基于代理的 Cookie 保护技术研究

徐 璐[1]，周安民[1]，刘 亮[1]，周 妍[2]
（1. 四川大学 信息安全研究所，成都 610064；2. 四川省信息安全测评中心，成都 610017）

摘 要：以窃取客户端 Cookie 为目的的跨站脚本(XSS)攻击，使互联网用户的个人隐私和利益不断遭受侵犯。如何防御 XSS 攻击，以保障网民和互联网公司的利益，成为亟待解决的问题。针对基于代理的 Cookie 保护技术进行了研究，设计了基于代理的 Cookie 保护框架，阐述了框架的基本原理，给出了关键技术的实现方法，并实现了一个基于代理的 Cookie 保护系统，最后对该保护系统的有效性进行了测试。测试结果表明本系统可对 Cookie 提供可靠的保护，为跨站防御的研究提供了新的方向。

关键词：小型文本文件；跨站脚本攻击；保护；代理
中图分类号：TP309 **文献标志码**：A **文章编号**：1001-3695(2012)08-3036-03
doi:10.3969/j.issn.1001-3695.2012.08.060

Research of proxy-based Cookie protection technology

XU Lu[1], ZHOU An-min[1], LIU Liang[1], ZHOU Yan[2]
(1. *Institute of Information Security, Sichuan University, Chengdu* 610064, *China*; 2. *Information Security Assessment Center, Chengdu* 610017, *China*)

Abstract: Cross site script (XSS) vulneralbility attack that aims to steal cookie violates Internet users' privacy and Interests. It is urgent to defend XSS from damaging the interests of netizens and Internet firms. This paper researched the proxy-based Cookie protection technology, designed a proxy-based cookie protection framework, and both basic theories and the realization of key technologies. Then it implemented a proxy-based Cookie protect system. And it carried out validation on the effectiveness of this technology and the result shows that it is robust to protect cookie, and provided a new direction for XSS research.

Key words: Cookie; XSS; protection; proxy

当前互联网环境下，跨站脚本攻击(XSS)漏洞日益猖獗，互联网用户的个人隐私和利益不断受到侵犯，各大互联网公司的 Web 产品均存在不同程度的 XSS 漏洞。2011 年 6 月 28 日，新浪微博爆发大规模 XSS 蠕虫事件，大量用户自动发布含有垃圾信息的微博，其中包括新浪微博高级认证用户，数万网民遭受攻击，此次攻击造成了极大的损失和恶劣的影响。如何防御利用此类漏洞发起的攻击，以保障网民和互联网公司的利益，成为亟待解决的问题。

XSS 攻击近几年成为最流行的 Web 漏洞，相关防御方面的研究也取得了一定成效，比如跨站脚本攻击服务端的检测[1]、基于 B/S 模式的跨站脚本防御[2]等。但是，这些研究成果均存在不同程度上的不足，特别是对 Cookie 保护的研究上仍很缺乏。笔者通过分析现有 Cookie 保护措施的不足，设计并提出了一种新的 Cookie 保护框架，并在实际的互联网环境中对其进行了实现和测试，最后得出结论。本文所讨论的基于代理的 Cookie 保护技术，弥补了以往的研究在 Cookie 保护方面的不足，为 XSS 防御的研究提供了新的方向。

1 Cookie 与跨站脚本攻击

Cookie，中文名称为小型文本文件或小甜饼，通常 Cookie 作为 HTTP 请求包头部的一部分、Set-Cookie 作为 HTTP 响应包头部的一部分用于记录并保持客户端的状态[3]。

下面将描述一个简单的用户登录过程，借此场景描述 Cookie 在网页浏览过程中所起的作用。首先，用户发起登录请求，其中包含登录认证所需的用户名、口令等信息；第二，服务端收到请求包，验证用户发送的数据，若用户合法，则返回登录结果，返回数据包头部包含 Set-Cookie 字段，该字段的信息能够唯一确定该登录用户，完成之后会话中客户端的验证；第三，浏览器接收到返回结果后，根据 Set-Cookie 的值设置 Cookie，之后客户端再向服务端发起新的 HTTP 请求时，HTTP 请求头部自动添加 Cookie 字段，此时服务端接收 HTTP 请求后，验证 Cookie 字段包含的 Cookie 值，若该 Cookie 合法，则认定该用户为合法用户，从而作为已登录用户响应此次 HTTP 请求。以上过程可用图 1 描述。

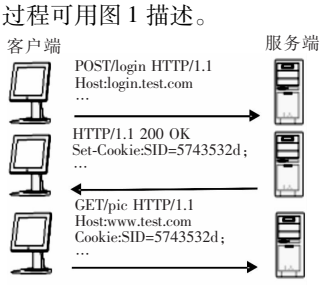

图 1 用户登录验证过程

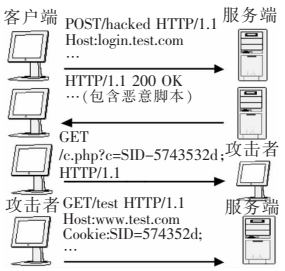

图 2 跨站脚本攻击

收稿日期：2011-12-09；**修回日期**：2012-01-15

作者简介：徐璐(1989-)，男，山东滕州人，硕士，主要研究方向为信息安全理论、网络攻防(scuxulu@qq.com)；周安民(1963-)，男，教授，主要研究方向为信息安全体系、关键技术和产品的研究、开发；刘亮(1982-)，男，讲师，主要研究方向为信息安全；周妍(1975-)，女，工程师，主要研究方向为信息安全.

攻击者利用跨站脚本攻击漏洞,向网页中嵌入恶意代码,用户浏览该网页时恶意代码一同被执行[4]。XSS 可被用于窃取用户 Cookie,如果不做任何防护措施,攻击者就可利用获取到的 Cookie 劫持用户与服务端的会话,或者进行其他网络攻击。通过 XSS 获取 Cookie 并劫持会话的过程可用图 2 描述。图 2 前三步为用户访问带有 XSS 攻击脚本的页面,第四步攻击者利用窃取到的 Cookie 伪装成合法用户访问服务器,此时服务器会验证 Cookie 为合法,进而认为攻击者为合法用户。

2　现有的 Cookie 保护机制

2.1　Cookie 与 IP 绑定

服务器将用户的 Cookie 与其登录 IP 绑定,当用户发起 HTTP 请求时服务端将同时对 Cookie 和 IP 进行验证[5],若不合法,则拒绝访问。这种措施可在一定程度上保护 Cookie,但当用户采用公共代理登录时就会失效,攻击者可采用同样的代理连接绕过这种防护。

2.2　Cookie 的 HttpOnly 属性

2002 年,微软的 Internet Explorer 研发工程师针对 Internet Explorer 6 SP1 版本增加了 Cookie 的 HttpOnly 属性[6]。Cookie 的 HttpOnly 属性若为 true,则表示该 Cookie 不能被客户端脚本获取,例如 JavaScript 中的 document. cookie 语句。

在其他方面,HttpOnly 属性值为 true 的 Cookie 与普通 Cookie 无异,用户端发起 HTTP 请求时同样会将 Cookie 附在 HTTP 头中。这种保护措施可以有效地保护 Cookie,但不适用于以下两种情况:第一,客户端使用的浏览器不支持 HttpOnly 属性;第二,Web 程序的开发者出于程序兼容性等原因没有采用 HttpOnly 属性。已有的统计数据表明,以上两种情况是非常常见的,HttpOnly 属性并没有发挥出想象中的功效[7]。

2.3　SSL 连接

当服务端采用加密连接时,服务端与客户端的所有通信会被加密,这种加密的连接可以有效地防止通信内容在通信线路上被窃取。但是这种保护对于 XSS 攻击则不起作用,攻击者同样可以获取到存在于浏览器中的明文 Cookie,从而绕过加密连接在通信线路上的保护[8]。

以上所提到的 Cookie 保护措施均存在不同程度的不足,都不能保证 Cookie 在 XSS 攻击中不被泄露。

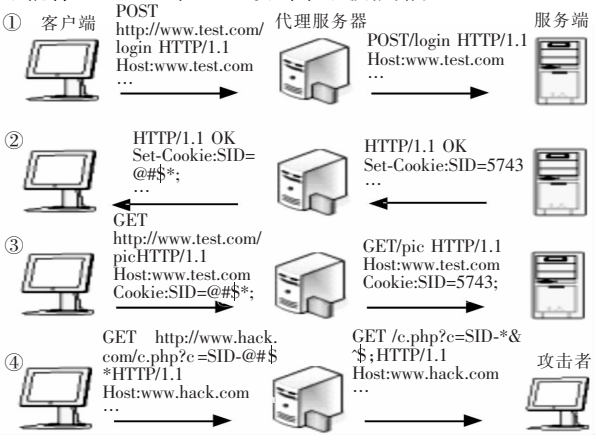

图 3　技术框架

3　基于代理的 Cookie 保护技术框架

本文提出了一种基于代理的 Cookie 保护措施,采用基于 DES 算法的 Cookie 加密(解密)方案,可以不用对服务端和客户端做任何改变,便于部署和维护,并提供可靠的保护。

以图 1 中的用户登录过程为例,采用基于代理的 Cookie 保护措施,用户经代理服务器向服务端发出登录请求,Web 服务器将处理结果返回代理服务器,代理服务器解析出返回的 Set-Cookie 值后,对其进行加密,将处理后的结果转发给客户端;同理,客户端发起 HTTP 请求时,代理服务器接收请求并解析数据包,将解析出的密文 Cookie 进行解密,并将处理结果转发给 Web 服务器。其中,加密(解密)密钥跟域名唯一对应,即针对不同的服务器采用不同的加密(解密)密钥。

以上整个过程,Cookie 由于经代理服务器加密,此时浏览器中储存的是密文 Cookie。假设用户被 XSS 攻击,由于攻击者的服务器不同于用户访问的服务器,其密钥不同,因此,攻击者窃取到的 Cookie 一定不会经过正确的密钥解密,从而有效保护了用户 Cookie。

基于代理的 Cookie 保护的工作过程可用图 3 描述:①用户发出登录请求;②服务器返回结果,其中 Set-Cookie 字段经代理服务器加密,图中 SID = 5743,经加密后为 SID = @#$*;③用户发出普通的 HTTP 请求,附带有加密过的 Cookie 值 SID = @#$*,经代理服务器解密后为 SID = 5743;④用户被 XSS 攻击,攻击者窃取到的 Cookie 由于没有被正确解密,发送给攻击者的 Cookie 值为 *&^$。

4　关键技术实现

4.1　代理功能的实现

本文所述的 Cookie 保护技术基于代理实现,因此首先利用编程语言实现代理功能。基本过程是,首先建立套接字,并根据输入参数监听指定端口,当监听到连接请求时,接受连接,并读取发送的数据。通过对数据包的分析,分离出数据包的各个字段,包括主机、端口、数据等信息,并将分离出的 Set-Cookie 字段或者 Cookie 字段发送给加密(解密)模块,进行加密或者解密。之后将处理完成的数据重新组包,并根据主机、端口等信息转发。

代理功能的程序流程如图 4 所示。

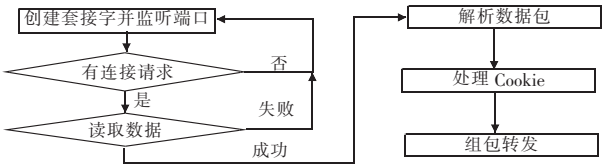

图 4　系统流程

4.2　加密解密的实现

Cookie 的加密和解密可采用任意的加密算法,在实际的工程环境中可以根据需要进行选择。本文以对称加密算法 DES 为例,阐述加密解密的实现。密钥存储在代理服务器中,当系统需要对 Cookie 进行加密或者解密时,通过查找系统数据库记录获取密钥值。密钥的产生通过预置的算法完成,可表述为取目标服务器域名加随机数的 MD5 散列值为密钥,其产生可

用如下公式表述：

$$K = \mathrm{MD5}(\mathrm{domain} + \mathrm{randomNum})$$

其中：K 表示密钥；MD5 表示散列函数；domain 表示域名；randomNum 表示随机数。

代理服务器解析数据源的 Set-Cookie 或者 Cookie 信息，并将此数据作为需要加密（解密）的数据。DES 加密（解密）程序获取要处理的字符串，将字符串分割成每 8 Byte 一组；然后分别对其进行一次标准 DES 加密（解密）；最后，将处理结果返回给代理模块。

5 测试

为了检验 Cookie 保护的有效性，本文使用 VS2010 的编译环境，对代码进行了编译，实现了可执行程序，并运行于实际网络环境中搭建的服务器上。测试机浏览器设置代理为代理服务器 IP，测试端口为 8086。测试方法：选取测试网站若干，在使用 Cookie 保护代理服务器和不使用 Cookie 保护代理服务器两种情况下分别访问并登录测试网站，通过比较检测 Cookie 保护系统是否将 Cookie 加密，若已加密，表示保护成功，反之保护失败；之后模拟一次获取 Cookie 的跨站脚本攻击，检测合法 Cookie 是否被攻击者获取，如果攻击者获取到的是非法 Cookie，则保护成功，反之保护失败。选取的测试网站为人人网、新浪网、天涯社区。测试结果如表 1 ~ 3 所示。

表 1 人人网测试结果

类别	Cookie 值
未保护	t = 197cc7995628221fb5058abf038440e5
保护	t = 2K 饢 S&8?? 鯈 N 卤 > D? 泙? 3 姐 u 赋
模拟 XSS	t = 垭；? b 饡鼹紳 & 粵訴蹌赚迭? 閻 5 闶? J?

表 2 新浪网测试结果

类别	Cookie 值
未保护	vjuids = -4293816cd. 133cf82c8c6. 0. d023
保护	vjuids = ? H 卑俚癒 g 騕腴 Y\S 跡? 8?% 繒?
模拟 XSS	vjuids = 夢騷 < 爆`| z 釩 2oE 紬勸椵? Z 啜

表 3 天涯网测试结果

类别	Cookie 值
未保护	sso = r = 861119083&sid = &wsid = D
保护	sso = zk] Gb 徹 4 淮琉閏? 7p 硕 O | @ . 黏?
模拟 XSS	sso = h? 漣嚪 iykW

测试结果表明，使用了本系统的网页 Cookie 均被加密后存储在浏览器中，且 XSS 攻击者获取到的 Cookie 没有被正确解密，Cookie 可被有效保护。

6 结束语

本文介绍了 Cookie 与 XSS 攻击，针对窃取 Cookie 类型的 XSS 攻击，提出了一种新的 Cookie 保护方法，并对其进行了实现。基于代理的 Cookie 保护，并利用 DES 算法或者其他加密算法对 Cookie 加密和解密，达到保护 Cookie 的目的。通过实际环境的测试，结果表明，可有效地防止 Cookie 被 XSS 攻击者窃取。这种 Cookie 保护框架可被有效地利用于 XSS 防御当中，以缓解目前日益泛滥的 XSS 攻击。

参考文献：

[1] JOHNS M, ENGELMANN B, POSEGGA J. XSSDS: server-side detection of cross-site scripting attacks[C]//Proc of Computer Security Applications Conference. Anaheim: IEEE, 2008: 335-344.

[2] GEBRE M T, LHEE K S, HONG M P. A robust defense against content-sniffing XSS attacks[J]. Digital Content, Multimedia Technology and its Applications, 2010, 9(1): 315-320.

[3] Network Working Group. RFC 2109, HTTP state management mechanism[S]. Washington DC: Internet Society, 1997.

[4] STEPHEN J, REDDY P, NAIDU D. Prevention of cross site scripting with E-Guard algorithm[J]. International Journal of Computer Applications, 2011, 22(5): 30-34.

[5] VARJABEDIAN R. Bullet proof cookies[EB/OL]. (2010-05)[2011-06]. http://www.codeproject.com/KB/aspnet/BulletProofCookies.aspx.

[6] JASON B, ELIE B, DIVIJ G. State of the art: automated black-box Web application vulnerability testing[J]. Security and Privacy, 2010, 5(1): 332-345.

[7] ZHOU Y, EVANS D. Why aren't HTTP-only cookies more widely deployed[C]//Proc of the 4th Web 2.0 Security and Privacy Workshop. 2010: 1-5.

[8] DIERKS T, RESCORLA E. RFC 4346, The transport layer security (TLS) protocol version 1.1[S]. US: IETF, 2006.