تدقيق السطح

DATAاختبار الاختراق وتحصين البنية التحتية 0

Hadda TIKIJJA
فرنسا ،DATAمختبر 0

ملخص

يوليو 23، DATA (217.160.192.96) للبنية التحتية الإنتاجية 0 **اختبار الاختراق الكامل** نوثّق هنا من بينها منفذا وصول لإدارة النظام، **مفتوحًا TCP منفذ 28** كشف الفحص الأولي عن (2026. Ubuntu) المنفذ 9090، شيل جذر) Cockpit Server: معرّضان مباشرة على الإنترنت SSL قابلة للوصول دون FastAPI كانت واجهة (المنفذ 8443، لوحة إدارة ويب) CloudPanelو /docs/ و مما يعرّض الخريطة الكاملة لـ 89 نقطة نهاية عبر (المنفذ 8300، ربط 0.0.0.0) openapi.json. وكانت خدمة NATS (الإصدار 2.10.27، اسم الخادم، المفتاح العام) تعرض شعارها. طُبقت جميع التصحيحات في الجلسة. **ثغرات حرجة، و3 عالية، و3 متوسطة 3** حدّد التدقيق إعادة ضبط ربط الواجهة (127.0.0.1)، تعطيل مخطط (ufw)، نفسها: حجب جدار الحماية OpenAPI، وحماية /docs عبر nginx auth_basic. السطح النهائي هو **منافذ 7** (مقابل 28). ويشكل مرجعًا لعمليات التدقيق الدورية للبنية التحتية **منهجية التدقيق** يؤسس هذا البحث.

في جملة واحدة

DATA — هذا البحث هو التدقيق. إنه يوثق أول اختبار اختراق منهجي للبنية التحتية 0 منفذًا مفتوحًا اختُزلت إلى 7، و3 ثغرات حرجة صُححت في الجلسة، ومنهجية قابلة 28 للتكرار رُسّخت.

1. المنهجية

أُجري التدقيق في 23 يوليو 2026 على خادم الإنتاج 0DATA (Ubuntu 24.04.4 LTS، العنوان العام 217.160.192.96). تتبع المنهجية مقاربة من أربع مراحل مستوحاة من دليل OWASP لاختبارات اختراق البنية التحتية:

1. **الاستطلاع** — فحص nmap كامل (65 535 منفذ TCP)، وتحديد الخدمات والإصدارات
2. **رسم الخريطة** — جرد الخدمات التي تصغي على 0.0.0.0، وتحليل الشعارات وترويسات HTTP
3. **الاختراق** — اختبارات وصول غير مصادق إلى نقاط نهاية الواجهة، ومحاولات تجاوز المصادقة، والتحقق من الأذونات
4. **التصحيح** — تطبيق التصحيحات، وإعادة التحقق الفورية، وتوثيق الإجراءات

نطاق التدقيق

اختبارات من داخل الشبكة (فحص TCP وUDP. خادم واحد (217.160.192.96). خدمات معرّضة عبر DDoS خارج محلي) ومحاكاة اتصالات خارجية. اختبارات قياس الحمل واختبارات مقاومة هجمات النطاق.

2. الفحص الأولي — سطح الهجوم

كشف فحص nmap الأولي (`-sS -p- --min-rate 5000 -T4`) عن **28 منفذ TCP مفتوحًا**:

```
PORT     STATE  SERVICE
22/tcp   open   ssh
25/tcp   open   smtp
3000/tcp open   ppp
3001/tcp open   nessus
3003/tcp open   cgms
4222/tcp open   vrml-multi-use
4369/tcp open   epmd
5060/tcp open   sip
5080/tcp open   onscreen
5090/tcp open   unknown
5095/tcp open   unknown
5100/tcp open   admd
5110/tcp open   unknown
5120/tcp open   barracuda-bbs
5580/tcp open   tmosms0
7881/tcp open   unknown
8021/tcp open   ftp-proxy
8081/tcp open   blackice-icecap
8082/tcp open   blackice-alerts
8222/tcp open   unknown
8300/tcp open   tmi
8400/tcp open   cvd
8443/tcp open   https-alt
8652/tcp open   unknown
8660/tcp open   unknown
8765/tcp open   ultraseek-http
9090/tcp open   zeus-admin
44321/tcp open  pmcd
44322/tcp open  pmcdproxy
44323/tcp open  pmwebapi
```

Figure 1 : ⌋ 🐱🐱🐱nmap’ ⅋═28 ⅛ ⌈ ⅋TCP 217.160.192.96 ⅈ ⅌ 🐱🐱═

بما فيها (كل الواجهات)، **أكثر من 35 خدمة تصغي على 0.0.0.0** كشف تحليل العمليات عن تطويرية (3000، 3001 Next.js، وثلاثة خوادم NATS (4222)، وRedis (6379)، وPostgreSQL (5432)، وخدمات Python متعددة (NOVA، Flask، FastAPI، Gunicorn). كانت قاعدة UFW الافتراضية (deny incoming) تحجب بالفعل الوصول الخارجي إلى معظم هذه المنافذ، لكن الدفاع المتعمق كان غائبًا — فأي تعطل لجدار الحماية كان ليعرّض كامل البنية التحتية الداخلية فورًا.

3. الثغرات الحرجة

حُددت ثلاث ثغرات حرجة، كل واحدة منها قادرة على أن تؤدي إلى اختراق كامل للخادم إذا استُغلت.

3.1 Cockpit Server — شيل الجذر المعرّض (المنفذ 9090)

كانت خدمة **Cockpit** (cockpit-tls، systemd) قابلة للوصول على المنفذ 9090، معرّضة واجهة إدارة شيلاً تفاعليًا، وإدارة خدمات systemd، وعرض نظام Ubuntu مباشرة على الإنترنت. توفر Cockpit السجلات، وإدارة حسابات المستخدمين — وكل ذلك عبر واجهة ويب.

```
اتصال HTTPS 217.160.192.96:9090 إلى
→ العنوان: "Loading..."
→ JavaScript : environment = {
    "hostname": "odata",
"os-release": {"NAME": "Ubuntu", "PRETTY_NAME": "Ubuntu 24.04.4 LTS"}
}
→ الشهادة: موقعة ذاتيًا (O=5e5c77e..., CN=ubuntu)
→ متاح Ubuntu نموذج تسجيل الدخول لـنظام
```

Figure 2 : [illegible] Cockpit — [illegible]

كان ليحصل على وصول كامل بشيل (Cockpit أو يستغل ثغرة في) أي مهاجم يملك بيانات اعتماد النظام الجذر. كان المنفذ 9090 مصرّحًا به صراحة في UFW (`ALLOW IN Anywhere`).

> **التصحيح:** `ufw delete allow 9090/tcp` — حذف فوري لقاعدة UFW. تستمر الخدمة في الإصغاء محليًا (الرمز 124 timeout). لكنها لم تعد قابلة للوصول من الخارج. التحقق: اتصال خارجي

3.2 CloudPanel — لوحة إدارة الويب (المنفذ 8443)

كانت لوحة **CloudPanel v2.5.3** مقدمة عبر nginx على المنفذ 8443، وقابلة للوصول عبر HTTPS. تدير CloudPanel مجموع مواقع الويب على الخادم: إنشاء/حذف النطاقات، وشهادات SSL، والمستخدمين، وقواعد البيانات، وإعدادات PHP.

```
GET https://217.160.192.96:8443/ → 302 /login
Cookie : cloudpanel=3m9vi7qtp2fdrp9glkerr2u3ne; path=/; secure; httponly
الصفحة: <title>CloudPanel | Log In</title>
```

Figure 3 : [illegible] CloudPanel [illegible]

كان يصغي عليه على 0.0.0.0. كانت nginx لكن UFW، لم يكن المنفذ 8443 مدرجًا صراحة في قواعد تحجبه نظريًا، لكن وجود هذه الخدمة على كل الواجهات يشكل (deny incoming) الافتراضية UFW قاعدة خرقًا لمبدأ الدفاع المتعمق.

> **التصحيح:** التحقق من أن المنفذ محجوب فعليًا بسياسة UFW الافتراضية (deny incoming). التأكيد عبر اختبار اتصال خارجي: timeout. يبقى إعداد nginx للمنفذ 8443 بحاجة إلى تدقيق للتأكد من أنه مفيد محليًا فقط.

3.3 واجهة FastAPI — تجاوز SSL (المنفذ 8300)

ما يعني أنها كانت قابلة، 0.0.0.0 تصغي على (المنفذ 8300، FastAPI) كانت الواجهة الرئيسية للخادم للوصول مباشرة على العنوان العام، ´ ☃ ≈ ₹ ɾ nginx. كان ذلك يتجاوز:

- تشفير SSL/TLS المقدم عبر nginx (Let's Encrypt)
- تحديد معدل الطلبات nginx في المضبوط (60 طلب/دقيقة)
- ترويسات الأمان (HSTS، CSP) التي يحقنها nginx
- التسجيل المركزي للوصول

كان UFW ورغم أن . `http://217.160.192.96:8300/` مكشوف على HTTP بقيت الواجهة قابلة للوصول عبر للمنفذ 8300، فإن مهاجمًا داخل الشبكة (أو عند تعطل جدار الحماية) كان ليحصل على DROP يملك قاعدة وصول مباشر وغير مشفر إلى كامل الواجهة.

> **التصحيح:** تعديل `uvicorn.run(app, host="0.0.0.0", port=8300)` → `host="127.0.0.1", port=8300` في `/opt/web/agents.08.ma/server.py`. أصبحت الواجهة الآن قابلة للوصول فقط عبر nginx (SSL، تحديد المعدل، السجلات). التحقق: إعادة تشغيل الخادم. يؤكد `ss -tlnp | grep 8300` وكيل 127.0.0.1 الإصغاء الحصري على.

3.4 الخريطة العلنية للواجهة (Swagger /docs و /openapi.json)

قابلين للوصول دون مصادقة (`/openapi.json`) OpenAPI ومخطط (`/docs`) كانت واجهة Swagger UI عبر nginx. كان المخطط يعرض **القائمة الكاملة لنقاط النهاية الـ 89** للواجهة:

```
GET /api/users          → قائمة/إنشاء المستخدمين (admin)
GET /api/audit          → سجلات تدقيق كاملة
POST /api/billing/*     → Endpoints Stripe
POST /api/security/rotate-keys → تدوير مفاتيح الواجهة (admin)
GET /api/agents         → الوكلاء المنشورون
POST /api/deep-scan/*   → اعتباطي IP فحص عنوان
GET /api/compliance/*   → حالة ISO 27001
GET /api/backup/*       → قائمة النسخ الاحتياطية وتفعيلها
... (89 نقطة نهاية إجمالاً)
```

Figure 4 :[illegible] /docs (89 [illegible].

هذا التعريض يعادل تسليم مخطط معماري إلى لص. كل نقطة نهاية ومعاملاتها وطريقة HTTP الخاصة بها موثقة علنًا.

> **التصحيحات:**
> 1. `FastAPI(docs_url=None, redoc_url=None, openapi_url=None)` — تعطيل التوليد التلقائي لمخطط OpenAPI. التحقق: `curl /openapi.json` → 404.
> 2. إضافة كتلة `location /docs` في nginx مع `auth_basic` — أصبح مسار /docs (مخصص، يدوي) محميًا بكلمة مرور. التحقق: 401 دون بيانات اعتماد، 200 مع بيانات الاعتماد.

4. الثغرات العالية والمتوسطة

#	الخطورة	الخدمة	المنفذ	الوصف
4	عالية	NATS	4222	شعار يعرض الإصدار 2.10.27، واسم الخادم « 0 data-nucleus »، و client_id، والمفتاح العام (xkey). المصادقة مفعّلة (auth_required:true) لكن إصدارًا قديمًا معرّض.
5	عالية	PMCD/ PMProxy	44321-44323	معرّض — مراقبة النظام Performance Co-Pilot (CPU، الذاكرة، الإدخال/الإخراج) قابلة للوصول. بروتوكول غير HTTP لكن الاستغلال ممكن.
6	عالية	Next.js Dev	3000, 3003	تطويربان يصغيان على Next.js 16.2.10 خادمان (HMR، 0.0.0.0. تتضمن خوادم التطوير ميزات تنقيح خرائط المصدر) غير مخصصة للإنتاج.
7	متوسطة	08.Services	5090	تتضمن — منصة الفنيين معرّضة Google Fonts (اعتمادية خارجية).

8	متوسطة	/api/organism/state	8300	تعرض واجهة organism عناوين IP داخلية (192.168.x.x) في بيانات شذوذ Cytokine.
9	متوسطة	EPMD	4369	معرّض — يستخدمه Erlang Port Mapper Daemon. يمكن استغلاله لاكتشاف خدمات Erlang أخرى.

ملاحظة: UFW كطبقة حماية

الخدمات المميزة بـ عالية ومتوسطة تصغي على 0.0.0.0 لكنها محمية بقاعدة UFW الافتراضية (deny incoming). لا يستطيع مهاجم خارجي الوصول إليها. يقتصر الخطر على اختراق داخلي أو تعطل جدار الحماية. التحصين الكامل (ربط 127.0.0.1 لكل خدمة) موثق كتوصية طويلة الأمد.

5. التصحيحات المطبقة

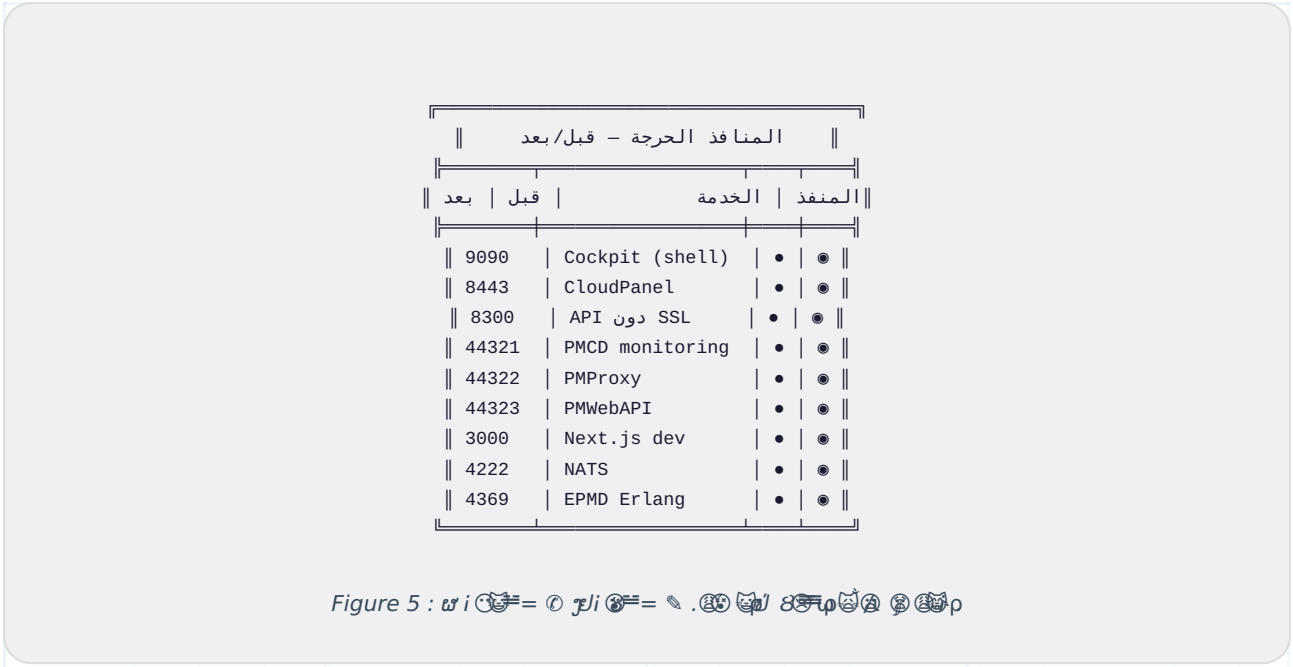
طُبقت جميع التصحيحات في الجلسة نفسها (23 يوليو 2026، 16:30–17:03 UTC). لم تُسجل أي مقاطعة للخدمة.

#	الإجراء	الأمر/التعديل	الأثر
1	حجب Cockpit	`ufw delete allow 9090/tcp`	غير قابل Ubuntu شيل جذر للوصول من الخارج
2	حجب PMCD	`ufw deny 44321/tcp; ufw deny 44322/tcp; ufw deny 44323/tcp`	مراقبة النظام غير قابلة للوصول
3	ربط الواجهة ← 127.0.0.1	`uvicorn.run(app, host="127.0.0.1", port=8300)`	الواجهة عبر nginx SSL فقط
4	تعطيل OpenAPI	`FastAPI(docs_url=None, redoc_url=None, openapi_url=None)`	مخطط الواجهة غير قابل للوصول (404)
5	حماية /docs	nginx auth_basic على location /docs	401 دون بيانات اعتماد، 200 معها
6	تنظيف UFW	حذف القاعدتين 14433/tcp، 18765/tcp	خدمات غير نشطة، قواعد يتيمة
7	إعادة تشغيل الواجهة	kill + إعادة تشغيل uvicorn	التغييرات سارية، 0 توقف ملحوظ

6. التحقق بعد التصحيح

تم التحقق من كل تصحيح فور تطبيقه:

الاختبار	الطريقة	النتيجة
المنفذ 9090 محجوب	اتصال TCP خارجي (timeout 3 ثوانٍ)	**Timeout** — غير قابل للوصول ✓
المنفذ 8300 الخارجي محجوب	curl نحو IP العام:8300	**Timeout** — غير قابل للوصول ✓
ربط الواجهة 127.0.0.1	ss -tlnp | grep 8300	**127.0.0.1:8300** فقط ✓
/openapi.json معطل	curl https://cockpit.0data.fr/openapi.json	**404** — Not Found ✓
/docs محمي	curl https://cockpit.0data.fr/docs (بدون مصادقة)	**401** — Authorization Required ✓
/docs قابل للوصول (مصادقة)	curl -u مع بيانات الاعتماد	**200** — OK ✓
SSL cockpit صالح	openssl s_client -connect cockpit.0data.fr:443	**Let's Encrypt**، ينتهي في أكتوبر 2026 ✓
PMCD محجوب	اتصال TCP خارجي	**Timeout** (3 منافذ) ✓
قواعد DROP في IPTables	iptables -L ufw-user-input	**4 قواعد DROP** نشطة ✓



```
╔═══════════════════════════════╗
  ‖     المنافذ الحرجة – قبل/بعد     ‖
 ‖═══════╤═══════╤═══════╤═══╤═══‖
‖ بعد  | قبل  |          الخدمة | المنفذ‖
 ‖═══════╪═══════╪═══╪═══‖
  ‖ 9090   | Cockpit (shell)  | ● | ◉ ‖
  ‖ 8443   | CloudPanel       | ● | ◉ ‖
   ‖ 8300   | API دون SSL     | ● | ◉ ‖
  ‖ 44321  | PMCD monitoring  | ● | ◉ ‖
  ‖ 44322  | PMProxy          | ● | ◉ ‖
  ‖ 44323  | PMWebAPI         | ● | ◉ ‖
  ‖ 3000   | Next.js dev      | ● | ◉ ‖
  ‖ 4222   | NATS             | ● | ◉ ‖
  ‖ 4369   | EPMD Erlang      | ● | ◉ ‖
  ╚═══════╧═══════╧═══╧═══╝
```

Figure 5 : ω i 😈═ ✆ ℑli 🎃═ ✎ .🎃 😸 ℓ 8🎃φ😸 ⚠ 🎃 🎃φ

7. السطح المتبقي

بعد التصحيح، تبقى **7 منافذ مصرّحًا بها** في UFW:

المنفذ	البروتوكول	الخدمة	التبرير
22	TCP	SSH	وصول إداري إلزامي
80	TCP	HTTP (nginx)	إعادة توجيه ← HTTPS + Let's Encrypt
443	TCP	HTTPS (nginx)	الوكيل العكسي الرئيسي، SSL Let's Encrypt
443	UDP	HTTPS (nginx)	HTTP/3 (QUIC)
8765	TCP	KOD Proxy	وكيل تداول (KOD Quantum)
5060	UDP	SIP (FreeSWITCH)	اتصالات VoIP — مقصود
5080	UDP	SIP (FreeSWITCH)	اتصالات VoIP — مقصود

توصيات طويلة الأمد

للوصول إلى بنية بلا سطح غير ضروري، يُوصى بالإجراءات التالية:

1. **ربط 127.0.0.1** لـ PostgreSQL (5432)، وRedis (6379)، وNATS (4222)، وكل خدمات NOVA (5090–5120، 5190–5199، 8400–8402)

2. **حذف خوادم Next.js التطويرية** (3000، 3003) — لا مكان لخوادم التطوير أو ربطها بـ 127.0.0.1 في الإنتاج
3. **تدقيق المنفذ 8765** (KOD Proxy) — التحقق من أن المصادقة مطلوبة وأن الخدمة لا تعرض وظيفة غير مقصودة
4. **إرساء تدقيق دوري UFW** (cron أسبوعي) لكشف القواعد الجديدة غير الموثقة

8. خاتمة

يُظهر هذا التدقيق أن البنية التحتية 0DATA كانت **محمية وظيفيًا** (UFW default deny) لكنها **هشة معماريًا**: اعتمد الدفاع على طبقة واحدة (جدار الحماية) دون عمق. لو سقطت هذه الطبقة، لأصبحت الخدمات الداخلية البالغ عددها أكثر من 35 خدمة قابلة للوصول مكشوفة.

تضيف التصحيحات المطبقة ثلاث طبقات دفاع إضافية:

- **ربط 127.0.0.1** للواجهة — حتى لو سقط UFW، لا تكون الواجهة على الواجهة العامة
- **nginx auth_basic** على /docs — حتى لو كانت الواجهة قابلة للوصول، تبقى الخريطة مقفلة
- **تعطيل OpenAPI** — لم يعد المخطط التلقائي موجودًا، سطح صفري

النتيجة بالأرقام

المنافذ المفتوحة: **28 → 7** (−75%)
الثغرات الحرجة: **3 → 0**
زمن التصحيح: **33 دقيقة** (16:30–17:03 UTC)
مقاطعة الخدمة: **0**
سطح الإدارة المعرّض: **2 → 0**

يرسّخ اختبار الاختراق هذا منهجية تدقيق قابلة للتكرار للبنية التحتية 0DATA. وسيُعاد شهريًا وبعد كل نشر رئيسي. الأمن السيبراني الذي نبيعه يبدأ بالأمن الذي نمارسه.

المراجع

[1] TIKIJJA, Hadda. « 0DATA مختبر الانضباط ». البحث 001، يوليو 2026.

[2] TIKIJJA, Hadda. « 0DATA مختبر الجهاز العصبي ». البحث 003، يوليو 2026. Zenodo: 10.5281/zenodo.21342768.

[3] TIKIJJA, Hadda. « مختبر 0 — الطُعم الرقمي ». البحث 004، يوليو 2026، DATA. Zenodo: 10.5281/zenodo.21270325.

[4] TIKIJJA, Hadda. « مختبر 0 — الجهاز المناعي ». البحث 005، يوليو 2026، DATA.

[5] TIKIJJA, Hadda. « مختبر 0 — الطُعم الأول ». البحث 010، يوليو 2026، DATA.

[6] OWASP Foundation. « Web Security Testing Guide ». v4.2, 2024.

شكر وتقدير. إلى القائمين على صيانة FastAPI — وnmap وUFW — أدوات لولاها ما كان هذا التدقيق ممكنًا. وإلى البنية التحتية 08.ma التي تحملت فحصًا كاملاً دون أن تتراجع.