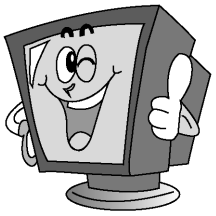

第 1 章　Web 信息安全基础

1.1　当前 Web 信息安全形势

从互联网诞生开始，Web 信息安全问题就随之产生，随着互联网的规模越来越大，应用越来越广，Web 信息安全的影响也越来越大，其破坏性造成的损失也越来越大。Web 信息安全从互联网产生至今经历了下面几个过程。

1. Web 信息安全“墙”时代亡羊补牢的无奈

说到 Web 信息安全，最早可追溯到互联网诞生时期，那时还是一个黑客备受尊敬和崇拜的时代。但由于 Web 业务所蕴含的信息量越来越大、价值越来越高，Web 平台不仅成为黑客们练手的“训练场馆”，还因为巨大的现实利益而沦为“黑色产业”中的“庄稼地”，不法分子利用各种网络平台的漏洞获得控制权，轻则留下“到此一游”的标记，重则将 Web 供应商保存的机密敏感信息、数据层层洗劫。并且，最恐怖的是大规模的恶意攻击，由于网络蠕虫病毒具有极大的传播性，其引发的恶劣后果每年都会引起全球媒体的关注。“世界上没有绝对安全的系统”这句名言揭示了这场对抗的起源。

Web 应用防火墙（WAF）的诞生在一定程度上加固了 Web 信息安全防护岌岌可危的堡垒，但这种从一开始就建立在规则匹配的亡羊补牢式防护，在很多情况下不尽如人意。所以，早期的 Web 防护形象地说就是一道形同马奇诺防线的“墙”，“墙”是死的而黑客是活的，很容易被绕过，这也意味着“墙”时代的传统 WAF 产品已经无法满足当前复杂的 Web 信息环境。

2. Web 信息安全“智”时代破釜沉舟的选择

业内对 NGWAF（应用程序防火墙）产品的呼唤由来已久，判断 WAF 产品核心性能——是否拥有智能化的攻击、检测、判断能力，如何提升这一核心竞争力？业内公认的方向集中在语义分析技术、机器学习技术和自学习技术上。机器学习技术和自学习技术是浅层次上的和基于概率控制的识别，相对而言较容易实现。而语义分析技术相较于机器学习、自学习技术而言是一个深度挖掘的过程，类似于人类认知、思考、判断的行为，自然也最难实现，毕竟在人工智能的终极问题没有解决之前，这一领域依旧停留在大数据层面。

但长亭科技这家公司从开发 WAF 产品开始，就是冲着 NGWAF 中最难落地的语义分析

技术去的，这家公司的员工显然抱着“不成功便成仁”的态度和拥有“破釜沉舟”的勇气，当然最重要的还是这群年轻人的集体智慧。经过潜心研究后，人工智能语义分析引擎 Demo 得以诞生。该引擎在 2015 年登上世界安全峰会 Black Hat USA 的舞台，其研究成果“新型 SQL 注入检测与防御引擎 SQLChop”被纳入军械库展示，这个针对黑客攻击实现智能识别和拦截的创新点与实际效果得到全球安全专家的一致认可。2016 年 7 月，长亭科技公司基于该技术的雷池（SafeLine）正式推出，相当于在平静多年的 Web 信息安全领域投下一颗石子，一石激起千层浪，以实力获得全球顶级安全赛事、峰会等主办方的青睐和赞誉。

3. Web 信息安全“云”时代创新能力的对决

这依然不是 Web 信息安全防护的最终形态，与其说 2018 年之前长亭科技公司的雷池是智能动态防御技术的顶尖代表，那么 2018 年宣布升级的雷池则将“云”时代的“智能”落地到更深的层次。

在企业对“云服务”接受程度不断提升的今天，如何从根本上规避云 WAF 轻易被绕过的风险，解决其可靠性低和保密性低的弊端，是业界长期以来的难题。长亭科技公司在雷池智能语义分析技术的基础上升级云端部署解决方案，不改变原有网络结构，实现了软件层面的灵活拓展。

雷池相比其他云 WAF 的优势在于：一方面，雷池的 WAF 服务器部署在企业私有云，与 Web Server 处于相同 VPC（Virtual Private Cloud，虚拟私有云）中，不需要通过解析用户的流量到云节点来实现防护，不存在强制解析域名问题；另一方面，处理过程只需一个环节，不需要过多的环节协同工作，最大限度地减小了出现问题的可能性。并且，雷池云端部署在非第三方云服务平台，数据处理转发完全在企业私有云内部，保密性自然毋庸置疑。此外，雷池（SafeLine）围绕不同类型用户业务类型而匹配的动态化、个性化、定制化模块和衍生服务，始终遵循一种“化繁为简”的科技理念。

显然，在 Web 信息安全“云”时代，这种技术创新能力的对决也深入到各个层面。以金融行业为例，目前主要面临数据泄露、APT 攻击、DDos 攻击、Web 攻击等多种网络威胁，其中绝大部分攻击均指向关键服务器，以试图获取包括客户信息、机密交易数据在内的重要商业信息。还有很大一部分则是僵尸网络、“羊毛党”们的“薅羊毛”行为（“薅羊毛”是目前绝大多数流量网站每时每刻都会遇到的难题），单一的 WAF 产品显然很难实现与客户的业务模式的产业交互，甚至传统 WAF 还会直接影响客户业务的正常运行，毕竟传统 WAF 的规则匹配和多环节协同加载是个难题。

长亭科技公司显然在这方面做足了功课，雷池能一炮而红不仅得益于其独一无二的技术创新，而且得益于其安全团队持续性、纵深化的服务能力，如雷池此次升级后成为业内首个支持私有云 WAF 部署的 NGWAF 产品、基于多年的顶级国际黑客赛事经验推出的洞鉴（X-Ray）安全评估系统，未来还会有相应的安全人才培养计划。长亭科技公司将不仅是一家初创型信息安全企业，而且承担着通过技术创新实现全球网络安全行业向前发展的重任，在网络安全上升为国家安全战略的当下，中国各行业、产业乃至整个社会，都乐见并希望这样的企业相继涌现。

现在人们在使用互联网时，个人的安全意识已经提高很多。大多数互联网企业为了自己的客户安全，在信息安全方面的投入也越来越多，但现今 Web 信息安全还存在下面几个问题。

（1）网络钓鱼激增

Webroot 调查研究发现，全球 IT 决策者认为，网络钓鱼已经取代了其他新型恶意软件，成为目前企业最容易遭受的攻击。虽然网络钓鱼已存在多年，但曾经不在网络钓鱼攻击者目标范围内的中小企业如今已不再获免，它们往往被当作进入大型企业的跳板而被攻击。

（2）勒索软件问题深化

Webroot 研究发现，在 WannaCry（一种勒索病毒）时代，中小企业心中的威胁排行榜上，勒索软件 2018 年从第五位升到了第三位，而英国的中小企业更是将勒索软件列在了最易遭受的攻击类型第一的位置。Webroot 称，这些结果遵循了他们看到的市场现象，过去的一段时间里他们的员工基本忙于处理勒索软件事件。

对很多小公司而言，被勒索软件攻击已成了他们的“恐慌事件”，很多情况下他们会选择支付赎金。然而，即便支付了赎金，诈骗犯们也可能只还给他们 50%的文件，有时候甚至一份文件都不恢复。

（3）内部威胁减少

距离斯诺登事件爆发已有 5 年，大部分中小企业不再对内部威胁毫无防备。Webroot 调查显示，全球仅 25%的公司称内部威胁依然成为问题。过去几年中大部分公司都开展了积极的信息安全教育项目，企业更小心谨慎地对待权限授予问题，雇员也更加了解来自内部的威胁。

相比大型咨询公司或拥有数千员工的国际承包商，中小企业这种员工间对彼此业务都很熟悉的环境，更容易遭受内部威胁。

（4）新型恶意软件担忧持续

Webroot 对 3 个国家安全人员的调查表明，新形式的恶意软件感染仍然是安全人员关心的重点。在美国，担忧新型恶意软件的人员占比为 37%，澳大利亚为 34%，英国为 32%。攻击者持续推出新型恶意软件，让安全公司忙于跟进。现在的情况显然与 5 年或 10 年前大不相同。在过去，安全人员添加一个病毒特征码就能挡住一个已知恶意软件。今天，很多新型恶意软件能动态改变特征码，当前威胁环境变得极为棘手。

（5）培训项目并不持续

大多数企业的信息安全培训项目没有保持连贯性。例如，信用卡公司就没跟进年度 PCI 培训。企业要么培训一遍就完事，要么只对 CEO 或董事做培训，而将负责具体事务的员工排除在外。

Webroot 信息安全培训的方法是在每次事件发生时插入培训内容。例如，当某员工单击了恶意链接，系统就会弹出一段 2 分钟的可疑链接单击后果教育视频。在事件发生时进行培训，会让员工更容易记住教训，也让公司节省了大量工作时间搞培训。而最糟糕的培训方式，就是所谓的“照单画钩”式培训——每年搞一两次形式化的培训，没人认真对待，效果很差。

（6）信息安全事件损失下降

Webroot 和卡巴斯基的研究在信息安全事件的损失额度上出现了分歧。Webroot 报告称信息安全事件平均损失为 52.7 万美元，下降了 9%，而卡巴斯基将这个数字定在了 12 万美元。不过，卡巴斯基称，企业规模不同，信息安全事件所致损失数额也有较大差异，员工数在 500 人以下的中小企业平均损失为 20 万美元，500～999 人规模的中小企业遭遇信息安全事件的平均损失约为 100 万美元。企业计算信息安全事件损失时，还必须考虑罚款、律师费、缓解工作开支和信誉损失所致的业务损失。

（7）信息安全预算增长

卡巴斯基指出，中小企业信息安全预算从 2017 年的 20.1 万美元增长到了 2018 年的 24.6

万美元。小微企业信息安全预算涨幅最大，从 2400 美元增加到 3900 美元。这表明，即便是最微小的公司，如今也开始正视 IT 安全问题。

卡巴斯基称，小公司往往负担不起聘请年薪 15 万～20 万美元的 CISO（首席信息安全官），但越来越多的小公司开始赞同业内流行的“CISO 租赁”概念。公司企业可以临时聘请 CISO 来搞培训，或者评估它们的整体安全准备程度，然后由 CISO 定期回访，查看公司安全的进展。

（8）代价最高昂的安全事件发生在云提供商身上

卡巴斯基的报告显示，影响第三方托管 IT 基础设施的攻击，是中小企业面临的代价最高昂的威胁之一。首先，中小企业平均要花费 11.8 万美元才能从此类攻击中恢复，其次就是涉及非计算型物联网设备的事件平均花费 9.8 万美元。AWS 和微软 Azure 之类的大型公有云提供商实力雄厚，而很多终端解决方案云提供商并没有把安全作为头等大事来对待。

（9）技术复杂性驱动安全投资

卡巴斯基报告称，超过 1/3 的企业将 IT 基础设施复杂度的增加和提升专业安全知识的需求作为投资网络安全的动机。在边界上搭建防火墙来保护护城河的时代一去不复返。今天，移动性驱动业务应用的方方面面都依赖 IT 技术。有太多的 IT 基础设施需要保护，太多的设备和应用需要锁定。于是，专门精于某方面安全技能的安全人员投入也就更大了，DDoS 攻击、网络钓鱼、Office 365、云、IoT，各方面都需要相应的安全技术人才。

1.2 Web 信息安全防护技术

针对 Web 信息安全常见的攻击方式有 SQL 注入攻击、文件上传攻击、XSS 跨站脚本攻击、CSRF（Cross-site Request Forgery）跨站请求伪造、程序逻辑漏洞、DDOS、暴力破解等。黑客在成功渗透到服务器后还可以进行 C 段攻击，某台服务器被攻陷后通过内网进行 ARP、DNS 等内网攻击。

针对 Web 信息安全问题的防御方法为强化口令、代码加固、网页防篡改、WAF、身份鉴别访问控制等。另外，对 Web 信息采用 Web 安全审计系统（WAS）进行安全审计，采用 Web 应用防护系统（HWAF）进行安全监控与恢复。最基本的防御原则就是永远不要相信用户提交的数据（包括 header\cookie\sessionid）。

下面以几个典型的 Web 信息安全攻击方法进行原理解析与防御方法分析。

1. SQL 注入

SQL 注入见表 1-1。

表 1-1　SQL 注入

漏洞原理	SQL 注入通过构建特殊的输入作为参数传入 Web 应用程序，而这些输入大都是 SQL 语法里的一些组合，通过执行 SQL 语句进而执行攻击者所要的操作，其主要原因是程序没有细致地过滤用户输入的数据，致使非法数据侵入系统
漏洞分类	（1）平台层 SQL 注入：由不安全的数据库配置或数据库平台的漏洞所致。 （2）代码层 SQL 注入：程序员对输入数据未进行细致的过滤从而执行了非法的数据查询

续表

产生原因	（1）不妥当的类型处理。 （2）不安全的数据库。 （3）不合理的查询集处理。 （4）不妥当的错误处理。 （5）转义字符处理不合适。 （6）多个提交处理不当
防范方法	（1）对用户的输入进行校验，可以通过正则表达式，或者限制长度；对单引号和双“-”进行转换等。 （2）不要使用动态拼装 SQL，可以使用参数化的 SQL。 （3）不要使用管理员权限的数据库连接，为每个应用使用单独的权限有限的账号进行连接。 （4）不要把机密信息直接存放，加密或用 hash 过滤密码和敏感信息。 （5）应用的异常信息应该给出尽可能少的提示，最好使用自定义的错误信息对原始错误信息进行包装
流程建议	（1）在部署应用系统前，始终要做安全评审。建立一个正式的安全过程，并且每次做更新时，要对所有的编码做评审。 （2）开发队伍在正式上线前会做很详细的安全评审，然后在几周或几个月之后他们做一些很小的更新时，他们会跳过安全评审这关，例如，“就是一个小小的更新，我们以后再做编码评审好了”。请始终坚持做安全评审
编码规范	Java：不允许直接根据用户输入的参数拼接 SQL 的情况出现，直接使用 PreparedStatement 进行 SQL 的查询；并且需要对输入的参数进行特殊字符的过滤。使用 Hibernate 等框架的，可以使用参数绑定等方式操作 SQL 语句。但是同样不允许直接使用拼接 SQL 语句。 PHP 编码规范：数据库操作，如使用框架进行处理，必须使用框架中提供的 sqlTemplate 或 paramBind、mysqli::preparesatement 等方式进行 SQL 语句的参数值注入（绑定），不要直接使用参数拼接原始 SQL 语句。不使用数据库操作类直接操作原始 SQL 语句，必须使用 Intval 对整数型参数过滤，使用 mysql_real_escape_string 对字符串型进行过滤，并要配合 mysql_set_charset 设置当前字符集
测试方法	基于编码规范部分进行 CODE REVIEW；小的项目边缘业务使用扫雷平台进行扫描；核心业务在扫雷平台的基础上使用 sqlmap 进行安全测试扫描

2. CSRF

CSRF 见表 1-2。

表 1-2 CSRF

攻击对象	应用程序的其他用户，属于客户端漏洞
漏洞原理	通过伪装成受信任用户的请求来利用受信任的网站，伪造客户端请求的一种攻击，攻击者通过一定技巧设计网页，强迫受害者的浏览器向一个易受攻击的 Web 应用程序发送请求，最后达到攻击者指定的操作行为
漏洞危害	在受害者毫不知情的情况下以受害者名义伪造请求发送给受攻击站点，从而在并未授权的情况下执行在权限保护之下的操作，危害用户资金信息安全
防范方法	验证 HTTP Referer 字段，存在问题：验证 Referer 值的方法，就是把安全性都依赖于第三方（浏览器）来保障，从理论上来讲，这样并不安全，因为浏览器也有漏洞，即 Refer 被篡改的情况下不可靠。对于提交的 form 表单服务器生成 CSRF TOKEN，不能使用 GET 请求更新资源，使用$_POST 请求获取 post 资源
测试方法	CODE REIVIEW，根据 Java 语言开发编码规范：对于改写数据类的提交请求，需要对请求的来源真实性进行验证。如果使用 struts 框架，可以使用框架提供的 TOKEN 机制。如未使用，可以参考其机制自行实现

3. URL 跳转

URL 跳转见表 1-3。

表 1-3　URL 跳转

攻击对象	客户端，该网站的其他用户
漏洞原理	服务器未对传入的跳转 URL 变量进行检查和控制，可能导致意外构造任意一个恶意地址，诱导用户跳转到恶意网站
漏洞危害	由于 URL 是从可信的站点跳转出去的，用户会比较信任，所以跳转漏洞一般用于钓鱼攻击，通过转到恶意网站欺骗用户输入用户名和密码盗取用户信息，或者欺骗用户进行金钱交易；也可能引发 XSS 漏洞（主要是跳转常常使用 302 跳转，即设置 http 响应头，Location:url，如果 URL 包含了 CRLF，则可能隔断了 http 响应头，使得后面部分落到了 http body 部位，从而导致 XSS 漏洞）

防范方法	① 如果需要跳转的 URL 可以确定，可在后台配置，客户端传入 URL 索引，服务器根据索引找到具体的 URL 再跳转。
	② 如果服务器生成 URL，链接生成的 URL 之后再进行签名，签名通过再跳转。
	③ 如果只能传入前端参数，是否符合授权白名单规则
	百度网站上的建议修复方案:function checkurl($url) {
	if ($url != '') {
	$urlParse = parse_url($url);
	$urlHost = strval($urlParse['host']);
	if
	(!preg_match("/\.baidu\.com$|\.baidu\.com\:|\.baidu\.com\.cn$|\.baidu\.com\.cn\:|\.baidu\.cn$|\.baidu\.cn\:/i", $ urlHost)) {
	return false;
	} else {
	return true;
	}
	} else {
	return false;
	}
	}

测试方法	使用工具进行黑盒扫描

4. 路径遍历

路径遍历见表 1-4。

表 1-4　路径遍历

攻击对象	服务器
攻击原理	Web 应用程序一般会对服务器的文件进行读取查看，大多会用到提交的参数来指明文件名，如 http://www.nuanyue.com/getfile=image.jpg，当服务器处理传送过来的 image.jpg 文件名后，Web 应用程序会自动添加完整路径，如 d://site/images/image.jpg，将读取的内容返回给访问者。由于文件名可以任意更改而服务器支持“~/”“/..”等特殊符号的目录回溯，从而使攻击者越权访问或覆盖敏感数据，如网站的配置文件、系统的核心文件，这样的缺陷被命名为路径遍历漏洞，例如，恶意攻击者会利用对文件的读取权限进行跨越目录访问，访问一些受控制的文件，如“.../etc/passwd”或“.../boot.ini”，如果对用户的下载路径不进行控制，将导致路径遍历攻击，造成系统重要信息泄露，并可能对系统造成危害

续表

防范方法	（1）数据净化，对网站用户提交的文件名进行硬编码或统一编码，对文件后缀进行白名单控制，对包含恶意的符号（反斜线或斜线）或空字节进行拒绝。 （2）Web 应用程序可以使用 chrooted 环境进入包含访问文件的目录，或者使用“绝对路径+参数”方式来控制访问目录，即使越权跨越目录也要在指定的目录下
测试方法	路径遍历漏洞允许恶意攻击者突破 Web 应用程序的安全控制，直接访问攻击者想要的敏感数据，包括配置文件、日志、源代码等，配合其他漏洞的综合利用，攻击者可以轻易地获取更高的权限，并且这样的漏洞也是很容易发现的，只要对 Web 应用程序的读写功能块直接手工检测，通过返回的页面内容来判断，这是很直观的，利用起来也相对简单

5. 文件上传漏洞

文件上传漏洞见表 1-5。

表 1-5　文件上传漏洞

攻击对象	Web 服务器
漏洞原理	由于服务器没有对用户上传的文件进行正确的处理，导致攻击者可以向某个可通过 Web 访问的目录上传恶意文件，并且该文件可以被 Web 服务器解析执行。 利用该漏洞产生攻击的条件：具体来说就是存放上传文件的目录要有执行脚本的权限，用户能够通过 Web 访问这个文件
漏洞危害	文件上传攻击是指攻击者利用 Web 应用对上传文件过滤不严，导致可以上传应用程序定义类型范围之外的文件到 Web 服务器上。如可以上传一个网页木马，如果存放上传文件的目录刚好有执行脚本的权限，那么攻击者就可以直接得到一个 Webshell。 Webshell 解释：以 asp、php、jsp 或 cgi 等网页文件形式存在的一种命令执行环境，取得对服务器某种程度上的操作权限，黑客在入侵了一个网站后，常常将这些 asp 或 php 木马后门文件放置在网站服务器的 Web 目录中，与正常的网页文件混在一起。然后黑客就可以用 Web 的方式，通过 asp 或 php 木马后门控制网站服务器，包括上传下载文件、查看数据库、执行任意程序命令等。再通过 DOS 命令或植入木马后门，通过服务器漏洞达到提权的目的。
防范方法	客户端检测：在上传页面里含有专门检测文件上传的 javascrIPt 代码，在文件被上传之前进行检测，最常见的就是检测上传文件的文件类型和规格是否合法。仅仅作为辅助手段，不完全可靠。 服务器检测：这类检测方法通过检查 http 包的 Content-Type 字段中的值来判断上传文件是否合法。 服务器文件扩展名检测：这类检测方法通过在服务端检测上传文件的扩展名来判断文件是否合法。 服务器目录路径检测：这类检测一般通过检测路径是否合法来判断。 服务器文件内容检测：这类检测方法相对于上面 4 种检测方法来说是最为严格的一种。它通过检测文件内容来判断上传文件是否合法。这里，对文件内容的检测主要有两种方法。其一，通过检测上传文件的文件头来判断。通常情况下，通过判断前 10 字节，基本就能判断出一个文件的真实类型。其二，文件加载检测，一般调用 API 或函数对文件进行加载测试。常见的是图像渲染测试，再严格点的甚至是进行二次渲染