2019 年全国大学生信息安全竞赛

作品报告

作品名称：基于网络流量监测与 RASP 技术的 Web 应用纵深防御解决方案

电子邮箱：leo_infosec@foxmail.com

提交日期：2019.5.26

填写说明

1. 所有参赛项目必须为一个基本完整的设计。作品报告书旨在能够清晰准确地阐述（或图示）该参赛队的参赛项目（或方案）。

2. 作品报告采用A4纸撰写。除标题外，所有内容必需为宋体、小四号字、1.5倍行距。

3. 作品报告中各项目说明文字部分仅供参考，作品报告书撰写完毕后，请删除所有说明文字。(本页不删除)

4. 作品报告模板里已经列的内容仅供参考，作者可以在此基础上增加内容或对文档结构进行微调。

5. 为保证网评的公平、公正，作品报告中应避免出现作者所在学校、院系和指导教师等泄露身份的信息。

目录

摘要

随着信息技术的迅速发展，大量互联网新兴产业也开始出现并迅速崛起。例如熟知的 Web 应用系统，其广泛应用于社交、银行、购物等重要业务，给人们的生活和工作带来了极大的便利，影响了现实空间中多个方面（经济、文化、科技等），在当今的网络资产中占有很高的比例。然而，相关的安全威胁也不容小觑，系统的广泛受攻击面以及攻击技术的多样化，降低了 Web 应用被攻击入侵的门槛，严重影响了社会稳定甚至国家安全。

已有 Web 应用安全防御与检测系统有着准确率低、维护困难、实时性差等常见缺陷，针对此现状，本系统以 Web 应用服务器环境结合网络流量监测和代码底层堆栈上下文检测为前提，以 Web 应用纵深防御为应用背景，以 RASP（应用运行时自保护）技术和 WAF（Web 应用防火墙）技术为思想借鉴，着眼于将 RASP 技术的相关防御思想结合 Web 应用环境来突破传统防御手段所无法解决的相关技术瓶颈，从而与传统 WAF 内外配合、相互补充，达到纵深防御的目的，积极应对针对 Web 应用的种种复杂的入侵攻击。

关键词：RASP ；WAF；应用运行时自保护 ；Web 应用防御；应用安全

第一章 作品概述

1.1 研究背景

目前，网络安全的威胁越来越突出，网络安全风险继续传递到政治、经济、文化、社会、生态、国防等领域，急需加强网络建设安全监管，并继续出台网络安全政策和法规。在中央网络安全与信息化委员会（原 "中央网络安全与信息化领导小组"）的统一领导下，中国进一步加强了网络安全和信息化管理工作，各行业主管部门共同努力，促进网络安全治理。

2018 年，我国进一步完善网络安全法律体系，改革网络安全管理体制机制，不断加强公共互联网安全治理，构建互联网发展安全基础，构筑安全上网环境，特别是在党政机关以及重要行业方面，网络安全应急响应能力进一步提升，恶意程序感染、网页篡改、网站后门等传统的安全问题得到有效控制。但是在关键信息基础设施与云平台等面临的安全风险依旧较为突出，APT 攻击、数据泄露和分布式拒绝服务攻击等问题也较为严重。

在 Web 应用安全方面，2018 年，CNCERT 监测发现境内外约 1.6 万个 IP 地址对我国境内约 2.4 万个网站植入后门。近三年来，我国境内被植入后门的网站数量持续保持下降趋势，2018 年的数量较 2017 年下降了 19.3%。其中，约有 1.4 万个（占全部 IP 地址总数的 90.9%）境外 IP 地址对境内约 1.7 万个网站植入后门，位于美国的 IP 地址最多，占境外 IP 地址总数的 23.2%，其次是位于中国香港和俄罗斯的 IP 地址，如图 12 所示。从控制我国境内网站总数来看，位于中国香港的 IP 地址控制我国境内网站数量最多，有 3,994 个，其次是位于美国和俄罗斯的 IP 地址，分别控制了我国境内 3,607 个和 2,011 个网站[1] 。

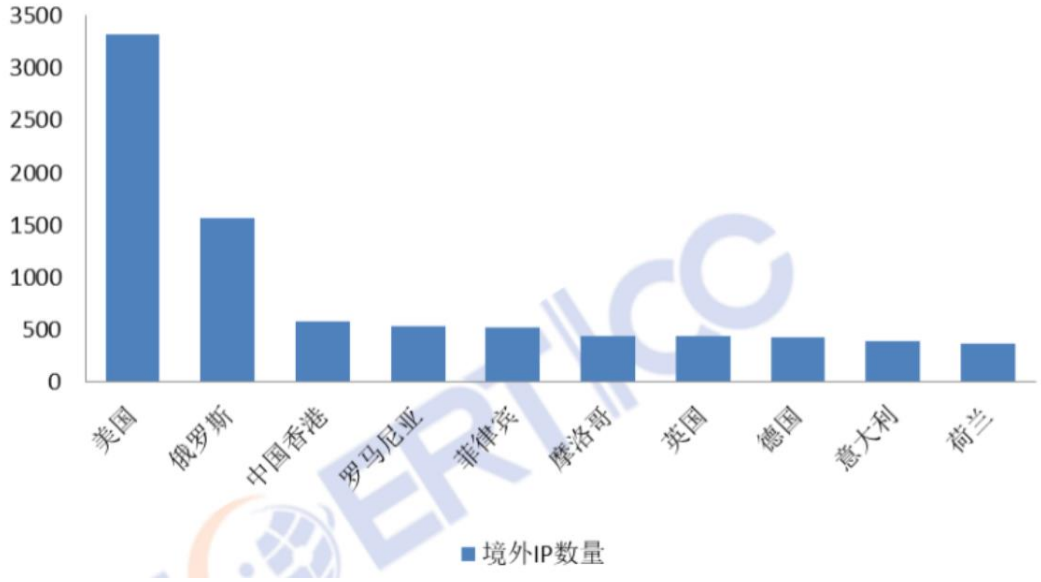

图 1.1 后门 IP 统计

2017 年 1-10 月，360 网站安全检测平台共扫描检测网站 104.7 万个，其中，扫出存在 漏洞的网站 69.1 万个（全年去重），占比为 66.0%，共扫描出 1674.1 万次漏洞。 从高危漏洞的检测情况来看，扫出存在高危漏洞的网站 34.5 万个（相比 2016 年的 14.0 万个网站漏洞，增长了约 2.5 倍），占扫描网站总数的 32.9%，共扫描出 247.0 万次高危漏洞。

图 1.2 17 年网站存在漏洞情况（扫描检测）

根据 360 网站安全检测平台扫描出高危漏洞的情况，跨站脚本攻击漏洞的扫出次数和 漏洞网站数都是最多的，稳居排行榜榜首。其次是 SQL 注入漏洞、SQL 注入漏洞（盲注）、 PHP 错误信息泄露等漏洞类型。

根据各类网站安全漏洞被扫出次数的统计情况。应用程序错误信息（287.7 万次）、 发现 敏感名称的目录漏洞（184.1 万次）和异常页面导致服务器路径泄露（122.7 万次）这三类安 全漏洞是占比最高的网站安全漏洞，三者之和超过其他所有漏洞检出次数的总和。 值得一提的是，“应用程序错误信息”漏洞等前四名都是低危漏洞，改变了前两年高危 漏洞稳居排名第一的局面。

2017 年 1-10 月，360 网站卫士共为 187.5 万个网站拦截各类网站漏洞攻击 26.4 亿次，较 2016 年 17.1 亿次，增长 54.4%，平均每天拦截漏洞攻击 868.9 万次。3 月、4 月是 2017 年攻击量最大的两个月，三月达到最高 6.3 亿次。网站每月遭遇漏洞攻击情况如下图所示。

图 1.3 16-17 年网站每月遭遇漏洞攻击次数

360 威胁情报中心监测显示，59.6%的网站漏洞攻击类型都为“SQL 注入”,稳坐第一。其次为“WebShell”和“通用漏洞”，占比分别为 8.2%和 3.6%。下图给出了网站漏洞攻击类型的具体分布情况[2]。

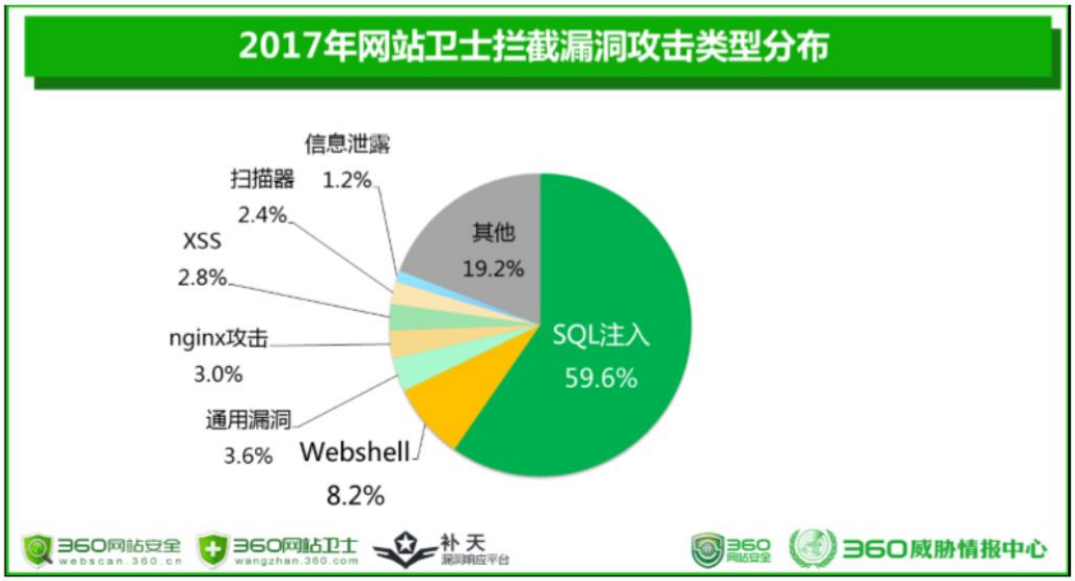

图 1.4 17 年网站卫士拦截漏洞攻击类型分布

1.2 研究现状

目前的 Web 应用安全研究主要集中在传统 Web 安全研究、AI 和大数据在安全方面的应用和拟态防御、可信计算等方面。

传统 Web 安全研究方面：

1.在代码层与网络层完善过滤和阻断策略（静态分析日志流量和文件样本和动态分析脚本执行流程和堆栈上下文）手段中：

作者基于用户服务的 WEB 应用保护墙框架中添加自学习模块，可以提高用户干预率，降低 WAF 误报。该框架还可以有效地处理安全问题，如支付密码泄露、数据泄露、表单篡改和 cookie 欺骗，并验证设计了 cookie 反欺骗整体方案和防篡改保护策略[8] 。

2.在安全开发生命周期(*SDL*)与安全编码、安全基线手段中：

作者提出了基于安全环的技术，以确保 web 应用程序从软件研发的生命周期开始时的安全性，以及应用程序安全性、安全建设管理、 安全操作和维护的各个方面，对 WEB 应用系统的安全保护进行了一站式部署，并从需求设计开始通过系统在线、

渗透测试、安全增强、计时检查、实时监控、安全通知提高了 WEB 进行攻击的可追溯性从而形成一个安全环路，提高了应用系统的整体防护能力[9] 。

另有作者介绍了漏洞消减插件的结构及模块功能，分析了安全检测模块的污点检测过程，为插件提供了具体的实现方法。 最后，测试插件的功能。 实验结果表明,漏洞消减插件工具在编码过程中检测到了恶意代码，并帮助开发人员在编码阶段的早期识别并了解漏洞，从而有效地减少大多数漏洞[10]。

机器学习、深度学习安全应用研究方面:

作者提出一种卷积神经网络的Web攻击检测方法,将Web访问流量转换为灰度图,并搭建基于空间金字塔池化的卷积神经网络,并从众多Web流量中识别出Web攻击，实现卷积神经网络在识别Web攻击时的更加高效准确[6] 。

作者提出了一种基于支持向量机的 web 攻击安全防御框架，并给出了支持向量机算法中的特征提取以及训练和分类的具体实施过程及检测模型[7] 。

可信计算与区块链安全应用研究方面:

作者介绍了一种基于区块链和可信计算技术的防篡改网页系统，并定义了受可信计算技术安全支持的 WEB 服务器模型从而确保单个节点的可靠性，另外本文提出了一个适用于该系统的区块链，解决了多个节点之间的信任问题，设计并实现了一个原型系统，证明了该网站可以阻断攻击，有效地解决了页面被篡改的问题[6]。

拟态防御安全研究方面：

作者提出了基于"动态异构冗余"结构的拟态防御模型,描述了拟态防御模型的防御原理然后基于拟态防御模型构建了拟态防御Web服务器,介绍了其架构,最后分析了拟态原理在Web服务器上的实现[5] 。

主动防御、态势感知（大数据、AI、安全社群）安全研究方面：

作者整合安全用户区域内用户终端和网络链路、应用系统以及数据流量等各类数据源,经统一汇聚和存储后,利用智能分析技术,结合数据处理、安全规则模型以及攻击推理模型等分析算法,将无序的安全日志和报警数据转化成直观的可视化安全事件信

息,从海量数据中挖掘威胁情报,从而实现发现、预警和感知,提升安全监测的攻击发现和安全态势感知的能力[3]。

作者在现有开源大数据组件的基础上，构建了数据收集与整理、数据存储、离线分析发现、实时相关检测、威胁预警和态势呈现等功能，并支持全过程安全事件处理，以及完整的网络安全态势感知和预警体系结构，与现有的类似平台体系结构相比，具有高可用性、可扩展性、易用性的特点部署等，并能更好地支持威胁情报的引入 [4]。

1.3 本文研究内容

传统的防御方法大多采用精确的特征匹配方法来匹配和检测所有用户发送的信息。 这种检测方法的问题在于，攻击者完全可以避免被特殊字符或编码检测。 除了基于精确模式匹配功能的方法限制所导致的大量误报之外，可能还有许多误报。

另一种方法是确保从 WEB 应用程序开发人员的角度验证所有用户提交的数据，指定用户输入的内容的格式和长度。 它是标准化最后收到的内容的确认。 但是，上述操作可能会降低应用程序系统的可用性，并且由于开发人员安全素养良莠不齐很难做到完全避免漏洞实现安全编码。

互联网安全其本质是黑客和开发者之间的攻防战争，既然是战争就有相通性，就可以借鉴战争的一些基本思想来实施防御。苏联元帅米·尼·图哈切夫斯基在对第一次世界大战以及国内战争经验的基础上，提出一种名为：大纵深作战理论的思想，该思想的核心是：多点布防，以点带面，多面成体，纵深打击及防御。

Gartner 在 2014 年提出了应用自我保护技术（RASP，Runtime Application Self-Protection ）的概念——“到 2020 年, 40% 从事 DevOps 的企业将通过采用应用程序安全自检、自我诊断和自我保护技术来确保开发后的应用程序的安全。

因为上述传统防御手段的局限性和纵深防御的思想以及最近的 RASP 技术的成熟，我们设计了基于网络流量监测与 RASP 技术的 Web 应用纵深防御解决方案。

本文包含以下内容：

分析针对 Web 应用的主要安全威胁和应对威胁、积极防御的安全需求，讨论了目前 Web 应用安全防御思想和方法，重点讨论了传统 Web 应用防火墙在安全防御的劣势与瓶颈。从而引出第二章对 RASP 技术的介绍，进而设计了基于 WAF 和 RASP 技术的 Web 应用防御系统的整体架构，进一步对系统组件进行功能设计，然后通过攻击防御测试和纵向对比验证了本次设计的可用性和相对传统防御方式的优越性，最后分析了本系统的缺陷，提出了改进方案，并展望了日后的 Web 应用安全防御的方向，重点介绍了威胁情报与态势感知在甲方安全防御的积极作用和巨大优势。

在研究针对 Web 应用的防御思想与方法过程中，详细讨论了目前 Web 应用安全防御的主流思想和方法，并分析了其优势与局限。

在研究目前 RASP 技术和 WAF 在 Web 应用安全防御方面应用的原理、效果的过程中，详细分析了 RASP 技术在 Web 应用服务器安全防御应用的主流方法及其优势，同时介绍了设计本系统所使用的相关技术的基本情况。

在具体的 RASP 技术和 WAF 结合的纵深防御系统中，首先以 Apache、php 环境为例，做出整体架构，然后对相关功能模块做出具体实现，再然后对本系统做了可用性测试和同传统防御方式的纵向对比并得出本系统的优势，最后总结经验提出了进一步的改进方案，展望了未来 Web 应用的安全防御蓝图。

第二章 相关技术

2.1 WAF 技术简介

Web 应用防护系统（也称为：网站应用级入侵防御系统。英文：Web Application Firewall，简称： WAF）。利用国际上公认的一种说法：Web 应用防火墙是通过执行一系列针对 HTTP/HTTPS 的安全策略来专门为 Web 应用提供保护的一款产品。WAF 具有以下特点：

异常检测协议：

Web 应用防火墙会对 HTTP 的请求进行异常检测，拒绝不符合 HTTP 标准的请求。并且，它也可以只允许 HTTP 协议的部分选项通过，从而减少攻击的影响范围。甚至，一些 Web 应用防火墙还可以严格限定 HTTP 协议中那些过于松散或未被完全制定的选项。

增强的输入验证：

增强输入验证，可以有效防止网页篡改、信息泄露、木马植入等恶意网络入侵行为。从而减小 Web 服务器被攻击的可能性。

及时补丁：

修补 Web 安全漏洞，是 Web 应用开发者最头痛的问题，没人会知道下一秒有什么样的漏洞出现，会为 Web 应用带来什么样的危害。WAF 可以为我们做这项工作了——只要有全面的漏洞信息 WAF 能在不到一个小时的时间内屏蔽掉这个漏洞。当然，这种屏蔽掉漏洞的方式不是非常完美的，并且没有安装对应的补丁本身就是一种安全威胁，但我们在没有选择的情况下，任何保护措施都比没有保护措施更好。

（附注：及时补丁的原理可以更好的适用于基于 XML 的应用中，因为这些应用的通信协议都具规范性。）

基于规则的保护和基于异常的保护：

基于规则的保护可以提供各种 Web 应用的安全规则，WAF 生产商会维护这个规则库，并时时为其更新。用户可以按照这些规则对应用进行全方面检测。还有的产品可以基于合法应用数据建立模型，并以此为依据判断应用数据的异常。但这需要对用户企业的应用具有十分透彻的了解才可能做到，可现实中这是十分困难的一件事情。

状态管理

WAF 能够判断用户是否是第一次访问并且将请求重定向到默认登录页面并且记录事件。通过检测用户的整个操作行为我们可以更容易识别攻击。状态管理模式还能检测出异常事件（比如登陆失败），并且在达到极限值时进行处理。这对暴力攻击的识别和响应是十分有利的。

其他防护技术：

WAF 还有一些安全增强的功能，可以用来解决 WEB 程序员过分信任输入数据带来的问题。比如：隐藏表单域保护、抗入侵规避技术、响应监视和信息泄露保护。

2.2RASP 技术简介

随着 Web 应用攻击手段变得复杂，基于请求特征的防护手段，已经不能满足企业安全防护需求。Gartner 在 2014 年提出了应用自我保护技术（RASP，Runtime Application Self-Protection ）的概念——“到 2020 年, 40% 从事 DevOps 的企业将通过采用应用程序安全自检、自我诊断和自我保护技术来确保开发后的应用程序的安全。推荐采用开发计划的运行时应用程序自我保护 (RASP)。评估不太成熟的供应商和提供商的潜在安全选项。即将防护引擎嵌入到应用内部，不再依赖外部防护设备。”

应用程序中 RSAP 将自己注入其中，与应用程序集成，实时监视和阻止攻击，并使程序本身具有保护自身的能力。而应用程序不需要在编码时进行任何修改，只需一个简单的配置就可以。

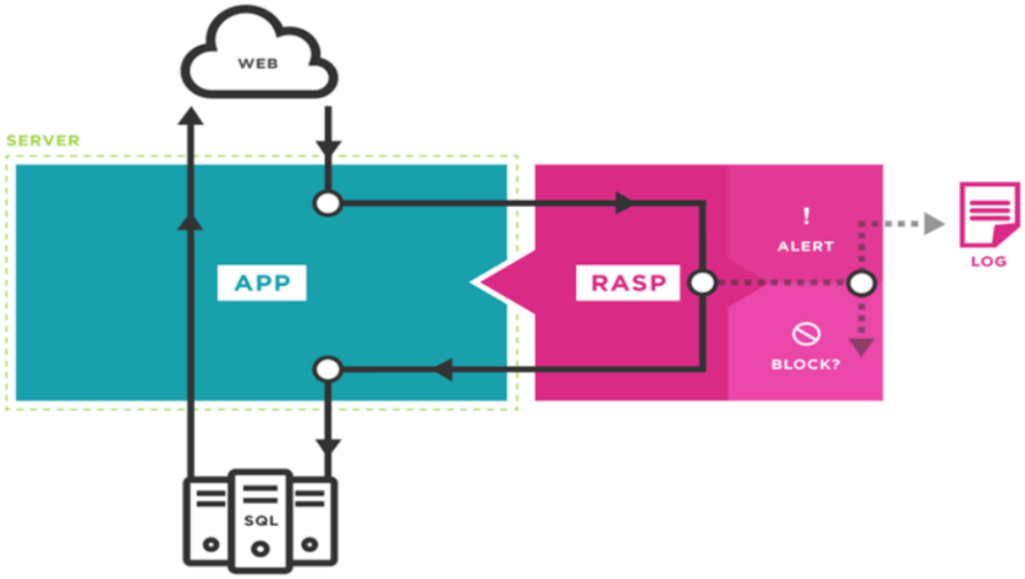

图 2.1 RASP 技术原理

RASP 具有以下几个特点：该系统运行在应用的内部；Hook 程序范围包括数据库、网络、文件系统等；HOOK 检测点位于应用程序的输入输出位置；输入点包括用户请求和文件输入等。

第三章 作品设计与实现

3.1 系统框架

本系统采用二级纵深防御设计方案，分别为一级防御子系统（WAF 子系统）和二级防御子系统（RASP 子系统），每个子系统采用模块化的结构由多个模块组成。

本系统的简易架构图如下：

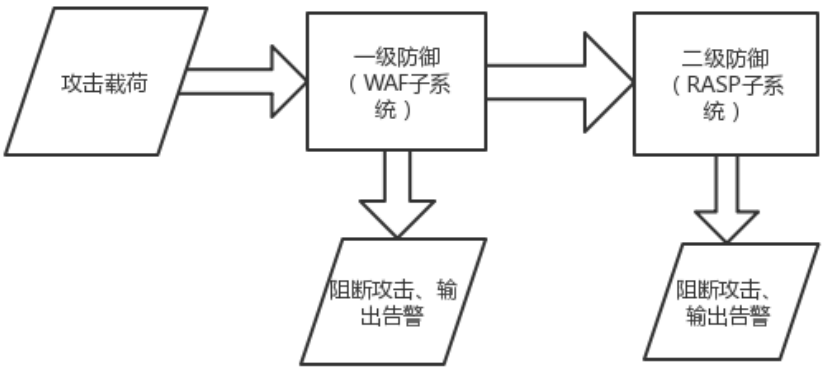

图 3.1 系统架构

以 PHP(mysqli) + MySQL 为例，WAF 子系统处理流程如下：

0.启动及初始化阶段：启动并加载配置文件，初始化全局变量，反向代理或者嵌入式方式侦听应用层 HTTP 流量。

1.请求头阶段：这个阶段的规则会在 apache 完成请求头的读取后立即被执行。

2.请求体阶段：对请求体进行规则匹配检测。

3.响应体阶段：这是通用输出分析阶段，可以插入运行规则截断响应体，输出的 HTML 信息公布、错误消息和失败的验证文字。

4.记录阶段：记录全阶段日志，方便审计。

主要流程如图：

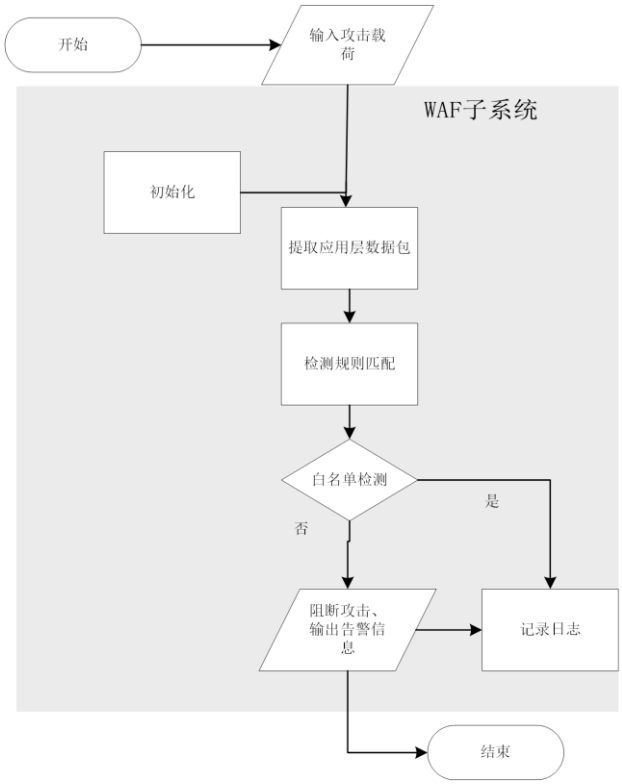

图 3.2 WAF 子系统

以 PHP(mysqli) + MySQL 为例，简要说明 RASP 子系统处理流程如下：

0. 初始化本系统所需全局变量，为新请求计算唯一的 request-id，设置 response header，方便记录各个恶意请求的日志。

1.使用 Fswatch 载入全局配置文件（插件名称、检测堆栈深度、检测速率、返回格式、记录格式）。

2.基线检测：当攻击者连接数据库，触发 mysqli_connect 函数 HOOK 点，收集 sql 连接配置信息然后调用 V8 模块驱动相关 JS 基线检测插件进行基线检测。

3.攻击检测：使用 Mysql HOOK 点进行数据库语句收集与检测，收集查询参数然后调用 V8 模块驱动相关 JS 攻击检测插件，使用插件定义的方法执行检测。

4.根据检测结果，记录告警日志，加载白名单判断是否阻断相关攻击。

5.析构阶段：释放请求相关资源，根据配置返回自定义响应结果并阻断或者放行本次攻击事务。

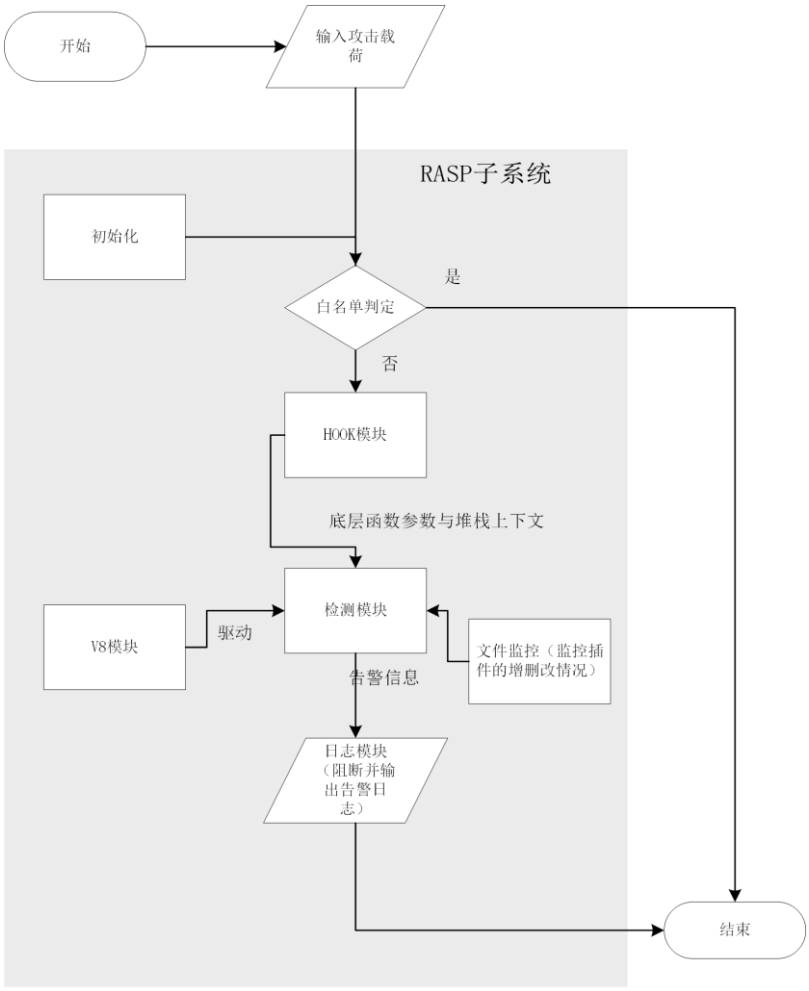

图 3.3 RASP 子系统

3.2 WAF 子系统设计

本 WAF 子系统使用了 ModSecurity 开源 WAF 组件，下面简单介绍该 WAF 组件的部署和安全模型。

ModSecurity 是一个入侵侦测与防护引擎，它主要是用于 Web 应用程序，所以也被称为 Web 应用程序防火墙。 它可以作为 Apache Web 服务器的模块或是单独的应用程序来运作。ModSecurity 的功能是增强 Web application 的安全性和保护 Web application 以避免遭受来自已知与未知的攻击。

ModSecurity 具有两种部署架构 1.与 Web Server 结合 。2.与 Apache 结合部署为网关，当作一个反向代理。

ModSecurity 能够立即针对你的 WEB 应用系统进行攻击防御，有三种通用的方法：

1、消极(negative)安全模型：消极安全模型监控那些异常的、不常用的和通用的 WEB 攻击类请求。它统计每个请求的有关 IP 地址、应该连接、和用户帐户的异常分数，当出现较高的异常分数时，会记录日志并完全的阻止访问。

2、积极安全模开型：部署积极安全模型后，只有那些明确的请求被允许通过，其它的一律禁止。这个模式要求你对需要保护的 WEB 应用要非常的了解。因此积极安全模式最好是用于那种大量访问却很少更新的系统，这样才能使这种模型的维护工作量降到最低。

3、已知漏洞攻击：其规则语言使 ModSecurity 成为一个理想的外部修补工具，外部修补（有时是指虚拟修补）可以减少机会之窗。一些组织修补这些应用的漏洞通常需要几周的时间，使用 ModSecurity，应用系统可以从外部修补，根本不用改应用的源码（甚至时不用去管它），可以保证你的系统安全直到有一个合适的补丁来应用到系统中。

3.3RASP 子系统设计

在 RASP 子系统中，主要包括初始化、HOOK、检测、记录等处理流程。下面针对构成该流程的几个主要的模块进行针对性说明：

3.3.1 白名单实现

白名单模块运行在检测动作之前，用于排除对信赖的动作和用户的检测。

白名单模块中的白名单列表包含放行函数、放行参数和受信赖的 IP，该列表是由用户自定义。本模块选择了 Double Array Trie 算法来匹配白名单，具体实现如下:

1.白名单放在共享内存里，并且通过读写锁更新

2.使用 Double Array Trie 算法在白名单里寻找匹配的项目，当进入检测点，根据 bitmask 来决定是否直接放行。

下面介绍以下 Double Array Trie 算法的相关原理：

Trie 是一种搜索树，词条的公共前缀是压缩存储在 Trie 树的逻辑存储词典里，只会存储一次，所以又称前缀树。如图所示：

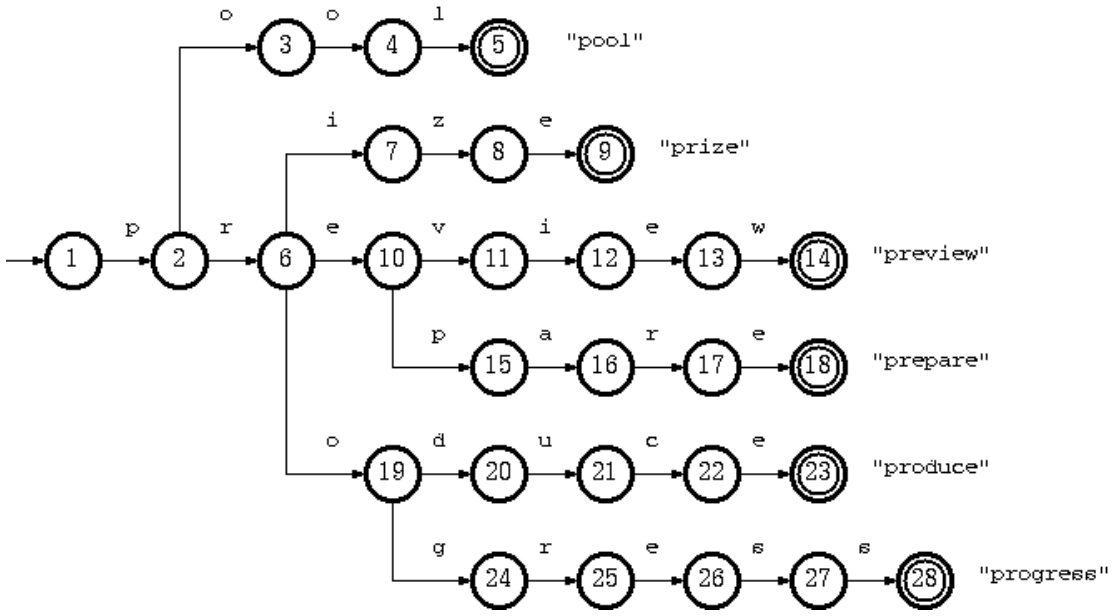

图 3.4 Trie

Trie 可以存储为树。每个节点包含 n 个指针，指向 n 个后续节点，每个节点对应一个输入字符。这样，每个节点的指针数与单词列表字符的大小相关。如果以链接列表的形式组织 n 个指针，则查询的效率较低，如果 n 个指针由固定长度数组表示，则占用的空间较大且在很大程度上是不可接受的。

为了解决上面的问题，有学者依次设计出了 Four-Array Trie，Triple-Array Trie 和 Double-Array Trie 结构，其得名源于内部采用的数组的个数。

Double-Array Trie：

Double-Array Trie 包含 base 和 check 两个数组。base 数组的每个元素表示一个 Trie 节点，即一个状态；check 数组表示某个状态的前驱状态。

base 和 check 的关系满足下述条件：

base[s] + c = t

check[t] = s

其中，s 是当前状态的下标，t 是转移状态的下标，c 是输入字符的数值。如图所示：

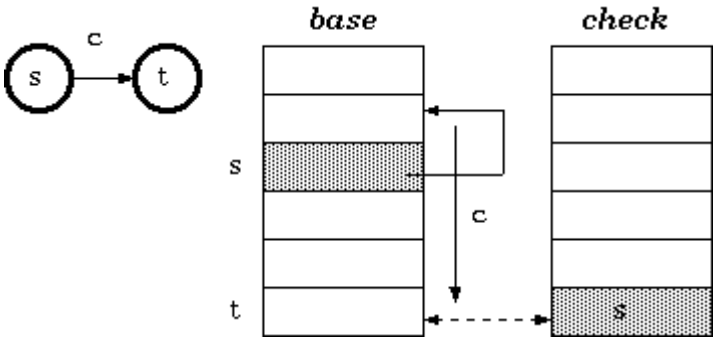

图 3.5 Double-Array Trie

查询过程：

根据上述公式，查找某个字符串就非常简单。假设初始状态为 t0，字符序列是(c1, c2, …, cn)。那么，输入 c1 后的状态为 t1 = base[t0] + c1，以此类推。

如果到某个状态是不合法的，那么查询失败；如果转到状态 tn = base[tn-1] + c，并且 tn 是结束状态，那么查询成功；如果 tn 不是结束状态，那么查询失败[13] 。

3.3.2HOOK 模块

本 HOOK 模块作用为在 PHP 的运行阶段中加入钩子，用于获取底层函数的参数和堆栈上下文。为便于理解该模块作用与流程，下面介绍 PHP 的执行生命周期。

以 cli SAPI 为例，其单个请求生命周期如下图所示：

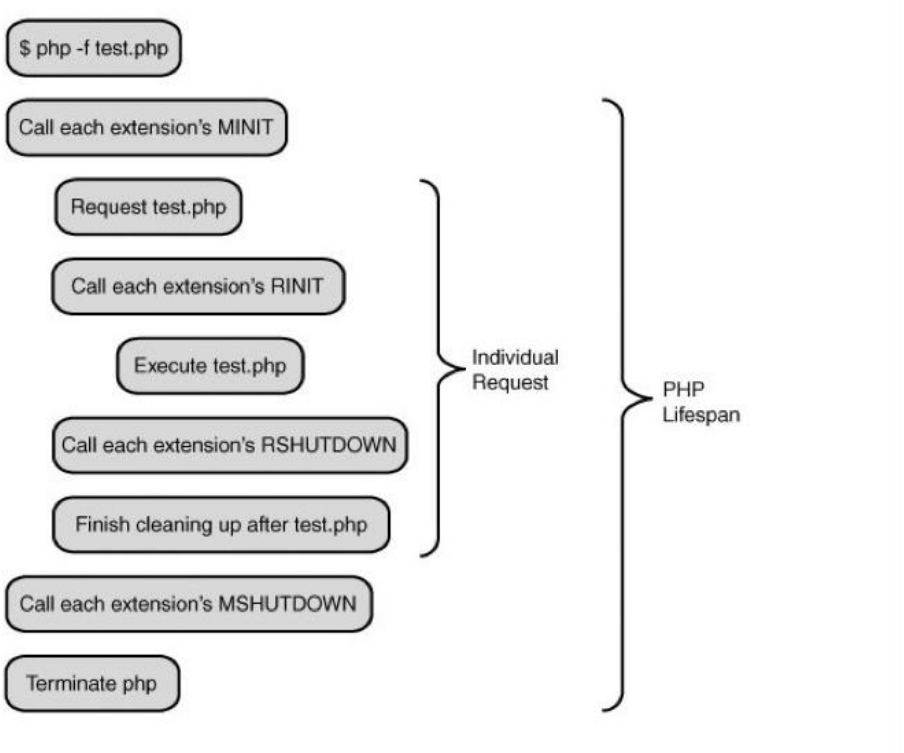

图 3.6 php 执行周期

RASP 系统核心原理为：在 MINIT 阶段，替换全局 compiler_globals 的 function_table 与 class_table 中特定 PHP_FUNCTION 对应的函数指针（封装原有 handler，增加前置、后置处理），由此实现对敏感函数的挂钩。通过敏感函数参数结合请求信息判断是否存在攻击行为，进而采取拦截或者放行操作。

HOOK 流程包含两类：compiler_globals 的 handler 替换和用户自定义 opcode_handler，启动流程如下

1.在全局 compiler_globals 对应的 hashtable(function_table 和 class_table)中查找函数对应 zend_function

2.封装原有 handler，根据需求增加前置、后置处理

3.针对指定 opcode（如 ZEND_INCLUDE_OR_EVAL）通过 zend_set_user_opcode_handler 自定义处理逻辑

3.3.3V8 模块

本子系统将 V8 嵌入到 本系统中作为 JavaScript 检测插件的执行引擎。该 V8 模块具体运行流程如下：

1.初始化阶段载入所有插件

2.每个请求线程在对应的 V8 驱动环境上执行检测逻辑

3.执行检测逻辑前向 V8 平台添加超时监控后台任务，超时后中断检测

4.析构阶段销毁线程对应 V8 平台申请的内存。

3.3.4 文件监控模块

Fswatch 是一个跨平台的文件更改监视，用于获取在指定的文件或目录的内容被改变或修改的通知警报。本系统利用该工具完成了检测模块，用于对自定义检测插件的增删改实时发现和自动加载。

在本系统的初始化阶段，通过使用 fswatch 工具，在后台进行插件文件及其目录的增删改监控。当目标文件与目录发生变化时，重载插件对象。 在系统停止和析构阶段，停止目标文件目录监控，销毁 fswatch 实例和后台线程

Fswatch 具有以下特点：

1.支持几种特定于 OS 的 API

2.允许递归目录监视

3.使用包含和排除正则表达式执行路径过滤

4.支持自定义记录格式

5.此外，它支持周期性空闲事件

3.3.5 检测插件

检测插件使用 JS 编写，由 V8 模块驱动，该模块定义了具体的检测函数和响应动作。

检测函数所使用的输入来自 HOOK 模块提供的底层函数参数和堆栈上下文并包含该请求对应的 HTTP 数据包，用户定义检测逻辑处理输入并规定相应的响应动作，最终输出告警信息到日志模块，并返回 HTTP 响应结果。

下面介绍检测插件的编写格式：

```
const plugin_version = '2018-1000-1000'
const plugin_name        = 'test-plugin'
```

```
'use strict'
var plugin    = new RASP(plugin_name)
const clean = {
action:          'ignore',
message:        'Looks fine to me',
confidence: 0
}
plugin.register('sql', function (params, context) {
plugin.log('SQL query: ' + params.query)
return clean
})
plugin.log('plugin-demo: plugin loaded')
```

首先，我们调用 `plugin.register` 注册了 SQL 查询的检测函数，并将 SQL 语句打印到插件日志。其中，`params` 为检查点提供的参数，如 SQL 语句、要读取的文件等等，`context` 为请求信息，如请求参数，服务器信息等等；

然后，我们在回调函数里，调用 `plugin.log` 打印了 SQL 查询语句；

最后，我们在回调函数里，返回检测结果，即 "放行"。

若要拦截危险操作 ，在回调函数里，将返回的 `action` 字段改为 `block` 即可拦截请求。比如，当 SQL 语句包含 union 注入特征，我们想拦截整个请求的话，可以这样写:

```
plugin.register('sql', function (params, context) {
    plugin.log('SQL tokens ', RASP.sql_tokenize(params.query, params.server))


    if (/union.*select.*from.*information_schema/.test(params.query)) {
        return {
```

```
                action:         'block',
                message:        '拦截 SQL 查询，因为 XXX',
                confidence: 90
            }
        }

        return clean
})
```

3.3.6 日志模块

日志模块用于记录检测模块提供的告警日志，方便日后的审计和集成在 SIEM 系统中。

日志模块启动流程如下：

1.初始化并载入日志模块所需的全局变量

2.申请共享内存从而用于部分日志的进/线程间同步

3.获取本机主机及网络信息，用于记录系统基线

4.获取插件反馈的日志信息

第四章 作品测试与分析

4.1 系统测试环境

表 4.1 测试环境

软硬件	配置
处理器	Intel Core i5
内存	4GB
操作系统	Ubuntu16.04LTS（64 位）
其他	VMWare(11),GoogleChrome(62.0.3202.94),Python(2.7)

4.2 测试数据与方案

可行性测试：

测试方案：使用 github 的 php 漏洞利用样例项目进行可行性测试，共 17 个，在每个测试用例里，我们都增加了说明，即如何模拟正常的请求，和不正常的攻击请求。

测试指标：在测试样例中记录样例的拦截个数，计算拦截率。

性能损耗测试：

测试方案：同一主机、同一网络、同一环境下，安装 discuz x3.2 ，控制变量为是否安装本系统，记录安装前后相关检测指标。

性能测试指标：记录安装系统前后的 QPS 和响应时间，对比时间差异。

注：QPS（每秒查询率）是对一个特定的查询服务器在规定时间内所处理流量多少的衡量标准。

纵向对比测试：

测试方案：使用 sqlmap 的绕过 WAF 功能，使用变形的 sql 注入对本系统进行攻击，查看 WAF 子系统和 RASP 子系统各自的拦截情况。

测试指标：两个子系统对变形的 sql 注入攻击的 payload 拦截个数，分别计算拦截率。

测试用例	用例路径
001 - 列目录操作 - scandir 方式	001-dir.php
002 - 任意文件读取 - file_get_contents	002-file-read.php
004 - 命令执行 - exec 方式，无回显	004-command-1.php
004 - 命令执行 - system 方式	004-command-2.php
005 - 任意文件写入 - file_put_contents 方式	005-file-write.php
008 - 任意文件上传 - move_uploaded_file 方式	008-file-upload.php
009 - 文件重命名 - rename 方式	009-file-rename.php
010 - 任意文件包含 - include 方式	010-file-include.php
012 - SSRF - cURL 方式	011-ssrf-curl.php
012 - SQL 注入测试- MySQLi 方式	012-mysqli.php
013 - WebShell - 回调类型后门	013-webshell-array_walk.php
014 - WebShell - 中国菜刀	014-webshell-eval.php
015 - WebShell - 文件上传小马	015-webshell-dropper.php
016 - WebShell - 文件包含方式	016-webshell-include.php
017 - XSS - 反射型XSS	017-xss.php

图 4.1 可行性测试样例

当攻击被拦截，RASP 系统会显示特定的拦截页面，以及当前 request ID，事后可根据这个 ID 去拦截日志里查找，定位原因。

4.3 测试结果分析

可行性测试结果，样例测试均通过，拦截率为 100%。

性能损耗测试测试结果如下表所示：

表 4.2 性能测试结果

指标	未安装 本系统	安装 本系统	影响比例
QPS	20550	20200	-1.7%
响应时间	233ms	237ms	-1.72%

在纵向对比中，对于变形的 sql 注入 payload，本系统的 WAF 组件被成功绕过，未能拦截住该攻击行为，拦截率为 0%：

利用 mysql 注释和注释特性绕过 WAF 策略：在 ModSecurity默认策略中，union select 这类 Sqli 注入语法一般是贪婪匹配 union 和 select 之间的字符、数字、下划线、左括号，很容易通过插入 /* */注释绕过;当变更规则过滤掉注释，依然可以通过在注释中使用 ！加版本号包裹关键词来绕过检测，因为只要 mysql 的当前版本等于或大于该版本号，则该注释中的 sql 语句将被 mysql 执行。

图4.2 绕过ModSec

但是尽管该攻击行为绕过了 WAF 组件，但是该 payload 并不能绕过 RASP 检测组件，被成功拦截，拦截率为 100%，由该横向对比表明该纵深防御系统相比传统 WAF 检测具有明显的优越性。


当前访问疑似黑客攻击，已被网站管理员设置为拦截
当前网址：http://www.leo-blog.cn/?id=1-1%23
客户端特征：Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/18.17763
拦截时间：2019-05-26 19:51:57 本次事件ID 31005

图 4.3　RASP 拦截

第五章 创新性说明

本系统提供的 Web 应用纵深防御解决方案，将 RASP（应用运行时自保护）技术和 WAF（Web 应用防火墙）技术相结合，将 RASP 技术的相关防御思想结合 Web 应用环境来突破传统防御手段所无法解决的相关技术瓶颈，二者结合达到纵深防御的目的，相较于传统的防御系统，极大地改善了误报率高、部署复杂、灵活性差等问题。总体而言，本作品的创新性体现在如下几个方面：

1.应用纵深防御思想的安全检测系统

本系统采用二级纵深防御设计方案，分别为一级防御子系统（WAF子系统）和二级防御子系统（RASP子系统），RASP技术与传统WAF技术内外配合、相互补充，达到纵深防御的目的，从而积极应对针对Web应用的种种复杂的入侵攻击。

2.广度与深度并存的保护引擎集成

本系统通过在 PHP 执行周期中挂钩关键函数，获取底层函数的参数和堆栈上下文，深度监控应用执行流。在数据库、网络、文件系统等多个层面，对应用进行全面的监控和防护，提高对 Web 应用的安全防御能力。

3.零规则的漏洞检测和防御方式

WAF 和数据库防火墙本质上是通过分析网络流量，通过正则识别 http 协议和数据库协议中的数据，通过静态规则识别注入的攻击代码， RASP 实现了降维防护，直接工作在后端应用的层面处理底层的数据库、文件读取等操作，成功的实现了多种零规则检测攻击算法从而有效对抗未知 0day 攻击。

4.高性能、低损耗的安全解决方案

在高并发压力下，本系统的接口响应时间延迟 1~7ms，可见本系统对宿主系统的性能损耗降到最低，此外本系统经过了小型高并发测试，基本通过所做测试指标，达到高性能和低损耗的目标。

第六章 总结

6.1 总结与改进

WAF 是一种传统的安全防御方案，对网络流量进行实时收集、匹配和阻断，WAF 系统需保持着对检测规则的实时更新，才可以防范更新的攻击行为。WAF 系统具有以下特点：1.对流量使用特定规则识别（也有一定程度智能识别变种的）最终基于特征 2.外部入口识别。WAF 具有部署简单、性能损耗小等优点，然而也存在变种可绕过规则、误报（攻击者的扫描行为）等缺陷。

RASP 是一种新型应用安全防护技术。这种技术直接将防护引擎嵌入到应用内部，能够感知应用上下文，并结合语义引擎、用户输入识别等能力，实现对攻击行为的检测。RASP 系统的主要有以下特点：1.基于攻击行为、动作(OGNL) 2.内部程序。具有无规则易维护、误报低、辅助开发等优点。然而 RASP 系统的引擎结合规则的检测本质导致了下面几处问题：信任问题：RASP 需要在服务器上增加各种客户端程序，相关重点安保单位会产生不信任感；问题及兼容性问题： WAF 虽然在安全检测方面较为落后，但是对客户来说，RASP 则需要适配不同的框架，不同的语言，这样看来是优势，其实也是最大的劣势。稳定性方面：由于 RASP 系统的深入定制特性，可能会导致因为和业务逻辑紧密结合导致的及其严重的稳定性问题（信任问题、性能问题、兼容问题、部署问题、稳定问题）。

在日后的改进中，对 WAF 系统来说，应配合机器学习和深度学习技术做到对规则的智能总结甚至自学习，从而提高 WAF 对未知攻击的检测能力；对 RASP 系统来说，应进一步提高系统兼容性，减轻部署压力，多次实验提高提高系统稳定性；对于整体纵深防御架构来说，应该增加对于接入层的准入子系统，进一步收集主机网络日志、部署蜜罐和构建安全情报中心、应急响应中心从而组建安全态势感知子系统，进一步扩充纵深防御的深度和广度。

6.2 展望

随着信息技术的飞速发展，网络安全技术也随之变幻。数十年来信息安全领域在以下方面不断演化：

存储技术从单纯的文件系统和关系型数据库到 HDFS 和 NoSQL 再到今天的 ELK 和区块链；分析技术从早初的正则表达式配合黑白名单到沙箱和威胁情报再到现在的语义分析与机器学习。而在这些技术演化中渗透着数据源的延伸和问题域的扩大：数据源从早期的安全设备日志、网络流量到 IoC 和漏洞情报等外部情报再延伸到现在的一整个社交网络和 IoT；问题域也从一开始的基础运维和基础攻防扩大到业务安全甚至企业运营方面。工程越来越大，处理越来越复杂， 防御越来越低效 ，攻防失衡并未得到扭转。

相信未来化繁为简、智能感知、主动防御，建设大数据与 AI 驱动的安全运营中心建设会成为大势所趋。

通过 ELK+HDFS 聚合主机、应用日志、网络流量，利用 AI 自动判断、简化威胁发现、聚合告警、事件合并。辨别主动化随机扫描，锚定针对性进攻，双向全流量交织验证，并主动进行内部资产梳理、智能辨认非合规利用，结合全球社区情报驱动，构建网络安全命运共同体。

攻击与防御相随相生，安全挑战与机遇永远并存。我们需要通过态势感知、威胁情报结合人工智能技术和拟态防御思想，筑建起网络安全长城，保障人民利益，构建网络安全命运共同体，不断深化网络安全交流合作。

参考文献

[1]国家计算机网络应急技术处理协调中心，2018 年我国互联网网络安全态势综述，2019.4

[2]360 威胁情报中心， 2017 中国网站安全形势分析报告， 2018.1

[3]管磊，胡光俊，王专 . 基于大数据的网络安全态势感知技术研究 [J]. 信息网络安全， 2016（9）：45-50.

[4]琚安康 郭渊博 朱泰铭.基于开源工具集的大数据网络安全态势感知及预警架构[J].计算机科学，2017,44(5):125-131

[5]仝青,张铮,张为华,邬江兴.拟态防御 Web 服务器设计与实现[J].软件学报,2017,28(4):883−897.

[6]石伟.一种新的 Web 服务防护模型[D],河北：河北大学，2018

[7]韩甜甜，基于 SVM 的 web 攻击防御框架扩展研究[D],兰州：兰州理工大学，2018

[8]杨爱华,张茂红,Web 应用防火墙在网上银行系统中的应用[J],计算机与网络,2019（4）：67-72

[9]陈英杰，李世武，基于安全环的一站式 Web 应用安全防范研究[J],计算机与网络，2019（6）：69-72

[10]林龙成，PHP Web 应用程序开发中漏洞消减技术研究[D],南京：南京师范大学，2014

[11]李双远，谈国胜.面向 Web 的攻击方式及防御对策[J],吉林化工学院学报，2019,36(3):47-53

[12]Gartner. Top 10 Security Predictions 2016，2016

[13]Aoe, J. An Efficient Digital Search Algorithm by Using aDouble-Array Structure[J]. IEEE Transactions on SoftwareEngineering. Vol. 15, 9 (Sep 1989). pp. 1066-1077.