DevOS

Операционная Система для Разработчиков

Industry 4.0 Ready • Developer First • Security Always

Проблемы Разработчиков

- Нестабильное окружение (зависимости от версий ОС)
- Сложная настройка инструментов и workflows
- Нарушение безопасности перед деплоем
- Ручная работа, которую можно автоматизировать

Решение: DevOS — ОС для DevSecOps

DevOS — это **полнофункциональная операционная система**, созданная на базе Fedora Linux, специально оптимизированная для разработчиков, DevOps инженеров и специалистов по безопасности.

9 встроенных утилит автоматизируют весь цикл разработки: от анализа проекта до мониторинга безопасности.

Архитектура DevOS

- ▶ Ядро: Linux LTS + Fedora Linux
- ▶ d-* Toolkit: 9 встроенных утилит для безопасности, автоматизации и управления.
- ▶ Контейнеризация: Docker, Kubernetes, GPU support
- ▶ DevSecOps: Встроенные security checks в процессы разработки.

d-env: Анализ Проекта за Секунду

Язык: Go

Мгновенный анализ проекта: Git статус, Docker контейнеры, зависимости, структура, поиск секретов

```
$ d-env
🔄 GIT: [main ↑3] | 🐳 Docker: 3/3 running | 📦 Python 3.11
🔓 ✔ No secrets | ⚠ Dependencies: 45 packages
```

d-env: Функции

Git Analysis: Бранч, коммиты, состояние репо

Dependency Detection: Python, Node.js, Go, Rust, Java

Docker/Kubernetes: Статус контейнеров и сервисов

Security Check: Сканирование состояния проекта.

d-guard: Безопасность Перед Коммитом

Язык: Go

Полный security линтер: секреты, уязвимости, SAST, Docker линтинг

```
$ d-guard scan --staged --fix
✅ Secrets: PASS | ⚠ CVE: 3 HIGH | ✅ SAST: PASS
✔ Auto-fixed dependencies in requirements.txt
```

d-guard: DevSecOps утилита

Secrets Scanning: gitleaks интеграция

Dependency Vulnerabilities: trivy, osv-scanner

SAST Analysis: semgrep для обнаружения уязвимостей

Auto-fix Mode: Автоматическое исправление проблем

d-ci: Мониторинг CI/CD Пайплайнов

Язык: Go

Real-time мониторинг GitLab, GitHub, Jenkins пайплайнов из терминала

```
$ d-ci view --refresh 10
main (#1234) ✓ | dev (#1233) ↻ | feat-x ✗
compile ✓ 4.2s | test ✓ 45.6s | deploy ↻
```

d-ci: CI/CD Control Plane

Multi-Provider: GitLab, GitHub поддержка

TUI Interface: Real-time обновления, функциональный вывод

Pipeline Control: Запуск, перезапуск, просмотр логов

Webhook Alerts: Мгновенные уведомления о результатах

d-recon: Сканирование Целей

АРХИТЕКТУРА: Go

Полное сканирование с одной команды: поддомены, порты, уязвимости, OSINT

```
$ d-recon -t example.com --profile full
Subdomains: 12 | Ports: 4 open | Vulns: 2 found
Report saved: report.html
```

d-recon: Multi-Engine Orchestrator

Parallel Scanning: Nmap, subfinder, bbot, массивное распараллеливание

Result Aggregation: Единая база данных результатов

Profiles: stealth, quick, full конфигурации

d-top: Мониторинг Инфраструктуры

Язык: Golang

Red Team + инфраструктура: мониторинг процессов, сокетов, скрытого трафика

```
$ d-top
CPU: 45% | RAM: 62% | Processes: 142 | Open Ports: 8
Suspicious: nginx (PID 1234) listening on 8.8.8.8:443
```

d-top: System & Security Intelligence

Local Monitor: CPU, RAM, I/O, сокеты каждого процесса

Remote Monitor: Аналогичный для подключаемого устройства.

Red Team Mode: «Тихое» убийство процесса, незаметность, zombie и fork bomb exploits.

Docker View: Визуализация Docker и взаимодействие с контейнерами.

d-shark: Анализ Сети & Фаервол

Язык: Rust

Real-time захват и анализ трафика(L2-L5) с динамическим фаерволом

```
$ sudo d-shark capture eth0 --firewall
In: 1.2 MB/s | Out: 0.8 MB/s | Threats: 2 detected
Rules: 24 active | Blocked: 12 connections
```

d-shark: Network Defense Layer

Packet Analysis: L2-L5 протоколы, DPI (Deep Packet Inspection)

Dynamic Firewall: nftables/iptables интеграция

Threat Detection: Автоматическое блокирование опасного трафика

Stratoshark: Связь пакетов с системными процессами

d-crypt: Шифрование Проектов

Язык: Rust

Встроенное шифрование с Shamir Secret Sharing: разделение ключей между USB ключами

```
$ d-crypt encrypt project/ --keys 3 --threshold 2
✔ Project encrypted (AES-256-GCM)
💾 Key shards written to 3 USB devices
```

d-crypt: Cryptographic Key Management

AES-256-GCM: Высокостойкое шифрование проектов

Shamir Sharing: Разделение ключа на N частей (M необходимо)

Audit Ledger: Логирование всех операций с ключами

USB Key Binding: Ключи хранятся на физических USB-ключах

Почему DevOS?

Полная ОС: Не просто набор утилит, а полнофункциональная система

DevSecOps Встроен: Безопасность в процессах разработки по умолчанию

Автоматизация: 9 мощных утилит покрывают весь цикл разработки

Производительность: Оптимизирована для разработки и deployment

Q&A

Вопросы?

DevOS — Будущее Разработки