->Wazuh/Suricata/Greenbone ->DFIR-IRIS ->IOC enrichment ->endpoint triage ->decision ->remediation ->regression/rescan ->lessons learned

Введение и границы проекта — Назначение книги

Данная книга является практической инструкцией по сборке и эксплуатации лабораторного SOC на базе Debian 12. Она построена не как каталог продуктов, а как воспроизводимый маршрут: подготовка платформы, развертывание компонентов, проверка цепочки событий, расследование, мониторинг и восстановление.

Главная цель - получить стенд, в котором сетевые события анализируются Suricata, события endpoint-ов и IDS-сенсора поступают в Wazuh, значимые alerts передаются в DFIR-IRIS, IOC проверяются в MISP, endpoint triage выполняется через Velociraptor, а состояние инфраструктуры контролируется Prometheus, Grafana, Blackbox Exporter и SOC Health Lite.

Введение и границы проекта — Фактическая редакция стенда

В актуальной редакции используются:

- Debian 12 как базовая ОС SOC-сервера;
- Docker и Docker Compose для серверных компонентов;
- Wazuh как SIEM/XDR-ядро;
- Suricata как IDS-сенсор зеркалируемого трафика;
- DFIR-IRIS как центр alerts, cases, tasks, observables, notes и решений;
- MISP как локальная Threat Intelligence-платформа;
- Velociraptor как DFIR, endpoint triage и threat hunting;
- Prometheus, Grafana, cAdvisor, Node Exporter и Blackbox Exporter как контур мониторинга;
- Suricata Admin Lite и SOC Health Lite как эксплуатационные панели;

• FinCERT IOC Pipeline как полуавтоматический импорт IOC;

• внешний AI-хост как опциональный помощник аналитика;

• JumpServer и WireGuard как контролируемый удаленный доступ;

• Greenbone/OpenVAS как внешний сетевой сканер уязвимостей.

Введение и границы проекта — Фактический профиль стенда.

На SOC-сервере локальные контейнеры Ollama и OpenWebUI не являются обязательными и могут быть полностью удалены. Интеграция custom-ai сохраняется, если она отправляет данные на отдельный AI-хост. Это уменьшает нагрузку и разделяет критичный SOC-контур и ресурсоемкий LLM-инференс.

Введение и границы проекта — Что не входит в эту книгу

Подробное администрирование JumpServer вынесено в отдельную книгу. Здесь описываются только точки интеграции: защищенная публикация интерфейсов, WireGuard-маршрутизация и подключение удаленных Wazuh Agents.

Книга не является разрешением на автоматические destructive-действия. Удаление файлов, изоляция хоста, блокировка пользователей, изменение firewall и массовые response-команды выполняются только после решения и фиксации в DFIR-IRIS.

Введение и границы проекта — Три режима работы

1. **Оператор**: проверяет NOC Wallboard, разбирает очередь IRIS и документирует результат.

2. **Аналитик**: использует Wazuh, MISP, IRIS modules, Velociraptor и AI для проверки гипотез.

3. **Администратор**: обслуживает Docker, systemd, сети, сертификаты, права, резервные копии и восстановление.

Смешивание ролей является типовой причиной ошибок. Оператор не должен перезапускать контейнеры только потому, что карточка Grafana стала красной; администратор сначала устанавливает причину, затем меняет конфигурацию.

Введение и границы проекта — Правило трех проверок

Компонент считается рабочим только если выполнены три проверки:

1. процесс или контейнер запущен и не находится в restart loop;

2. порт или сокет доступен из нужной зоны;

3. прикладной сценарий работает: API отвечает, alert создается, IOC импортируется, правило загружается или probe возвращает успех.

```
docker ps --format "table {{.Names}}\t{{.Status}}\t{{.Ports}}"
docker stats --no-stream
ss -lntp
systemctl --failed
```

Введение и границы проекта — Правило диагностики по слоям

При отказе нельзя начинать с переустановки. Проверка идет снизу вверх:

1. питание, кабель, link и IP-адрес;

2. маршрут и firewall;

3. Docker network и порт;

4. состояние контейнера и restart count;

5. права на bind mount, volume, сертификат или базу;

6. конфигурация и секреты;

7. прикладной запрос и интеграция.

Введение и границы проекта — Журнал изменений

Рекомендуется вести эксплуатационный журнал:

```
mkdir -p /opt/soc/docs
nano /opt/soc/docs/operations-log.md
```

Минимальная запись содержит дату, компонент, симптом, причину, изменение, проверку результата и способ отката.

Введение и границы проекта — Что нового в издании 5

Издание включает полный материал предыдущей редакции и расширяет его практическими главами о качестве детектирования, production-grade интеграции Wazuh с IRIS, удаленном VPS, Greenbone/OpenVAS, реальных ошибках внедрения и управлении изменениями. Особый акцент сделан на доказательной проверке: alert не считается инцидентом только из-за тревожного названия.

Архитектура SOC Lab — Целевая схема

SOC Lab представляет собой связанный контур обнаружения, расследования, Threat Intelligence и контроля собственной работоспособности.

Endpoint telemetry ->Wazuh Agent ->Wazuh Manager

Mirrored network traffic ->Suricata ->eve-wazuh.json ->Wazuh Agent

Suricata severity 1 ->Wazuh rule 100601 level 8 ->IRIS / AI

Suricata severity 2 ->Wazuh rule 100602 level 4 ->Wazuh only

IOC archives ->FinCERT pipeline ->rules / lists / MISP Events

Endpoint triage ->Velociraptor ->evidence ->IRIS Case

Health monitoring ->Prometheus / Grafana / Blackbox / Health Lite

Remote agents ->WireGuard ->Jump ->SOC 10.77.77.30 ->Wazuh

Vulnerability scanning ->Greenbone/OpenVAS ->autonomous reports / manual transfer

Схема на рисунке ниже показывает не только основные

компоненты, но и границу автоматизации. Сплошные линии обозначают постоянные потоки событий и результатов обработки. Пунктиром показаны ручные или опциональные действия. Greenbone/OpenVAS намеренно остается автономным: он может сканировать как собственные активы, так и внешние клиентские объекты, которые не входят в инвентарь Wazuh и не должны автоматически создавать карточки домашнего SOC.

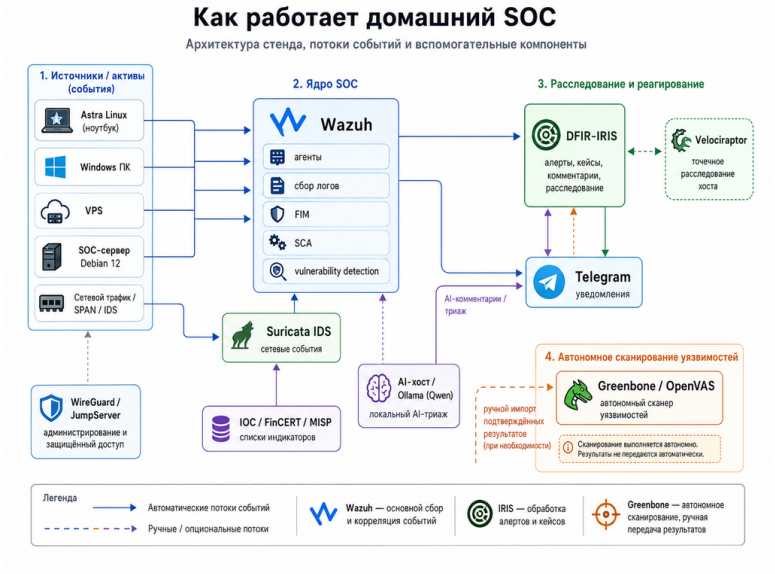

Рисунок. Архитектура домашнего SOC Lab и основные потоки данных.

Архитектура SOC Lab — Фактиче

Конец ознакомительного фрагмента.