SIBUNA

محرك لحماية الويب مشروط بالعمل وبلا تخصيص للذاكرة

قبول مشروط بالعمل، وفحص في زمن خطي، وحالة محدودة، وتخزين عبر SQL

Vikrant Rathore

بمساعدة Ronak Rathore وKanak Rathore

Insan AI | الإصدار beta.3-1.0.0

الملخص

تعاني الجدران النارية لتطبيقات الويب ودفاعات الحافة في الغالب من ثلاث مشكلات في بنيتها. الأولى انقلاب الكلفة: لا يكاد المرسل ينفق شيئًا ليبعث طلبًا، بينما يحلله المدافع، ويفحصه بتعابير نمطية قد تتراجع (backtrack)، ويستعلم عنه مخزنًا خارجيًا. والثانية تقلب السلوك وقت التشغيل: فتخصيص الذاكرة في الكومة على مسار الطلب، وتوقفات جمع المهملات، والنشر في عدة حاويات، كلها تجعل ذيل توزيع زمن الاستجابة عصيًا على الحصر. والثالثة الثقل التشغيلي: كثيرًا ما تقيم الحالة المشتركة للحظر والرموز والحدود في خدمات يتعين على المشغل تشغيلها بجانب الجدار الناري. **Sibuna** محاولتنا لمعالجة هذه المشكلات بعدد قليل من الآليات البسيطة في ملف تنفيذي واحد مكتفٍ بذاته، وتشرح هذه الورقة المنطق وراء كل منها. (i) القبول المشروط بالعمل: يكسب المتصفح جلسة بحل لغز Hashcash أو إثبات عمل متسلسل على طريقة Cohen-Pietrzak، يُختار أحدهما لكل قاعدة، ويتحقق الخادم من الإجابة في 70.2 ns أو 22.3 us مقابل تحدٍّ لم يخزنه قط. (ii) مسار طلب محدود بلا تخصيص للذاكرة: تطابق آلة Aho-Corasick ذات فئات البايتات 40 توقيعًا للروبوتات في 55.7 ns، أي أسرع بـ 13.7 مرة من البحث التسلسلي؛ ويقرر محدِّد المعدل GCRA في 6.2 ns بـ 16 بايتًا من الحالة لكل عميل؛ وتسجل مجموعة Robin Hood للحلول المُنفَقة حلًا في 21.0 ns. ويجري فحص حقن SQL والبرمجة النصية عبر المواقع وحقن الأوامر واجتياز المسارات في المرور نفسه، بزمن خطي في طول المدخل. (iii) حالة مشتركة بلا بروتوكول إجماع من صنعنا: يخدم ملف SQLite مدمج عقدةً واحدة، وتخدم قاعدة بيانات PostgreSQL مشتركة واحدة عنقودًا، ويقرؤهما خيط تخزين واحد لا تنتظره الطلبات أبدًا. على معالج AMD Ryzen 9 5950X واحد، والبرنامج الخدمي على أربع وحدات CPU، يجيب Sibuna عن نحو 200,000 طلب forward auth في الثانية (من 197,532 إلى 211,636 عبر أعباء العمل) بقيمة p99 من 305 إلى 327 us، ويُبقي p99.9 عند 62 us تحت حمل مفتوح الحلقة قدره 100,000 طلب في الثانية، ولا تتجاوز ذروة ذاكرته المقيمة نحو 16 MiB. ونبيّن كيف حصلنا على كل رقم، ونسرد ما لم نقسه.

ns 6.2	ns 55.7	µs 22.3	MiB 16
فحص المعدل بـ GCRA	**مطابق الروبوتات**	**التحقق من PoSW**	**ذروة RSS للبرنامج الخدمي**
عميل مُعرَّف بتجزئة | 16 شريحة	40 توقيعًا | Aho-Corasick	العمق 13 | 16 عملية فتح	تحت حمل 4 | wrk وحدات CPU

Sibuna على نواة واحدة من AMD Ryzen 9 5950X، أو على أربع وحدات CPU للبرنامج الخدمي، 15 أكتوبر 2026: benchmarks/results/primitives-latest.md وproduct-latest.md؛ ويشرح قسم التقييم المنهجية.

1 استهلال: اقتصاديات الطلب

كل طلب يصل إلى خادم يطالبه بشيء من انتباهه. ولا تكفي طريقة HTTP ومسار وبضع ترويسات لإثبات أن الطلب جدير بأن يُخدَم. لذلك يضع الدفاع قرارًا بين الطلب والخادم الأصلي: أن يقبله، أو يرفضه، أو يطلب من المرسل أن يؤدي أولًا عملًا يمكن التحقق منه. وتتتبع هذه الورقة ذلك القرار في Sibuna، من رياضياته إلى البايتات في الذاكرة.

> **فاينمان والحدس الفيزيائي: من يدفع ثمن العمل؟**
>
> تخيّل بوابة دوّارة تدور عند أخف لمسة. ويجلس خلفها حارس عليه، مع كل لمسة، أن ينهض ويراجع ثلاثة دفاتر ويتصل بالمكتب الرئيسي. يستطيع طفل في الخارج أن يلمس البوابة طوال العصر دون عناء، بينما يُنهَك الحارس قبل الغداء. هكذا رُتّب كثير من الجدران النارية: الجانب الرخيص هو جانب المهاجم. وتقترح الفيزياء ترتيبًا أفضل. صِل البوابة الدوارة بمضخة، بحيث يتعين على كل من يريد العبور أن يرفع دلوًا من الماء إلى خزان فوق المدخل. يتطلب الرفع قليلًا من الجهد الصادق؛ ولا يفعل الحارس إلا أن يلقي نظرة على الخزان ليرى أن الماء ارتفع. لا تكاد النظرة تكلف الحارس شيئًا، أما الرفع فيكلف الزائر شيئًا حقيقيًا. يرفع المسافر اليومي الدلو مرة واحدة ويعبر طوال اليوم؛ أما العابث فسرعان ما يعود إلى بيته.

الصناديق التي تتكلم في هذه الورقة بصوت أحد العلماء هي من كلامنا نحن، كتبناها بروح ريتشارد فاينمان ودونالد كنوث وليزلي لامبورت، ونحاول أن نتبع طريقتهم في الشرح. وليست اقتباسات، إلا حيث يذكر الصندوق مصدره.

يدفع المرسل الآلي القليل جدًا عن كل طلب يبعثه. وعندما يصل طلب إلى جدار ناري تقليدي، قد يُتم المدافع مصافحة TLS، ويخصص ذاكرة لنسخ الطلب وتقطيعه، ويمرر عليه قائمة من التعابير النمطية، ويجري رحلة ذهاب وإياب عبر الشبكة إلى مخزن مشترك ليحدّث عدادًا. ولا ننسب إلى هذا اللاتماثل نسبة عامة، لأنه لا توجد نسبة كهذه: فهو يتوقف على الخادم الأصلي وذاكرة التخزين المؤقت والهجوم. وبدلًا من ذلك يجعل



الشكل 1 أين يذهب العمل. في الأعلى، يُنفق الدفاع الذي يبدأ بالفحص من عمله هو على كل طلب يبعثه المرسل. وفي الأسفل، يطلب Sibuna من المرسل 2^b محاولة تجزئة متوقعة، ويتحقق من الإجابة بتجزئة واحدة. مؤشرات العمل نموذج؛ أما زمن التحقق فمقيس (التحقق من Hashcash عند 16 بتًا، `benchmarks/results/primitives-latest.md`).

نموذج بسيط القرار واضحًا. لتكن c_o عمل الخادم الأصلي لكل طلب مقبول، وc_g عمل البوابة لكل طلب، وa نسبة الطلبات التي تقبلها البوابة. لا تسترد البوابة كلفتها إلا عندما

$$c_g < (1 - a)\, c_o,$$

أي عندما يفوق العمل الذي توفره على الخادم الأصلي كلفتها هي[1]. ويترتب على ذلك أمران. يجب أن تكون كلفة البوابة نفسها c_g صغيرة ومحدودة مهما كان المدخل، ويجب أن يكون قرار القبول مكلفًا للمرسل لا للخادم.

يقلب Sibuna هذا اللاتماثل بأن يجعل القبول مشروطًا بإثبات عمل. فلكي يكسب المتصفح جلسة، يجد قيمة مؤقتة (nonce) يبدأ ملخص SHA-256 الخاص بها بـ b بتًا صفريًا، أو يحسب إثبات عمل متسلسل. البحث المتوقع 2^b محاولة تجزئة، أي 65,536 عند $b = 16$؛ أما الخادم فيفحص مرشحًا واحدًا. وفحص الخادم عديم الحالة أيضًا: لا يكتب إصدار التحدي شيئًا، ولا يشغل الذاكرة إلا حلٌّ دُفع ثمنه.

لا يدل إثبات العمل على أن المرسل إنسان، ولا على هويته أو نيته. فالمرسل يستطيع أن يستأجر حواسيب، أو يتشارك جلسة طوال مدة صلاحيتها، أو يوزع العمل على عدة أجهزة. يغيّر اللغز ثمن القبول؛ أما ما يحق للطلب المقبول أن يفعله فتبقى السياسة والفحص هما من يقرره.

2 نظرة عامة على التصميم: ثلاثة قرارات في مرور واحد

يُبقي Sibuna القبول والفحص والسمعة ثلاثة قرارات منفصلة، ويتخذها في مرور واحد على الطلب، داخل عملية واحدة، قبل أن يؤدي الخادم الأصلي أي عمل. ويقدمها في وضعين للمنتج. يوفر *Gate* القبول بإثبات العمل مع القواعد وسمعة نطاقات CIDR وضوابط محلية للإغراق. ويضيف *Shield*، وهو الوضع الافتراضي، فحصًا محدودًا لحمولة الطلب. ويعمل كل من الوضعين وكيلًا عكسيًا أمام خادم أصلي، أو نقطة نهاية forward auth يستشيرها وكيل دخول مثل Caddy أو Nginx. والتخزين الدائم للسياسة والسمعة وسجل الحوادث خيار لكل منهما: محليًا على عقدة واحدة، أو مشتركًا في عنقود[2].

2.1 خيارات التصميم

معظم الأفكار أدناه مأخوذة من الأدبيات البحثية أو من أنظمة سابقة، ويتتبع الجزء الثاني من الكتاب أصولها. وما حاولناه هو أن نجمعها في مسار طلب واحد محدود، وأن نُبقي كل خطوة رخيصة، وأن نقيس النتيجة بأمانة. ونسرد الخيارات مع سبب كل منها.

[1] الكتاب، الجزء الأول، "القبول قرار اقتصادي". المتباينة نموذج للسعة، لا قياس؛ وهي تُغفل عرض النطاق وحالة الاتصال وتأخر العميل.

[2] Decision" SID 0002، "الكتاب، الجزء الثاني، "ثلاث أفكار في ملف تنفيذي واحد"؛ الكتاب، الجزء السادس، "محرك السياسة التصريحي". والمثال الخاص بكل قاعدة، وتذكرة المتطلبات، والجلسة ذات مستوى العمل مأخوذة من الجزء الرابع من الكتاب؛ والخط الأساسي التكيفي من الجزء الرابع من الكتاب ومن SID 0006 (ملاحظة المراجعة، Part H).

الجدول 1 وضعا المنتج (SID 0002؛ الجزء الثاني من الكتاب)

الإمكانية	Gate (`--gate`)	Shield (`--shield`، الافتراضي)
القبول	تحدٍّ في المتصفح بـ Hashcash أو بالعمل المتسلسل؛ ورمز جلسة مرتبط بالعميل وبالعمل الذي دفعه	نفسه
السياسة	قواعد تصريحية (المسار، وUser-Agent، والترويسات، ونطاقات CIDR لـ IPv4 وIPv6، وبصمة JA4 لاتصال TLS من وكيل دخول موثوق) مع ALLOW وDENY وCHALLENGE وWEIGH؛ وتوقيعات الروبوتات؛ والسمعة	نفسه
ضوابط الإغراق	حدود المعدل بـ GCRA، والمصيدة، والحظر المؤقت	نفسه
فحص الحمولة	معطَّل	حقن SQL، والبرمجة النصية عبر المواقع، وحقن الأوامر، واجتياز المسارات، في أول 8 KiB من الجسم
الحالة الاختيارية	السياسة والسمعة في SQLite، أو مشتركة عبر PostgreSQL	وأيضًا الأدلة الجنائية للحوادث: البحث في النص الكامل وتجميع الحملات

1. تحديات لا يكلّف إصدارها الخادمَ شيئًا. التحدي سجل يصادق على نفسه، مختوم بتجزئة ذات مفتاح، فلا يكتب إصداره أي حالة؛ ولا يشغل الذاكرة إلا حلٌّ دُفع ثمنه، ويغادرها حين تنتهي صلاحية التحدي (الخوارزميات وضماناتها).
2. مستويان من العمل، يُختار أحدهما لكل قاعدة. إلى جانب Hashcash، يقدم Sibuna إثبات العمل المتسلسل لـ Cohen-Pietrzak، الذي بُرهنت سلامته (soundness) حتى في مواجهة مُبرهِنين كموميين، ويُتحقق من إثباته في بضع ميكروثوانٍ. قد يطلب مسار الدفع 20 بت عمل من العمل المتسلسل، بينما تطلب مدونة 16 بتًا من Hashcash (إثباتات العمل: البحث المتوازي والعمل المتسلسل).
3. جلسات تتذكر العمل المدفوع. يسجل رمز الجلسة الآلية وبتات العمل التي حلها حامله. ولا يقبل إلا المسارات التي لا تطلب أكثر من ذلك، فلا تستطيع جلسة رخيصة أن تفتح مسارًا مكلفًا.
4. رموز بمفاتيح متماثلة. تُختم الجلسات بـ BLAKE3 ذي المفتاح، الذي يُتحقق منه في 155.8 ns، أي بعمل أقل بنحو 260 مرة من توقيع Ed25519، ولا يمنح البحث الكمومي في مواجهته إلا تسريعًا تربيعيًا.
5. فحص لا تستطيع الجلسة تجاوزه. يجري القبول والفحص في المرور نفسه، ولا تلبي الجلسة الصالحة إلا التحدي: فالطلب الذي يحمل جلسة و`q=' OR 1=1--` يُرفض مع ذلك.
6. فحص خطي الزمن مدرك للبنية. مسح واحد لفئات البايتات، ومرور واحد للآلة، ومقسِّمات بنيوية صغيرة، تقرر كلها بزمن خطي في طول المدخل، دون تعابير نمطية قد تتراجع.
7. حالة محدودة ومواضع تحددها تجزئة ذات مفتاح. لكل جدول سعة ثابتة وسلوك معلن عند امتلائه، وتوضع المفاتيح بتجزئة ذات مفتاح، فلا يستطيع المهاجم أن يوجه التصادمات نحو ضحية (التجزئة ذات المفتاح والمراجعة الأمنية).
8. لا تخصيص للذاكرة لكل طلب، وتحديثات للسياسة بلا توقف. يخدم الاتصال كل قرار من مخزن مؤقت ثابت واحد، ويُنشر محرك سياسة جديد دون أن ينتظر أي طلب أو كاتب (الانسجام مع العتاد: بلا تخصيص للذاكرة وبحالة محدودة).
9. صعوبة تتبع الحمل. فوق خط أساس من 50 حلًا متحققًا منه في الثانية، تضيف كل مضاعفة للمعدل بت عمل واحدًا، حتى ستة بتات، فيدفع الإغراق أكثر بصورة أسية بينما تحتفظ الحركة الهادئة بالكلفة الأساسية.
10. حملات من الحوادث. تُسجَّل الطلبات المرفوضة مع تضمين (embedding) صغير، وتُجمع الحوادث المتشابهة في حملات يستطيع المشغل التحقيق فيها من وحدة التحكم.

3 الانسجام مع العتاد: بلا تخصيص للذاكرة وبحالة محدودة

> **كنوث والدقة الميكانيكية: أسطر الذاكرة المخبئية والرياضيات الملموسة**
>
> المبرمج الذي يستدعي مخصِّص ذاكرة عام الأغراض داخل الحلقة الداخلية لنظام يعمل في الزمن الحقيقي قد سلّم جزءًا من التصميم إلى غريب. قد يكون المخصِّص سريعًا في المتوسط، لكن كلفته تتوقف على تاريخ الكومة، وهذا التاريخ يكتبه من يرسل الطلبات، أيًّا كان. وقبل أن نُحسّن أي شيء ينبغي أن نعدّ: كم بايتًا تلمس كل خطوة، وكم سطرًا من الذاكرة المخبئية، وكم عملية تحميل يعتمد بعضها على بعض. ثم ينبغي أن نقيس، لأن العدّ قد يخطئ هو أيضًا. يكون البرنامج أنيقًا حين يكون لكل بايت يحتاجه موطن معروف قبل أن يُفتح أول مقبس، وحين تؤدي كل حلقة العمل الذي تقتضيه رياضياتها بالضبط.

3.1 لامتغير انعدام التخصيص

كل تخصيص ديناميكي للذاكرة على مسار الطلب رافعة لحجب الخدمة: تجزؤ على مدى أيام، وتنافس على أقفال المخصِّص، وتضخيم بطلبات مصممة لتستدعي التخصيص. وقاعدة Sibuna أن التصنيف والتحقق وتسليم الطلب إلى الوكيل تجري كلها دون تخصيص في الكومة. يملك كل اتصال مخزن قراءة من 64 KiB ومخزن كتابة من 16 KiB طوال عمر مقبسه، ومساحة عمل للوكيل من 40 KiB تُخصَّص مرة واحدة عند بدء الاتصال. وكل ما عدا ذلك مقاطع (slices) منها: يحدد المحلل مواضع الطريقة والمسار والاستعلام والترويسات وملفات تعريف الارتباط داخل مخزن القراءة ولا ينسخ شيئًا منها[3]. ويفرض نظام الأنواع في التنفيذ ألا يعيش الطلب المحلَّل أطول من البايتات التي يصفها، فلا يمكن إعادة ملء المخزن ما دام طلب يقرأ منه.

هذه القاعدة عقدٌ يخص مسار الطلب، وليست ادعاءً بشأن العملية كلها. فإنشاء الخيوط وبدء التشغيل والتخزين تخصص الذاكرة. ويُتحقق من العقد باختبارات تعدّ عمليات التخصيص: يعدّ اختبار الحالة المستقرة للبرنامج الخدمي عمليات تخصيصه، ولا يجد اختبار الوكيل أيًّا منها على مدى 500 طلب keep-alive يمر عبر الوكيل[4].

3.2 خيوط محدودة يُعاد استخدامها

يُخدَم كل اتصال مستقبَل على خيط خاص به، فلا يؤخر العميل البطيء إلا نفسه، ويبقى مسار الطلب متزامنًا. نظرنا في حلقة أحداث واحتفظنا بنموذج الخيوط: فسرعة مسار الطلب تأتي من محلل بلا نسخ، ومخزن ثابت لكل اتصال، وقرارات بلا تخصيص للذاكرة، ولا تحسّن حلقة الأحداث أيًّا منها[5]. عدد خيوط الاتصالات محدود بـ `--max-connections` (1,024 افتراضيًا)؛ وبعد هذا الحد تجيب حلقة استقبال الاتصالات بـ `503` دون إطلاق أي شيء. ولا يحتفظ عنوان نظير واحد للمقبس افتراضيًا بأكثر من 64 اتصالًا، ويُحتسب نظير IPv6 مع شبكته /64.

تأتي الخيوط من جدول ثابت من المواضع القابلة للضم (joinable). يحتفظ الخيط المنتهي بمكدسه حتى يُضم، لذا تضم المهمة الجديدة أدنى موضع منتهٍ وتأخذه بدل أن تتقدم بمؤشر؛ وعندئذ تتبع الذاكرة المقيمة عدد الاتصالات المفتوحة، لا عدد الاتصالات المخدومة. والخيط الذي ينهي اتصاله يتوقف منتظرًا ثم يأخذ الاتصال التالي. ويروي القسم ما الذي غيّره القياس كيف علّمنا قياسٌ أن نضيف هذه القاعدة الأخيرة.

3.3 خط معالجة الطلب

يمر الطلب بالفحوص أدناه بترتيب ثابت. تأتي الفحوص الرخيصة أولًا، وكل قرار يُنهي الطلب يُنهيه قبل أن تعمل أي مرحلة أغلى.

الجلسة الصالحة تلبي التحدي؛ لكنها لا تتجاوز أبدًا رفضًا ولا نتيجة من WAF. أما الوكيل فيقرأ بدوره رأس استجابة الخادم الأصلي، ويصنف تأطيرها (بلا جسم، أو بطول محدد، أو بترميز chunked، أو حتى الإغلاق)، وينقل الجسم كما هو بالضبط، ويعيد مقابس الخادم الأصلي التي تُركت عند حد نظيف إلى مجمّع مقسم إلى شرائط يتسع لـ 256 مقبسًا، فيبقى اتصال العميل الممرَّر مفتوحًا ولا يُعاد الاتصال بالخادم الأصلي مع كل طلب.

3.4 نشر السياسة دون إيقاف الطلبات

يُعاد بناء محرك السياسة خارج مسار الطلب حين تتغير السياسات، ويجب ألا ترى الطلبات أبدًا محركًا نصف مبني. يبني خيط التخزين محركًا كاملًا وينشره، بعدّ المراجع، في موضع مع عداد للأجيال. ويحتفظ كل اتصال بمخبأ خاص: يقرأ الجيل بعملية تحميل واحدة بترتيب acquire، ولا يأخذ مرجعًا جديدًا إلا بعد نشر جديد. ويُحرَّر المحرك القديم حين يتخلى عنه آخر مخبأ يمسكه، ولا ينتظر أي كاتب أبدًا أن يفرغ القراء. وتحت التنافس، تكون القراءة من المخبأ أرخص بأكثر من 400 مرة من أخذ قفل قراءة وكتابة. ويحمل المحرك نفسه نحو 200 KiB من الجداول المتراصّة (209,052 بايت)، بسعة 128 قاعدة و8,192 عقدة في الشجرة.

3.5 أقفال في شرائح

الجداول التي تتشاركها خيوط الاتصالات (مجموعة الحلول المُنفَقة، ومحدِّد المعدل، وجدول الحظر) مقسمة إلى 16 شريحة بتجزئة المفتاح، وتقع كل شريحة على سطر خاص بها من الذاكرة المخبئية. أقسامها الحرجة بطول عشرات التعليمات، لذا قسنا قفل الدوران (spin lock) مقابل كائن استبعاد متبادل (mutex) متكيف، يدور برهة ثم ينام. يوفر قفل الدوران نحو نانوثانية حين لا ينافسه أحد، لكن خادم الحافة يشغّل خيوطًا أكثر من وحدات CPU، وقفل الدوران الذي يُنتزع المعالج من حامله يجعل كل منتظر يحرق حصته الزمنية. وقد فاز mutex تحت التنافس، فأبقينا عليه.

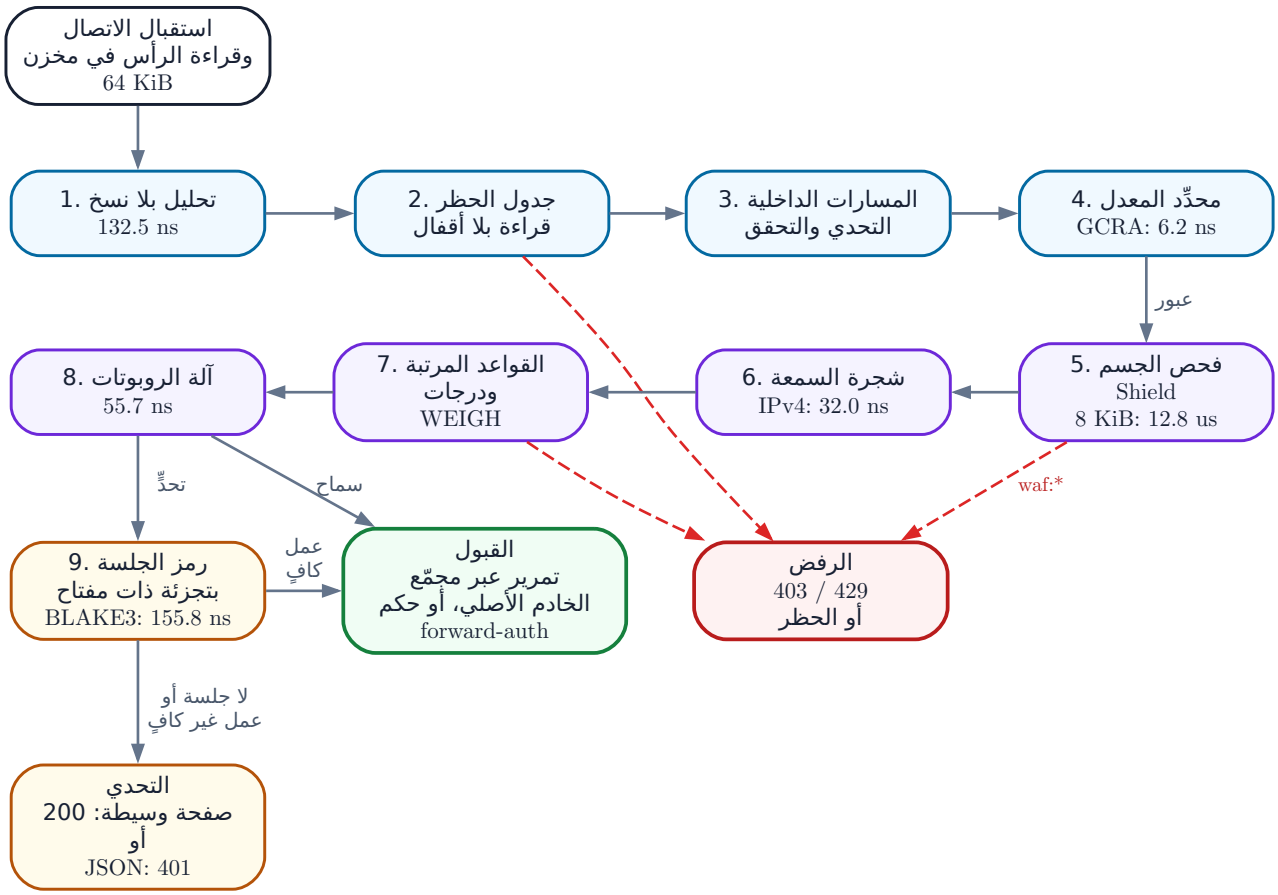
4 الخوارزميات وضماناتها

لكل عملية أولية على مسار الطلب بيان بما تكلفه وما تضمنه. ونقدم ثلاثًا منها في صورة مبرهنات، مع موجز للحجة؛ ويقدم SID 0006 والأجزاء من الثالث إلى السادس من كتاب Sibuna البراهين الكاملة والتعريفات والمراجع.

[3] الكتاب، الجزء الخامس، "خيط لكل اتصال، بحد أعلى" و"مخزن مؤقت واحد لكل اتصال".

[4] الكتاب، الجزء الخامس، "حلقة الاتصال"؛ `cmd/sibuna/tests/steady_state.rs` و`crates/net/tests/steady_state.rs`.

[5] "Concurrency model" SID 0010، و"Rationale and alternatives"؛ "Request contract" SID 0002، وملاحظات مراجعته (الاتصالات لكل عنوان، وبادئة IPv6، ومجمّع اتصالات الخادم الأصلي المقسم إلى شرائح).



الشكل 2 خط معالجة الطلب، مع الزمن الوسيط لكل عملية أولية على نواة واحدة حيث تقيسها مجموعة قياسات العمليات الأولية (benchmarks/results/primitives-latest.md). تعمل المراحل بالترتيب المرقّم؛ وتُظهر الأسهم المتقطعة حالات الرفض المبكر (ويرفض محدّد المعدل GCRA وشجرة السمعة بالطريقة نفسها). يستغرق التصنيف الكامل لطلب متصفح 75.9 ns بإعدادات Gate و790.8 ns بإعدادات Shield.

مبرهنة 1 (فحص خطي الزمن بآلة Aho-Corasick ذات فئات البايتات). لتكن $\mathcal{P}$ مجموعة أنماط طولها الكلي m، تقع بايتاتها في $c \le 256$ صنفًا، وليكن طول نصٍّ M. يبني Sibuna الآلة في زمن ومساحة $O(mc)$، ويبيّن ما إذا كان أي نمط يظهر في النص في زمن $O(M)$، بعملية تحميل واحدة من الجدول لكل بايت من المدخل يعتمد كلٌّ منها على سابقتها، بصرف النظر عن عدد الأنماط. ولا يوجد مدخل يستطيع أن يجعلها تتراجع.

مخطط البرهان. تمسح آلة Aho وCorasick النص مرة واحدة بعد بناء شجرة بادئات بروابط فشل. ويدمج Sibuna روابط الفشل في جدول الانتقال وقت البناء، فيكون كل بايت من المدخل انتقالًا واحدًا، ولا تُتبع أي سلسلة فشل أبدًا. وفئة البايت إما بايت يظهر في نمط ما، وتجتمع

الجدول 2 كلفة قراءة واحدة للمحرك مع خيط واحد ومع 16 خيطًا تقرأ في حلقة، على Ryzen 9 5950X (الجزء الخامس من الكتاب، أكتوبر 2026)

القارئ	خيط واحد	16 خيطًا
قفل قراءة وكتابة، حارس القراءة فقط	ns 4.2	ns 752
قفل قراءة وكتابة ومرجع جديد	ns 7.6	ns 833
فحص الجيل مع مرجع مخبأ	ns 0.8	ns 1.8

الجدول 3 إنتاجية أقفال الشرائح بملايين العمليات في الثانية على Ryzen 9 5950X (الجزء الخامس من الكتاب؛ SID 0010)

عبء العمل	mutex متكيف	قفل دوران
مجموعة الحلول المُنفَقة، 16 خيطًا	10.8	8.6
مجموعة الحلول المُنفَقة، 64 خيطًا	11.2	6.9
GCRA على ستة عناوين ساخنة، 16 خيطًا	44.9	19.3
GCRA على ستة عناوين ساخنة، 64 خيطًا	37.4	6.5

حالتا الحرف الواحد في فئة واحدة، وإما الفئة الوحيدة التي تضم سائر البايتات؛ ولذلك يضم صف الجدول c مدخلًا بدل 256. وتُرقَّم حالات التطابق بعد سائر الحالات، فيكفي لكشف التطابق مقارنة واحدة.

القياس. على 40 توقيعًا للروبوتات تستغرق الآلة 55.7 ns مقابل 763.0 ns للبحث المتسلسل عن السلاسل الفرعية بالأنماط نفسها ودلالة "أي تطابق" نفسها، أي أنها أسرع 13.7 مرة. وبفضل فئات البايتات يشغل جدول توقيعات الروبوتات كله 42,244 بايت[6].

□

مبرهنة 2 (حد الفترة في GCRA). لنثبّت حدًّا قدره N طلبًا لكل نافذة W، بفاصل إصدار $T = \max(1, \lceil W/N \rceil)$ وتسامح مع الدفعات $\tau = (N-1)T$. يحتفظ كل عميل بزمن وصول نظري واحد TAT. يكون الوصول في الزمن t مطابقًا إذا وفقط إذا كان $\mathrm{TAT} \le t + \tau$، وبعده يصير $\mathrm{TAT} \leftarrow \max(\mathrm{TAT}, t) + T$. عندئذ تحوي أي فترة طولها L تبدأ بوصول مطابق $N + \lfloor L/T \rfloor$ وصولًا مطابقًا على الأكثر.

مخطط البرهان. لنرقّم الوصولات المطابقة $1, \dots, k$ في الأزمنة $t_1 \le \dots \le t_k$. يرفع كل وصول مطابق قيمة TAT بمقدار T على الأقل، فبعد الوصول ذي الرتبة j يكون $\mathrm{TAT}_j \ge t_1 + jT$. وبما أن الوصول k كان مطابقًا، فإن $\mathrm{TAT}_{k-1} \le t_k + \tau \le t_1 + L + \tau$. ومن المتباينتين معًا نحصل على $(k-1)T \le L + (N-1)T$، أي $k \le N + L/T$، وk عدد صحيح.

التنفيذ. حالة العميل خلية واحدة من 16 بايتًا، بلا سجل لكل طلب ولا أثر لحدود النوافذ الثابتة. وتتقاسم الشرائح الـ 16 خلايا `--rate-table-cells` (262,144 خلية افتراضيًا، أي 4 MiB)، وتُوضع الخلايا بتجزئة ذات مفتاح، وتُستعاد الخلية المستنفدة في مكانها دون خيط كنس. ويستغرق الفحص لعميل حسب البرنامج الخدمي تجزئته مسبقًا 6.2 ns[7].

□

مبرهنة 3 (لا تنمو الحالة إلا بالعمل المدفوع). لتكن S مجموعة وسوم التحديات التي يتذكرها البرنامج الخدمي. لا يدخل الوسم إلى S إلا بعد قبول إثبات صالح لتحديه، ويخرج منها عند انتهاء صلاحية التحدي. فإذا حدّ r معدل الحلول المتمايزة المقبولة خلال عمر تحدٍّ واحد، فإن $|\,S\,| \le r \cdot \mathrm{TTL}$. أما التحدي الذي جُلب ولم يُحل فلا يشغل أي ذاكرة.

مخطط البرهان. معرّف التحدي سجل يصادق على نفسه: الإصدار، والخوارزمية، والصعوبة، وعدد عمليات الفتح، ووقت الإصدار، وبصمة العميل ذات المفتاح، وقيمة مؤقتة مشتقة بدالة شبه عشوائية، وتجزئة القاعدة التي طلبته، في 36 بايتًا يليها وسم BLAKE3 ذو مفتاح من 16 بايتًا، أي 70 محرفًا بترميز base64 الآمن لعناوين URL. ولا يكتب إصداره شيئًا. ويفحص التحقق الوسم في زمن ثابت، ثم العمر وارتباط العميل، ثم الإثبات، ولا يُدرج الوسم في مجموعة الحلول المُنفَقة إلا بعد ذلك كله. وإذا عددنا BLAKE3 في وضع المفتاح دالة شبه عشوائية، فإن الخصم الذي يجري q استعلام تحقق يزوّر وسمًا باحتمال لا يتجاوز $q \cdot 2^{-128}$ مضافًا إليه أفضلية الخصم على الدالة شبه العشوائية (PRF).

التنفيذ. مجموعة الحلول المُنفَقة 16 شريحة بعنونة مفتوحة على طريقة Robin Hood. ولا يجوز الكتابة فوق مدخل منتهي الصلاحية في مكانه إلا لمفتاح يبعد عن موضعه الأصلي مسافة لا تقل عن مسافة ذلك المدخل، وإلا فقد يتوقف بحث لاحق قبل أوانه ويفوّت وسمًا حيًّا؛ وبهذه القاعدة لا نحتاج إلى خيط كنس، ولا يطرد الإدراج المرفوض وسمًا حيًّا أبدًا. ويستغرق الإنفاق والبحث معًا 21.0 ns[8].

□

4.1 الجدار الناري الدلالي

يفحص Shield المسار، والاستعلام، وUser-Agent، والترويسات غير البنيوية، وأول 8 KiB من الجسم. وقد علّمتنا نسخته الأولى درسًا. فقد كانت تطابق قائمة مسطحة من السلاسل الفرعية مع كل ترويسة، وكانت القائمة تضم `*/*`، وكل متصفح يرسل `Accept: text/html,...,*/*;q=0.8`. فصُنّف كل متصفح حقيقي على أنه حقن SQL. ومنذ ذلك الحين لا يحق للكاشف أن ينطلق إلا على البنية، ويمرر اختبارٌ طلبَ Chrome كاملًا عبر المحرك ويشترط أن يخرج سليمًا[9].

للفحص أربع مراحل، كل منها خطية في طول الحقل.

1. مسح أولي لفئات البايتات. يسجل مرور واحد بجدول من 256 مدخلًا فئات البايتات الموجودة: علامات الاقتباس، و=، و>، وفواصل الصدفة، و%، وNUL، ومحفزات التطبيع. ويستشير كل كاشف لاحق هذه الفئات بدل إعادة المسح، فلا يدخل حقل خالٍ من علامة الاقتباس وعلامة المساواة إلى مقسّم SQL أبدًا.

[6] `benchmarks/results/primitives-latest.md` (55.7 و763.0 ns)؛ وحجم الجدول من `primitives-latest.json` كما يرد في الجزء الثامن من الكتاب، "الحجم والذاكرة"؛ وSID 0006، Part F، وملاحظة مراجعته عن فئات البايتات.

[7] SID 0006، Part D (Definition 22، Theorem 23)؛ SID 0004، ملاحظات المراجعة (المواضع المحددة بتجزئة ذات مفتاح، و`--rate-table-cells`)؛ `benchmarks/results/primitives-latest.md` (`gcra_check_hashed`).

[8] الكتاب، الجزء الثالث (التمهيدية 3 والتمهيدية 4، وتنسيقات النقل)؛ الكتاب، الجزء الخامس، "مجموعة Robin Hood للحلول المُنفَقة"؛ SID 0006، Part E (Lemma 28)؛ `benchmarks/results/primitives-latest.md` (`robin_hood_spend_and_lookup`).

[9] "Revision note (2026-09-07)" وSID 0004، "Verification"؛ الكتاب، الجزء السادس، "درس في الإنذارات الكاذبة".

2. آلة توقيعات موسومة. تقيم كل التوقيعات القوية لجميع الفئات في آلة واحدة (المبرهنة 1). والإصابة رفض نهائي يُسمّى `waf:sqli` أو `waf:xss` أو `waf:path-traversal` أو `waf:rce`.
3. مقسِّمات بنيوية. يبحث مقسِّم SQL عن كل كلمة قصيرة في جدول الكلمات المفتاحية، ويفحص العبارات الصحيحة دائمًا من شكل `literal = literal` بعد `or` و`and`؛ وفيما عدا ذلك تكون علامة الاقتباس شرطًا لازمًا، ويجب أن تبلغ القرائن درجة صغيرة. فالعبارة "it's a group order from the shop" تحصل على 3 وتمر. ويتطلب كشف البرمجة النصية عبر المواقع (XSS) وسمًا يمكن أن يستضيف سكربتًا مع سمة معالج أحداث في موضع السمة، ويتطلب حقن الأوامر فاصلًا يليه اسم أمر ليس إسنادًا، فيكون `a=1&id=5` معاملًا لا أمرًا.
4. التطبيع. الحقل الذي يحوي تهريبات النسبة المئوية، أو بادئات التعليقات، أو تهريبات JSON أو HTML، أو مسافات بيضاء قابلة للدمج، يُفك ترميزه ويُفحص مرة أخرى، فتؤول `1%27/**/UnIoN/**/SeLeCt` و`%252e%252e/` إلى التوقيعات الخام.

يُمسح جسم من 8 KiB في 12,784.3 ns، أي نحو 1.56 ns لكل بايت. ويمكن للمشغّل أن يضع فئة في وضع التدقيق بدل الإنفاذ، فتُسجَّل نتائجها دون أن يُتخذ بشأنها إجراء. وعندئذ يجب الحكم على الحقل لكل فئة على حدة، كي لا تخفي نتيجةٌ مدققة نتيجةً مُنفَذة. وبدل مسح الحقل مرة لكل فئة، يمسحه Sibuna مرة واحدة ليجمع الفئات التي تظهر توقيعاتها، ثم لا يبحث إلا عن الفئات المطلوبة من بينها؛ ويستغرق جسم الـ 8 KiB نفسه مع وضع حقن SQL في التدقيق 15,485.4 ns[10]. ويبقى Shield فاحصًا استدلاليًا محدودًا، لا محللًا كاملًا لـ SQL أو HTML، ولا تُفحص البايتات التي تتجاوز حد الفحص.

4.2 بصمات TLS من وكيل الدخول

ترويسة User-Agent سلسلة يكتبها أي عميل كما يشاء. أما بصمة JA4 فتزويرها أصعب. فهي تلخص الطريقة التي يفتح بها العميل اتصال TLS: البروتوكول، والإصدار، ومجموعات التشفير، والامتدادات، وأول قيمة ALPN. وتثبّت مكتبة TLS لدى العميل هذه القيم قبل إرسال أول بايت من HTTP. ولا يرى المصافحةَ إلا المكوّن الذي يُنهي TLS، فهو وحده القادر على حساب JA4. وبما أن Sibuna لا يُنهي أي اتصال TLS وارد، فإنه لا يحسب JA4. بل يقرأ البصمة التي حسبها وكيل دخول موثوق ومررها في ترويسة طلب يسميها `--ja4-header`[11].

- الثقة. لا تُقرأ الترويسة إلا حين تكون الترويسات الممررة موثوقة، بالقرار نفسه الذي يحكم `X-Forwarded-For`. وتُتجاهل إذا جاءت من نظير غير موثوق، وفي وضع الوكيل العكسي تُزال كل نسخها قبل أن يصل الطلب إلى الخادم الأصلي.
- التحقق. لا تُقبل القيمة إلا بصيغة JA4 ذات الـ 36 بايتًا، وإلا إذا وُجد حقل واحد بالضبط بذلك الاسم. والقيمة المشوهة أو المكررة تُعامل كأنها غائبة، وتُعدّ.
- السياسة. تطابق القاعدة بصمة كاملة، أو بادئة يليها *؛ ولا يطابق طلب بلا بصمة مثل هذه القاعدة أبدًا. والمطابقة مقارنة بايتات، ويظل مسار الطلب الدافئ بلا أي تخصيص في الكومة.
- السجلات. تخزن الحوادث بصمة طلبها، وتعرضها وحدة التحكم في أدلة الحادث وتصفّي الأحداث بها. أما الحظر والسمعة فيبقيان مفهرسين بالعنوان، لأن كثيرًا من العملاء المشروعين يتشاركون مكتبة TLS واحدة.

4.3 التجزئة ذات المفتاح والمراجعة الأمنية

كشفت مراجعة عدائية لمسار الطلب في أكتوبر 2026 نقاط ضعف في آليات كنا نعدّها محسومة. فأصلحنا كلًّا منها، وسجلنا الإصلاح في سجل التصميم الخاص بالآلية[12].

- مواضع جداول يمكن التنبؤ بها. كان محدِّد المعدل وجدول الحظر يضعان المفاتيح بتجزئة تحت بذور مكتوبة في الشيفرة المصدرية، فكان بوسع أي أحد أن يحسب عناوين تتشارك نافذة السبر الخاصة بضحية. أما الآن فيعتمد الوضع على SipHash-1-3 تحت مفتاح مأخوذ من عشوائية نظام التشغيل، ويشتق كل جدول مفتاحه الخاص. ويجزّئ البرنامج الخدمي العميل مرة واحدة لكل طلب (8.7 ns) ويمرر التجزئة إلى كل جدول؛ فانخفض مسار التخزين للطلب من 43 إلى 26 ns.
- طرد الحظر. الحظر الساري لا يُطرد أبدًا. وحين تكون كل المواضع التي يمكن أن يشغلها المفتاح تحمل حظرًا حيًّا، يُرفض الحظر الجديد ويُعدّ، فلا يستطيع العميل المحظور أن يطرد حظره.
- الهوية الممررة. يستطيع العميل أن يكتب المدخل الأيسر في `X-Forwarded-For`. لذلك يأخذ Sibuna المدخل الذي يبعد `--trusted-hops` موضعًا عن اليمين، وهو الذي أضافه أقرب وكيل موثوق.
- رمي ملفات تعريف الارتباط *(cookie tossing)*. قد يحمل الطلب عدة قيم لملف تعريف ارتباط الجلسة؛ ويجرب البرنامج الخدمي حتى ثماني منها، فلا يستطيع ملف تعريف ارتباط زرعه نطاق شقيق أن يُبقي الزائر في حلقة من التحديات.

[10] `benchmarks/results/primitives-latest.md` (`8kb_body_semantic_scan` و`8kb_body_sqli_audit_other_enforce`)؛ SID 0004، ملاحظة المراجعة "inspection modes"؛ الكتاب، الجزء الثامن، "فحص WAF".

[11] SID 0014، "Specification" و"Security and operational considerations"؛ الكتاب، الجزء التاسع، "بصمات JA4 من وكيل الدخول"؛ `github/release-notes.txt`..

[12] SID 0002، ملاحظات المراجعة (الهوية الممررة؛ قيم ملفات تعريف الارتباط)؛ SID 0004، ملاحظات المراجعة (المواضع المحددة بتجزئة ذات مفتاح، وجدول الحظر، والتكاليف المقيسة، والجداول المصفَّرة)؛ SID 0006، ملاحظات المراجعة (سعة مجموعة الحلول المُنفَقة، ومدخلات المتحكم التكيفي)؛ SID 0010، سجل المراجعة (المراجعة الأمنية، 2026-10-15)؛ `benchmarks/results/primitives-latest.md` (`gcra_check` و`gcra_check_hashed`).

- سعة مجموعة الحلول المُنفَقة. استُنفد جدول من 65,536 وسمًا بنحو 220 حلًا متحققًا منه في الثانية خلال عمر تحدٍّ مدته خمس دقائق، فصارت الحلول النزيهة تُرفض بعد ذلك. والسعة الآن قابلة للضبط، وقيمتها الافتراضية 1,048,576 وسمًا (24 MiB).
- الصعوبة التكيفية. صارت وحدة التحكم التي ترفع العمل تحت الحمل تعدّ الحلول المتحقق منها بدل التحديات الصادرة، التي لا تكلف المُغرِق شيئًا.

للوضع ذي المفتاح ثمن، ونحن ندفعه مرة لكل طلب لا مرة لكل جدول. ففحص محدِّد المعدل الذي يجزّئ بايتات العميل الخام بنفسه يستغرق 12.8 ns؛ أما الفحص على عميل مُعرَّف بتجزئة محسوبة مسبقًا، وهو المسار الذي يسلكه البرنامج الخدمي، فيستغرق 6.2 ns. ولأن محدِّد المعدل ومجموعة الحلول المُنفَقة يُخصَّصان مصفَّرين، فإن الجداول الخاملة لا تقيم في الذاكرة، ولا تكلف سعتها المضبوطة أي ذاكرة حتى تُستخدم.

5 إثباتات العمل: البحث المتوازي والعمل المتسلسل

5.1 مستوى Hashcash

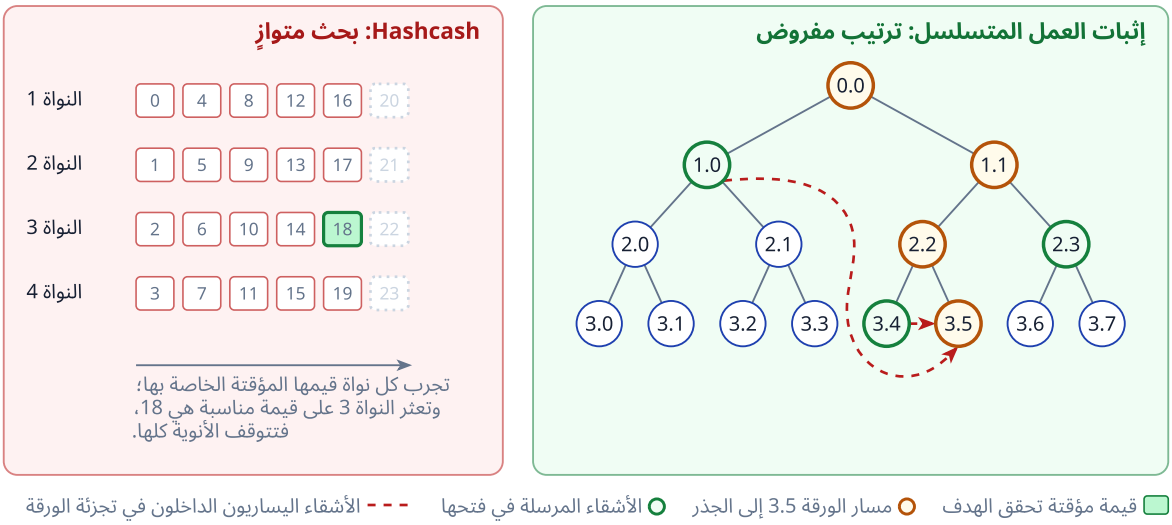
المستوى الأول هو Hashcash بدقة البت الواحد. فلسلسلة التحدي C والصعوبة b، يجد العميل قيمة مؤقتة N بحيث تبدأ $\mathrm{SHA\text{-}256}\,(C \,\|\, \text{":"} \,\|\, \mathrm{dec}\,(N))$ بـ b بتًا صفريًا. وتبقي الكتابة العشرية للقيمة المؤقتة صيغة النقل متطابقة بين برنامج الحل بـ WebAssembly، والبديل بـ JavaScript، والمتحقق الأصلي. وفي حالة الملخصات المثالية يتبع عدد المحاولات K توزيعًا هندسيًا مع $p = 2^{-b}$:

$$P(K > k) = (1-p)^{k}, \qquad E[K] = 2^{b}, \qquad \mathrm{median}\,(K) \approx 2^{b} \ln 2.$$

للبحث حدّان. فتباينه كبير، ولذلك لا يعني حلٌّ بطيء واحد أن العامل معطَّل. والمحاولات المستقلة تجري على التوازي، فينتهي المرسل الذي يملك أنوية أكثر في وقت أقصر، مع أن العمل الكلي المتوقع واحد[13].

5.2 مستوى العمل المتسلسل

يتبع المستوى الثاني إثبات العمل المتسلسل لـ Cohen وPietrzak (EUROCRYPT 2018). تشكّل تسمياته شجرة ثنائية تامة عمقها n. تجزئ العقدة الداخلية ابنيها. وتجزئ الورقة الأشقاء اليساريين لأسلافها حيثما اتجه مسارها من الجذر إلى اليمين. وهذه الحواف إلى الأشقاء اليساريين هي التي تجعل الحساب متسلسلًا: فلا يمكن تسمية ورقة قبل اكتمال كل شجرة فرعية على يسارها، ولذلك تعتمد تسمية الجذر على التسميات الـ $2^{n+1}-1$ كلها بالترتيب. وبعد حساب الجذر φ، يشتق المُبرهِن t ورقة بطريقة Fiat-Shamir، ويرسل لكل منها تسمية الورقة وتسميات الأشقاء الـ n على طول مسارها. ويعيد المتحقق حساب كل ورقة مفتوحة ويجزئ صعودًا حتى φ، بكلفة $O(tn)$ من حسابات التسميات في مساحة عمل ثابتة. اختيرت المعلمات بحيث تبقى الأرقام صغيرة. تُحوَّل بتات العمل إلى عمق بالعلاقة $n = b - 3$، مع حصرها في $[4, 24]$. ويبلغ حجم الإثبات $32(1 + t(n+1))$ بايتًا، أي $32(1 + 16 \times 14) = 7200$ بايت عند العمق 13 مع 16 عملية فتح. ولا يحتفظ المُبرهِن إلا بالمستويات العليا من الشجرة، ويعيد حساب الشجرة الفرعية التي يجب أن يفتحها، فتبقى مساحة عمله كلها، ومعها الإثبات، دون 100 KB؛ ويستخدم تبويب المتصفح



الشكل 3 البحث المتوازي مقابل العمل المتسلسل. على اليسار، Hashcash: محاولات مستقلة على القيم المؤقتة، فتنتهي M نواة أسرع بنحو M مرة. وعلى اليمين، شجرة Cohen-Pietrzak بعمق 3: تجعل حواف الأشقاء اليساريين المتقطعة الورقة 3.5 تنتظر الأشجار الفرعية التي على يسارها، ويرسل فتح الورقة 3.5 تسميتها والأشقاء الثلاثة الخضر (عن الجزء الثالث من الكتاب).

[13] الكتاب، الجزء الثالث، "Hashcash: رياضيات المستوى الأول"؛ الكتاب، الجزء الأول، "البحث عن التجزئة متغيرًا عشوائيًا".

والاختبارات الأصلية المُبرهِن نفسه[14]. وقد برهن Blocki وLee وZhou (ITC 2021) سلامة البناء في مواجهة مُبرهِنين كموميين في نموذج الأوراكل العشوائي الكمومي.

ينبغي أن نحذر مما يثبته هذا. فنتائج الأمان صحيحة تحت افتراضات الأوراكل العشوائي في تلك الأوراق، وللبنى والحدود التي تنص عليها. وهي ليست تدقيقًا مستقلًا لبروتوكولنا الملموس القائم على SHA-256، ولا لتحويل معلماته، ولا لتوقيته في المتصفح. أما حد أخذ العينات الأولي $(1-\alpha)^t$، أي احتمال أن تخطئ t عملية فتح مستقلة كلها جزءًا سيئًا ثابتًا α، فيفسر لماذا تفيد زيادة عمليات الفتح؛ لكنه وحده ليس برهانًا في مواجهة مُبرهِن يكرر المحاولة على جذور كثيرة. ولا ندّعي ادعاءً غير مشروط بأن الغش لا يكون أرخص أبدًا.

5.3 رحلة الذهاب والإياب والجلسة

يتلقى التنقل الذي لا يحمل جلسة صالحة الصفحةَ الوسيطة، وهي تحمل تذكرة متطلبات: التحدي الذي يفرضه هذا القرار (الخوارزمية، وبتات العمل، وعمليات الفتح) وتجزئة القاعدة، مختومة لهذا العميل وصالحة مدة عمر تحدٍّ واحد. وتجلب الصفحة تحديها بهذه التذكرة، فيكون العمل محددًا بالقرار الذي طلبه. ويحل Web Worker التحدي بالمُبرهِن المبني بـ WebAssembly، أو ببديل JavaScript حيث لا يتوفر WebAssembly، ثم يرسل الجواب. وعند النجاح يضبط الخادم ملف تعريف ارتباط للجلسة من 75 محرفًا، مربوطًا ببصمة ذات مفتاح بعنوان العميل وUser-Agent، ثم تعيد الصفحة التحميل. والمُبرهِن في المتصفح مترجَم من الشيفرة المصدرية نفسها التي يُترجم منها متحقق الخادم، فلا يمكن أن يتباعدا[15].

5.4 كم يكلف التحقق

يقيّد SHA-256 المستويين كليهما، ويختار Sibuna تعليمات SHA في المعالج وقت التشغيل حيثما وُجدت، فيكون الملف التنفيذي الواحد سريعًا على كل جهاز يملكها. يستغرق التحقق من Hashcash عند 16 بتًا 70.2 ns، ويستغرق التحقق من إثبات بعمق 13 مع 16 عملية فتح 22,272.4 ns. والمُبرهِن في المتصفح وحدة WebAssembly حجمها 14,744 بايت. وفي فحص المتصفح من طرف إلى طرف، حل Chromium بلا واجهة تحدي الصفحة الوسيطة في 29 إلى 34 ms بـ WebAssembly، وفي 84 إلى 91 ms ببديل JavaScript[16]. ولا تقيس مجموعة الأوليات إلا التحقق؛ والتوقيت الأصلي لا يمثل زمن الحل في المتصفح.

تستخدم الجلسات رمزًا متماثلًا. فالمتحقق هو المُصدِر، أو عنقود يتشارك بذرة واحدة، ولذلك توفر دالة شبه عشوائية ذات مفتاح المصادقة المطلوبة بالضبط. يُتحقق من رمز BLAKE3 ذي المفتاح في 155.8 ns ومن رمز Ed25519 في 40,587.5 ns، أي أن التحقق من توقيع واحد يكلف قرابة ما يكلفه 260 تحققًا من MAC. وللاختيار سبب ثانٍ: فخوارزمية Shor تكسر اللوغاريتمات المنفصلة التي يقوم عليها Ed25519، بينما لا يمنح البحث الكمومي في مواجهة التجزئة ذات المفتاح إلا تسريعًا تربيعيًا على الأكثر. ويبقى Ed25519 خيارًا لعمليات النشر التي تحتاج إلى متحققين لا يستطيعون سك الرموز.

6 حالة مشتركة دون بروتوكول إجماع خاص بنا

> **لامبورت واللامتغيرات الموزعة: السلامة والحيوية ومن يملك السجل**
>
> في عام 1987 وصف ليزلي لامبورت النظام الموزع بأنه نظام قد يجعل فيه تعطّلُ حاسوب لم تكن تعلم بوجوده حاسوبَك أنت غير صالح للاستعمال. وإذا قرأناه نصيحةً، فهو يطلب منا أن نحصي الحواسيب التي نعتمد عليها، وأن نذكر ما يجب أن يبقى صحيحًا حين يتعطل أيٌّ منها. ويكفي نوعان من العبارات. السلامة: لا يحدث شيء سيئ أبدًا، كأن لا تعمل عقدة أبدًا بسياسة لم تكتبها معاملة مودَعة. الحيوية: يحدث شيء جيد في النهاية، كأن تخدم العقدة التي تستطيع الوصول إلى قاعدة البيانات أحدث سياسة في النهاية. وبروتوكول الإجماع طريقة جميلة للحصول على الاثنتين معًا. لكنه أيضًا شيء كبير على من يمتلكه، والسؤال المطروح على الجدار الناري ليس هل الإجماع جيد، بل من ينبغي أن يتولى صيانة السجل.

6.1 لماذا SQL، ولماذا لا نبني بروتوكول إجماع خاصًا بنا

لا يخزن Sibuna إلا القليل: بضع مئات من صفوف السياسات، وجدول سمعة، وحوادث للتحقيق الجنائي يصدّرها المشغلون أيضًا إلى أنظمة أخرى. ولا تحتاج هذه الحالة إلى محرك نسخ متماثل مصنوع خصيصًا. بل تحتاج إلى مخزن SQL موثوق، محلي للعقدة الواحدة ومشترك بين العقد الكثيرة. لذلك تحفظ العقدة المنفردة حالتها في ملف SQLite مدمج في دليل بياناتها، في وضع WAL، ويوجّه العنقود كل عقدة إلى قاعدة بيانات PostgreSQL واحدة[17].

اخترنا ألا نشغّل بروتوكول إجماع داخل الجدار الناري. فمحرك الإجماع نظام كبير على من يمتلكه بجانب الجدار الناري، له صيغه الخاصة وأنماط إخفاقه الخاصة، ويضيف عملًا إلى الإيقاف وإلى كل ترقية. وتصبح إتاحة حالة العنقود ونسخها من شأن PostgreSQL بدلًا من ذلك، بما يوفره

[14] الكتاب، الجزء الثالث، "إثبات العمل المتسلسل: المستوى الثاني" و"ذاكرة المُبرهِن" و"المعلمات ليست أزمنة"؛ SID 0006، Part B ومراجعة تنفيذه بتاريخ 2026-09-07.

[15] الكتاب، الجزء الرابع، "دورة التحدي" و"ملف تعريف ارتباط الجلسة"؛ SID 0010، "The browser solver".

[16] benchmarks/results/primitives-latest.md؛ وحجم WebAssembly من primitives-latest.json كما يرد في الجزء الثامن من الكتاب، "الحجم والذاكرة"؛ وأزمنة الحل في المتصفح من SID 0010، المعيار 3 (Chromium 155، cargo xtask browser)؛ والكشف عن تعليمات SHA وقت التشغيل من SID 0006، ملاحظة المراجعة (Parts A to C).

[17] "Abstract"، SID 0011 و"Motivation and scope" و"Why PostgreSQL for clusters"؛ ويصف SID 0005 التصميم المكرَّر السابق.

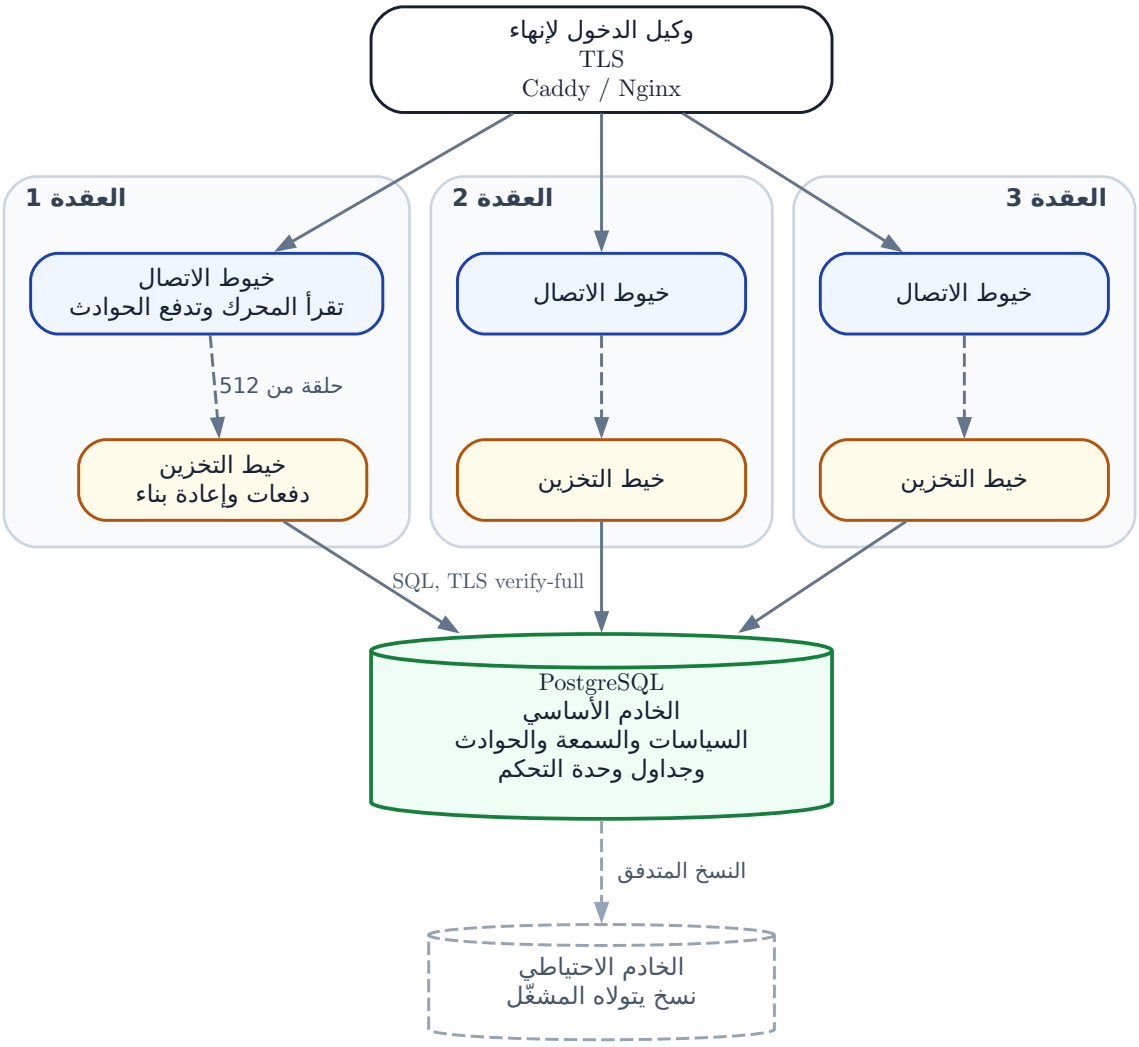
من النسخ المتدفق وأدوات تجاوز الفشل، أو عبر خدمة مُدارة. ونرى أن هذا هو الخيار الأكثر تواضعًا: فقد اختُبر النسخ في PostgreSQL في عمليات نشر تفوق بكثير ما يمكن أن تبلغه عمليات نشرنا، ومعدل الكتابة في Sibuna (حوادث مجمّعة في دفعات وتغييرات نادرة في السياسات) صغير بما يكفي لئلا يكون الخادم الأساسي الواحد عنق زجاجة.

ويبقى التخزين اختياريًا كذلك. فمن دون `--data-dir` أو `--database-url` يكون البرنامج الخدمي جدارًا ناريًا في الذاكرة تأتي سياساته من ملف.

6.2 خيط تخزين واحد

يمتلك خيط تخزين واحد اتصالات قاعدة البيانات؛ ولا يلمس SQL شيء غيره. ولا تنتظره خيوط الطلبات أبدًا. فهي تدفع سجلات الحوادث إلى حلقة محدودة من 512 موضعًا، تُسقط السجل وتعدّه حين تمتلئ، وتقرأ المحرك المنشور. وكل 500 ms افتراضيًا ينفذ خيط التخزين نبضة واحدة: يكتب ما يصل إلى 32 حادثًا منتظرًا في معاملة واحدة، ويقرأ مراجعة السياسة، وإذا تغيرت أعاد بناء المحرك ونشره. وتقرأ إعادة البناء المراجعة قبل التحميل وبعده؛ فإن أودع كاتب تغييرًا في الأثناء، استُبعد المحرك المرشح وبقي المحرك القديم ساريًا. ويمكن للمشغلين الكتابة في جداول السياسات مباشرة: إذ تقدّم القوادح (triggers) المراجعة داخل معاملة الكاتب، وفي PostgreSQL يتيح `NOTIFY` للعقد أن تنفذ نبضتها فورًا، بينما يبقى الاستطلاع الدوري هو الضمان[18].

لكل عقدة في العنقود معرّف `--cluster-node` مميز بين 1 و8,388,607. ترتبط مفاتيح التحديات بالعقدة التي أصدرتها، فيجب أن يعود الحل إلى العقدة التي أصدرت تحديه، بينما تصلح رموز الجلسات وتذاكر المتطلبات على كل عقدة تشترك في البذرة نفسها. وتحمل معرفات الحوادث رقم العقدة في بتاتها العليا، وتحتفظ كل عقدة بإيصال دائم يتقدم في المعاملة نفسها التي تُكتب فيها دفعتها، فلا تكتب الدفعة المعادة بعد عطل ملتبس شيئًا مرتين.



الشكل 4 عنقود Sibuna. تُبقي كل عقدة مسار طلباتها في الذاكرة وتُسند كل عمليات SQL إلى خيط تخزين واحد؛ وتتشارك العقد قاعدة بيانات PostgreSQL واحدة، يتولى PostgreSQL نسخها المتماثل وتجاوز أعطالها (SID 0011). أما العقدة المفردة فتستخدم بدلًا من ذلك ملف SQLite مدمجًا.

[18] "The storage thread"، "SID 0011" و"Policy revision" و"Incidents and campaigns".

اللامتغيرات في طبقة التخزين (SID 0011). **S1 (مالك واحد).** تجري كل عمليات SQL على خيط التخزين؛ ولا ينتظر أي خيط طلبات قاعدة البيانات أبدًا. **S2 (محركات مكتملة).** كل محرك يقرؤه طلب بُني كاملًا قبل نشره، من مراجعة واحدة متسقة لجداول السياسات. **L1 (النشر في نهاية المطاف).** يصل كل تغيير يُثبَّت في جداول السياسات إلى كل عقدة تستطيع الوصول إلى قاعدة البيانات عند نبضتها التالية. **F1 (فقد سلس).** حين تتعطل قاعدة البيانات، يواصل آخر محرك منشور الخدمة، وتصطف الحوادث في الحلقة المحدودة وتُسقط مع عدّاد حين تمتلئ، ويعيد خيط التخزين المحاولة مع تأخير متزايد بين المحاولات.

6.3 مخطط محدد الأنواع

يعبّر المخطط بلغة SQL عما تعبّر عنه أنواع البرنامج، فترفض قاعدة البيانات أي صف لا تستطيع طبقة التخزين تمثيله. جداول SQLite من نوع `STRICT`؛ ولكل عمود تعدادي قيد `CHECK` يسرد بالضبط الأسماء التي يكتبها البرنامج؛ ولكل عدد صحيح محدود فحص للمدى؛ ويكون العمود `NOT NULL` حين لا تغيب قيمته أبدًا، ولا يكون كذلك إلا حينئذ. تربط كل عبارة قيمها بوصفها معاملات (parameters)، فلا تُبنى أي شفرة SQL بإدراج القيم بين علامات الاقتباس. ويتبع تخزين وحدة التحكم القواعد نفسها: ترحيل أساسي واحد لـ 33 جدولًا محدد الأنواع، ولا مشغّلات (triggers). ويفرض كل لامتغير من لامتغيرات وحدة التحكم، مثل "كل تغيير في رمز يكتب صف تدقيق"، دالة واحدة بالضبط تنفذ عباراتها في معاملة واحدة، على SQLite وPostgreSQL على السواء[19].

تُحمى الاتصالات بـ PostgreSQL كما يحدد عنوان URL، عبر `sslmode` و`sslrootcert` في libpq، حين يُبنى البرنامج الخدمي بالميزة `postgres-tls`. ويرفض البناء الخالي من هذه الميزة القيم `require` و`verify-ca` و`verify-full` عند بدء التشغيل؛ ولا يتراجع أبدًا إلى نص غير مشفر. وتستخدم عناقيد الإنتاج `sslmode=verify-full`.

تُجمّع الحوادث المتشابهة في حملات. يحمل كل حادث تضمينًا (embedding) من 64 بُعدًا يُخزَّن في 256 بايت. ويُبقي خيط التخزين في الذاكرة أحدث 16,384 تضمينًا من جميع العقد (4 MiB)، ويُسند الحادث إلى حملة أقرب جيرانه حين لا تتجاوز مسافة جيب التمام 0.35. أما التجميع مقابل الحوادث الأقدم فإغفال متعمد: فالحملات حديثة بطبيعتها، والحد يُبقي الذاكرة والعمل ثابتين.

6.4 ما لا يفعله هذا التصميم

ينبغي أن نكون صريحين بشأن حدود هذا الاختيار. لا يتضمن Sibuna بروتوكول إجماع خاصًا به، واتساق العنقود هو اتساق PostgreSQL: فالعقدة التي تفقد قاعدة البيانات تحتفظ بآخر سياسة لديها حتى تعيد الاتصال. والجلسة التي تُصدَر على عقدة تقبلها كل العقد، والحل المُنفَق على عقدة يُرفض إذا أُعيد تقديمه إلى أخرى، لأن العقد تتشارك بذرة واحدة ولأن مفاتيح التحديات مرتبطة بالعقدة المُصدِرة؛ ولا تحتاج أي من الخاصيتين إلى قاعدة البيانات. حالة SID 0011 هي Committed: يُظهر اختبار وظيفي بثلاث عقد على قاعدة بيانات PostgreSQL واحدة أن السياسة المُدرجة على عقدة تفرضها العقد الثلاث كلها خلال نبضة واحدة، وأن حظر المصيدة على عقدة يصل إلى الأخريات، وأن معرفات الحوادث لا تتصادم أبدًا. ولا توجد بعدُ أداة قياس تشغّل عدة عقد على قاعدة بيانات PostgreSQL واحدة تحت الحمل، لذا لا تعرض هذه الورقة أي قياسات للعنقود. وتُبنى الملفات التنفيذية للإصدار دون دعم PostgreSQL؛ ويبني العنقود Sibuna من المصدر بالميزة `postgres-tls`[20].

7 وحدة التحكم

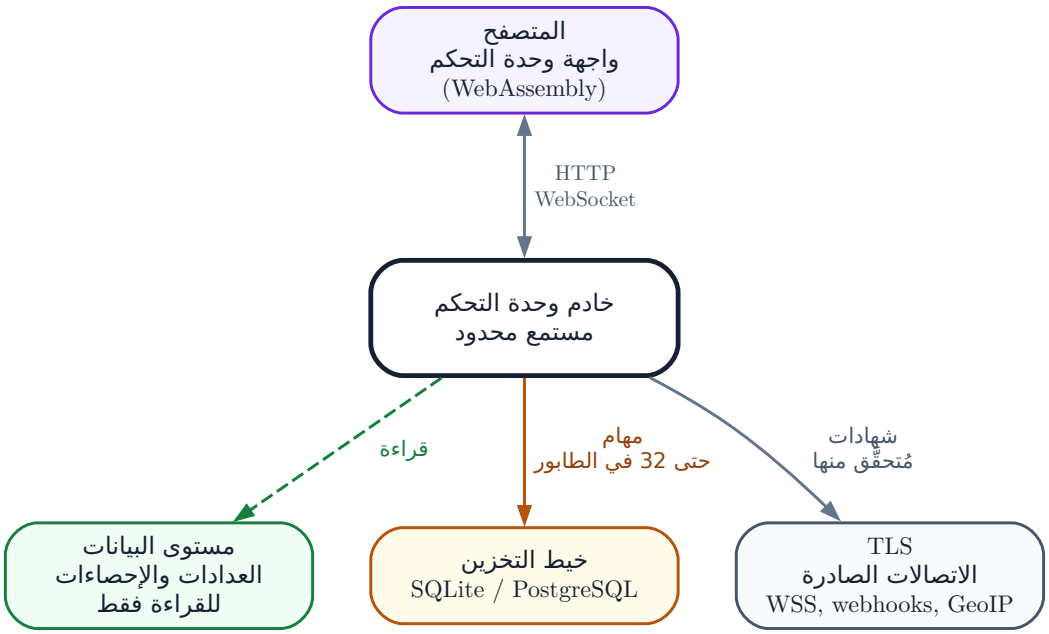
يتضمن Sibuna وحدة تحكم إدارية اختيارية: لوحات معلومات بمصادقة تعرض حركة المرور الحية، ومسار تحقيق يقود من الرفض إلى أدلته المسجلة، وتحرير السياسات، ومجموعات العناوين، وإثراء البيانات بالبلد من بيانات GeoIP المفتوحة، ومستخدمين بكلمات مرور Argon2id وعامل تحقق ثانٍ TOTP، وسجل تدقيق، وخطافات ويب (webhooks)، وإدارة العنقود. وهي مُضمَّنة في الملف التنفيذي نفسه، وتعمل واجهتها في المتصفح وحدةَ WebAssembly[21].

القيد الحاكم لوحدة التحكم هو عزلها عن مسار الطلب. فلها أن تقرأ عدادات مستوى البيانات وقاعدة بياناتها وأن تكتب في قاعدة البيانات، لكنها لا تدخل خيط طلبات أبدًا، ولا تخصص ذاكرة على أي منها، ولا تمسك قفلًا يحتاج إليه خيط اتصال. ويقبل مستمعها 96 اتصالًا على الأكثر، يُحجز منها 16 لطلبات HTTP، ويمكن أن يكون 64 منها تدفقات متصفح و16 تدفقات نظراء. في beta.3-1.0.0 يبلغ حجم وحدة الواجهة الكاملة 724,330 بايت (272,568 بايت بعد الضغط بـ gzip). وهذا دون السقف البالغ 768 KiB الذي يفرضه البناء، وفوق عتبة المراجعة البالغة 640 KiB، وهو

[19] "Schema and migrations"، SID 0011، وسجل مراجعته (أمان النقل)؛ "Abstract"، SID 0013.

[20] "Security and operational considerations"، SID 0011، و"Acceptance criteria and implementation status" (المعيار 2، `crates/storage/tests/multi_node.rs`)؛ "Known limitations"، `github/release-notes.txt`؛ "Authentication and work"، SID 0002 (مفاتيح التحديات المرتبطة بالعقدة)؛ الكتاب، الجزء الثامن، "ما لم يُقس".

[21] "Abstract"، SID 0007 وملاحظات مراجعته (القبول على المستمع، ووحدة الواجهة، وGeoIP وكلمات المرور)؛ SID 0010، سجل المراجعة (نقل وحدة التحكم، وTOTP)؛ SID 0012؛ SID 0013. وقياس العزل موجود في سجل مراجعة SID 0010 (نقل وحدة التحكم)؛ ويشير الجزء الثامن من الكتاب، "التكلفة الإضافية لوحدة التحكم"، إلى أن السجل لا يسمي ملف نتائج له.



الشكل 5 وحدة التحكم. يشغّل المتصفح الواجهة وحدةَ WebAssembly؛ ويقرأ خادم وحدة التحكم عدادات مستوى البيانات ويُسند عمليات SQL الخاصة به إلى خيط التخزين؛ وتستخدم الاتصالات الصادرة إلى النظراء وخطافات الويب وناشري GeoIP بروتوكول TLS بشهادات مُتحقَّق منها (SIDs 0007 و0012 و0013).

ما يسجله SID 0007 للمراجعة[22].

تُنشئ وحدة التحكم اتصالات TLS صادرة لتدفقات النظراء طويلة العمر، وخطافات الويب، وتنزيلات GeoIP. ويُتحقق من الشهادات مقابل جذور النظام أو ملف CA يحدده المشغّل، ولا يمكن تعطيل التحقق من اسم المضيف، ولا يُسمح بـ HTTP العادي إلا إلى عناوين الاسترجاع (loopback) المكتوبة حرفيًا، ولا يتراجع خطأ الشهادة أبدًا إلى نص غير مشفر. ومع ذلك لا يُنهي Sibuna أي اتصال TLS وارد؛ فتلك تبقى مهمة وكيل الدخول. يحدد SID 0007 هدف الأداء لوحدة التحكم: لا يجوز أن يكلف تفعيلها أكثر من 1% من إنتاجية مستوى البيانات ولا أكثر من 10% من زمن الاستجابة p99. ولدينا قياس واحد: مع تضمين وحدة التحكم في البناء وهي معطلة، أو للتخزين فقط، أو مفعلة وخاملة، بقيت أعباء العمل الستة عشر للمنتج الكامل في حدود نحو 1% من برنامج خدمي مبني دونها، وهذا ضمن الضجيج. أما معيار الاجتياز الرسمي، مع ثماني لوحات معلومات نشطة تحت الحمل، فلم يُشغَّل بعد، لذا لا نزعم أن الهدف قد تحقق. وقد نجح تشغيل وظيفي بثماني لوحات معلومات حية.

الجدول 4 ثماني لوحات معلومات Traffic على عقدة SQLite واحدة مع وحدة تحكمها، في Chromium 155 بلا واجهة (`benchmarks/results/console-8-dashboards-2026-10-15.md`، سُجل في 15-10-2026). النتيجة: PASS

المقياس	القيمة
طلبات مستوى البيانات	6,000 في 30.0 s (200 في الثانية، اتصال واحد لكل طلب)
المُجاب عنها	6,000: 200 x 1,200، و401 x 3,000، و403 x 600، و404 x 600، و500 x 600
الأخطاء (بلا رد)	0
زمن الاستجابة p99 / p90 / p50 / الأقصى	246 / 311 / 373 / 552 us (يشمل الاتصال)
الذاكرة المقيمة للبرنامج الخدمي عند البدء، ومع 8 لوحات معلومات، وبعد حركة المرور	36,152 KiB؛ 38,028 KiB؛ 41,496 KiB
لوحات المعلومات	3 جلسات للمسؤولين و5 شاشات عرض جدارية؛ تلقت كل منها من 134 إلى 139 تحديثًا تفاضليًا لـ `stats` وانتهت بالمجاميع نفسها، دون أي استثناء أو خطأ في وحدة التحكم أو انتهاك لسياسة Content-Security-Policy

[22] SID 0007، سجل المراجعة (النقل إلى Rust، والدمج؛ والتكامل مع البرنامج الخدمي)، ويسجل 719,406 بايت قبل حقول JA4 التي أضافها SID 0014. والحجم في الإصدار 1.0.0-beta.3 هو حجم الوحدة مبنيةً كما يبنيها سكربت البناء الخاص بالبرنامج الخدمي (`cargo rustc --lib -p sibuna-console-ui --target wasm32-unknown-unknown --profile wasm -- -C link-arg=--compress-relocations`)، مضغوطةً بـ `gzip -9`.

8 التقييم التجريبي

8.1 المنهجية

تقيس أداء Sibuna ثلاث أدوات قياس تُشغَّل عبر `cargo xtask bench`. تقيس مجموعة العمليات الأولية زمن عمليات مسار الطلب في سبع دفعات مؤقَّتة بعد إحماء غير مؤقَّت، وتعيد بناء الحالة القابلة للتغيير خارج المؤقِّت، وتعرض الدفعة الوسيطة لكل عملية مع أسرع دفعة وأبطئها بجانبها. وتشغّل أداة المنتج الكامل البرنامج الخدمي بـ `wrk` (خيطان، و64 اتصالًا، و5 ثوانٍ، و3 تكرارات) في وضعي forward auth والوكيل العكسي، على الوضعين Gate وShield، عبر أربعة أعباء عمل: `admitted` (جلسة صالحة)، و`challenged` (بلا جلسة)، و`allowed_static` (`/robots.txt`)، و `attack` (جلسة مع `q=' OR 1=1--`، لا يرفضها إلا Shield). أما أداة زمن الاستجابة فتستخدم مولّدًا مفتوح الحلقة يرسل بمعدل ثابت أيًّا كان ما يفعله البرنامج الخدمي، ويقيس زمن كل طلب من موعد استحقاقه، فيُحسب الانتظار في طوابير البرنامج الخدمي على البرنامج الخدمي نفسه[23]. سُجلت التشغيلات في 15 أكتوبر 2026 على مضيف واحد: معالج AMD Ryzen 9 5950X (16 نواة، و32 خيطًا) يعمل بنظام Linux 7.0.0-28. شُغّل البرنامج الخدمي على وحدات المعالجة من 0 إلى 3 بأربعة عمال، ومولّد الحمل وخادم أصلي بديل (stub) على وحدات المعالجة من 8 إلى 15 (على CCD الآخر)، ومجموعة العمليات الأولية على وحدة المعالجة 4. وكان البرنامج الخدمي بناء الإصدار الافتراضي، مع تضمين SQLite ووحدة التحكم، وعمل في الذاكرة دون تخزين.

نطاق القياس. يستخدم كل تشغيل مضيفًا واحدًا عبر واجهة الاسترجاع (loopback): بلا TLS، ولا شبكة واسعة، ولا متصفحات حقيقية. لا تعدّ أدوات القياس عمليات التخصيص؛ بل تتحقق اختبارات الحالة المستقرة من خلو مسار الطلب من التخصيص. وهي تشغّل الملف التنفيذي للإصدار دون تعديل، مُعرَّفًا بتجزئة SHA-256 الخاصة به. والترتيب على هذا المضيف ليس ترتيبًا على غيره.

8.2 أزمنة العمليات الأولية

الجدول 5 أزمنة العمليات الأولية في Sibuna بالنانوثانية لكل عملية على نواة واحدة من Ryzen 9 5950X: وسيط سبع دفعات، مع أسرع دفعة وأبطئها (`benchmarks/results/primitives-latest.md`، أُنشئ في 2026-10-15T22:34:59Z)

النظام الفرعي	عبء العمل	الوسيط	الأدنى-الأقصى
إثبات العمل	تحقق Hashcash، 16 بتًا	70.2	69.9-71.0
إثبات العمل	تحقق PoSW، العمق 13، و16 عملية فتح	22,272.4	22,257.0-22,316.4
مطابِق الروبوتات	Aho-Corasick، 40 توقيعًا	55.7	55.3-57.2
مطابِق الروبوتات	بحث تسلسلي عن السلاسل الجزئية، 40 توقيعًا (للمقارنة)	763.0	762.2-763.5
مصنِّف IP	بحث CIDR في IPv4	32.0	31.8-32.2
مصنِّف IP	بحث CIDR في IPv6	79.3	78.5-79.6
الرمز	رمز جلسة BLAKE3 ذو مفتاح	155.8	155.3-156.4
الرمز	رمز جلسة Ed25519	40,587.5	40,544.6-40,631.1
مجموعة الحلول المُنفَقة	الإنفاق والبحث في جدول Robin Hood	21.0	20.8-21.3
محدِّد المعدل	GCRA، عميل مُعرَّف بتجزئة	6.2	6.1-6.3
محدِّد المعدل	GCRA، أربعة نطاقات قواعد، عميل مُعرَّف بتجزئة	6.3	6.2-6.4
محدِّد المعدل	GCRA، العميل بايتاتٍ خامًا (تشمل التجزئة ذات المفتاح)	12.8	12.7-12.8
محدِّد المعدل	GCRA، أربعة نطاقات قواعد، العميل بايتاتٍ خامًا	13.1	13.0-13.2
محلل HTTP	طلب وملف تعريف ارتباط بلا نسخ	132.5	132.3-133.2
محرك السياسات	تصنيف كامل، بإعدادات Shield	790.8	787.2-793.1
محرك السياسات	التصنيف، بإعدادات Gate	75.9	75.6-76.1
WAF	جسم 8 KiB، فحص دلالي	12,784.3	12,771.5-12,809.2
WAF	جسم 8 KiB، حقن SQL في وضع التدقيق	15,485.4	15,460.9-15,530.1

[23] `benchmarks/results/README.md`، "Method" و"Provenance and limits" (rustc 1.99.0؛ والبرنامج الخدمي مبني بـ `cargo build --release -p sibuna`، وحجمه 49,113,856 بايت)؛ الكتاب، الجزء الثامن، "المنهجية".

التشتت ضيق: لا تبعد أي دفعة في أي صف أكثر من 4.2% عن وسيط صفها[24]. يستخدم البرنامج الخدمي صفوف العميل المُعرَّف بتجزئة، لأنه يجزّئ العميل مرة واحدة لكل طلب؛ أما الصفوف التي تمرر البايتات الخام فتشمل تلك التجزئة ذات المفتاح في كل فحص. وتحيط المؤقتات بدفعات كاملة، لا بعمليات منفردة، لذا لا يعرض الجدول أي مئينات لكل عملية.

8.3 المنتجات الكاملة

الجدول 6 Sibuna تحت wrk: الطلبات في الثانية، وزمن الاستجابة بالميكروثانية، وزمن CPU لكل طلب بالميكروثانية، وذروة الذاكرة المقيمة؛ وسطاء ثلاثة تشغيلات (benchmarks/results/product-latest.md، أُنشئ في 2026-10-15T23:15:24Z)

وضع النشر	الوضع	عبء العمل (الحالة)	req/s	p50	p99	CPU	ذروة KiB
forward auth	Gate	(200) admitted	211,345	151	305	11.84	15,912
forward auth	Gate	(401) challenged	198,315	160	326	13.15	16,176
forward auth	Gate	(200) allowed_static	211,053	152	305	11.73	16,176
forward auth	Gate	(200) attack	211,618	151	305	11.84	16,176
forward auth	Shield	(200) admitted	208,101	152	308	12.87	15,928
forward auth	Shield	(401) challenged	197,532	161	327	13.77	16,192
forward auth	Shield	(200) allowed_static	211,636	151	305	11.78	16,192
forward auth	Shield	(403) attack	210,549	152	307	12.32	16,192
الوكيل العكسي	Gate	(200) admitted	142,498	427	768	27.68	15,896
الوكيل العكسي	Gate	(200) challenged	199,465	160	324	12.98	16,160
الوكيل العكسي	Gate	(200) allowed_static	147,303	411	759	26.88	16,324
الوكيل العكسي	Gate	(200) attack	143,900	422	745	27.67	16,348
الوكيل العكسي	Shield	(200) admitted	138,965	438	857	28.57	16,064
الوكيل العكسي	Shield	(200) challenged	195,188	162	328	13.73	16,328
الوكيل العكسي	Shield	(200) allowed_static	145,673	415	746	27.27	16,328
الوكيل العكسي	Shield	(403) attack	209,507	151	306	12.17	16,328

في وضع forward auth يسأل وكيل الدخول Sibuna هل يقبل الطلب، ولا يوجد خادم أصلي؛ وتقع جميع أعباء العمل بين 197,532 و211,636 طلبًا في الثانية مع p99 من 305 إلى 327 us. ولا تكون وحدات المعالجة الأربع للبرنامج الخدمي مشبعة هناك (ينشغل منها ما بين 2.47 و2.72)، مما يشير إلى أن مولّد الحمل هو السقف. وفي وضع الوكيل العكسي تتضمن صفوف الطلبات المقبولة والملفات الثابتة الخادم الأصلي البديل وقفزة loopback إضافية، وتكون وحدات المعالجة للبرنامج الخدمي مشبعة (ينشغل منها 3.95 إلى 3.98 من أربع)؛ أما الطلبات التي يجيب عنها Sibuna بنفسه (التحديات، ورفض Shield للهجوم) فتظل تعمل بنحو 195,000 إلى 210,000 طلب في الثانية. وتبقى ذروة الذاكرة المقيمة بين 15,896 و16,348 KiB في كل عبء عمل[25].

كانت بضع إجابات من نحو مليوني إجابة في ثلاثة صفوف للوكيل العكسي 503: فقد أعاد wrk الاتصال قبل أن يعيد اتصالٌ يُغلَق مكانه ضمن حد الاتصالات لكل عنوان. وصار البرنامج الخدمي الآن يعيد المكان قبل أن يكتب استجابة الإغلاق؛ وهذه الملفات أقدم من ذلك الإصلاح.

يُسقط تغيير لاحق في ملف تعريف الإصدار (release profile) معلومات التصحيح لمكتبة SQLite المضمنة، وهي نحو 13 MB من أصل 49 MB وفق الملاحظة الواردة في بيان البناء؛ ولم تُعَد ملفات النتائج منذ ذلك الحين، لذا نذكر الحجم المقيس[26].

8.4 زمن الاستجابة عند حمل معروضٍ متساوٍ

يُبقي wrk 64 طلبًا قيد التنفيذ، فيتوقف معدل الطلبات على البرنامج الخدمي، وتُسقط مئينات wrk الطلبات التي لم يرسلها أثناء انتظاره. أما الأداة مفتوحة الحلقة فتثبّت المعدل المعروض بدلًا من ذلك وتحسب كل تأخير على البرنامج الخدمي، وهذا ما يختبره الزائر فعلًا.

في وضع forward auth يبقى p99.9 دون 75 us حتى 150,000 طلب في الثانية. وفي وضع الوكيل العكسي، مع وجود الخادم الأصلي في المسار، يبلغ p99 قيمة 100 us وp99.9 قيمة 150 us عند 100,000 طلب في الثانية. وعند 150,000 طلب في الثانية في وضع الوكيل العكسي يتجاوز البرنامج الخدمي سعته: فهو ينجز 134,344 طلبًا في الثانية، قريبًا من 138,965 التي قاسها wrk لعبء العمل نفسه، ويرتفع زمن الاستجابة

[24] الكتاب، الجزء الثامن، "أزمنة العمليات الأولية"، محسوبة من primitives-latest.json.

[25] benchmarks/results/product-latest.md؛ ووحدات CPU المشغولة من cores_busy_median في product-latest.json؛ والإجابات 503 وإصلاحها من benchmarks/results/README.md ومن SID 0002، ملاحظة المراجعة (إصلاح علة).

[26] الكتاب، الجزء الثامن، "الحجم والذاكرة"؛ والملف Cargo.toml في الجذر، الملف الشخصي [profile.release.package.libsqlite3-sys].

الجدول 7 حجم البرنامج الخدمي لـ Sibuna واستهلاكه للذاكرة (product-latest.json وprimitives-latest.json، 15 أكتوبر 2026، كما في جداول الجزء الثامن من الكتاب)

المقياس	القيمة
الملف التنفيذي للبرنامج الخدمي، البناء الافتراضي مع SQLite ووحدة التحكم	49,113,856 بايت
الذاكرة المقيمة في الخمول، تشغيلات المنتج	5,592 إلى 5,788 KiB
ذروة الذاكرة المقيمة أثناء عبء عمل	15,896 إلى 16,348 KiB
الذاكرة المقيمة بعد أعباء العمل	16,176 إلى 16,348 KiB
جداول محرك السياسات	209,052 بايت
جدول توقيعات الروبوتات	42,244 بايت
جدول توقيعات WAF	164,908 بايت
المُبرهِن في المتصفح (WebAssembly)	14,744 بايت

الجدول 8 Sibuna، Shield، طلبات مقبولة بمعدلات معروضة ثابتة؛ زمن الاستجابة بالميكروثانية وزمن CPU لكل طلب، وسطاء ثلاثة تشغيلات مدة كل منها 10 ثوانٍ (benchmarks/results/latency-latest.md، أُنشئ في 2026-10-15T23:07:14Z)

وضع النشر	req/s المعروضة	req/s المحققة	p50	p90	p99	p99.9	الأخطاء	CPU us
forward auth	10,000	10,000	30	32	37	58	0	16.92
forward auth	25,000	25,000	28	31	34	45	0	15.83
forward auth	50,000	50,000	25	27	30	40	0	13.83
forward auth	100,000	99,998	24	26	41	62	0	13.48
forward auth	150,000	149,996	25	36	51	74	0	14.16
الوكيل العكسي	10,000	10,000	56	61	65	76	0	32.25
الوكيل العكسي	25,000	25,000	51	54	58	72	0	28.63
الوكيل العكسي	50,000	50,000	47	50	60	74	0	27.45
الوكيل العكسي	100,000	99,996	55	75	100	150	0	28.42
الوكيل العكسي	150,000	134,344	583,934	1,544,160	1,906,273	1,967,134	4	29.22

إلى ثوانٍ، كما ينبغي أن يُظهر اختبار مفتوح الحلقة فوق السعة. ووقعت الأخطاء الأربعة في ذلك الصف أثناء الحمل الزائد[27].

8.5 ما الذي غيّره القياس

غيّرت أداة قياس زمن الاستجابة تصميم خيوط الاتصال. ففي تشغيلاتها الأولى، مع 64 اتصالًا يغلق البرنامج الخدمي كلًّا منها بعد 4,096 طلبًا، كان المئين 99.9 عند 100,000 طلب مقبول في الثانية بين 0.3 و2 ms، وهو ذيل لم يُظهره wrk. فإنشاء الخيط وإنهاؤه يربط (map) مكدس إشاراته ثم يفك ربطه ويعيد صفحات مكدسه، وتلك التغييرات في فضاء العناوين كانت توقف الخيوط الأخرى. وحسمت تجربة واحدة التشخيص: بناءً لا يغلق الاتصالات أبدًا، ومن ثم لا ينشئ أي خيط أثناء التشغيل، أعطى 54 إلى 60 us عند المعدل نفسه. ويتيح الإصلاح للخيط الذي ينهي اتصاله أن ينتظر متوقفًا ويأخذ الاتصال التالي؛ والخيط الذي يبقى متوقفًا 10 ثوانٍ ينتهي، فلا تترك الدفعة المفاجئة مكدساتها مقيمة في الذاكرة. وبعد الإصلاح صار p99.9 عند 100,000 طلب في الثانية 62 us في وضع forward auth و150 us في وضع الوكيل العكسي[28].

يحرّك التوقف النادر الدوري الذيلَ البعيد قبل أن يحرك p99 بزمن طويل. فعند 100,000 طلب في الثانية عبر 64 اتصالًا، يدوم كل اتصال 4096/1562.5 ≈ 2.6 s، فيبدأ نحو 24 خيطًا وينتهي كل ثانية؛ وفي تشغيل من مليون طلب لا تضم أعلى 0.1% سوى 1,000 طلب، وتكفي بضعة طلبات متأخرة عند كل تبديل للخيوط لملئها. ومثل هذا الخلل لا يُرى إلا بالمئين p99.9 وبمؤقّت على كل طلب.

8.6 أبرز الخلاصات

1. تصنيف في أقل من ميكروثانية. بإعدادات Gate يصنّف محرك السياسات طلب متصفح في 75.9 ns؛ ومع الفحص الكامل بإعدادات Shield يستغرق 790.8 ns.
2. تحقق متناظر. يُتحقق من رمز جلسة BLAKE3 ذي مفتاح في 155.8 ns، مقابل 40,587.5 ns لرمز Ed25519: أي عمل أقل بنحو 260 مرة لقاء سلامة البيانات نفسها ضمن نطاق ثقة واحد.

[27] benchmarks/results/latency-latest.md؛ والصفوف عند 25,000 طلب في الثانية وصفوف الوكيل العكسي عند 150,000 موجودة في ذلك الملف.

[28] SID 0002، ملاحظة المراجعة (الأداء، وإعادة استخدام خيوط الاتصال)؛ الكتاب، الجزء الثامن، "ما الذي غيّره القياس"؛ benchmarks/results/latency-latest.md.

3. أثر صغير ومستقر على الموارد. نحو 200,000 طلب forward auth في الثانية على أربع وحدات معالجة، مع ذروة ذاكرة مقيمة قرب 16 MiB في كل عبء عمل ودون 6 MiB في الخمول.
4. الذيول تحتاج إلى قياس مفتوح الحلقة. كان توقف تبديل الخيوط غير مرئي لأداة القياس مغلقة الحلقة، وواضحًا عند p99.9 تحت مولّد مفتوح الحلقة.

9 مقارنة سوقية: Sibuna ونظراؤه

نقارن Sibuna بالأدوات التي يُرجَّح أن يوازن المشغّل بينها وبينه. قُرئت الحقائق عن المنتجات الأخرى من وثائقها العامة وملفات إصداراتها في سبتمبر 2026، ويسجلها كتاب Sibuna مع مصادرها في الجزء الثاني[31]. يخدم كل من هذه المنتجات مستخدميه جيدًا، وقد اتخذ كل منها خيارات تصميمية لا تقيس أداتنا فوائدها. ونحن نصف الفروق في البنية، لا ترتيبًا للمنتجات.

9.1 مقارنة مقيسة مع Anubis

قسنا الملف التنفيذي الرسمي لإصدار Anubis v1.27.0 بجانب Sibuna بأداة القياس في المنتجات الكاملة: أعباء العمل الأربعة نفسها، ووحدات المعالجة الأربع نفسها لكل خيط في كل منتج، وحمل `wrk` نفسه، وعمل Hashcash متكافئ (8 بتات صفرية لـ Sibuna، ورقمان ست عشريان صفريان لـ Anubis)، وجلسة يُحصل عليها بحل تحدي كل منتج كما يفعل المتصفح، و`caddy respond` خادمًا أصليًا[29]. يعمل Sibuna هنا بوضع Shield، فهو يفحص كذلك كل حمولة، وهو ما لا يفعله Anubis.

الجدول 9 مقارنة الميزات (الجزء الثاني من الكتاب؛ قُرئت الحقائق عن المنتجات الأخرى من وثائقها العامة في سبتمبر 2026، وصف JA4 في أكتوبر 2026)

الميزة	Sibuna 1.0.0-beta.3	Anubis 1.27	SafeLine CE 9	Cloudflare WAF
طريقة التشغيل	ملف تنفيذي واحد؛ وكيل عكسي أو forward auth؛ ملف SQLite اختياري أو قاعدة بيانات PostgreSQL مشتركة	ملف تنفيذي واحد بلغة Go؛ وكيل عكسي أو forward auth	سبع حاويات Docker (tengine، وdetector، وmgt، وluigi، وfvm، وchaos، وPostgreSQL)	شبكة مستضافة؛ يشير DNS إلى Cloudflare
بيئة التشغيل	شفرة أصلية بلا جامع نفايات؛ بلا أي اعتماد على طرف ثالث حين يكون التخزين معطلًا	Go، مع جمع النفايات	وكيل وكاشف بلغة C/C++، وإدارة بلغة Go، وPostgreSQL	برمجيات حافة مملوكة
الرخصة	LGPL-3.0-or-later	MIT	شفرة إدارة بترخيص GPL-3.0؛ صور كاشف مغلقة	خدمة مملوكة
القبول بإثبات العمل	Hashcash (على مستوى البتات) والعمل المتسلسل لـ Cohen-Pietrzak، يُختار لكل قاعدة	Hashcash بـ SHA-256 بأنصاف بايت ست عشرية؛ تحديات بلا عمل	تحدي JavaScript وCAPTCHA؛ غير مشروط بالعمل	Managed challenge، وJS challenge، وTurnstile
حالة التحدي على الخادم	لا شيء حتى الحل؛ الوسوم المحلولة في جدول Robin Hood ثابت السعة	خلفية التخزين: الذاكرة، أو bbolt، أو Valkey، أو S3	تديرها المجموعة البرمجية	تديرها Cloudflare
رمز الجلسة	BLAKE3 ذو مفتاح، 75 محرفًا، مرتبط بالعمل المدفوع؛ Ed25519 اختياري	JSON Web Token موقّع بـ Ed25519؛ HS512 اختياري	ملف تعريف ارتباط تصدره المجموعة البرمجية	ملف تعريف الارتباط `cf_clearance`
فحص الحمولة	Shield: آلة موسومة مع مقسّمات بنيوية، أول 8 KiB من الجسم	لا يوجد	تحليل دلالي عبر فئات OWASP	مجموعات قواعد مُدارة (Pro وما فوق)
بصمة TLS (JA4)	تطابق القواعد قيمة JA4 يمررها وكيل دخول موثوق؛ لا يحسبها Sibuna	غير موثق؛ قد تشير السياسات إلى بصمة HTTP من نوع JA4H، التي يحسبها Anubis	تُحسب عند إنهاء TLS الخاص به؛ قواعد مخصصة وقائمة رفض مدمجة	JA3 وJA4 في القواعد المخصصة والتحليلات (Enterprise مع Bot Management)
الحالة المشتركة	قاعدة بيانات PostgreSQL واحدة للسياسات والسمعة والأدلة الجنائية	مفتاح توقيع مشترك؛ مخزن Valkey مشترك	مجموعة برمجية واحدة لكل مضيف	شبكة anycast عالمية
إنهاء TLS	لا (وكيل الدخول)	لا (وكيل الدخول)	نعم	نعم

[31] الكتاب، الجزء الثاني، "مقارنة الميزات" و"الصفحات الوسيطة وشبكات الحافة"؛ وترخيص Sibuna هو قيمة `license` لمساحة العمل في `Cargo.toml`.

[29] `benchmarks/results/comparison-latest.md` و`admission-latest.md`، سُجّلا في 16 أكتوبر 2026 بالأمرين `cargo xtask bench product --anubis` و`cargo xtask bench admission --anubis`؛ ويسرد الجزء الثامن من الكتاب، "Sibuna وAnubis"، الطريقة وكل اختلاف بين المنتجين.

الجدول 10 Sibuna (Shield) و Anubis 1.27.0 تحت wrk: الطلبات في الثانية وزمن الاستجابة p99 بالميكروثانية، وهي وسيط ثلاث جولات مدة كل منها خمس ثوانٍ (benchmarks/results/comparison-latest.md). رمز الحالة 200 ما لم يُذكر غيره؛ ويرفض Shield الهجوم بالرمز 403.

وضع النشر	عبء العمل	Sibuna (طلب/ث)	Sibuna (p99)	Anubis (طلب/ث)	Anubis (p99)
forward auth	admitted	208,635	308	36,509	7,477
forward auth	challenged	193,087	342	113,170	3,781
forward auth	allowed_static	212,444	304	89,904	4,547
forward auth	attack	211,347 (403)	306	36,853	7,283
الوكيل العكسي	admitted	103,298	3,321	19,398	11,062
الوكيل العكسي	challenged	197,327	328	28,842	9,546
الوكيل العكسي	allowed_static	106,819	3,316	28,862	30,124
الوكيل العكسي	attack	210,483 (403)	306	23,399	14,124

قراءة هذه الصفوف. تصف هذه الأرقام أربعة أعباء عمل على جهاز واحد، لا كل نشر، وعدة صفوف ليست مقارنة بين متماثلين. ففي وضع forward auth يجيب Anubis بنفسه عن /robots.txt وعن مسار الهجوم؛ ولا يفحص Anubis أي حمولة، فيؤدي رفض Shield عملًا أكبر؛ ويتطلب بدء التحدي طلبين في Sibuna وصفحة واحدة في Anubis؛ ويفسر الرمز المتناظر جزءًا كبيرًا من فجوة فحص الجلسة، إذ إن مخطط HS512 الخاص بـ Anubis أسرع بنسبة 55% من مخططه الافتراضي Ed25519. وAnubis أداة واسعة الاستخدام وقادرة، وقد اتخذ القائمون عليها خيارات مختلفة لأسباب وجيهة. أما ما يُظهره القياس فعلًا فهو أن مسار طلب صغيرًا ومحدودًا يمكن أن يجعل القبول رخيصًا على المدافع.

ينفق Anubis من وقت المعالج 106 us لكل طلب مقبول في وضع forward auth، مقابل 13 us لـ Sibuna؛ وبلغت ذروة ذاكرته المقيمة نحو 32 MiB في وضع forward auth، ونمت إلى نحو 600 MiB في وضع الوكيل العكسي، في حين بقيت ذاكرة Sibuna قرب 16 MiB طوال القياس. وخلف caddy respond تحدّ قدرةُ الخادم الأصلي صفوفَ Sibuna التي يمرر فيها الطلبات، لأنه يتقاسم وحدات CPU مع مولّد الحمل: فمع الخادم الأصلي البديل الأخف الموصوف في المنتجات الكاملة تبلغ الصفوف نفسها نحو 139,000 طلب في الثانية.

9.2 مقارنة معمارية مفصلة

SafeLine (Chaitin Technology) 9.2.1

ينقل SafeLine الفكرة الدلالية التي قام عليها libinjection إلى منتج متكامل مزود بوحدة تحكم إدارية، ويذكر مورّده معدلات كشف قدرها 71.65% (في الوضع المتوازن) و76.17% (في الوضع الصارم) مع معدلات إنذار كاذب قدرها 0.07% و0.22%[32]. هذه قياسات لجودة الكشف، ولم ينشر Sibuna قياسات مثلها لمحركه.

- النشر. يُشحن الإصدار المجتمعي في سبع حاويات Docker على مضيف Linux فيه نواة معالج واحدة على الأقل، و1 GB من الذاكرة، و5 GB من القرص. وهو ينهي TLS بنفسه، وهو ما لا يفعله Sibuna.
- الحالة. يحفظ سجل الهجمات في PostgreSQL داخل الحزمة نفسها، بحزمة واحدة لكل مضيف.
- ما يختلف فيه Sibuna. *Sibuna* ملف تنفيذي واحد يبدأ بملف SQLite اختياري، ويتشارك العنقود قاعدة بيانات PostgreSQL واحدة قد يكون المشغّل يديرها أصلًا. وتراوحت ذاكرته المقيمة في حال الخمول بين 5,592 و5,788 KiB في جولات قياس المنتج. لم نشغّل SafeLine على مضيفنا، إذ لم يكن عليه برنامج Docker الخدمي، فلا نقدم مقارنة للأداء.

Anubis وCoraza 9.2.2

شاعت الصفحة الوسيطة القائمة على إثبات العمل في عام 2025، حين جعلت حركة الزواحف التي تجمع البيانات لتدريب النماذج اللغوية تشغيلَ الخوادم الأصلية العادية مكلفًا، وAnubis أشهر تطبيقاتها المفتوحة. وهو وكيل عكسي مكتوب بلغة Go، قدير وواسع الانتشار: يقدم لغزًا بلغة

الجدول 11 عمليات القبول في الثانية على وحدتي CPU، وهي وسيط سبع دفعات (benchmarks/results/admission-latest.md). يُشغَّل Anubis مرة بكلّ من مخطّطي الرموز لديه.

المنتج (الرمز)	فحص الجلسة	فحص غير موثق	تمهيد التحدي	التحقق من الإثبات	الذاكرة المقيمة
Sibuna (BLAKE3 ذو مفتاح)	88,104	78,997	42,664	57,924	MiB 7.6
Anubis (Ed25519)	10,951	28,619	7,893	12,668	MiB 32.6
Anubis (HS512)	16,976	27,785	7,952	19,261	MiB 29.1

[32] الكتاب، الجزء الثامن، "ما لم يُقس": أرقام أعلنها المورّد وتُحقق منها في سبتمبر 2026.

JavaScript، ويوقّع رمز JSON Web Token بخوارزمية Ed25519، ويحفظ حالة التحديات في مخزن. أما Coraza فينفذ لغة قواعد ModSecurity (SecLang) ويشغّل مجموعة OWASP Core Rule Set؛ وهو إحدى طرق إضافة فحص قائم على القواعد إلى جانب بوابة كهذه.

- أين يذهب العمل. يجعل Anubis وSibuna Gate كلاهما العميلَ يدفع قبل أن يفعل الخادم الأصلي أي شيء. ويضيف Sibuna مستوى ثانيًا من العمل المتسلسل، يُختار لكل قاعدة، للمسارات التي تستحق بوابة أقوى، وتتذكر جلساته مقدار العمل المدفوع.
- ما يتذكره الخادم. يخزن Anubis التحديات؛ أما Sibuna فيصدر سجلًا يصادق على نفسه، ولا يتذكر إلا وسوم الحلول، في جدول ثابت الحجم (المبرهنة 3). ورمزه الافتراضي متماثل، فالتحقق منه أرخص، ولا يكسب منه المهاجم الكمّي إلا تسريعًا تربيعيًا.
- الفحص. يجري Sibuna Shield فحصه البنيوي الخاص في المرور نفسه الذي يجري فيه القبول، فتخدم عملية واحدة ومحلل واحد الدورين معًا. ولا يشغّل مجموعة OWASP Core Rule Set. قيّمنا محركات SecLang القائمة، ومنها الارتباطات بـ libmodsecurity وCoraza، ولم نقبل أيًا منها: فمحرك الفحص على خادم الحافة شيفرة حرجة أمنيًا نفضّل أن نملكها ونراجعها بأنفسنا. وإذا أضاف Sibuna توافقًا مع Core Rule Set، فسيحدد سجل تصميم المجموعة الجزئية المدعومة أولًا[33].
- القياس. على مضيف واحد وتحت الحمل نفسه، أجاب Sibuna في وضع forward auth عن أكثر من خمسة أضعاف ما أجاب عنه Anubis من الطلبات المقبولة، بزمن p99 أقل ونحو ثُمن وقت المعالج لكل طلب (مقارنة مقيسة مع Anubis).

9.2.3 جدران WAF السحابية على الحافة (Cloudflare وغيرها)

تضع شبكات الحافة المستضافة الفحصَ وتحديد المعدل وتقييم الروبوتات في نقاط حضور كثيرة تديرها مستوى تحكم عالمي. وهي تمتص الهجمات الحجمية التي لا يقدر عليها أي خادم منفرد، ولا تحتاج من المشغّل إلى أي عتاد. ويستعير نشر Sibuna العنقودي شكلها، أي مستوى سياسات مشتركًا يغذي لقطات محلية غير قابلة للتغيير، مع بقائه مستضافًا ذاتيًا.

- إقامة البيانات. ينهي WAF المستضاف اتصالات TLS الخاصة بالمشغّل، ويرى حمولات الطلبات على بنية تحتية مشتركة، وقد يتعارض ذلك مع متطلبات إقامة البيانات.
- الخطط والحصص. ترتبط الميزات بالخطط: تسمح حصص Cloudflare المنشورة بـ 5 و20 و100 و1,000 قاعدة مخصصة، وبـ 1 و2 و5 و100 قاعدة لتحديد المعدل، في خطط Free وPro وBusiness وEnterprise على الترتيب.
- ما يختلف فيه *Sibuna*. يعمل Sibuna على أجهزة المشغّل نفسه دون رسوم على كل طلب، وتصل تغييرات سياسته إلى كل عقدة في نبضة التخزين التالية (500 ms افتراضيًا). وهو لا يخفف الهجمات الشبكية الحجمية، ولا يقيّم الروبوتات بنموذج مدرَّب، ولا ينهي TLS؛ والنشر الذي يحتاج إلى ذلك يظل محتاجًا إلى وكيل دخول أو شبكة حافة أمامه.

10 التنفيذ: حزم صغيرة واعتماديات قليلة

كُتب Sibuna بلغة Rust، على هيئة مساحة عمل من حزم صغيرة تتجه اعتمادياتها في اتجاه واحد: المكتبات القابلة لإعادة الاستخدام، ثم نواة للإعدادات والتشخيص، ثم مكتبات التطبيق، ثم البرنامج الخدمي ومُبرهِن المتصفح. وتهم المشغّلَ ثلاثةُ خيارات هندسية، فنسجلها باختصار[34].

اعتماديات قليلة. نعامل كل مكتبة من طرف خارجي على أنها عبء: يجب أن تُراجَع وتُحدَّث ويُوثَق بها. ولا يستخدم مستوى البيانات أيًا منها. فالتجزئة والتواقيع وbase64 وJSON وDEFLATE وAho-Corasick ومحلل HTTP/1.1 مكتبات صغيرة من صنعنا، يمكن لمشاريع أخرى استخدامها مستقلة. ولا تُقبل اعتمادية إلا إذا كانت ناضجة ومكتفية بذاتها وتحل مشكلة كبيرة حقًا، ويُكتب سبب قبولها بجانبها في ملف تعريف البناء وفي سجل تصميم. وقد قُبلت اثنتان، كل منهما محصورة في مكتبة واحدة: `sqlx` لتخزين SQLite وPostgreSQL، وتجلب نحو مئة حزمة متعدية[35]، و`rustls` لاتصالات TLS الصادرة من وحدة التحكم ولاتصال TLS مع PostgreSQL، وهي ثلاث عشرة حزمة وقت التشغيل، لأن عميل TLS مكتوبًا داخليًا سيكون مشروعًا حرجًا أمنيًا أكبر من مستوى بيانات Sibuna كله[36]. والبناء الذي لا يضم التخزين ولا وحدة التحكم لا يعتمد على أي طرف خارجي إطلاقًا.

أمان الذاكرة. الشيفرة التي تتجاوز فحوص أمان الذاكرة في المترجم مرفوضة افتراضيًا، والوحدة التي تحتاج إليها يجب أن تطلبها صراحةً باسمها وأن تبرر كل استخدام لها في تعليق. وبعض الوحدات تفعل ذلك: المسارات التي تستخدم تعليمات SHA في المعالج، وعملية كتابة واحدة تمحو الأسرار، وتصريحات بضع دوال من مكتبة C لمعالجة الإشارات، وصادرات وحدتي WebAssembly، والحلقة التي تنقل الحوادث إلى خيط التخزين. والحلقة هي الموضع الوحيد الذي أبقينا فيه شيفرة كهذه من أجل السرعة، وبعد قياس فقط: إذ كلّفت القناة المحدودة في المكتبة القياسية أكثر

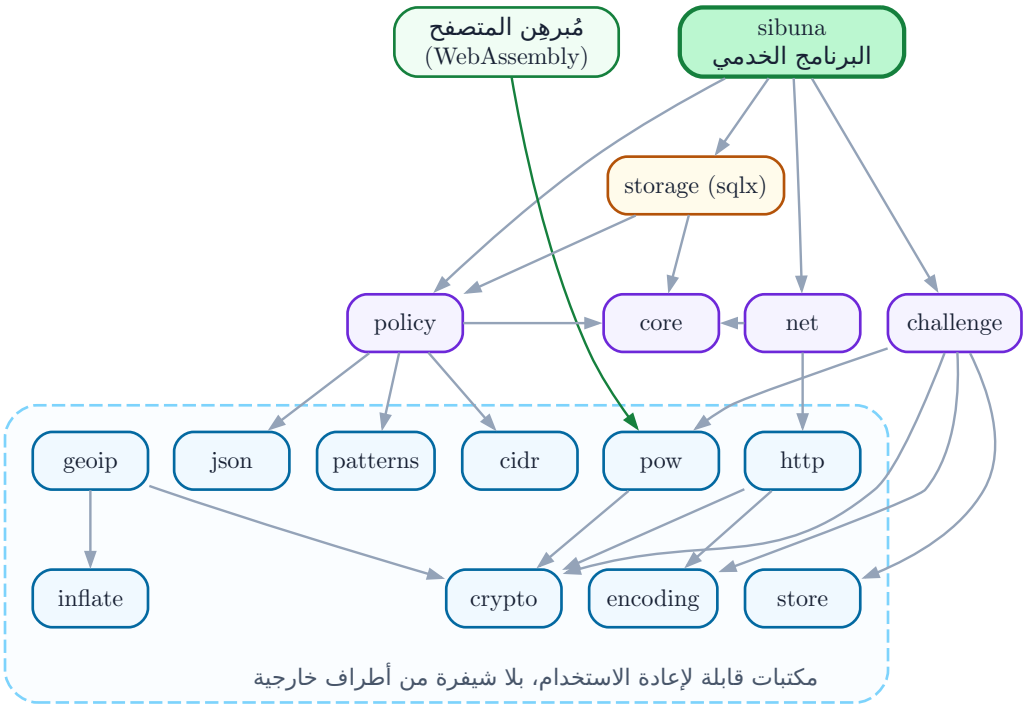
[33] "The OWASP Core Rule Set"، SID 0010.

[34] "Workspace"، SID 0010؛ و"Dependency policy" و"Engineering standards".

[35] "Why sqlx"، SID 0011؛ و`Cargo.toml`، حيث تحمل كل حزمة من طرف خارجي سبب قبولها.

[36] "Specification"، SID 0012؛ و"Rationale and alternatives" وسجل المراجعة (بناء TLS لـ PostgreSQL).

[37] SID 0010، سجل المراجعة (المراجعة الأمنية، 2026-10-15): القناة المحدودة القياسية مقابل حلقة Vyukov، `tests/timing.rs`. والوحدات التي تطلب ذلك صراحةً هي `crates/crypto/src/sha256/` (x86-64 وAArch64)، و`crates/crypto/src/ct.rs`، و`cmd/sibuna/src/signals.rs`، و`cmd/sibuna-pow-wasm/src/lib.rs`، و`cmd/console-ui/src/abi.rs`، و`crates/store/src/ring.rs`؛ وتضبط مساحة العمل `unsafe_code = "deny"` و`undocumented_unsafe_blocks = "deny"` في Clippy.



الشكل 6 مكتبات Sibuna مبسطة (عن SID 0010). تتجه الأسهم من المكتبة إلى ما تستخدمه. لا تعتمد المكتبات القابلة لإعادة الاستخدام على أي مكتبة من مكتبات التطبيق، ولا تجلب شيفرة من أطراف خارجية إلا مكتبتا التخزين ووحدة التحكم.

بنسبة 20 إلى 45% لكل إدخال مع 16 منتجًا[37]. وفي ما عدا ذلك تؤدي الملكية العمل الذي كانت إدارة الذاكرة اليدوية ستؤديه: فلا يمكن لطلب محلَّل أن يعيش أطول من مخزنه المؤقت، ويُحرَّر المحرك مع آخر قارئ له، ولا يمكن إغلاق مقبس ما دام أي خيط يستطيع الوصول إليه.

الانضباط والاختبارات. لا يتجاوز جسم الدالة 70 سطرًا منطقيًا، ولا يتجاوز الملف 1,408 أسطر، ويفرض ذلك أداة فحص. تعيد شيفرة المكتبات أخطاء منظمة ولا تطبع شيئًا أبدًا، وكل خطأ يراه المستخدم يذكر موضعه، ويعرض مقتطفًا، ويقدم تلميحًا. تشغّل مجموعة اختبارات مساحة العمل في الإصدار beta.3-1.0.0 نحو 2,200 اختبار على SQLite ونحو 700 اختبار آخر على PostgreSQL؛ ولواجهة وحدة التحكم 298 اختبارًا، منها تجهيزات مرجعية ذهبية؛ وتقود فحوص المتصفح Chromium بلا واجهة عبر الصفحة الوسيطة وعبر كل صفحة من صفحات وحدة التحكم[30].

11 حدود العمل وما بقي منه

نسرد هنا ما لا تثبته هذه الورقة، كي لا يُفهم أي ادعاء على نطاق أوسع من أدلته.

- منافس واحد مقيس. قسنا Anubis على مضيف واحد (مقارنة مقيسة مع Anubis). ولم نقس SafeLine ولا أي WAF مستضاف.
- لا عنقود تحت الحمل. يشغّل اختبار وظيفي ثلاث عقد على قاعدة بيانات PostgreSQL واحدة (SID 0011، بحالة Committed)، لكن لا توجد أداة قياس تشغّل عدة عقد تحت الحمل.
- يحتاج *PostgreSQL* إلى البناء من المصدر. تتضمن الملفات التنفيذية للإصدار تخزين SQLite ووحدة التحكم، لكنها لا تدعم PostgreSQL؛ ويبني العنقودُ Sibuna بالميزة `postgres-tls`.
- معيار اجتياز العزل في وحدة التحكم، مع ثماني لوحات معلومات نشطة تحت الحمل، لم تُشغَّل بعد (وحدة التحكم).
- واجهة الاسترجاع فقط. لا يغطي أي قياس TLS، ولا شبكة واسعة النطاق، ولا متصفحات حقيقية تحت الحمل؛ وعلى كل نشر أن يقيس مساره هو.
- فحص محدود. Shield فاحص استدلالي لأول 8 KiB من الجسم، وليس محللًا كاملًا لـ SQL أو HTML، وفحص forward auth لا يرى إلا البايتات التي يرسلها وكيل الدخول.
- نطاق البروتوكول. يتحدث Sibuna بروتوكولي HTTP/1.0 وHTTP/1.1؛ ولا يدعم HTTP/2 ولا إنهاء TLS للاتصالات الواردة.

[30] "Acceptance criteria and implementation status", SID 0010 (المعايير 1 و3 و5)، وقد سجّل نحو 2,100 اختبار لمساحة العمل و290 اختبارًا للواجهة قبل SID 0014؛ والأعداد في الإصدار beta.3-1.0.0 مأخوذة من `cargo test --workspace -- --list`.

- بصمات *TLS* تأتي من وكيل الدخول. لا يستطيع Sibuna حساب JA4 بنفسه. والبصمة لا تكون موثوقة إلا بقدر وكيل الدخول الذي يكتبها، وعلى وكيل الدخول هذا أن يستبدل أي نسخة أرسلها العميل. ويمكن للعميل أن يقلد مصافحة متصفح بمكتبات متخصصة، لذا ترفع JA4 تكلفة الأتمتة، لكنها لا تثبت الهوية[11].
- الضمان التشفيري. يقوم مستوى العمل المتسلسل على براهين منشورة في نموذج الأوراكل العشوائي (random oracle)؛ وبروتوكولنا الملموس له اختبارات أمام مولّد تسميات مرجعي، لا تدقيق تشفيري مستقل.
- جهاز واحد. أُخذ كل رقم على جهاز AMD Ryzen 9 5950X واحد. والترتيب على هذا المضيف ليس ترتيبًا على مضيف آخر.

12 الخاتمة

كثيرًا ما نمت حماية الويب بالتراكم: بيئات تشغيل مفسَّرة، وحاويات، ومجموعات قواعد من التعبيرات النمطية، ومخازن خارجية تتكدس أمام التطبيق. وفي Sibuna جربنا الاتجاه المعاكس، مسترشدين ببضع أفكار بسيطة:

- اجعل المرسل يدفع ثمن القبول بعمل يمكن التحقق منه، وأبقِ فحص الخادم رخيصًا وعديم الحالة، ودع الجلسة تتذكر مقدار العمل الذي دُفع؛
- افحص البنية في زمن خطي، في المرور نفسه الذي يجري فيه القبول، كي لا تستطيع أي جلسة تجاوز الفحص؛
- احترم الجهاز: مخزن مؤقت واحد لكل اتصال، ولا تخصيص للذاكرة على مسار الطلب، وجداول محدودة بمواضع تحددها تجزئة ذات مفتاح، وخيوط يُعاد استخدامها بدل إعادة إنشائها؛
- ودع قاعدة بيانات ناضجة تتولى الحالة المشتركة الدائمة، خلف خيط تخزين واحد لا ينتظره مسار الطلب أبدًا.

على جهاز واحد من 16 نواة، مكّنت هذه الخيارات Sibuna من خدمة أكثر من 200,000 طلب مقبول في الثانية في وضع forward auth على أربع وحدات CPU، بنحو 16 MiB من الذاكرة المقيمة تحت ذلك الحمل، وبزمن p99.9 قدره 62 us عند 100,000 طلب في الثانية. هذه أرقام مضيف واحد عبر واجهة الاسترجاع، ونقدمها دليلًا على تماسك التصميم، لا ترتيبًا للمنتجات. لقد أزلنا بروتوكول الإجماع الخاص بنا ونقول ذلك صراحةً؛ وفي المقابل صار النظام أصغر، ولم يعد لمستوى بياناته أي اعتماد على طرف خارجي. ولا يزال الكثير بانتظار القياس، وقد حاولنا أن نقول ما هو بالضبط.

13 المصادر وإعادة الإنتاج

كل رقم في هذه الورقة مأخوذ من ملف نتائج مودَع في مستودع Sibuna، أو من سجل Sibuna Discussion (SID)، أو من كتاب Sibuna، ويذكر كل جدول أو حاشية مصدره. تكتب ملفاتِ النتائج الأوامرُ `cargo xtask bench primitives` و`cargo xtask bench product` و`cargo xtask bench latency`، ويقدم `benchmarks/results/README.md` الأوامر ومجموعات CPU ومصدر كل ملف.

- `benchmarks/results/primitives-latest.md` و`product-latest.md` و`latency-latest.md` وملفات `.json` المقابلة لها؛ و `console-8-dashboards-2026-10-15.md`.
- SID 0002 (المعمارية الأساسية وملاحظات مراجعتها)، وSID 0004 (وضع Shield)، وSID 0006 (الأسس الرياضية)، وSID 0007 (وحدة التحكم)، وSID 0010 (مساحة العمل والحزم وسياسة الاعتماديات)، وSID 0011 (التخزين عبر sqlx)، وSID 0012 (TLS لوحدة التحكم)، وSID 0013 (تخزين وحدة التحكم)، وSID 0014 (بصمات JA4 من وكيل الدخول)، في `docs/sid/records/`.
- كتاب Sibuna، `docs/book/`: الجزء الأول (تكلفة الطلب)، والجزء الثاني (الأعمال السابقة)، والجزء الثالث (التشفير)، والجزء الرابع (البروتوكول)، والجزء الخامس (الهندسة بلا تخصيص للذاكرة)، والجزء السادس (الآلات والجدار الناري الدلالي)، والجزء الثامن (التقييم التجريبي)، والجزء التاسع (التشغيل).
- الأدبيات الأصلية: Back، Hashcash (1997)؛ وCohen وPietrzak، Simple Proofs of Sequential Work (EUROCRYPT 2018)؛ وBlocki وLee وZhou، On the Security of Proofs of Sequential Work in a Post-Quantum World (ITC 2021)؛ وAho وCorasick، Efficient String Matching (1975)؛ وCelis، Robin Hood Hashing (1986)؛ وخوارزمية Generic Cell Rate Algorithm من ATM Forum (1996).

الورقة التقنية لـ Sibuna | نوفمبر 2026 | الإصدار beta.3-1.0.0
المواصفات والشيفرة المصدرية وقياسات الأداء: https://github.com/insanai/sibuna