# 基于追踪标记的 WAF 设计思路

发布于：https://www.freebuf.com/articles/web/338814.html

## 一 相关背景

目前，市面上的 WAF 产品通常采用”发现即阻断“的策略，以防护针对业务系统的 Web 攻击行为。虽然该策略可及时阻断攻击，但形式上过于简单，并不能有效掌握攻击者进一步的攻击意图，也不能有效提高攻击者的成本投入。本文借鉴蜜罐的思想，构思一种融合欺骗技术的 WAF 系统，目的是为现有 WAF 提供一种设计思路。相对于传统 WAF，本文所述 WAF 不仅具有传统 WAF 的功能，同时可识别并追踪攻击者。

如图 1 所示，WAF 系统以串联方式部署在业务系统的前面，用于对来自互联网的流量进行检测：当发现客户请求为正常流量时，则将其转发给业务服务区部署的“真实业务系统”，从而为正常用户提供所需的互联网服务；当发现客户请求为恶意流量时，则会对攻击源进行唯一性标记，并把该客户端的流量通过 NGINX 集群牵引到镜像服务区部署的“镜像业务系统”。

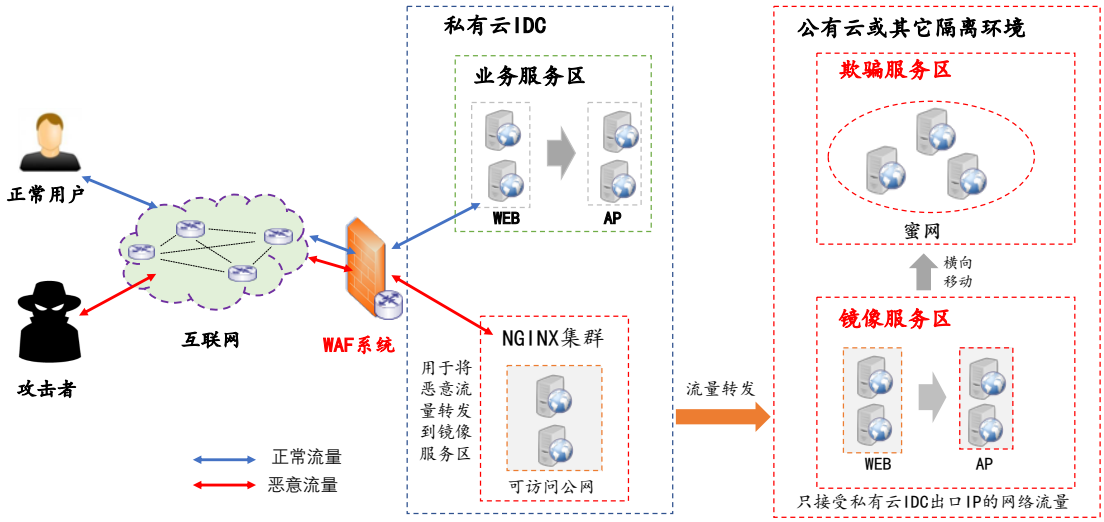


图 1 WAF 部署模式和运行流程

**1 业务服务区：**部署有“真实业务系统”，用于处理正常用户的访问请求。如果“潜在攻击者”不对系统发起攻击，那么其访问请求会被转发到真实的业务服务区。只是一旦其发起攻击行为，便会被 WAF 标记为“攻击源”，并进行唯一性标记，那么其当前、及后续访问请求都会被牵引到镜像服务区。由于“镜像业务系统”的前端展示和业务功能与“真实业务系统”没有区别，所以整个牵引过程对攻击者是无感的。

**2 镜像服务区：**部署有“镜像业务系统”，用于处理攻击者的访问请求。如前所述，该系统的前端展示和业务功能与“真实业务系统”没有区别，在使用过程中很难区分，同时做了数据脱敏处理、伪造了一些常见漏洞点。此外，为了防止攻击者在获取镜像服务区的机器权限后，以此为跳板，对业务服务区发起攻击，部署时要求镜像服务区与业务服务区需完全隔离，最好是物理层面的隔离。

**3 欺骗服务区：**部署有各种“欺骗性系统”，用于引诱黑客在内网横行渗透过程中的攻击。“欺骗服务区”本质是一个蜜网，其目的是为了更长时间的浪费攻击者的时间与精力，更多的掌握攻击者的行为、手段、技能，更大程度上使攻击者漏出马脚、从而溯源其身份。比如：在数据库中存储一些伪造的、带有标记性特征的“高价值客户数据”，一旦被攻击者拿到并在网上售卖，则可对攻击者做进一步的溯源；在系统中放置一些伪造的、带有木马功能的“敏感文件”，一旦被攻击者拖回并打开，则可反向控制攻击者的客户端。

备注：镜像服务区、欺骗服务区可部署在阿里云、腾讯云等公有云平台上（或者其它隔离环境），本地 IDC 通过 NGINX 集群将识别到的攻击流量转发到这些区域。

# 二 系统设计

本文所述 WAF 的核心功能包括五个模块，分别是：流量检测模块、终端标记模块、流量分发模块、漏洞配置模块、指纹采集模块。（只是个人想法，可根据实际扩展）

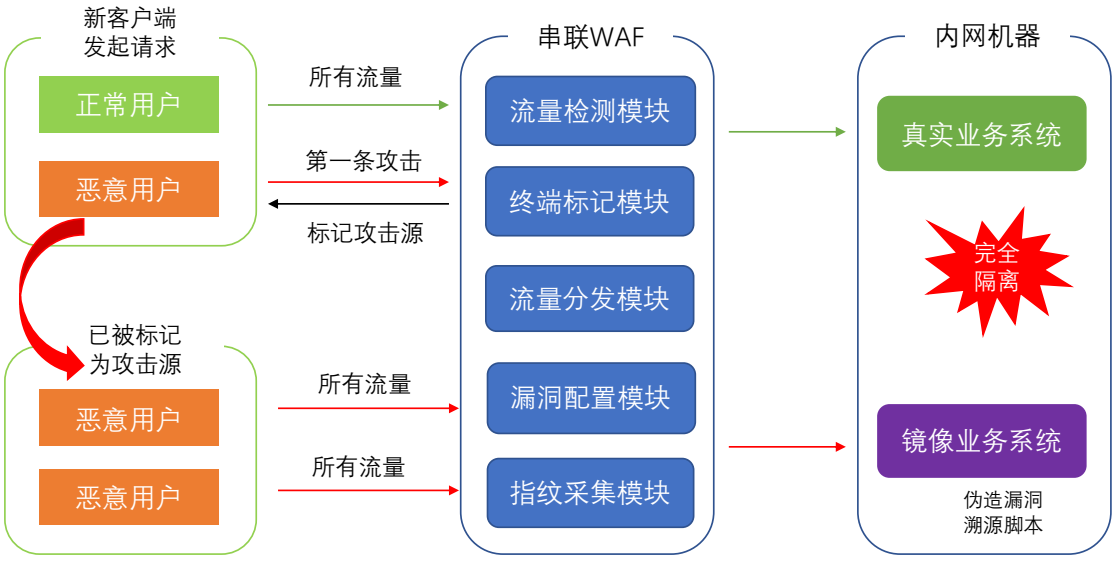


图 2 WAF 概要设计和核心模块

**1 流量检测模块：**不仅具备常规 WAF 的通用功能，如：SQL 注入检测、XSS 漏洞检测、代码执行检测、文件上传检测等，同时还具备“攻击源标记查验”功

能，目的是判断请求流量中是否带有“终端标记模块”下发的攻击源标记字段。如果请求流量中带有攻击源标记字段，表明该请求是“已被标记为攻击源”的客户端发起的，则直接将该请求经 NGINX 集群牵引至“镜像业务系统”；如果请求流量中没有攻击源标记，表明该请求是正常的客户端发起的，则会继续判断是否带有攻击特征。

**2 终端标记模块：**用于对发起恶意请求的客户端进行标记。当流量检测模块发现当前请求为恶意请求时，便会调用终端标记模块在 HTTP 响应包中插入标记字段，以对发起该请求的客户端进行唯一性标记。其中，插入标记字段的功能是通过 set-cookie 实现的，该特征字符串会反向到达、并存储在攻击者的浏览器中。由于该 cookie 值的过期时间设置为永久，因此只要不手动清除，浏览器随后访问目标网站的所有请求都会带上该字段。（格式： trackid=32 位随机字符串，形如：trackid =92f4ac47-527b-11eb-ba1b-f45c89c42263）

**3 流量分发模块：**用于根据“流量检测模块”的判定结果信息、以及 WAF 系统中配置的路由信息，将访问请求分发到“真实业务系统”或“镜像业务系统”。其中，如果是“已知攻击源发起的请求”或“某客户端初次发起的恶意请求”，则将其经 NGINX 集群牵引到“镜像业务系统”；如果是 “正常客户端发起的请求”，则将其牵引到“真实业务系统”。对客户端而言，整个过程是完全透明的，且需要攻击者通过浏览器发起请求。

**4 漏洞配置模块：**用于在“镜像业务系统中”中以“插件化”形式配置伪造的漏洞点，目的就是让黑客发现并对其攻击。其中，在部署漏洞前，可通过虚拟机镜像手段将业务系统从“业务服务区”克隆到了“镜像服务区”，并做好数据脱敏工作。在配置漏洞点的时候，漏洞配置模块可将事先构建好的漏洞文件下发到“镜像业务系统”服务器。（该功能可作为 WAF 的一部分，或者独立于 WAF 部署）

**5 指纹采集模块：**用于采集攻击者的客户端设备指纹信息，攻击者的第三方平台账号、键盘记录、访问过的网站等信息。其中，整个过程是通过在“镜像业务系统”的返回页面中插入溯源脚步（JavaScript 代码）实现的。当溯源脚本反向到达攻击源客户端、并被浏览器解析执行后，便会将相关的指纹信息回送给 WAF 系统。其中，采集信息的丰富度一定程度上依赖于攻击者的操作。

# 三 运行流程

WAF 在对互联网边界流量处理的过程，其主要流程如图 3 所示，具体步骤如下：

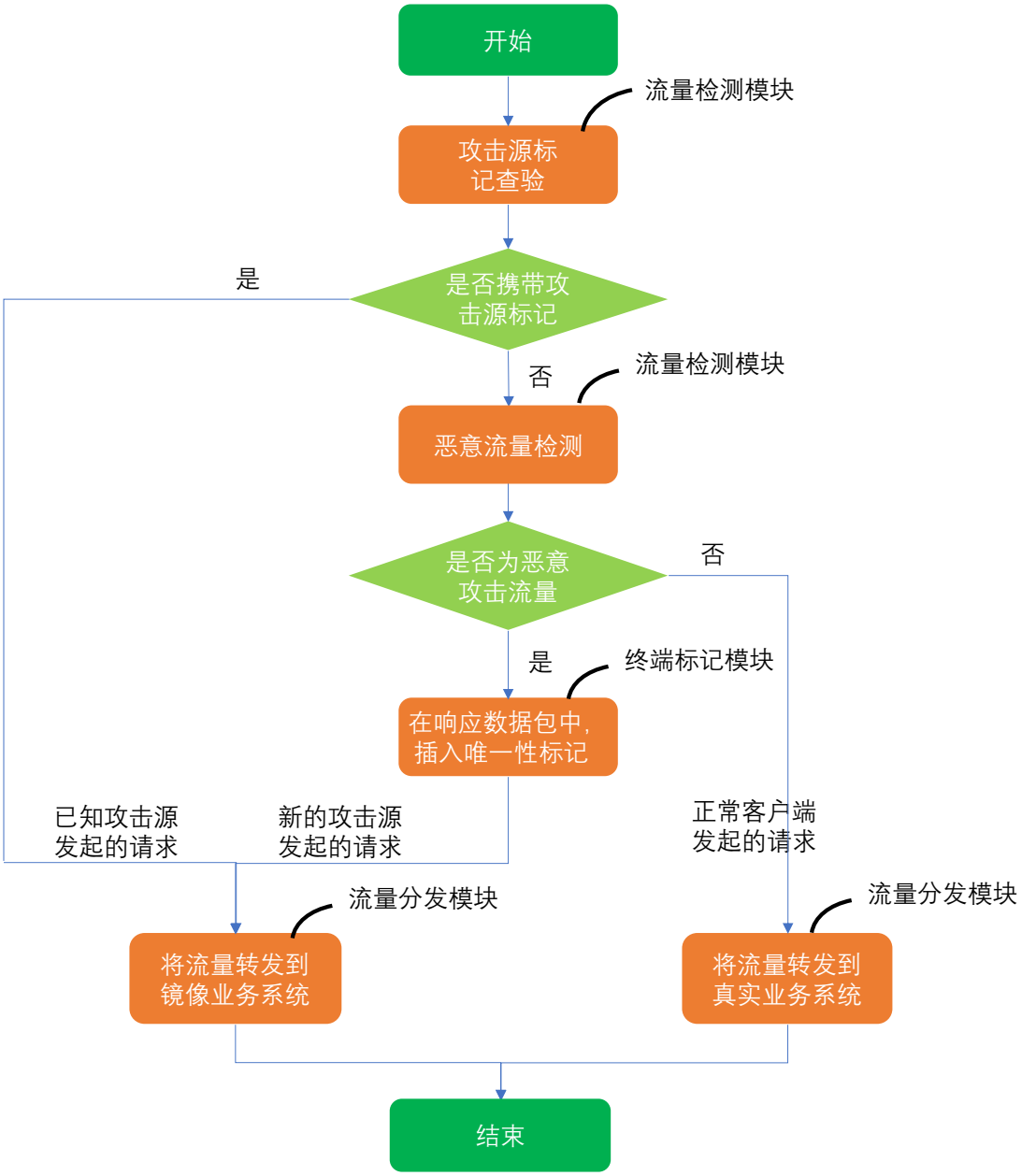


图 3 WAF 对网络流量处理流程

1 基于“流量检测模块”查验流量中是否带有“攻击源标记”。如果没有，则进入步骤 2；如果有，则表明当前请求是已被标记为攻击源的客户端发起的，则跳转到步骤 4。

2 基于“流量检测模块”检测流量中是否带有攻击行为特征。如果没有，则表明当前请求是正常用户发起的，则跳转至步骤 5；如果有，则进入步骤 3。

3 基于“终端标记模块”对当前网络请求进行标记，并在对应的响应报文中通过 set-cookie 的方式插入攻击源标记字符串。

4 基于“流量分发模块”模块将网络流量转发至“镜像业务系统”。其中，该步骤处理的流量为“已知攻击源”与“新的攻击源”发起的网络请求。

5 基于“流量分发模块”模块将网络流量转发至“真实业务系统”。其中，该步骤处理的流量为正常用户使用的客户端发起的网络请求。备注：对于攻击源而言，一旦被 WAF 标记，其当前请求以及后续发起的网络请求，无论是否带有恶意特征，都会被转发到“镜像业务系统”。

# 四 总结归纳

整体而言，本文所述 WAF 从“潜在攻击识别、溯源取证分析”等方面弥补了传统 WAF 的不足，即保护了业务系统的安全性，又具备一定的溯源取证的能力。其优点在于：

1 可以避免攻击者对业务系统的潜在攻击：当识别到攻击流量后，WAF 可基于“终端标记模块”对攻击源进行唯一性标记，进而将同一攻击源的当前流量、以及后续流量牵引到“镜像业务系统”，从而避免攻击者对真实业务系统的潜在攻击行为。

2 可消耗攻击者的时间以及攻击者的精力：无论何种目的、何种手段，所有的攻击行为都是需要投入时间和精力的。由于攻击行为被识别、牵引到了“镜像业务系统”，使得攻击者将时间、精力消耗在对“镜像业务系统”的攻击上，因此整个攻击过程对真实的业务系统是无效的，相对降低了黑客的有效攻击成果。

备注：本文所描述的思路需要一定的先验条件，即：攻击者使用浏览器（或配合 Burp）对目标系统进行测试（目的是在 Cookie 中设置追踪 ID）。除了基于追踪 ID 来判断请求是否由攻击者发起之外，也可基于“时间、源 IP、目标 IP”的方式进行判断，这样就不用考虑是否使用浏览器的问题了，只是有一定的误报。当然，没有一种技术是完美的，本文只是描述一种 Web 防护的思路。