Web 攻击及安全防护技术研究

随着企业和政府越来越多的业务系统采用基于 WEB 服务方式，互联网在为用户提供方便快捷的同时，针对 WEB 业务的攻击亦在迅猛增长。一方面，基于新技术的突破和应用，WEB 技术提供的快捷性、交互性和通用性被越来越多的业务系统所采用；而另一方面，病毒、木马蠕虫、钓鱼软件等却又通过 WEB 服务的方式大肆在互联网上传播，严重威胁到了 WEB 服务的业务系统。

1 .WEB 攻击

常见的 WEB 攻击可分为三类：

一是利用 WEB 服务器的漏洞进行攻击，如 CGI、缓冲区溢出、目录遍历漏洞利用等攻击；

二是利用网页自身的安全漏洞进行攻击，如 SQL 注入、跨站脚本攻击、Cookie 假冒、认证逃避、非法输入、强制访问、隐藏变量篡改等；

三是利用僵尸网络的分布式 DOS 攻击，造成网站拒绝服务。

1.1 表单篡改

HTML 表单是应用层的安全隐患。主要原因是 WEB 应用的设计者总是信任用户输入是在 HTML 表单的格式限定内的良好的、无恶意的数据。

HTTP 协议的无状态使得设计者需要管理多次 HTTP 请求间的应用状态。通过使用 HTTP 表单的隐藏域可以把一系列的请求响应串起来，比通过使用后端的数据库的方法要容易。可是这样使用隐藏域就使得客户端可以修改应用的状态。例如：目前 Internet 上不需要特殊技巧实现对有些 WEB 站点的攻击，攻击者可以把 HTML 表单存到本地磁盘，然后用文本编辑器改变隐藏在表单中隐

藏域的商品价格或是商品的打折比例，再把修改了的 HTML 表单装入浏览器发送回 WEB 服务器就可以买到便宜的商品。 有不少的商业站点可以用这种简单的方法进行攻击，修改表单的攻击通常和其他的攻击方法混合使用。

1.2 SQL 注入攻击

WEB 应用一般是用客户输入数据来构造 SQL 查询语句，用户输入数据如果不经过严格的检查，就可以利用 SQL 语言来攻击后端数据库[2]。 例如，服务器端的 SQL 语句如下：

```
SQL="SELECT count(*)
FROM client
WHERE name=$name and pwd=$pwd "
```

将用户输入的姓名和密码传给变量$name 和$pwd，如果返回的结果大于等于 1 就认为用户是合法的用户。

用户输入如下的字符串 'my'or'1'='1' 得到的 SQL 语句就是：

```
SQL="SELECT count(*)
FROM client
WHERE name='my'or'1'='1'and pwd='my'or
'1 '='1 '"
```

WHERE 表达式总为真，所以攻击者在不知道有效的用户名和密码的情况下就可以通过身份验证。 此种攻击方法是多种多样的。

1.3 跨站点的脚本攻击

跨站点的脚本攻击指的是用户向动态的 WEB 应用提交有恶意的 HTML

语言，通常还包括恶意的脚本。 恶意的 HTML 可能嵌入在 URL 的参数、表单域或 cookie 中[3]。其他用户浏览来自动态 WEB 应用带有恶意脚本的页面，当然用户是认为自己看的页面是一个可以信任的站点，这样攻击者通过这个动态 WEB 站点攻击了另外一个使用该站点的用户，而这些暗藏的跨越站点的脚本是很危险的。 跨站点脚本的攻击包括窃取秘密信息，改变表单的行为并且暴露 SSL 的连接。 跨站点的脚本攻击可以通过严格限定用户输入来防止恶意脚本，但是对于大型的应用，确保所有用户的输入都是符合要求是一个艰巨的任务。

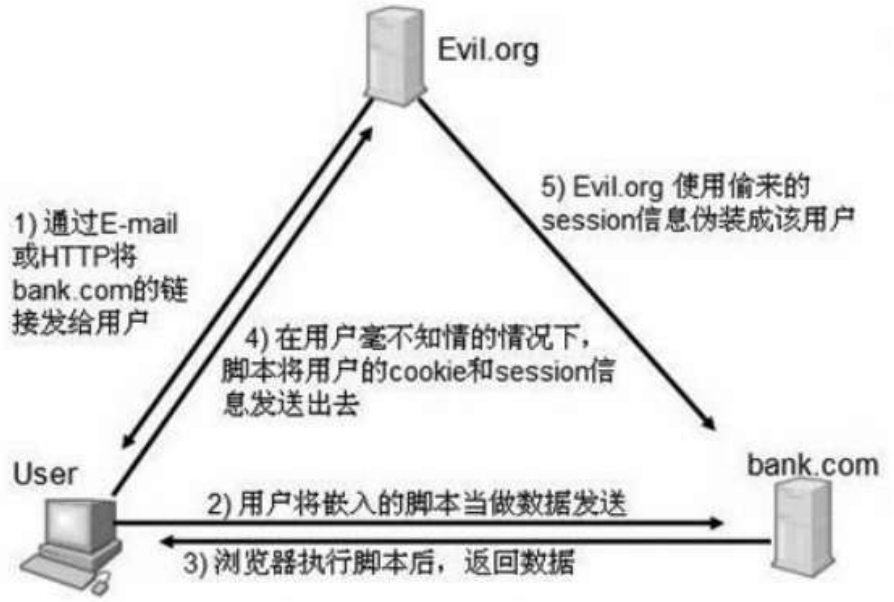

图 1 跨站点脚本攻击的过程

跨站点脚本攻击的过程如图 1 所示。

恶意攻击者（这里使用 Evil.org 表示）通过 E-mail 或 HTTP 将某银行的网址链接发给用户（银行用 bank.com 表示），该链接中附加了恶意的脚本（图 1 步骤一）；

用户访问发来的链接，进入银行网站，同时，嵌在链接中的脚本被用户的浏览器执行（图 1 步骤二、三）；

用户在银行网站的所有操作，包括用户的 cookie 和 session 信息，都被脚本收集到，并且在用户毫不知情的情况下发送给恶意攻击者 (图 1 步骤四)；

恶意攻击者使用偷来的 session 信息，伪装成该用户，进入银行网站，进行非法活动（图 1 步骤五）。

只要 Web 应用中，有可被恶意攻击者利用执行脚本的地方，都存在极大的安全隐患。黑客们如果可以让用户执行他们提供的脚本，就可以从用户正在浏览的域中偷到他的个人信息、可以完全修改用户看到的页面内容、跟踪用户在浏览器中的每一个动作，甚至利用用户浏览器的缺陷完全控制用户的机器。目前，跨站点脚本攻击是最大的安全风险。

1.4 缓冲区溢出

攻击者利用超出缓冲区大小的请求和构造的二进制代码让服务器执行溢出堆栈中的恶意指令[4]。 缓冲区溢出的目的在于扰乱具有某些特权运行程序的功能，这样就可以让攻击者取得程序的控制权，如果该程序具有足够的权限，那么整个主机甚至服务器就被控制了。

在被攻击程序地址空间里安排攻击代码的方法有两种：

一是植入法，攻击者向被攻击的程序输入一个字符串，程序会把这个字符串放到缓冲区里，这个字符串所包含的数据是可以在这个被攻击的硬件平台运行的指令流，在这里攻击者用被攻击程序的缓冲区来存放攻击代码；

二是利用已经存在的代码，有时候攻击者所要的代码已经存在于被攻击的程序中了，攻击者所要做的只是对代码传递一些参数，然后使程序跳转到想要执行的代码那里。

2 网页安全防护技术

2.1 网页防篡改系统

对基于 OS（Windows、Linux、Unix）的 WEB 服务器、数据库支持系统和特定的中间件处理服务器进行底层加固、上层过滤，控制系统中进程的运行类别以及权限，对进程启动所需要的 exe、dll、sys 等文件进行完整性度量。 如符合策略标准，则允许其正常启动，否则阻止该程序的启动。当应用程序请求访问某些文件时，需要通过度量模块进行度量，判断是否满足访问权限。 这样在权限控制合理的前提下，应用程序只拥有最小权限，即使其被攻击者利用，对系统的威胁也是有限的.

同时，网页防篡改系统会限制 WWW 服务、FTP 服务等访问与 WEB 无关的其他任何目录及文件，有效的实现了 WWW 服务和系统中其他应用的逻辑隔离，从而降低 WWW 服务受攻击的概率。当终端用户的 HTTP 请求被 WWW 服务解析之前，过滤引擎首先将其捕获，并与策略规则库中进行特征匹配，如果确定该请求有攻击性质，则丢弃。 这样就可以有效防御诸如 SQL 注入、跨站脚本漏洞、CGI 等流行攻击，利用文件过滤驱动技术保护网页和动态脚本不被非法篡改，从而从源头上杜绝各类非法篡改行为。

2.2 Web 安全网关

Web 安全网关是是一种边界应用安全网关。 其主要功能包括防病毒、URL 过滤、Internet 应用控制和带宽管理等。 网关防病毒主要针对 HTTP/HTTPS、FTP、SMTP、POP3 等协议流量进行双向的过滤扫描，来达到对企业内网用户和服务器的保护，并防止内网已感染病毒的客户端和服务器对外扩散病毒。 HTTP 协议的检测性能是网关防病毒的关键性能指标。 网关 URL 过滤的实现机制是将客户端请求的 URL 与网关中的 URL 过滤策略进行匹配，从而达到过

滤控制的目的。

2.3 应用层防火墙

传统的网络安全设备对于应用层的攻击防范，作用十分有限。 所以，虽然在网络中部署了多层的防火墙，入侵检测系统（IDS），入侵防御系统（IPS）等设备，但是基于 WEB 应用的攻击事件仍然不断发生，其根本的原因在于目前的大多防火墙都是工作在网络层，通过对网络层的数据过滤（基于 TCP/IP 报文头部的 ACL）实现访问控制的功能；通过状态防火墙保证内部网络不会被外部网络非法接入。 所有的处理都是在网络层，而应用层攻击的特征在网络层次上是无法检测出来的。

随着攻击向应用层发展，传统网络安全设备不能有效的解决目前的安全威胁，网络中的应用部署面临的安全问题必须通过一种全新设计的高性能防护应用层攻击的安全防火墙——应用防火墙来解决。 应用防火墙通过执行应用会话内部的请求来处理应用层。 应用防火墙专门保护 Web 应用通信流和所有相关的应用资源免受利用 Web 协议发动的攻击。 应用防火墙可以阻止将应用行为用于恶意目的的浏览器和 HTTP 攻击。这些攻击包括利用特殊字符或通配符修改数据的数据攻击，设法得到命令串或逻辑语句的逻辑内容攻击，以及以账户、文件或主机为主要目标的目标攻击。 IDS，IPS 通过使用深包检测的技术检查网络数据中的应用层流量，和攻击特征库进行匹配，从而识别出以知的网络攻击，达到对应用层攻击的防护。

2.4 XML 文件验证技术

XML 作为一种专门在互联网上传递信息的语言，已经被广泛认为是继 Java 之后 Internet 上最激动人心的新兴技术。 XML 的语法与 HTML 的语法十分相近，事实上 W3C（World Wide Web Consortium，万维网联盟）发

展 XML 的目的并不是为了要取代 HTML，相对于 HTML 描述网页要如何呈现数据的样式，XML 则是描述数据的结构，同时发展 XML 是为了要提供使用者更有弹性也更具扩充性的标记语言，为此 W3C 在开发 XML 时的做法是不再提供预先定义的标记，XML 标记交由使用者自行定义；同时 XML 决定采用更严格的语法格式，以防止文件结构过于松散[5]。

XML 是具有可扩展性和协同合作能力的计算机语言。 XML 的设计人员在创造这种语言时，制定了相当严格的结构和语法规则。它有两个重要的规则，第一个规则是提供可扩充的标记定义及严格的语法格式(Well-Formed)；另一个规则是 XML 文件是可验证的(Valid)。 因为，XML 标记并不像 HTML 是已经预先决定好用途的，因此 XML 提供文件验证的机制来检查 XML 文件是否符合由使用者自行定义的文件结构与标记规则。 XML 文件的验证机制主要有二种，分别为 DTD(Document Type Definition) 与 XML Schema[6]。由于 XML 良好的文件结构定义，XML 为电脑文件开创了一个新的视野，只要配合相关的技术，XML 便可以用在文件内容的显示、作为数据交换格式、储存数据的格式及连结其他的资源。 相关的技术关联见表 1。

表 1 XML 相关技术表

XML用途	相关技术	说明
文件验证	DTD XML Schema	检查XML文件结构和标记内容是否符合规则
显示文件	CSS (Cascading Style Sheets) XSL / XSLT (Extensible Stylesheet Language)	产生报表或依指定条件提取出XML文件的内容
数据交换	SOAP(Simple Object Access Protocol)	在商业应用上，XML可用于分布式系统数据交换的格式
连结到其他资源	XPoint (XML Pointer Language) Xlink (XML Linking Language)	支持连结到XML、非XML文件，甚至是更复杂的连结

由上面所述可知，DTD 与 XML Schema 均为 XML 文件的验证机制。由于这两种文件验证机制均是描述 XML 文件架构的文件，因此可以把这两种文件验证机制视为一种 Metadata。

DTD 为 XML 最初的验证机制，目的是定义 XML 文件中元素的结构、元素标记与属性。 DTD 还是有一些不足，它采用独有的非 XML 语法，不能很轻易地支持命名空间，而且提供非常有限的数据类型，仅适用于属性。 因此 W3C 开始发展替代 DTD 的解决方案。

XML Schema 原先为由微软所提出的规格草稿， 如上述 XML Schema 原先是 W3C 为了改善 DTD 的问题而为 W3C 所大力推动的，与 DTD 相较，XML Schema 有许多优点，因此目前成为 XML 新一代的验证机制。

XML 为未来 HTML 4.0 最后的终结语言，可知 HTML 将不会有 5.X 以后的版本， 取而代之的是 XHTML 1.0 (HTML in XML)，

XHTML 是 HTML 与 XML 相互辅助的网页程序语言。

3 结束语

该文对 Web 攻击及 Web 网站安全技术进行了比较详细地分析。 随着新的安全问题不断出现，构建一个面面俱到的网站安全体系，仍需要不断地学习与实践。 网站的安全稳定运行，应侧重于预防，不断增强安全意识，及时地堵上各种安全漏洞，采取各种预防措施施，才能及时有效地排除安全隐患。