

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2025 年 6 月 19 日 (19.06.2025)



(10) 国际公布号
WO 2025/124249 A1

- (51) 国际专利分类号:
H04L 9/40 (2022.01)
- (21) 国际申请号: PCT/CN2024/136670
- (22) 国际申请日: 2024 年 12 月 4 日 (04.12.2024)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
202311707378.0 2023年12月13日 (13.12.2023) CN
- (71) 申请人: 天翼云科技有限公司(CHINA TELECOM CLOUD TECHNOLOGY CO., LTD.) [CN/CN]; 中国北京市东城区青龙胡同甲1号、3号2幢2层205-32室 100007 (CN)。
- (72) 发明人: 杨洋(YANG, Yang); 中国北京市东城区青龙胡同甲1号、3号2幢2层205-32室 100007 (CN)。辛晨(XIN, Chen); 中国北京市东城区青龙胡同甲1号、3号2幢2层205-32室 100007 (CN)。吴雷(WU, Lei); 中国北京市东城区青龙胡同甲1号、3号2幢2层205-32室 100007 (CN)。
- (74) 代理人: 华进联合专利商标代理有限公司(ADVANCE CHINA IP LAW OFFICE); 中国广东省广州市天河区珠江东路6号4501房(部位: 自编01-03和08-12单元)(仅限办公用途) 510623 (CN)。

- (81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。
- (84) 指定国(除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:
— 包括国际检索报告(条约第21条(3))。

(54) Title: HYPERSCAN-BASED WEB APPLICATION FIREWALL RULE MATCHING METHOD

(54) 发明名称: 基于Hyperscan的Web应用防火墙规则匹配方法

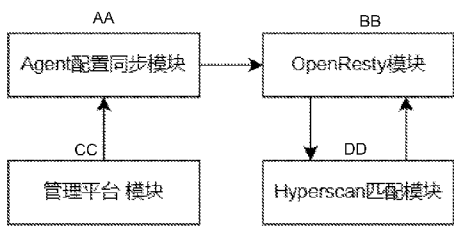


图 3

AA Agent configuration synchronization module
BB OpenResty module
CC Management platform module
DD Hyperscan matching module

(57) Abstract: Disclosed in the present application is a Hyperscan-based Web application firewall rule matching method, mainly comprising the following steps: step 1, an OpenResty server using a Hyperscan regular expression match library for rule matching; step 2, adding an HTTP API interface into a Lua module of the OpenResty; step 3, invoking an interface provided by the Hyperscan to compile regular expressions in the latest configuration, and continuing to invoke the interface to serialize a compiling result into a character string and then write same into a shared memory; step 4, each time when a request arrives, each worker updating the configuration to the current worker on the basis of a semaphore in the shared memory; step 5, storing a deserialized database pointer in the shared memory; step 6, storing in the shared memory a pointer of an allocated scratch memory; step 7, after completing the above steps, each worker retrieving the database pointer and the scratch pointer from the shared memory, and then invoking a scan interface provided by the Hyperscan for scanning and acquiring a matching result; and step 8, on the basis of the matching result, each worker determining whether to release or intercept the current request.

(57) 摘要: 本申请公开了一种基于Hyperscan的Web应用防火墙规则匹配方法, 主要包括以下步骤: 步骤一: OpenResty服务器采用Hyperscan正则匹配库来进行规则匹配; 步骤二: 在OpenResty的Lua模块中新增一个HTTP API接口; 步骤三: 调用Hyperscan提供的接口编译最新配置中的正则表达式, 并继续调用接口将编译结果序列化成字符串后写入共享内存中; 步骤四: 每个worker在每次请求到来时根据共享内存中的信号量来更新配置到当前worker中; 步骤五: 将反序列化后的database指针保存在共享内存中; 步骤六: 将分配到的scratch内存的指针保存在共享内存中; 步骤七: 每个worker在完成以上步骤后从共享内存中取出database指针和scratch指针, 然后调用Hyperscan提供的scan接口进行扫描和获取匹配结果; 步骤八: 每个worker根据匹配结果来判断放行还是拦截当次请求。

基于 Hyperscan 的 Web 应用防火墙规则匹配方法

相关申请

本申请要求 2023 年 12 月 13 日申请的，申请号为 202311707378.0，名称为“一种基于 Hyperscan 的 Web 应用防火墙规则匹配方法”的中国专利申请的优先权，在此将其全文引入作为参考。

技术领域

本申请涉及信息安全技术和计算机技术领域，尤其涉及一种基于 Hyperscan 的 Web 应用防火墙规则匹配方法。

背景技术

WAF，全称 Web Application Firewall，即 Web 应用防火墙，是一种特殊的应用防火墙。它主要用于保护网络应用程序，防止各种类型的攻击，如跨站脚本（XSS）、SQL 注入和跨站点请求伪造（CSRF）等。

WAF 的工作原理是在应用程序和互联网之间建立一个保护层，对所有进出的数据进行检查和过滤。它可以识别并阻止那些可能利用应用程序漏洞进行攻击的恶意数据包。

目前市面上大多数 WAF 会采用正则匹配来命中规则，常常会用 Hyperscan 来提高规则匹配的效率。

Hyperscan 是一种高效的正则表达式匹配库，它可以处理多个正则表达式和大量数据。Hyperscan 使用了一种复杂的有限状态机（Finite State Machine, FSM）和多模式匹配算法，可以在单个扫描过程中匹配多个模式。

Hyperscan 一次完整的匹配流程包括编译期（编译正则表达式）、分配 scratch 空间和运行期（扫描并报告匹配）。而传统的方案是先将正则表达式的编译结果放在 Redis 数据库中，然后在每次请求到来时再从 Redis 数据库中取出来后进行分配 scratch 空间和扫描，由于每次请求都需要读取 Redis 和分配 scratch 空间，并且 scratch 内存分配不一定是高效的操作，这样会导致 WAF 的匹配效率大大降低，限制了 WAF 的 QPS。

发明内容

本部分的目的在于概述本申请的实施例的一些方面以及简要介绍一些较佳实施例。在

本部分以及本申请的说明书摘要和发明名称中可能会做些简化或省略以避免使本部分、说明书摘要和发明名称的目的模糊，而这种简化或省略不能用于限制本申请的范围。

为解决上述技术问题，本申请提供如下技术方案：一种基于 Hyperscan 的 Web 应用防火墙规则匹配方法，主要包括：

步骤一：OpenResty 服务器采用 Hyperscan 正则匹配库来进行规则匹配；

步骤二：在 OpenResty 的 Lua 模块中新增一个 HTTP API 接口；

步骤三：将最新配置写入共享内存中后，调用 Hyperscan 提供的接口编译最新配置中的正则表达式，并继续调用接口将编译结果序列化成字符串后写入共享内存中；

步骤四：每个 worker 在每次请求到来时根据共享内存中的信号量来更新配置到当前 worker 中；

步骤五：每个 worker 取出步骤三中写入共享内存中的序列化字符串，然后调用 Hyperscan 的 deserialize_db 接口进行反序列化，并将反序列化后的 database 指针保存在共享内存中；

步骤六：每个 worker 在反序列化完后再分配 scratch 内存，并将分配到的 scratch 内存的指针保存在共享内存中；

步骤七：每个 worker 在完成以上步骤后从共享内存中取出 database 指针和 scratch 指针，然后调用 Hyperscan 提供的 scan 接口进行扫描和获取匹配结果；

步骤八：每个 worker 根据匹配结果来判断放行还是拦截当次请求。

在其中一些实施例中，所述 WAF 至少包括 OpenResty 服务器、Agent 配置同步组件和管理系统平台；

所述 Agent 配置同步组件和管理系统平台可以是任何配置同步系统；

所述 OpenResty 服务器采用 Hyperscan 正则匹配库来进行规则匹配；

所述 OpenResty 服务器可以是任何基于 nginx 的支持 lua 插件的系统；

所述 Agent 组件负责在配置变动时调用 OpenResty 的 API 接口将最新配置同步到 OpenResty 中。

在其中一些实施例中，所述 OpenResty 服务器的硬件部分包括处理器、存储设备、输入和输出接口、通信接口以及总线；

其中处理器采用中央处理器；

存储设备采用随机存取存储器、SSD、HDD 中一种；

输入和输出接口用于连接输入/输出模块，以实现信息输入及输出，输入/输出模块可以作为组件配置在设备中，也可以外接于设备以提供相应功能，其中输入设备包括键盘、

鼠标以及各类传感器，输出设备包括显示器、扬声器以及指示灯；

通信接口用于连接通信模块，以实现本设备与其他设备的通信交互，其中通信模块可以通过有线方式实现通信；

总线包括一通路，在设备的各个组件之间传输信息。

在其中一些实施例中，所述 Hyperscan 的匹配流程是一个从编译正则表达式到在输入数据中搜索匹配的模式的过程，Hyperscan 一次完整的匹配流程包括编译期、分配 scratch 空间和运行期，运行时包括但不限于一个 worker 工作进程。

在其中一些实施例中，所述步骤二中，在 OpenResty 的 Lua 模块中新增一个 HTTP API 接口，便于 Agent 将最新的配置同步到 WAF 中，在 API 接口中判断如果是配置的增、删、改的时候将当前配置写入 OpenResty 的共享内存中，并在共享内存中写入信号量便于其他 worker 同步最新配置。

在其中一些实施例中，所述步骤四中，由于 OpenResty 的每个 worker 进程的地址空间是相互独立的，所以采取的方案是在一个 worker 上进行编译，并将编译后的规则库序列化成字符串保存在共享内存中，其他 worker 再去共享内存中取出字符串反序列化后使用。

在其中一些实施例中，所述步骤 4 中，每个 worker 在每次请求到来时先去共享内存中获取信号量判断是否需要更新配置，如果需要更新配置，则先从共享内存中获取最新配置并替换当前 worker 独立内存中的配置。

在其中一些实施例中，所述步骤五中，由于每个 worker 的内存空间是独立的，每个 worker 的 database 指针都不一样，所以在共享内存中的 key 按照 db_[worker_id]来命名。

在其中一些实施例中，所述步骤六中，由于每个所述 worker 的内存空间是独立的，每个 worker 的 scratch 指针都不一样，所以在共享内存中的 key 按照 scr_[worker_id]来命名。

在其中一些实施例中，所述步骤六，每个 worker 在反序列化完后，再通过调用 Hyperscan 的 make_scratch 接口来分配 scratch 内存。

本申请的有益效果：

1、与传统的方案相比，在每次请求中去掉了从 Redis 数据库中获取规则库的序列化后的字符串，取而代之的是能够从读取速度更快的共享内存中获取序列化后的字符串。

2、与传统的方案相比，预先在反序列化后分配好 scratch 内存并将 database 指针和 scratch 指针保存到共享内存中，避免在每次请求中都去反序列化和分配 scratch 内存，提高了 WAF 正则匹配效率，且有效提升了 WAF 的处理性能。

附图说明

为了更清楚地说明本申请实施例的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其它的附图。其中：

图 1 为根据本申请一实施例的一种基于 Hyperscan 的 Web 应用防火墙规则匹配方法的流程图。

图 2 为根据本申请一实施例的 OpenResty 服务器的硬件框图。

图 3 为本申请实施例 I 中防火墙规则匹配系统的结构框图。

具体实施方式

为使本申请的上述目的、特征和优点能够更加明显易懂，下面结合说明书附图对本申请的具体实施方式做详细的说明。

在下面的描述中阐述了很多具体细节以便于充分理解本申请，但是本申请还可以采用其他不同于在此描述的其它方式来实施，本领域技术人员可以在不违背本申请内涵的情况下做类似推广，因此本申请不受下面公开的具体实施例的限制。

其次，此处所称的“一个实施例”或“实施例”是指可包含于本申请至少一个实现方式中的特定特征、结构或特性。在本说明书中不同地方出现的“在一个实施例中”并非均指同一个实施例，也不是单独的或选择性的与其他实施例互相排斥的实施例。

再其次，本申请结合示意图进行详细描述，在详述本申请实施例时，为便于说明，表示器件结构的剖面图会不依一般比例作局部放大，而且所述示意图只是示例，其在此不应限制本申请保护的范围。此外，在实际制作中应包含长度、宽度及深度的三维空间尺寸。

实施例 I

参照图 1 至 3，本申请在一个实施例中提供了一种基于 Hyperscan 的 Web 应用防火墙规则匹配方法，如图 1，主要包括以下步骤一至步骤八。

步骤一：OpenResty 服务器采用 Hyperscan 正则匹配库来进行规则匹配。

步骤二：在 OpenResty 的 Lua 模块中新增一个 HTTP API 接口，便于 Agent 将最新的配置同步到 WAF 中，这个 API 接口中判断如果是配置的增、删、改的时候先将配置更新到当前 worker 进程中，再将配置写入共享内存中，然后在共享内存中再申请一个共享字典，key 为[worker_id]，值为 true，便于后续其他 worker 同步最新配置。其中共享内存（Shared Memory）是一种特殊的内存存储，允许多个 worker 进程间共享数据。由于 OpenResty 是

多进程模型的，而不是多线程的，所以不同的 worker 进程之间默认是不能共享状态或内存的。但是，使用共享内存，这些进程可以在此预分配的内存区域中共享信息。

步骤三：将最新配置写入共享内存中后，调用 Hyperscan 提供的 compile 接口编译最新配置中的正则表达式，并继续调用 Hyperscan 提供的 serialize_db 接口将编译结果序列化成字符串后写入共享内存中。由于 OpenResty 的每个 worker 进程的地址空间是相互独立的，所以本申请采取的方案是在一个 worker 上进行编译，并将编译后的规则库序列化成字符串保存在共享内存中，其他 worker 再去共享内存中取出字符串反序列化后使用。

步骤四：每个 worker 在每次请求到来时先去共享内存中获取信号量判断是否需要更新配置，如果需要更新配置，则先从共享内存中获取最新配置并替换当前 worker 独立内存中的配置。即每个 worker 在每次请求到来时先去共享内存中根据 worker_id 获取对应的值，如果为 true 则需要更新配置，从共享内存中获取最新配置并更新到当前 worker 进程中。

步骤五：每个 worker 取出步骤三中写入共享内存中的序列化字符串，然后调用 Hyperscan 的 deserialize_db 接口进行反序列化，并将反序列化后的 database 指针保存在共享内存中；

由于每个 worker 的内存空间是独立的，每个 worker 的 database 指针都不一样，所以在共享内存中的 key 按照 db_[worker_id]来命名。

步骤六：每个 worker 在反序列化完后通过调用 Hyperscan 的 make_scratch 接口来分配 scratch 内存，并将分配到的 scratch 内存的指针保存在共享内存中。由于每个所述 worker 的内存空间是独立的，每个 worker 的 scratch 指针都不一样，所以在共享内存中的 key 按照 scr_[worker_id]来命名。

步骤七：每个 worker 在完成以上步骤后从共享内存中取出 database 指针和 scratch 指针，然后调用 Hyperscan 提供的 scan 接口进行扫描和获取匹配结果；

步骤八：每个 worker 根据匹配结果来判断放行还是拦截当次请求。

具体的，所述 WAF 至少包括 OpenResty 服务器、Agent 配置同步组件和管理系统平台，其中 Agent 配置同步组件和管理系统平台可以是任何配置同步系统；

所述 Agent 配置同步组件和管理系统平台可以是任何配置同步系统；

所述 OpenResty 服务器采用 Hyperscan 正则匹配库来进行规则匹配，所述 Hyperscan 的匹配流程是一个从编译正则表达式到在输入数据中搜索匹配的模式的过程，Hyperscan 一次完整的匹配流程包括编译期(编译正则表达式)、分配 scratch 空间和运行期(扫描并报告匹配)，运行时包括但不限于一个 worker 工作进程；

所述 OpenResty 服务器可以是任何基于 nginx 的支持 lua 插件的系统,本申请对此不做限制;

所述 Agent 组件负责在配置变动时调用 OpenResty 的 API 接口将最新配置同步到 OpenResty 中。

更进一步地,如图 2,所述 OpenResty 服务器的硬件部分可以包括处理器、存储设备、输入和输出接口、通信接口以及总线;

其中处理器可以采用通用的中央处理器,用于执行相关程序,以实现本说明书实施例所提供的技术方案;

存储设备可以采用随机存取存储器(Random Access Memory, RAM)、SSD(Solid State Drive)、HDD(Hard Disk Drive)等形式实现;

输入和输出接口用于连接输入/输出模块,以实现信息输入及输出,输入/输出模块可以作为组件配置在设备中,也可以外接于设备以提供相应功能,其中输入设备可以包括键盘、鼠标、各类传感器等,输出设备可以包括显示器、扬声器、指示灯等;

通信接口用于连接通信模块,以实现本设备与其他设备的通信交互,其中通信模块可以通过有线方式(例如 USB、网线等)实现通信;

总线包括一通路,在设备的各个组件(例如处理器、存储设备、输入/输出接口和通信接口)之间传输信息。

需要说明的是,尽管上述设备仅示出了处理器、存储设备、输入/输出接口、通信接口以及总线,但是在具体实施过程中,该设备还可以包括实现正常运行所必需的其他组件。此外,本领域的技术人员可以理解的是,上述设备中也可以仅包含实现本说明书实施例方案所必需的组件,而不必包含全部组件。

如图 3,本申请还提供一种基于 Hyperscan 的 Web 应用防火墙规则匹配的系统,该系统应用于上述方法,系统包括:

OpenResty 模块: 对所有进出的数据进行检查和过滤。它可以识别并阻止那些可能利用应用程序漏洞进行攻击的恶意数据包;

Agent 模块: 负责实时将最新的配置通过 API 接口同步到 OpenResty 模块中;

管理平台模块: 负责将网页上的最新配置项下发到 Agent 模块;

Hyperscan 正则匹配模块: 负责 WAF 防护规则的正则匹配,返回匹配结果。

综上,与传统的方案相比,在每次请求中去掉了从 Redis 数据库中获取规则库的序列化后的字符串,取而代之的是从读取速度更快的共享内存中获取序列化后的字符串,因为

Openresty 中的共享内存设计为工作进程之间的高速数据共享机制。它允许在工作进程之间存储和检索数据，而无需进行进程间通信（IPC）。与传统的方案相比，预先在反序列化后分配好 scratch 内存并将 database 指针和 scratch 指针保存到共享内存中，避免在每次请求中都去反序列化和分配 scratch 内存，明显的提高了 WAF 的处理速度，因为内存分配和释放本身就是一种相对昂贵的操作，如果在每次请求时都进行，那么这个开销就可能会变得很大。此外，频繁的内存分配和释放还可能导致内存碎片，进一步降低性能。

本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程，是可以通过计算机程序来指令相关的硬件来完成，所述的计算机程序可存储于一非易失性计算机可读存储介质中，该计算机程序在执行时，可包括如上述各方法的实施例的流程。其中，本申请所提供的各实施例中所使用的对存储器、存储、数据库或其它介质的任何引用，均可包括非易失性和易失性存储器中的至少一种。非易失性存储器可包括只读存储器（Read-Only Memory, ROM）、磁带、软盘、闪存或光存储器等。易失性存储器可包括随机存取存储器（Random Access Memory, RAM）或外部高速缓冲存储器。作为说明而非局限，RAM 可以是多种形式，比如静态随机存取存储器（Static Random Access Memory, SRAM）或动态随机存取存储器（Dynamic Random Access Memory, DRAM）等。

以上实施例的各技术特征可以进行任意的组合，为使描述简洁，未对上述实施例中的各个技术特征所有可能的组合都进行描述，然而，只要这些技术特征的组合不存在矛盾，都应当认为是本说明书记载的范围。

以上所述实施例仅表达了本申请的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对发明专利范围的限制。应当指出的是，对于本领域的普通技术人员来说，在不脱离本申请构思的前提下，还可以做出若干变形和改进，这些都属于本申请的保护范围。因此，本申请专利的保护范围应以所附权利要求为准。

权利要求

1. 一种基于 Hyperscan 的 Web 应用防火墙规则匹配方法，包括：

步骤一：OpenResty 服务器采用 Hyperscan 正则匹配库来进行规则匹配；

步骤二：在 OpenResty 的 Lua 模块中新增一个 HTTP API 接口；

步骤三：将最新配置写入共享内存中后，调用 Hyperscan 提供的接口编译最新配置中的正则表达式，并继续调用接口将编译结果序列化成字符串后写入共享内存中；

步骤四：每个 worker 在每次请求到来时根据共享内存中的信号量来更新配置到当前 worker 中；

步骤五：每个 worker 取出步骤三中写入共享内存中的序列化字符串，然后调用 Hyperscan 的 deserialize_db 接口进行反序列化，并将反序列化后的 database 指针保存在共享内存中；

步骤六：每个 worker 在反序列化完后再分配 scratch 内存，并将分配到的 scratch 内存的指针保存在共享内存中；

步骤七：每个 worker 在完成以上步骤后从共享内存中取出 database 指针和 scratch 指针，然后调用 Hyperscan 提供的 scan 接口进行扫描和获取匹配结果；

步骤八：每个 worker 根据匹配结果来判断放行还是拦截当次请求。

2. 根据权利要求 1 所述的方法，其中所述 WAF 至少包括 OpenResty 服务器、Agent 配置同步组件和管理系统平台；

所述 Agent 配置同步组件和管理系统平台可以是任何配置同步系统；

所述 OpenResty 服务器采用 Hyperscan 正则匹配库来进行规则匹配；

所述 OpenResty 服务器可以是任何基于 nginx 的支持 lua 插件的系统；

所述 Agent 组件负责在配置变动时调用 OpenResty 的 API 接口将最新配置同步到 OpenResty 中。

3. 根据权利要求 2 所述的方法，其中所述 OpenResty 服务器的硬件部分包括处理器、存储设备、输入和输出接口、通信接口以及总线；

其中处理器采用中央处理器；

存储设备采用随机存取存储器、SSD、HDD 中一种；

输入和输出接口用于连接输入/输出模块，以实现信息输入及输出，输入/输出模块可以作为组件配置在设备中，也可以外接于设备以提供相应功能，其中输入设备包括键盘、

鼠标以及各类传感器，输出设备包括显示器、扬声器以及指示灯；

通信接口用于连接通信模块，以实现本设备与其他设备的通信交互，其中通信模块可以通过有线方式实现通信；

总线包括一通路，在设备的各个组件之间传输信息。

4. 根据权利要求 2 所述的方法，其中所述 Hyperscan 的匹配流程是一个从编译正则表达式到在输入数据中搜索匹配的模式的过程，Hyperscan 一次完整的匹配流程包括编译期、分配 scratch 空间和运行期，运行时包括但不限于一个 worker 工作进程。

5. 根据权利要求 1 所述的方法，其中所述步骤二中，在 OpenResty 的 Lua 模块中新增一个 HTTP API 接口，便于 Agent 将最新的配置同步到 WAF 中，在 API 接口中判断如果是配置的增、删、改的时候将当前配置写入 OpenResty 的共享内存中，并在共享内存中写入信号量便于其他 worker 同步最新配置。

6. 根据权利要求 1 所述的方法，其中所述步骤四中，由于 OpenResty 的每个 worker 进程的地址空间是相互独立的，所以采取的方案是在一个 worker 上进行编译，并将编译后的规则库序列化成字符串保存在共享内存中，其他 worker 再去共享内存中取出字符串反序列化后使用。

7. 根据权利要求 1 所述的方法，其中所述步骤 4 中，每个 worker 在每次请求到来时先去共享内存中获取信号量判断是否需要更新配置，如果需要更新配置，则先从共享内存中获取最新配置并替换当前 worker 独立内存中的配置。

8. 根据权利要求 1 所述的方法，其中所述步骤五中，由于每个 worker 的内存空间是独立的，每个 worker 的 database 指针都不一样，所以在共享内存中的 key 按照 db_[worker_id] 来命名。

9. 根据权利要求 1 所述的方法，其中所述步骤六中，由于每个所述 worker 的内存空间是独立的，每个 worker 的 scratch 指针都不一样，所以在共享内存中的 key 按照 scr_[worker_id] 来命名。

10. 根据权利要求 1 所述的方法，其中所述步骤六，每个 worker 在反序列化完后，再通过调用 Hyperscan 的 make_scratch 接口来分配 scratch 内存。

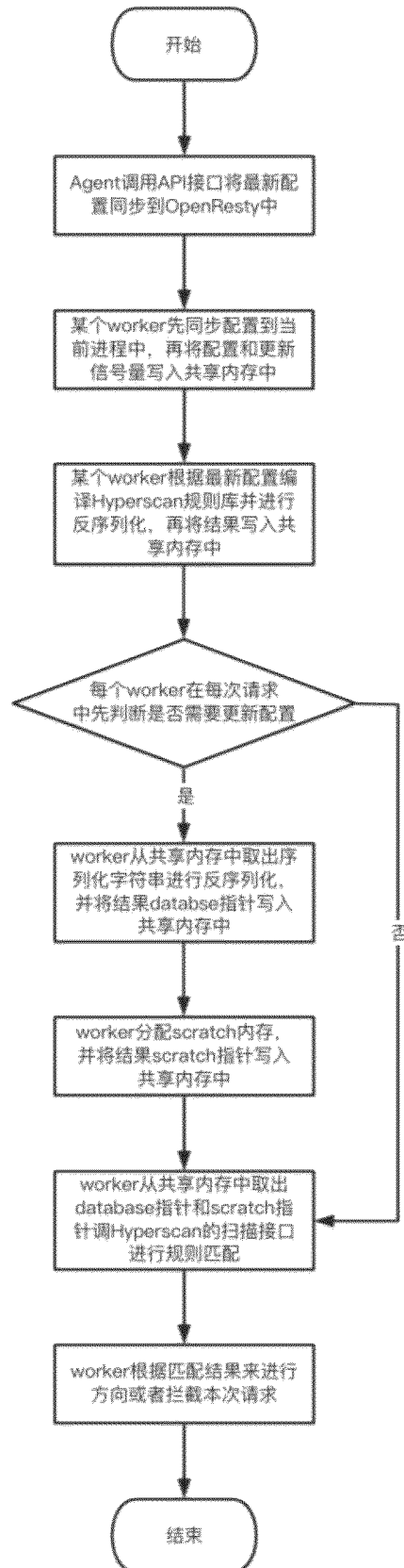


图 1

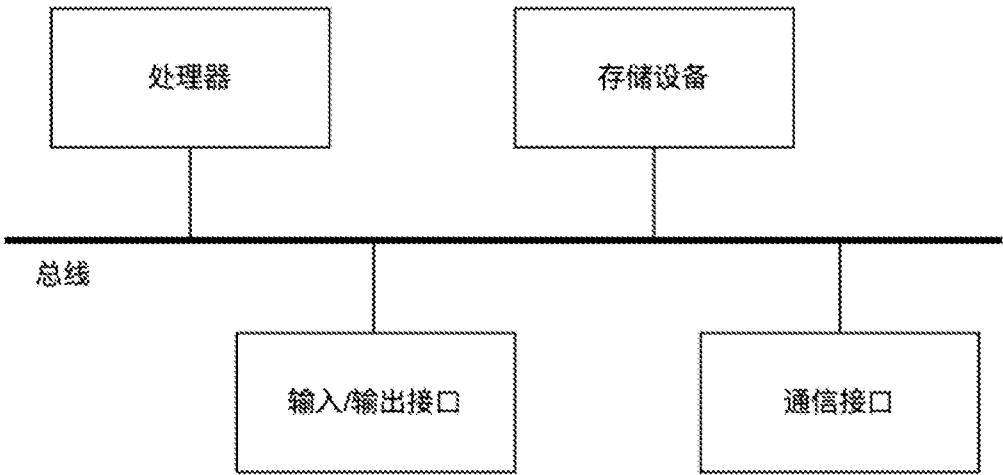


图 2

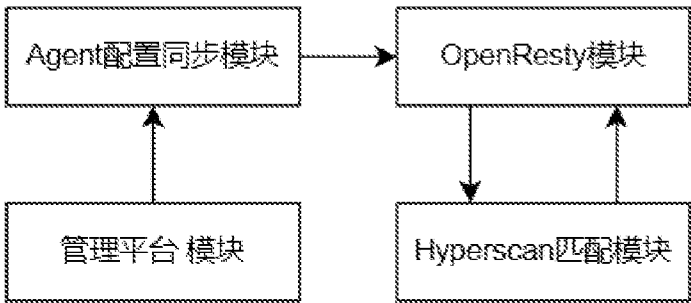


图 3

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2024/136670

A. CLASSIFICATION OF SUBJECT MATTER

H04L9/40(2022.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNTXT, ENTXT, DWPI, CNKI, 必应, BING: WAF, WEB应用防火墙, Hyperscan, OpenResty, 共享内存, 序列化, 字符串, 反序列化, database, scratch, shared, memory, deserialize, serialize, scan

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 117857124 A (TIANYI CLOUD TECHNOLOGY CO., LTD.) 09 April 2024 (2024-04-09) claims 1-10	1-10
Y	GITHUB. "fesily/lua-resty-hs" https://github.com/fesily/lua-resty-hs , 19 August 2022 (2022-08-19), README.md, hs.lua code, lines 1-215	1-10
Y	CN 111786959 A (CHINA MOBILE (HANGZHOU) INFORMATION TECHNOLOGY CO., LTD. et al.) 16 October 2020 (2020-10-16) description, paragraphs 0006 and 0032-0074	1-10
Y	CN 112351020 A (BEIJING JIANKANG ZHIJIA TECHNOLOGY CO., LTD.) 09 February 2021 (2021-02-09) description, paragraphs 0043-0064	1-10
A	CN 116846613 A (YUNZHILIAN NETWORK TECHNOLOGY (BEIJING) CO., LTD.) 03 October 2023 (2023-10-03) entire document	1-10



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"D" document cited by the applicant in the international application

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

13 March 2025

Date of mailing of the international search report

18 March 2025

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
China No. 6, Xitucheng Road, Jimenqiao, Haidian District,
Beijing 100088

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2024/136670

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2022113969 A1 (INTEL CORP.) 14 April 2022 (2022-04-14) entire document	1-10

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2024/136670

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN	117857124	A	09 April 2024	None	
CN	111786959	A	16 October 2020	None	
CN	112351020	A	09 February 2021	None	
CN	116846613	A	03 October 2023	None	
US	2022113969	A1	14 April 2022	None	

A. 主题的分类

H04L9/40(2022.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

IPC: H04L

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNTEXT, ENTXT, DWPI, CNKI, 必应: WAF, WEB应用防火墙, Hyperscan, OpenResty, 共享内存, 序列化, 字符串, 反序列化, database, scratch, shared, memory, deserialize, serialize, scan

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
PX	CN 117857124 A (天翼云科技有限公司) 2024年4月9日 (2024 - 04 - 09) 权利要求1-10	1-10
Y	GITHUB. "fesily/lua-resty-hs" https://github.com/fesily/lua-resty-hs , 2022年8月19日 (2022 - 08 - 19), README.md, hs.lua代码第1-215行	1-10
Y	CN 111786959 A (中移(杭州)信息技术有限公司等) 2020年10月16日 (2020 - 10 - 16) 说明书第0006、0032-0074段	1-10
Y	CN 112351020 A (北京健康之家科技有限公司) 2021年2月9日 (2021 - 02 - 09) 说明书第0043-0064段	1-10
A	CN 116846613 A (云智联网络科技(北京)有限公司) 2023年10月3日 (2023 - 10 - 03) 全文	1-10
A	US 2022113969 A1 (INTEL CORPORATION) 2022年4月14日 (2022 - 04 - 14) 全文	1-10

☐ 其余文件在C栏的续页中列出。

☒ 见同族专利附件。

* 引用文件的具体类型:

"A" 认为不特别相关的表示了现有技术一般状态的文件

"D" 申请人在国际申请中引证的文件

"E" 在国际申请日的当天或之后公布的在先申请或专利

"L" 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)

"O" 涉及口头公开、使用、展览或其他方式公开的文件

"p" 公布日先于国际申请日但迟于所要求的优先权日的文件

"T" 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

"X" 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

"Y" 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

"&" 同族专利的文件

国际检索实际完成的日期

2025年3月13日

国际检索报告邮寄日期

2025年3月18日

ISA/CN的名称和邮寄地址

中国国家知识产权局
中国北京市海淀区蓟门桥西土城路6号 100088

授权官员

李昕萌

电话号码 (+86) 010-53961596

PCT/ISA/210 表(第2页) (2022年7月)

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2024/136670

检索报告引用的专利文件			公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN	117857124	A	2024年4月9日	无	
CN	111786959	A	2020年10月16日	无	
CN	112351020	A	2021年2月9日	无	
CN	116846613	A	2023年10月3日	无	
US	2022113969	A1	2022年4月14日	无	