基于 HTTP 协议 Host 头二义性问题带来的一种漏洞挖掘新思路

李文皓　李斌勇
（成都信息工程大学信息安全工程学院　成都　610225）

【摘　要】针对 HTTP 协议 Host 头二义性的相关安全问题，深入分析了近期互联网络上与 HTTP 协议有关的已知与潜在漏洞。围绕这些漏洞，分析并发现了其本质是因 RFC 的制定与具体实施之间的差异所致。利用所造成上述差异的问题本质，在此基础上创新性地提出了一种重新组合与利用的漏洞挖掘新思路。论文所提出的这种漏洞挖掘新思路，对网络安全研究人员发现与挖掘新的安全漏洞，具有可行的参考意义。

【关键词】通信协议；二义性；网络安全；漏洞挖掘
【中图分类号】TP393
【文献标识码】A

A Novel Method of Vulnerability Detection Based on Multiple Host Ambiguities in HTTP

Li Wen-hao　Li Bin-yong
(Chengdu University of Information Technology,College of Information Security Engineering SichuanChengdu 610225)

【Abstract】Aiming at understanding the security issues related to multiple host ambiguities in HTTP, an in-depth analysis is carried out on the recent vulnerabilities caused by the flaws of HTTP protocol which are revealed and potentially exist on the Internet. According to the research, the differences between specifications and implementations of RFC essentially contribute to the problem. A novel method of vulnerability detection by reforming and exploiting the flaws is creatively put forward. The method mentioned in this paper carries some practical significance for security researchers to find the new potential vulnerabilities in cyberspace.

【Keywords】communication protocol; ambiguity; network security;vulnerability detection

1　引言

HTTP（Hyper Text Transfer Protocol，超文本传输协议）位于应用层，是用于 WWW 服务器传输到用户本地浏览器的无状态传送协议。服务器、浏览器以及相关 Web 应用的通信无不依赖于 HTTP 协议[1]。RFC（Request For Comments）是一系列以编号排定的文件，基本的互联网通信协议在 RFC 中均有详细的说明描述。而在 RFC2616 中定义了目前 Web 世界中最广泛使用的 HTTP/1.1。

最近，清华大学、Berkeley 大学以及华为安全人员研究发现[2]：在 HTTP 协议具体实施中，因没有严格去遵循 RFC 中相关定义，HTTP 协议中的 Host 头具有二义性，从而造成了一系列安全问题。针对该问题，研究人员向相关网络安全机构报告了该问题。

本文旨在从 Web 安全人员的角度来分析上述文献中所提到的问题以及其所带来的一种新的漏洞挖掘思路。尤其是从网络安全渗透测试的角度来分析该安全问题的具体威胁与利用场景,以便引起业界对该安全问题的关注。

2 Host 头的二义性

为了深入剖析 Host 头,首先需要明确 HTTP 的报文结构及其各结构要素的意义。通常 HTTP 的报文都有一条来自客户端的请求或者一条来自服务端的响应。而这两者分别由起始行、首部、主体三部分构成。起始行的作用是对报文的一个描述,而首部则包含了一些属性,主体则是包含着所要传递的数据信息。这里需要着重关注请求报文的首部,因 Host 头就在请求首部中。就请求首部中的 Host 头而言,其主要用于描述接收请求的服务器及其相应端口(若无端口信息,对于 HTTP URL 来说便是 80 端口)。其实,根据 RFC2616 协议的"5.2"请求资源的识别[3],更准确的来说确定所请求的服务器以及相应的端口不是由 Host 头唯一确定,还需要决定于 Request-URL。若 Request-URL 是绝对地址,那么即使 Host 存在值,也依然会被忽略掉。反之,若 Request-URL 是一个相对地址,且请求头中包含有 Host,那么请求的资源位置就会由 Host 来确定。

由此分析可见 Host 头在请求中扮演着相当重要的角色。除此而外,Host 头也是很多安全检测的关键切入点,接下来会详细讨论该问题。

在分析 Host 基础上,还需了解 Host 的二义性。由语言学对于二义性的定义,可较为容易地理解二义性的概念。例如,一句话在同一情景下可能存在两种(或以上)的含义,就称这句话是具有二义性的,或者说这句话是存在歧义的。

本质上而言,Host 的二义性是因为 HTTP 协议在具体媒介中实施的时候并未完全遵循 RFC 标准,在两个媒介处理同一个 HTTP 的请求时,对同一个 Host 头存在不同的两种分析处理或解释行为,从而导致了 Host 头可能存在二义性,利用这种差异的不同组合便为一些潜在安全漏洞利用提供了可乘之机。

如图 1 所示为攻击者利用 Host 头二义性绕过 CDN 防护。攻击者构造了一个含有两个 Host 头,且第二个 Host 头前存在一个空格的请求数据包。对于 CDN 来说接受了第一个 Host,而服务器接收到了前面有空格的第二个 Host 头,从而绕过了 CDN。同样的思路,CDN 这里还可以是防火墙等。

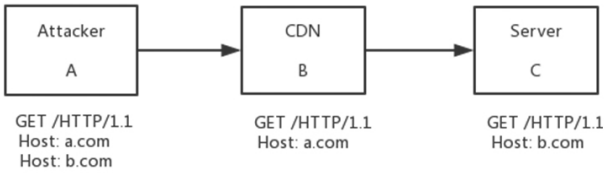

图 1 利用 Host 头二义性绕过 CDN 防护示意图

来自清华大学、Berkeley 大学以及华为安全研究人员在论文中给出了阐述:其通过实验发现有 33 个流行应用的 HTTP 实施对于 Host 头的解释有显著不同的行为,从而导致了三种利用的方法:(1)多个 Host 头的利用方法;(2)包含空格的 Host 头利用方法;(3)含有绝对路径请求的利用方法。

针对上述三种利用方法,给出具体阐述。

多个 Host 头的利用方法:事实上在 RFC2616 和 RFC7230[4]中都或间接或直接的说明了不允许有多个 Host 头,而安全人员实验发现:测试的 33 个样本中其中有 25 个都不遵循 RFC 的规定,没有拒绝这种具有多个 Host 的请求。那么就可能存在这种情况:对于 A,会接受第一个 Host,对于 B,会去接受第二个 Host。因此,就可以利用这样一些问题去绕过一些安全限制,就如在上文中所提到的一样。

包含空格的 Host 头利用方法:这种类型具体而言会有三种构造的方式。尽管在 RFC2616 和 RFC7230 中也几乎有相应对策,但是测试人员发现具体的实施之中,33 个样本中只有 5 个遵循 RFC2616,2 个遵循 RFC7230。

含有绝对路径请求的利用方法:在前文介绍 Host 之时,可知根据 RFC2616 请求的资源位置,在 Request-URL 含有绝对路径之时,Host 头将会被忽视。而在 RFC7230 中则更进一步说明了,即使 Request-

URL 中存在绝对路径，请求中仍然应该具有明确的 Host。这里研究者们又发现一些非常有趣的事情了：就是不论在 RFC2616 或是 RFC7230 中，都没有明确的指出 Request-URL 中绝对路径中允许存在的协议类型。发现在具体的实施中：有些是支持多种类型的协议，有些是仅仅支持 HTTP 或者 HTTPS 其中之一的。在所做的测试之中，同样发现遵循 RFCs 协议的非常少。

3 基于 HTTP 协议 Host 头二义性的漏洞挖掘新思路

在详细阐述 HTTP 协议 Host 头的二义性问题，以及举例说明利用这种二义性可在某些特定环境下进行绕过 CDN 和防火墙等进行攻击的基础上，本文将论述该问题对于漏洞挖掘所提供的一种新的研究思路。

除了已经提到的 Host 头，事实上 Cookie 头也会存在这样的问题。若出现两个同名的 Cookie 头，那么不同 Server 语言在处理过程中将会出现差异。如图 2、图 3 所示分别给出了 PHP 和 Python 对于两个同名 Cookie 头的渗透处理过程及结果。

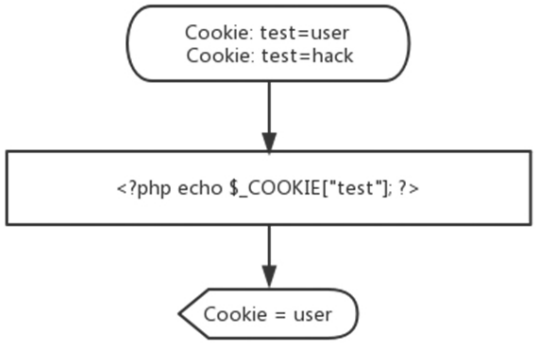

图 2 PHP 渗透测试过程及结果

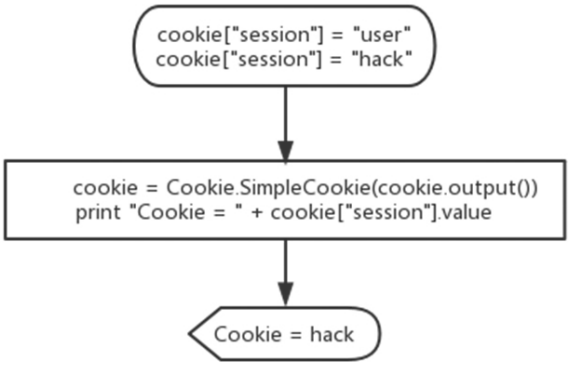

图 3 Python 渗透测试过程及结果

显然，可以注意到 PHP 解析后是接收第一个 Cookie 而 Python 接收了后者。其实，这与 HTTP 协议 Host 头二义性的问题在本质上是一致的。协议的说明指定与实际实施之间存在的差异，导致了上述测试差异的出现。

对于使用了 Google Analytics 的 Python Django 也存在 CSRF 绕过问题。首先，通过 Google Analytics 进行 Cookie 注入（由于 Cookie 中存在一个 Path 参数，而该参数又直接来源于 Referer 中，未过滤）。由于 Web Server 以及浏览器对于 Cookie 的解析过程存在不符合 RFC 标准的问题，因而最终导致成功的注入。从而可以实现删除已有 Cookie 或者覆盖已存在的 Cookie。由此不难发现，很大程度上导致该问题也是因协议在具体指定和实施之间所存在的差异所致。

Internet Explorer/Edge 存在的一个“结合 host 头注入和 host 的不严谨解析，提供恶意数据服务”[5]问题，并且利用这个问题可以成功在 Github 上面实施了 OAuth token 的窃取以及对 Heroku 和 Fastly 路由混淆[5][6]。

利用这种解析差异，我们甚至还能构造出 XSS 漏洞等。由此可见，所探索上述问题无疑为漏洞研究提供了一条新的思路，并且说明了对于这类漏洞的研究在工业领域也具有实际运用意义。Browser 在 Web 生态中扮演着重要的角色，且其与 Web Server 休戚相关。而有意思的是，在实际的实施、解析之中往往都没有去遵循 RFC 协议规定，往往这之间所造成的差异也各具不同。运用这样的思路，就可以利用实现的不同，从而造成具有现实意义的攻击。

互联网世界里还存在更多的协定，但是实际应用中很可能大量都并没有去遵循这样的协定。对于整个 Web 互联网世界而言，单独个体对于协议的不遵循（如浏览器和 Web Server，Web Server 与 Web Server 等）于自身而言就是漏洞，对这些漏洞点加以组合利用，便最终构成了多种潜在的网络安全攻击。为此，寻找协议的制定与实施之间的差异，并在此基础上加以组合利用，便成为了一种新的漏洞挖掘思路。

【下转第 56 页】

模糊测试。模糊测试是基于缺陷注入的一种自动软件测试技术,通过将大量精心构造的数据作为应用程序的输入,通过观察程序是否出现异常来判断应用程序中可能存在的漏洞。

动态污染传播。这一技术也被称为动态信息流分析方法,是通过在程序运行时标记变量、存储单位等信息,对攻击路径进行跟踪,从而获得存在漏洞信息。[5]

4 结束语

源代码漏洞无处不在,在开发过程中,软件开发人员应高度重视漏洞这一问题,根据软件开发源代码安全指南等规范标准进行代码编写。我们只是对源代码中具有代表性的漏洞进行了简单分析,了解其侵入原理及防范。源代码安全漏洞的防治修复是一项持续不断的工作,需要开发人员不断对其完善修复,同时用户在程序应用过程中也应加强漏洞防范措施,避免应漏洞的存在造成的损失。

参考文献

[1] 朱圣才,徐御,王火剑.常见源代码安全漏洞分析与研究[J].信息网络安全,2014,(02):49-52

[2] 赵晖.面向军工应用软件的源代码漏洞分析系统的研究与实现[C].北京交通大学,2015-7-14

[3] 叶亮.基于安全规则的源代码分析方法研究[C].华中科技大学,2013-10-13.

[4] 王跃.源代码安全漏洞检测方法研究[C].华中科技大学,2013-27-41.

[5] 周诚,张涛,马媛媛,李伟伟.一种高效检测源代码安全漏洞的代码审查方法[J].现代电子技术,2015,(05):83-86.

作者简介:

黎新(1987-),男,汉族,工程师,研究生,硕士学位;主要研究方向和关注领域:信息安全。

【上接第 52 页】

4 结束语

本文主要分析了一种由于 HTTP 协议 Host 头具有二义性所造成的安全问题,阐述了具体攻击场景和利用方式,并且进一步分析了由 HTTP 协议 Host 头二义性所带来的一种漏洞挖掘研究的新思路,并辅以网络上具体存在的漏洞案例进行论述。事实上,很可能找到诸多不遵循 RFC 或其他标准的实施,而研究的难点和关键在于:如何将这一系列的网络安全漏洞与缺陷进行有效地联系或组合起来,从而构造出合适的网络攻击场景并发掘更为深层次的潜在安全危害。这是需要进一步思考与研究的问题;同时通过本文,希望能有更多的安全研究人员能够关注协议与实现之间存在的差异等,以及与之相关的安全漏洞利用问题。

参考文献

[1] Gourley D, Totty B. HTTP: the definitive guide [M]. "O'Reilly Media, Inc.",2002.

[2] Chen J, Jiang J, Duan H, et al. Host of Troubles: Multiple Host Ambiguities in HTTP Implementations [C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM,2016: 1516-1527.

[3] Fielding R, Gettys J, Mogul J, et al. Hypertext transfer protocol-HTTP/1.1, 1999[J]. RFC2616, 2006.

[4] Fielding R, Reschke J. RFC 7230: Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing[J]. 2014.

[5] Labsdetectify:Combining host header injection and lax host parsing serving maliciousData.https://labs.detectify.com/2016/10/24/combining-host-header-injection-and-lax-host-parsing-serving-malicious-data/.

[6] XSS Jigsaw:Internet Explorer has a URL problem.http://blog.innerht.ml/internet-expl-orer-has-a-url-problem/.

基金项目:

成都信息工程大学科研基金资助项目(KYTZ201618)。

作者简介:

李文皓(1996-),男,四川成都人,大学本科;主要研究方向和关注领域:信息安全。

李斌勇(1983-),男,四川江油人,博士,讲师;主要研究方向和关注领域:信息安全、云服务技术。