راهنمای جامع نصب، راه‌اندازی و بهره‌برداری · نسخه ۱٫۰ · آبان ۱۴۰۵

VulnScope

سامانه‌ی متمرکز بررسی آسیب‌پذیری‌ها (CVE) برای سرورها و ایستگاه‌های کاری

جمع‌آوری خودکار موجودی نرم‌افزار، تطبیق با پایگاه‌های داده‌ی معتبر جهانی و اولویت‌بندی بر پایه‌ی شواهد واقعی بهره‌برداری و احتمال سوءاستفاده — در یک داشبورد فارسی.

macOS | Linux | Windows 11 / 10 | Windows Server 2016–2025

NVD API 2.0
اطلاعات و امتیاز CVSS

CISA KEV
شواهد بهره‌برداری فعال

FIRST EPSS
احتمال سوءاستفاده

OSV.dev
بولتن‌های توزیع لینوکس

فهرست مطالب

۱	معرفی و قابلیت‌ها	VulnScope چیست و چه مسئله‌ای را حل می‌کند
۲	معماری سامانه	Agent، سرور، موتور تطبیق، داشبورد
۳	منابع داده: NVD، CISA KEV، EPSS، OSV	هر منبع چه می‌دهد و چطور استفاده می‌شود
۴	روش تطبیق برای هر سیستم‌عامل	ویندوز، لینوکس، مک و نرم‌افزارها
۵	نصب و راه‌اندازی سرور مرکزی	TLS، توکن، کلید NVD، سرویس systemd
۶	نصب Agent روی Windows Server و Windows 11	PowerShell و Scheduled Task
۷	نصب Agent روی لینوکس و macOS	systemd timer و launchd
۸	کار با داشبورد	میزبان‌ها، یافته‌ها، فیلترها، خروجی
۹	اولویت‌بندی و برنامه‌ی رفع آسیب‌پذیری	از KEV تا EPSS
۱۰	سفارشی‌سازی و افزودن نرم‌افزار	cpe_map.json
۱۱	مرجع API	برای اتوماسیون و یکپارچه‌سازی
۱۲	امنیت، نگهداری، محدودیت‌ها و عیب‌یابی	
پ	پیوست: نمونه‌ی نتایج واقعی	داده‌های واقعی NVD، KEV و EPSS

پیش از شروع

این ابزار فقط «نسخه»ها را می‌خواند و هیچ آزمون نفوذ، اسکن پورت یا تغییری در سیستم‌ها انجام نمی‌دهد. بنابراین روی سرورهای عملیاتی بی‌خطر است و بار پردازشی ناچیزی دارد. با این حال، طبق سیاست سازمان، پیش از استقرار گسترده مجوز لازم را بگیرید.

۱ معرفی و قابلیت‌ها

VulnScope یک سامانه‌ی خودمیزبان (Self-hosted) برای کشف آسیب‌پذیری‌های شناخته‌شده (CVE) در سیستم‌عامل‌ها و نرم‌افزارهای نصب‌شده است. روی هر سیستم یک **Agent** سبک اجرا می‌شود که نسخه‌ی دقیق سیستم‌عامل (برای ویندوز تا سطح Build و UBR)، فهرست بسته‌ها و نرم‌افزارها و هات‌فیکس‌ها را جمع می‌کند و به **سرور مرکزی** می‌فرستد. سرور این موجودی را با پایگاه‌های داده‌ی معتبر تطبیق می‌دهد و نتیجه را در داشبورد فارسی، به ترتیب اولویت واقعی، نمایش می‌دهد.

چرا فقط CVSS کافی نیست؟

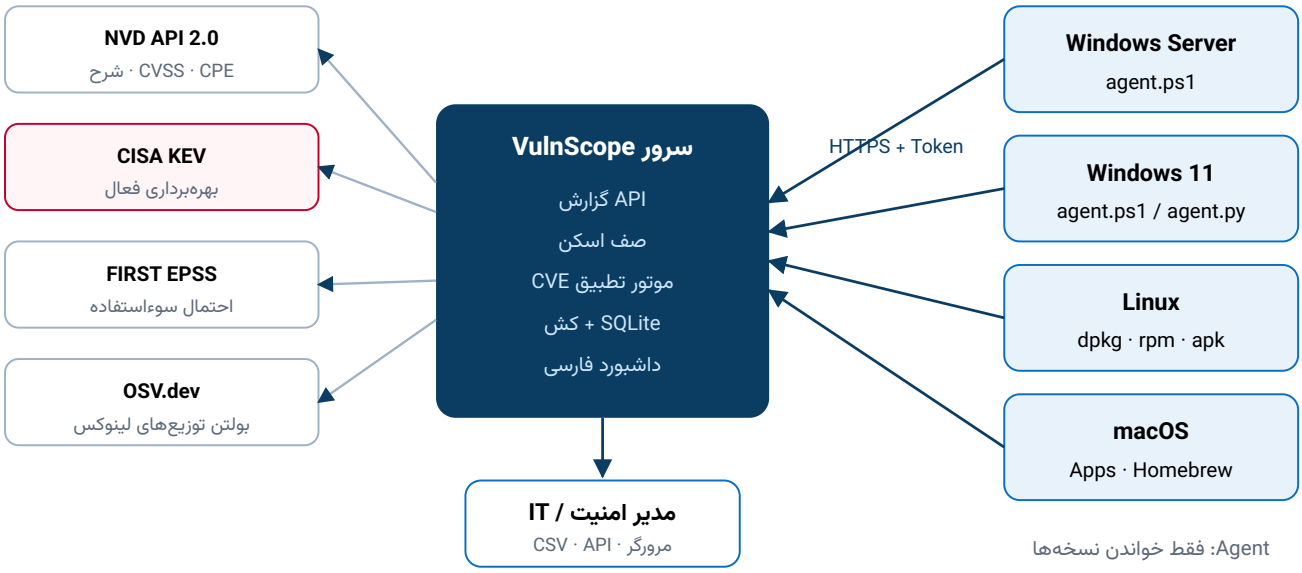
در یک شبکه‌ی معمولی هزاران CVE «بالا» و «بحرانی» پیدا می‌شود؛ رفع همه‌ی آن‌ها همزمان ممکن نیست. پژوهش‌ها نشان می‌دهند تنها بخش کوچکی از CVEها واقعاً مورد بهره‌برداری قرار می‌گیرند. به همین دلیل VulnScope سه سیگنال را کنار هم می‌گذارد:

سیگنال	پرسشی که پاسخ می‌دهد	منبع
KEV شواهد واقعی	آیا مهاجمان همین حالا از آن استفاده می‌کنند؟	کاتالوگ Known Exploited Vulnerabilities سازمان CISA آمریکا
EPSS احتمال	چقدر احتمال دارد ظرف ۳۰ روز آینده سوءاستفاده شود؟	مدل Exploit Prediction Scoring System از FIRST
CVSS شدت	اگر سوءاستفاده شود، اثرش چقدر جدی است؟	پایگاه ملی آسیب‌پذیری NVD (NIST)

قابلیت‌های اصلی

- **چندسکویی:** Windows Server 2016/2019/2022/2025، Windows 10/11، Ubuntu، Debian، Alpine، Rocky، AlmaLinux و macOS.
- **Agent ویندوز بدون Python:** نسخه‌ی PowerShell برای سرورهای ویندوزی که نصب Python روی آن‌ها مجاز نیست.
- **دقت در سطح Build ویندوز:** مقایسه‌ی `10.0.20348.2340` با بازه‌ی `versionEndExcluding` هر CVE در NVD؛ بنابراین نصب یک Cumulative Update بلافاصله در نتایج دیده می‌شود.
- **درک Backport در لینوکس:** بسته‌های توزیع با بولتن‌های رسمی همان توزیع (از طریق OSV) تطبیق داده می‌شوند، نه با نسخه‌ی بالادستی؛ این کار مثبت کاذب را به شدت کم می‌کند.
- **نشان باج‌افزار:** CVEهایی که CISA استفاده‌ی آن‌ها در کارزارهای باج‌افزاری را تأیید کرده، جداگانه علامت می‌خورند.
- **نسخه‌ی اصلاح:** برای هر یافته، نسخه یا Buildای که آسیب‌پذیری در آن رفع شده نمایش داده می‌شود.
- **امتیاز ریسک میزبان، خروجی CSV، اسکن مجدد خودکار روزانه و کش ۲۴ ساعته** برای رعایت محدودیت نرخ APIها.
- **بدون وابستگی:** فقط کتابخانه‌ی استاندارد Python و پایگاه داده‌ی SQLite؛ مناسب شبکه‌های با دسترسی محدود.

۲ معماری سامانه



جریان کار

1. **جمع‌آوری:** Agent طبق زمان‌بندی (پیش‌فرض روزانه) اجرا می‌شود، موجودی را می‌سازد و با توکن مشترک از طریق HTTPS به `/api/report` می‌فرستد.
2. **صف:** سرور گزارش را ذخیره می‌کند و میزبان را در صف اسکن قرار می‌دهد. اسکن‌ها یکی‌یکی اجرا می‌شوند تا محدودیت نرخ NVD رعایت شود.
3. **تطبیق:** موتور تطبیق بسته به نوع سیستم‌عامل از OSV یا NVD استفاده می‌کند (فصل ۴).
4. **غنی‌سازی:** برای هر CVE وضعیت KEV (به‌همراه سررسید و استفاده در باج‌افزار) و امتیاز EPSS اضافه می‌شود.
5. **نمایش:** داشبورد هر ۱۵ ثانیه به‌روز می‌شود؛ هر ۲۴ ساعت همه‌ی میزبان‌ها با داده‌ی تازه دوباره اسکن می‌شوند، حتی اگر Agent گزارش جدیدی نفرستاده باشد.

ساختار فایل‌ها

```
vulnscope/
├── agent/agent.py            Agent لینوکس، macOS و ویندوز (Python 3.8+)
├── agent/agent.ps1           Agent ویندوز بدون Python (PowerShell 5.1+)
├── server/server.py          API، داشبورد، صف اسکن، TLS
├── server/matcher.py         موتور تطبیق: NVD / OSV / KEV / EPSS
├── server/db.py              SQLite، کش و امتیاز ریسک
├── server/cpe_map.json       نگاشت نام نرم‌افزار به CPE
├── server/demo.py            داده‌ی نمایشی آفلاین
├── server/static/index.html  داشبورد فارسی
├── install/                  نصب‌کننده‌های Agent برای هر سیستم‌عامل
└── tests/                    تست‌ها + داده‌ی واقعی NVD، KEV و EPSS
```

۳ منابع داده

همه‌ی منابع زیر عمومی، رایگان و معتبرند. سرور VulnScope باید به آدرس‌های آن‌ها دسترسی خروجی HTTPS داشته باشد (Agentها نیازی به اینترنت ندارند).

NVD — National Vulnerability Database (NIST)

`https://services.nvd.nist.gov/rest/json/cves/2.0`

مرجع اصلی اطلاعات CVE: شرح، تاریخ انتشار، امتیاز و بردار **CVSS** (نسخه‌های 3.1، 3.0، 4.0 و 2) و مهم‌تر از همه **پیکربندی CPE**؛ یعنی دقیقاً کدام محصول در کدام بازه‌ی نسخه آسیب‌پذیر است. VulnScope همه‌ی CVEهای هر محصول را یک بار دریافت، خلاصه و ۲۴ ساعت کش می‌کند و سپس برای هر میزبان، بازه‌ها را به‌صورت محلی ارزیابی می‌کند.

محدودیت نرخ: بدون کلید ۵ درخواست در ۳۰ ثانیه، با کلید رایگان ۵۰ درخواست. دریافت کلید: nvd.nist.gov/developers/request-an-api-key

CISA KEV — Known Exploited Vulnerabilities

`https://www.cisa.gov/sites/default/files/feeds/known_exploited_vulnerabilities.json`

فهرست رسمی CVEهایی که سازمان امنیت سایبری آمریکا (CISA) **شواهد قطعی بهره‌برداری فعال** از آن‌ها را تأیید کرده است. برای هر مورد این اطلاعات استفاده می‌شود: تاریخ افزوده‌شدن، **سررسید رفع** (dueDate) برای نهادهای دولتی آمریکا، اقدام لازم و این‌که آیا در **کارزارهای باج‌افزاری** استفاده شده است. اگر cisa.gov در دسترس نباشد، آینه‌ی رسمی CISA در GitHub (cisagov/kev-data) به‌طور خودکار استفاده می‌شود.

در زمان تهیه‌ی این راهنما، نسخه‌ی کاتالوگ 2026.10.30 با ۱۷۶۶ CVE بود.

FIRST EPSS — Exploit Prediction Scoring System

`https://api.first.org/data/v1/epss`

مدل یادگیری ماشین FIRST که هر روز برای همه‌ی CVEها **احتمال مشاهده‌ی تلاش برای سوءاستفاده در ۳۰ روز آینده** را (عددی بین ۰ و ۱) و **صدک** آن را منتشر می‌کند. مثلاً EPSS برابر ۰٫۷۰ با صدک ۹۹٫۴ یعنی این CVE از ۹۹٫۴٪ کل CVEها پرخطرتر است. VulnScope امتیازها را دسته‌ای (۱۰۰ CVE در هر درخواست) می‌گیرد.

OSV.dev — Open Source Vulnerabilities

`https://api.osv.dev/v1/querybatch`

پایگاه داده‌ی گوگل که بولتن‌های امنیتی رسمی توزیع‌های لینوکس (Debian DSA/DLA، Ubuntu USN، Alpine، Rocky، AlmaLinux) را در قالب یکسان گردآوری می‌کند. چون توزیع‌ها وصله‌ها را **Backport** می‌کنند (مثلاً OpenSSH نسخه‌ی 8.9p1 در Ubuntu وصله شده ولی شماره‌ی بالادستی‌اش تغییر نکرده)، تطبیق بسته‌های لینوکس فقط از این طریق دقیق است.

> **نکته‌ی مهم درباره‌ی شبکه‌های با دسترسی محدود**
>
> اگر سرور به اینترنت دسترسی مستقیم ندارد، از طریق Proxy سازمانی (متغیرهای `HTTPS_PROXY`) به این چهار دامنه دسترسی بدهید: `services.nvd.nist.gov`، `www.cisa.gov` (یا `raw.githubusercontent.com`)، `api.first.org` و `api.osv.dev`.

۴ روش تطبیق برای هر سیستم‌عامل

سیستم	چه چیزی خوانده می‌شود	منبع تطبیق	شیوه
Windows Server	ProductName، InstallationType، CurrentBuild، **UBR** از رجیستری؛ برنامه‌ها از کلیدهای Uninstall (۳۲ و ۶۴ بیتی)؛ Get-HotFix	NVD	Build به CPE نگاشت می‌شود (مثلاً 20348 ← `windows_server_2022`) و نسخه‌ی کامل `10.0.Build.UBR` با بازه‌ی هر CVE مقایسه می‌شود.
Windows 11 / 10	مانند بالا + DisplayVersion (23H2، 24H2…)	NVD	22631 ← `windows_11_23h2`، 26100 ← `windows_11_24h2`، 19045 ← `windows_10_22h2`؛ معماری (x64/arm64) هم بررسی می‌شود.
Linux	/etc/os-release، بسته‌های apk / dpkg / rpm با نام و نسخه‌ی **بسته‌ی منبع**	OSV + NVD (برای تکمیل CVSS)	پرس‌وجوی دسته‌ای ۵۰۰تایی در اکوسیستم دقیق توزیع، مثل `Ubuntu:22.04:LTS` یا `Debian:12`.
macOS	sw_vers، برنامه‌های Applications/ (system_profiler)، بسته‌های Homebrew	NVD	نسخه‌ی macOS با `apple:macos`؛ برنامه‌ها و بسته‌های Homebrew از طریق cpe_map.json.
نرم‌افزارها	Chrome، Firefox، Edge، 7-Zip، WinRAR، PuTTY، WinSCP، VMware Tools/Workstation، Veeam، AnyDesk، TeamViewer و …	NVD	نام با regex به vendor:product نگاشت و نسخه با بازه‌های NVD مقایسه می‌شود.

ارزیابی بازه‌ی نسخه

هر CVE در NVD یک یا چند «معیار تطبیق» (cpeMatch) دارد. مثلاً برای CVE-2024-38063 (آسیب‌پذیری اجرای کد از راه دور در TCP/IP ویندوز) معیار ویندوز سرور ۲۰۲۲ این است:

```
criteria:            cpe:2.3:o:microsoft:windows_server_2022:*:*:*:*:*:*:*:*
versionEndExcluding: 10.0.20348.2655
```

یعنی هر Build کمتر از `20348.2655` آسیب‌پذیر است و همین عدد به‌عنوان «نسخه‌ی اصلاح» در داشبورد نمایش داده می‌شود. موتور مقایسه، نسخه‌ها را جزءبه‌جزء (عددی و حرفی) مقایسه می‌کند، پس `9.8p1 < 9.3p1` و `116.0.5845.187 < 116.0.5845.96` درست تشخیص داده می‌شوند.

> **برای جلوگیری از مثبت کاذب**
>
> معیارهایی که «همه‌ی نسخه‌ها» را بدون هیچ بازه‌ای آسیب‌پذیر اعلام می‌کنند نادیده گرفته می‌شوند، و CVEهایی با وضعیت Rejected در NVD حذف می‌شوند. در پیکربندی‌های ترکیبی (مثلاً «برنامه X روی سیستم‌عامل Y») فقط بخش مربوط به محصول بررسی می‌شود؛ ممکن است گاهی یافته‌ای نشان داده شود که فقط روی سکوی دیگری اثر دارد — ستون «شرح» را بخوانید.

۵ نصب و راه‌اندازی سرور مرکزی

پیش‌نیازها

- یک ماشین مجازی لینوکس (Ubuntu 22.04/24.04 یا Rocky 9) با ۲ vCPU، ۲ گیگابایت RAM و ۱۰ گیگابایت دیسک برای چند صد میزبان.
- Python 3.9 یا بالاتر (بدون نیاز به pip و بسته‌ی اضافی).
- دسترسی HTTPS خروجی به منابع فصل ۳، و دسترسی ورودی از Agentها روی پورت انتخابی (پیش‌فرض 8443).

گام ۱ — کپی فایل‌ها و ساخت کاربر سرویس

```
sudo useradd --system --home /opt/vulnscope --shell /usr/sbin/nologin vulnscope
sudo mkdir -p /opt/vulnscope && sudo cp -r server agent install /opt/vulnscope/
sudo chown -R vulnscope: /opt/vulnscope
```

گام ۲ — گواهی TLS

بهترین گزینه گواهی صادرشده از CA داخلی سازمان است. برای شروع می‌توانید گواهی خودامضا بسازید:

```
cd /opt/vulnscope/server
sudo -u vulnscope openssl req -x509 -newkey rsa:3072 -nodes -days 825 \
  -keyout key.pem -out cert.pem -subj "/CN=vulnscope.corp" \
  -addext "subjectAltName=DNS:vulnscope.corp,IP:10.10.1.50"
```

اگر گواهی خودامضاست، فایل cert.pem را به Agentها بدهید (`cafile`-) یا در Trusted Root ویندوز وارد کنید. از `insecure`- فقط برای تست استفاده کنید.

گام ۳ — توکن و تنظیمات

```
# ساخت توکن تصادفی قوی
python3 -c "import secrets; print(secrets.token_urlsafe(32))"

sudo tee /etc/vulnscope.env >/dev/null <<'EOF'
VULNSCOPE_TOKEN=<توکن ساخته‌شده>
NVD_API_KEY=<کلید رایگان NVD>
VULNSCOPE_DASH_PASSWORD=<رمز داشبورد>
VULNSCOPE_DB=/opt/vulnscope/vulnscope.db
EOF
sudo chmod 600 /etc/vulnscope.env
```

گام ۴ — سرویس systemd

```
sudo tee /etc/systemd/system/vulnscope.service >/dev/null <<'EOF'
[Unit]
Description=VulnScope server
After=network-online.target
[Service]
User=vulnscope
EnvironmentFile=/etc/vulnscope.env
WorkingDirectory=/opt/vulnscope/server
ExecStart=/usr/bin/python3 server.py --port 8443 --cert cert.pem --key key.pem
Restart=on-failure
NoNewPrivileges=true
ProtectSystem=strict
ReadWritePaths=/opt/vulnscope
[Install]
WantedBy=multi-user.target
EOF
sudo systemctl daemon-reload && sudo systemctl enable --now vulnscope
sudo ufw allow 8443/tcp
```

گزینه‌های خط فرمان سرور

گزینه	توضیح
`--token`	توکن مشترک Agentها (حداقل ۱۶ کاراکتر؛ یا متغیر VULNSCOPE_TOKEN)
`--nvd-key`	کلید API رایگان NVD؛ اسکن اول را حدود ۱۰ برابر سریع‌تر می‌کند
`--dashboard-password`	فعال‌سازی ورود (Basic Auth) برای داشبورد و APIهای مدیریتی
`--cert / --key`	فعال‌سازی HTTPS با حداقل TLS 1.2
`--rescan-hours`	بازاسکن خودکار همه‌ی میزبان‌ها (پیش‌فرض ۲۴؛ ۰ = خاموش)
`--port / --host / --db`	پورت (8443)، آدرس شنود (0.0.0.0) و مسیر پایگاه داده

آزمایش بدون اینترنت

با `python3 demo.py --db demo.db` پنج میزبان نمایشی ساخته می‌شود و موتور تطبیق واقعی روی رکوردهای واقعی NVD، کاتالوگ کامل CISA KEV و امتیازهای EPSS موجود در پوشه‌ی tests اجرا می‌شود. سپس سرور را با `-- db demo.db --` اجرا کنید.

۶ نصب Agent روی Windows Server و Windows 11

Agent ویندوز (`agent.ps1`) با PowerShell 5.1 که در همه‌ی نسخه‌های پشتیبانی‌شده‌ی ویندوز موجود است کار می‌کند و به Python نیاز ندارد.

روش ۱ — نصب خودکار با Scheduled Task (پیشنهادی)

پوشه‌ی `vulnscope` را روی سیستم کپی کنید و PowerShell را **Run as Administrator** باز کنید:

```
Set-ExecutionPolicy -Scope Process Bypass -Force
cd C:\Temp\vulnscope\install
.\install-agent-windows.ps1 -Server https://vulnscope.corp:8443 -Token <TOKEN> -Time 02:30
```

این اسکریپت:

- Agent را در `C:\ProgramData\VulnScope` کپی می‌کند؛
- توکن را در فایلی ذخیره می‌کند که فقط SYSTEM و Administrators به آن دسترسی دارند؛
- یک Scheduled Task با نام «VulnScope Agent» با حساب SYSTEM برای اجرای روزانه می‌سازد و یک بار هم بلافاصله اجرا می‌کند.

روش ۲ — استقرار گروهی با GPO یا SCCM/Intune

1. پوشه‌ی vulnscope را در یک Share فقط‌خواندنی (مثلاً `\\fs01\deploy$\vulnscope`) قرار دهید.
2. در GPO: **Computer Configuration** ← **Policies** ← **Windows Settings** ← **Scripts** ← **Startup** یک اسکریپت PowerShell با پارامترهای بالا اضافه کنید؛ یا در SCCM/Intune آن را به‌صورت Script/Win32 App منتشر کنید.
3. چون Task با `-Force` ثبت می‌شود، اجرای مکرر اسکریپت مشکلی ایجاد نمی‌کند.

اجرای دستی و آزمایش

```
# فقط ساخت گزارش محلی برای بررسی
.\agent.ps1 -Output C:\Temp\report.json

# ارسال به سرور
.\agent.ps1 -Server https://vulnscope.corp:8443 -Token <TOKEN>

# بررسی Task
Get-ScheduledTaskInfo -TaskName 'VulnScope Agent'
```

> **نکته برای Windows 11**
>
> رجیستری ویندوز ۱۱ در مقدار ProductName هنوز «Windows 10» می‌نویسد. Agent با بررسی Build (۲۲۰۰۰ به بالا) نام درست را تشخیص می‌دهد. همچنین نسخه‌ی کامل `10.0.22631.4317` شامل UBR است؛ همین عدد پس از نصب Cumulative Update ماهانه افزایش می‌یابد و یافته‌های وصله‌شده در اسکن بعدی خودبه‌خود حذف می‌شوند. روی Server Core هم Agent بدون تغییر کار می‌کند.

۷ نصب Agent روی لینوکس و macOS

لینوکس (systemd)

پیش‌نیاز: python3 (در همه‌ی توزیع‌های رایج به‌طور پیش‌فرض نصب است). توزیع‌های پشتیبانی‌شده‌ی کامل: Ubuntu، Debian، Rocky Linux، Alpine و AlmaLinux.

```
sudo ./install/install-agent-linux.sh https://vulnscope.corp:8443 <TOKEN>

# بررسی
systemctl list-timers vulnscope-agent.timer
journalctl -u vulnscope-agent.service -n 20

# اجرای دستی (لینوکس و مک): فقط گزارش محلی، یا ارسال با گواهی خودامضا
python3 agent/agent.py --output report.json
python3 agent/agent.py --server https://vulnscope.corp:8443 --token <TOKEN> --cafile cert.pem
```

اسکریپت Agent را در `/opt/vulnscope` قرار می‌دهد، توکن را در `/etc/vulnscope-agent.env` با دسترسی 600 ذخیره می‌کند و یک systemd timer روزانه با تأخیر تصادفی تا ۲ ساعت می‌سازد تا همه‌ی سرورها هم‌زمان گزارش نفرستند.

استقرار گروهی با Ansible

```
- hosts: linux_servers
  become: true
  tasks:
    - copy: { src: vulnscope/, dest: /tmp/vulnscope/, mode: preserve }
    - command: /tmp/vulnscope/install/install-agent-linux.sh {{ vs_server }} {{ vs_token }}
```

چه چیزی جمع می‌شود؟

مدیر بسته	اطلاعات
`dpkg`	نام بسته، نسخه، **نام و نسخه‌ی بسته‌ی منبع** (همان چیزی که بولتن‌های Debian/Ubuntu بر اساس آن نوشته می‌شوند)
`rpm`	نام، Epoch، Version-Release، معماری و SRPM
`apk`	از پایگاه داده‌ی `/lib/apk/db/installed` به‌همراه بسته‌ی origin

مثال واقعی: اجرای Agent روی یک Ubuntu 24.04.5 در زمان آزمایش این ابزار، ۱۰۴۴ بسته را در کمتر از یک ثانیه فهرست کرد.

macOS (launchd)

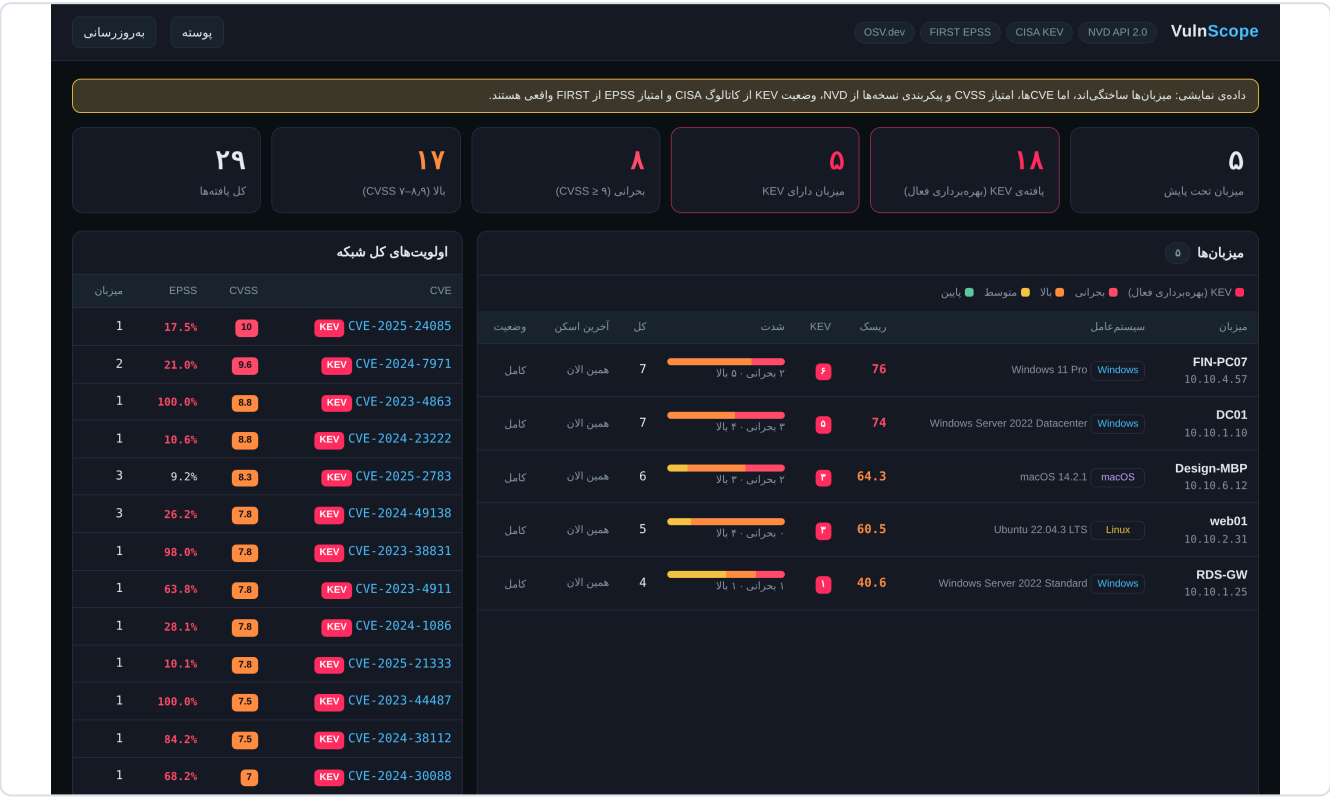
پیش‌نیاز: python3 (با نصب Xcode Command Line Tools با دستور `xcode-select --install` یا از python.org).

```
sudo ./install/install-agent-macos.sh https://vulnscope.corp:8443 <TOKEN>
tail -f /var/log/vulnscope-agent.log
```

یک LaunchDaemon با نام `com.vulnscope.agent` ساخته می‌شود که روزانه ساعت ۳ بامداد و هنگام راه‌اندازی اجرا می‌شود. Agent نسخه و Build سیستم‌عامل، برنامه‌های نصب‌شده (به‌جز برنامه‌های داخلی `System/` که با خود macOS به‌روز می‌شوند) و بسته‌های Homebrew را جمع می‌کند. برای استقرار گروهی می‌توانید همین اسکریپت را از طریق Jamf یا Kandji به‌عنوان Policy Script اجرا کنید.

۸ کار با داشبورد

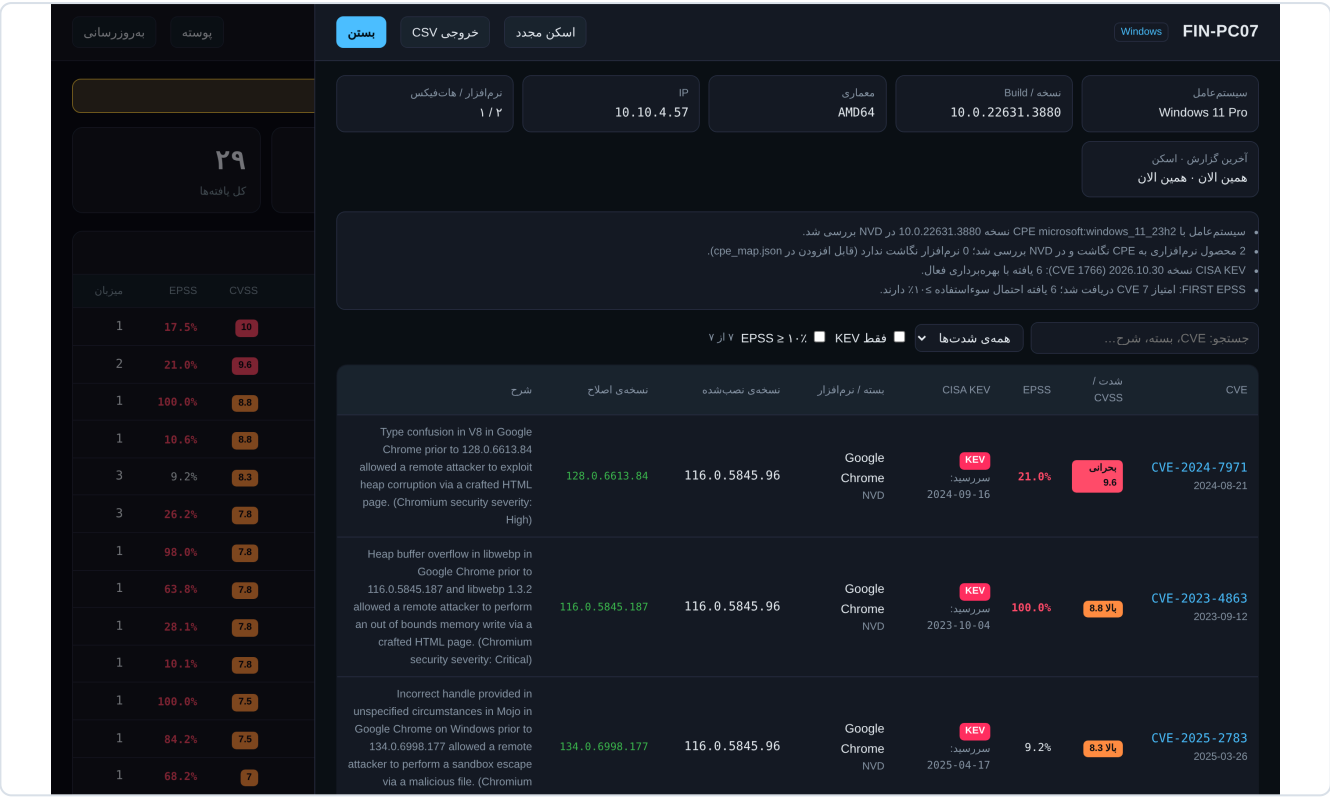
آدرس `https://vulnscope.corp:8443` را در مرورگر باز کنید. اگر رمز داشبورد تعیین شده باشد، با هر نام کاربری و همان رمز وارد شوید.



شکل ۱ — صفحه‌ی اصلی: کارت‌های خلاصه، جدول میزبان‌ها به ترتیب ریسک و اولویت‌های کل شبکه

بخش‌های صفحه‌ی اصلی

- **کارت‌ها:** تعداد میزبان‌ها، یافته‌های KEV، میزبان‌های دارای حداقل یک KEV، تعداد یافته‌های بحرانی و بالا.
- **جدول میزبان‌ها:** مرتب‌شده بر اساس KEV و سپس شدت. امتیاز **ریسک** (۰ تا ۱۰۰) ترکیبی وزن‌دار است: هر KEV ۲۵، بحرانی ۱۰، بالا ۴، متوسط ۱ و پایین ۰٫۲ واحد، که به‌صورت اشباع‌شونده به ۱۰۰ نزدیک می‌شود. نوار رنگی، ترکیب شدت‌ها را نشان می‌دهد و ستون «وضعیت» مرحله‌ی اسکن (در صف، EPSS، KEV، NVD، OSV، کامل یا خطا) را.
- **اولویت‌های کل شبکه:** CVEهایی که بیشترین اهمیت را دارند، به‌همراه تعداد میزبان‌های درگیر؛ مناسب برای تصمیم‌گیری درباره‌ی وصله‌های گسترده.



شکل ۲ — جزئیات یک میزبان Windows 11: یادداشت‌های اسکن، فیلترها و جدول یافته‌ها

صفحه‌ی جزئیات میزبان

- **یادداشت‌های اسکن:** نشان می‌دهد با کدام CPE و نسخه تطبیق انجام شد، چند نرم‌افزار نگاشت نداشتند، نسخه‌ی کاتالوگ KEV و تعداد امتیازهای EPSS دریافتی. اگر منبعی در دسترس نبود، خطای آن همین‌جا نمایش داده می‌شود.
- **فیلترها:** جستجو در CVE، نام بسته و شرح؛ فیلتر شدت؛ «فقط KEV»؛ «EPSS ≥ ۱۰٪».
- **ستون‌ها:** CVE (پیوند به NVD یا OSV) و تاریخ انتشار، شدت و امتیاز CVSS (با نگه‌داشتن ماوس بردار کامل دیده می‌شود)، EPSS (با صدک)، نشان KEV با سررسید و برچسب «باج‌افزار»، بسته، نسخه‌ی نصب‌شده، **نسخه‌ی اصلاح** و شرح.
- **اسکن مجدد:** بلافاصله پس از نصب وصله، Agent را اجرا کنید یا این دکمه را بزنید.
- **خروجی CSV:** فایل UTF-8 با BOM که مستقیماً در Excel درست باز می‌شود؛ مناسب برای ارجاع به تیم‌ها یا ثبت در سیستم تیکتینگ.

۹ اولویت‌بندی و برنامه‌ی رفع آسیب‌پذیری

VulnScope یافته‌ها را با این ترتیب مرتب می‌کند:

```
ORDER BY KEV desc → CVSS desc → EPSS desc
```

پیشنهاد ما برای سیاست رفع (SLA) سازمانی بر پایه‌ی همین سیگنال‌ها:

سطح	شرط	مهلت پیشنهادی	اقدام
P0	در CISA KEV و با برچسب باج‌افزار، یا KEV روی سرور در معرض اینترنت	۴۸ ساعت	وصله‌ی اضطراری؛ اگر وصله ممکن نیست، Mitigation فوری (بستن سرویس، قانون فایروال، غیرفعال‌سازی قابلیت)
P1	سایر موارد KEV	۷ روز	وصله در اولین پنجره‌ی نگه‌داری؛ پیگیری سررسید KEV
P2	CVSS ≥ ۹ **یا** (CVSS ≥ ۷ **و** EPSS ≥ ۱۰٪)	۱۴ روز	وصله‌ی برنامه‌ریزی‌شده با اولویت
P3	CVSS بین ۷ و ۸٫۹ با EPSS پایین	۳۰ روز	چرخه‌ی وصله‌ی ماهانه (Patch Tuesday)
P4	متوسط و پایین	۹۰ روز	به‌روزرسانی عادی

مثال واقعی از داده‌های این راهنما

چرا CVSS به‌تنهایی گمراه‌کننده است

CVE-2024-30088 (آسیب‌پذیری Race Condition در کرنل ویندوز) امتیاز CVSS نسبتاً پایین **۷٫۰** دارد، اما در کاتالوگ CISA KEV ثبت شده، **استفاده‌ی آن در باج‌افزار تأیید شده** و EPSS آن حدود **۶۸٪** است. در مقابل CVE-2024-38063 با CVSS **۹٫۸** (بحرانی) هنوز در KEV نیست. در سیاست بالا اولی P0 و دومی P2 است، در حالی که مرتب‌سازی صرف بر اساس CVSS ترتیب را برعکس می‌کرد.

گردش کار ماهانه‌ی پیشنهادی

1. **هر روز:** بررسی کارت «یافته‌ی KEV» و فهرست اولویت‌های کل شبکه. CISA تقریباً هر هفته موارد جدید اضافه می‌کند و VulnScope هر ۱۲ ساعت کاتالوگ را تازه می‌کند.
2. **سه‌شنبه‌ی دوم هر ماه (Patch Tuesday):** پس از نصب Cumulative Update ویندوز، اسکن مجدد بزنید و کاهش یافته‌های Build را تأیید کنید.
3. **هفتگی:** خروجی CSV میزبان‌های پرریسک را برای مالکان سیستم‌ها بفرستید.
4. **ماهانه:** روند امتیاز ریسک میزبان‌ها و تعداد موارد KEV باز را به مدیریت گزارش دهید.

برای محیط VMware

VMware Workstation، VMware Tools و Remote Console در cpe_map.json تعریف شده‌اند. برای ESXi و vCenter که Agent روی آن‌ها نصب نمی‌شود، بولتن‌های VMSA را جداگانه دنبال کنید یا vCenter را از طریق API به‌صورت دستی در گزارش‌ها وارد کنید (فصل ۱۱).

۱۰ سفارشی‌سازی و افزودن نرم‌افزار

نرم‌افزارهای ویندوز و مک نام استانداردی ندارند؛ بنابراین VulnScope با فایل `server/cpe_map.json` نام نمایشی هر نرم‌افزار را به شناسه‌ی vendor:product در NVD نگاشت می‌کند. در یادداشت‌های اسکن هر میزبان تعداد نرم‌افزارهای بدون نگاشت گزارش می‌شود.

افزودن یک نرم‌افزار ویندوز یا مک

1. در `nvd.nist.gov/products/cpe/search` نام محصول را جستجو کنید و vendor و product را از CPE بردارید. مثال: `cpe:2.3:a:veeam:backup_and_replication:…`
2. یک سطر به آرایه‌ی `apps` اضافه کنید. ستون اول یک عبارت منظم روی DisplayName است:

```
"apps": [
  ["^Google Chrome$",            "google",    "chrome"],
  ["^Veeam Backup & Replication", "veeam",     "veeam_backup_\\&_replication"],
  ["^MyCompany ERP Client",       "myvendor",  "erp_client"]
]
```

افزودن بسته‌ی Homebrew

```
"homebrew": { "openssl@3": ["openssl", "openssl"], "nginx": ["f5", "nginx"] }
```

پسوند نسخه (مثل `python@3.12`) خودکار حذف می‌شود.

پس از ویرایش، سرور را دوباره راه‌اندازی کنید (`systemctl restart vulnscope`) و میزبان‌ها را دوباره اسکن کنید.

افزودن نسخه‌ی جدید ویندوز

وقتی نسخه‌ی تازه‌ای از ویندوز منتشر می‌شود، نگاشت Build به CPE را در تابع `windows_cpe` در فایل matcher.py اضافه کنید؛ مثلاً:

```
26200: "windows_11_25h2",        # کلاینت
26100: "windows_server_2025",    # سرور
```

۱۱ مرجع API

همه‌ی پاسخ‌ها JSON با کدگذاری UTF-8 هستند. ارسال گزارش با توکن Agent (`Authorization: Bearer`) و سایر مسیرها با رمز داشبورد (Basic Auth، اگر فعال باشد) محافظت می‌شوند.

متد	مسیر	کاربرد
POST	`/api/report`	دریافت گزارش Agent (حداکثر ۵۰ مگابایت) و قرار دادن در صف
GET	`/api/hosts`	فهرست میزبان‌ها با شمارش شدت‌ها، KEV و امتیاز ریسک
GET	`/api/hosts/{id}`	جزئیات میزبان و همه‌ی یافته‌ها
GET	`/api/hosts/{id}/export.csv`	خروجی CSV یافته‌ها
POST	`/api/hosts/{id}/scan`	اسکن مجدد با داده‌ی تازه
DELETE	`/api/hosts/{id}`	حذف میزبان (مثلاً سرور از رده خارج شده)
GET	`/api/top`	۱۵ CVE مهم کل شبکه
GET	`/api/health`	وضعیت سرویس و طول صف (بدون احراز هویت، برای مانیتورینگ)

نمونه‌ها

```
# میزبان‌هایی که KEV دارند
curl -s -u admin:PASS https://vulnscope.corp:8443/api/hosts \
  | jq '.hosts[] | select(.kev>0) | {hostname, kev, risk}'

# PowerShell: همه‌ی یافته‌های KEV یک میزبان
$c = Get-Credential
(Invoke-RestMethod https://vulnscope.corp:8443/api/hosts/demo-dc01 -Credential $c).findings |
   Where-Object kev -eq 1 | Select cve, cvss, epss, kev_due, fixed_version
```

قالب گزارش Agent

برای ورود دستی سیستم‌هایی که Agent ندارند (مثل appliance یا vCenter) می‌توانید JSON زیر را بسازید و POST کنید:

```
{
  "schema": 1, "agent_version": "manual",
  "host": { "machine_id": "vcsa01", "hostname": "vcsa01", "ips": ["10.10.1.5"],
            "os": { "family": "linux", "name": "Photon OS", "distro_id": "photon",
                    "version": "4.0", "arch": "x86_64" } },
  "packages": [ { "name": "Google Chrome", "version": "126.0.6478.127", "manager": "windows" } ],
  "hotfixes": []
}
```

۱۲ امنیت، نگهداری، محدودیت‌ها و عیب‌یابی

امنیت

- همیشه با **TLS** اجرا کنید؛ داده‌ی موجودی نرم‌افزار برای مهاجم ارزشمند است (نقشه‌ی دقیق نقاط ضعف شبکه).
- توکن را تصادفی و حداقل ۳۲ کاراکتر انتخاب کنید و به‌طور دوره‌ای عوض کنید. مقایسه‌ی توکن در سرور زمان‌ثابت است.
- رمز داشبورد را فعال کنید و دسترسی به پورت داشبورد را با فایروال به شبکه‌ی مدیریت محدود کنید.
- سرویس با کاربر غیرممتاز و محدودیت‌های systemd (`ProtectSystem=strict`) اجرا می‌شود.
- از فایل `vulnscope.db` نسخه‌ی پشتیبان بگیرید و دسترسی به آن را محدود کنید.

نگهداری

- کش: NVD و OSV ۲۴ ساعت، KEV ۱۲ ساعت، EPSS ۲۴ ساعت. اسکن اول هر محصول کند است (به‌ویژه بدون کلید NVD)؛ اسکن‌های بعدی چند ثانیه طول می‌کشند.
- حجم پایگاه داده برای چند صد میزبان معمولاً کمتر از چند صد مگابایت است.
- برای به‌روزرسانی برنامه، فایل‌ها را جایگزین و سرویس را راه‌اندازی مجدد کنید؛ ساختار پایگاه داده حفظ می‌شود.

محدودیت‌های شناخته‌شده

- تطبیق بر اساس نسخه است؛ اگر آسیب‌پذیری با تغییر تنظیمات (Mitigation) خنثی شده باشد، همچنان نمایش داده می‌شود.
- نرم‌افزارهای بدون نگاشت در cpe_map.json بررسی نمی‌شوند. کتابخانه‌های داخل برنامه‌ها (مثل Log4j درون یک JAR) و Containerها پوشش داده نمی‌شوند.
- RHEL و SUSE در OSV پشتیبانی مستقیم ندارند؛ برای آن‌ها نرم‌افزارهای نگاشت‌شده بررسی و در یادداشت‌ها هشدار داده می‌شود.
- کیفیت پیکربندی CPE در NVD برای همه‌ی CVEها یکسان نیست؛ CVEهای بسیار تازه ممکن است هنوز تحلیل نشده باشند (وضعیت Awaiting Analysis).

عیب‌یابی

نشانه	علت و راه‌حل
وضعیت «خطا» با پیام «درخواست ناموفق»	سرور به منابع فصل ۳ دسترسی ندارد. DNS، Proxy و فایروال خروجی را بررسی کنید.
اسکن طولانی در مرحله‌ی NVD	محدودیت نرخ NVD بدون کلید؛ کلید رایگان را با `nvd-key --` اضافه کنید.
Agent ویندوز: unauthorized	توکن اشتباه یا فاصله‌ی اضافه در فایل token.txt.
Agent: خطای گواهی SSL	cert.pem را با `cafile --` بدهید یا در Trusted Root ویندوز وارد کنید.
ویندوز «نگاشت CPE تعریف نشده»	Build جدید یا قدیمی است؛ فصل ۱۰ را ببینید.
یافته‌ای پس از وصله باقی مانده	Agent را دوباره اجرا کنید تا UBR یا نسخه‌ی جدید گزارش شود، سپس «اسکن مجدد».

پ پیوست: نمونه‌ی نتایج واقعی

جدول زیر خروجی موتور تطبیق VulnScope روی پنج میزبان نمایشی است. میزبان‌ها و نسخه‌های نصب‌شده فرضی‌اند، اما **رکوردهای CVE و پیکربندی نسخه‌ها از NVD، وضعیت KEV از کاتالوگ CISA (نسخه 2026.10.30) و امتیاز EPSS از FIRST (دریافت‌شده در ۱۱ آبان ۱۴۰۵) واقعی هستند** و محاسبه‌ی «آسیب‌پذیر بودن» و «نسخه‌ی اصلاح» توسط خود موتور انجام شده است.

میزبان	CVE	محصول / نسخه	اصلاح در	CVSS	EPSS	KEV
DC01 Server 2022 · 20348.2340	CVE-2024-30088	Windows Kernel	20348.2522	7.0	68.2%	KEV باج‌افزار
	CVE-2024-38112	MSHTML	20348.2582	7.5	84.2%	KEV
	CVE-2024-38077	Remote Desktop Licensing	20348.2582	9.8	84.2%	—
	CVE-2024-38063	TCP/IP (IPv6)	20348.2655	9.8	70.6%	—
FIN-PC07 Windows 11 23H2 · 22631.3880	CVE-2023-4863	Chrome 116.0.5845.96	116.0.5845.187	8.8	100.0%	KEV
	CVE-2023-38831	WinRAR 6.22	6.23	7.8	98.0%	KEV باج‌افزار
	CVE-2024-49138	CLFS Driver	22631.4602	7.8	26.2%	KEV
	CVE-2024-38063	TCP/IP (IPv6)	22631.4037	9.8	70.6%	—
web01 Ubuntu 22.04	CVE-2023-4911	glibc 2.35	2.39	7.8	63.8%	KEV
	CVE-2024-1086	Linux kernel 6.5.0	6.6.15	7.8	28.1%	KEV باج‌افزار
	CVE-2024-6387	OpenSSH 9.3p1	بالاتر از 9.8	8.1	99.5%	—
Design-MBP macOS 14.2.1	CVE-2025-24085	macOS (CoreMedia)	14.7.5	10.0	17.5%	KEV
	CVE-2024-23222	macOS (WebKit)	14.3	8.8	10.6%	KEV
	CVE-2024-3094	xz 5.6.1 (Homebrew)	—	10.0	90.9%	—

نکته‌ها: (۱) در این نمونه، بسته‌های لینوکس برای کار بدون اینترنت مستقیماً با NVD تطبیق داده شدند؛ در استقرار واقعی از بولتن‌های Ubuntu در OSV استفاده می‌شود که Backportها را لحاظ می‌کند و ممکن است نسخه‌ی اصلاح متفاوتی (مثل `2.35-0ubuntu3.4`) نشان دهد. (۲) برای CVE-2024-3094 در NVD فقط نسخه‌های دقیق 5.6.0 و 5.6.1 آسیب‌پذیر اعلام شده‌اند، بنابراین «نسخه‌ی اصلاح» خالی است. (۳) امتیاز EPSS روزانه تغییر می‌کند. (۴) همین سناریوها در تست‌های خودکار پوشه‌ی `tests` بررسی می‌شوند: `python3 -m unittest discover -s tests -v`