文章编号:1001 - 9081(2007)03 - 0616 - 03

一种基于 SIP 安全认证机制的研究

姬 宁,林 晓,普杰信
(河南科技大学 电子信息工程学院,河南 洛阳 471003)
(hkdjining@ yahoo. com. cn)

摘 要:目前,会话初始协议(SIP)大部分认证机制只提供了服务器到客户端的认证,HTTP 摘要认证便是其中的一种。分析了这种机制容易遭受服务器伪装攻击和密码窃取攻击的缺陷,提出了一种弥补这些缺陷的安全认证机制。试验表明该算法具备较高的效率。

关键词:会话初始协议;认证;HTTP 摘要;安全
中图分类号: TP393.08 **文献标识码**:A

Research on secure authentication based on session initial protocol

JI Ning, LIN Xiao, PU Jie-xin
(*Electronic Information Engineering College, Henan University of Science & Technology, Luoyang Henan* 471003, *China*)

Abstract: At present, most of the session initial protocol (SIP) authentication schemes only provide client-to-server authentication. And HTTP digest authentication is one of them. The weakness of this scheme in server spoofing and password guessing attacks was analyzed, and a more secure strategy was proposed which avoided these shortcomings. The experimental result illustrates that this proposed algorithm is effective.

Key words: Session Initial Protocol (SIP); authentication; HTTP digest; security

0 引言

会话初始协议(Session Initial Protocol,SIP)是由 IETF(The Internet Engineering Task Force)提出并主持研究的一种基于应用层的多媒体会话控制协议。SIP 是实现新一代多媒体通信和软交换的关键技术,用来创建、修改和终止一个或多个多媒体会话[1,2]。目前,SIP 不仅被用于网络电话和软交换中,3GPP(3rd Generation Partnership Project)及 3GPP2(3rd Generation Partnership Project 2)等组织也已经确定 SIP 为未来多媒体子系统的核心呼叫控制协议。

SIP 采用文本形式表示消息的词法和语法,因此容易被攻击者模仿、篡改,从而被非法利用。IETF 在最初设计 VoIP 体系及其信令协议 SIP 时,把重点放在了提供信令的动态的、强大的业务的可能性和简单性方面,很少注意安全特征,没有为 SIP 指定专门的安全协议[3]。不少文献针对 SIP 的这一安全性问题提出了自己的解决方案。文献[3]分析了有状态无状态服务器在承载层分别为 TCP、UDP、TLS 及有无 HTTP 摘要认证情况下的各种处理性能。文献[4]定义了在 SIP 用户代理和下一跳实体之间协商安全策略的一种机制。文献[5]提出了一种基于 SIP 的媒体通信安全框架结构,它能保证跳到跳间的信令安全及端到端的媒体通信安全,但结构特别复杂。文献[6]提出在注册服务器和代理服务器实现为一体时,如何减少认证流程以提高性能的策略。文献[7]提出了双向 HTTP 摘要认证和密钥协商机制,虽有效避免了服务器伪装攻击,但认证流程过于繁杂。这些文献或者没有涉及用户服务器双向认证的问题,或者提出的机制过于复杂。

本文分析了 HTTP 摘要认证方式的流程及存在的缺陷,并参考文献[8]中描述的 3G/UMTS 认证和密钥同意机制的原理,结合 HTTP 摘要认证,提出了一种既能实现服务器和客户端之间的双向认证,又能比较有效地避免密码猜测攻击的安全高效的认证策略。

1 SIP 协议

SIP 运用一种分布式的控制模式,采用 Client/Server 结构的消息机制,将对语音通信的控制信息封装到消息的头域中,通过消息的传递来实现[4]。SIP 的体系结构由五种网络实体构成:用户代理(User Agent, UA)、代理服务器(Proxy Server)、注册服务器(Register Server)、重定向服务器(Redirect Server)和位置服务器(Location Server)。

用户代理是客户端终端系统的应用程序,它代表要加入呼叫的用户。用户代理包含两个部分:用户代理客户端(UAC)用来初始化一个呼叫,发起 SIP 请求,并作为该用户的呼叫代理;用户代理服务器(UAS)用来接收请求,代表用户发出响应,并作为被叫用户代理。

代理服务器在主叫 UA 和被叫 UA 之间转发请求和回复。代理服务器分为有状态代理和无状态代理两种。其区别在于有状态代理能记住它接收到的入请求,以及回送的响应和它转送的出请求;无状态代理一旦转送请求或回复后就忘记所有的信息。无状态代理服务器效率高于有状态代理服务器,成为 SIP 结构的骨干。

注册服务器接受客户机的注册请求,并将注册地址写入位置服务器。当用户代理客户端地址改变了,它用来发送一个注册请求到注册服务器来更新在位置服务器中的记录的位置信息。

收稿日期:2006 - 09 - 20;**修订日期**:2006 - 12 - 12 **基金项目**:河南省科技攻关项目(0524220019);洛阳市科技攻关项目(50225)
作者简介:姬宁(1982 -),男,河南柘城人,硕士研究生,主要研究方向:计算机网络与通信; 林晓(1978 -),女,河南南阳人,讲师,主要研究方向:人工智能与模式识别、计算机网络、信息安全; 普杰信(1959 -),男,河南鹿邑人,教授,博士,主要研究方向:人工智能与模式识别、计算机网络、信息安全.

重定向服务器接受 SIP 请求,将请求的地址映射为一个或多个联系地址,然后通知呼叫方,呼叫方直接与这些地址进行联系。

位置服务器用来维护和保存用户代理的当前位置信息,同时为代理服务器、重定向服务器、注册服务器查询或注册用户代理当前地址提供服务。

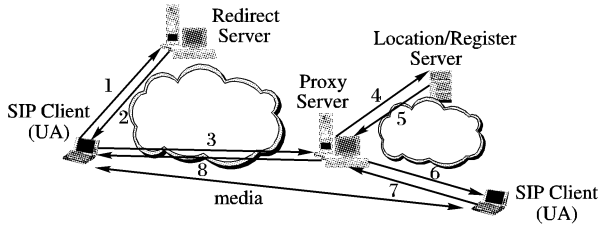

图 1　SIP 协议基本网络模型

当前,SIP 安全变得越来越重要。SIP 和 IP 电话标准化的过程中的一个热点问题就是如何加强安全支持以满足业务需要[4]。当客户机请求得到 SIP 服务时,它需要首先得到服务器的认证。基于以上的原因,RFC2543 提出了 HTTP Digest 和 HTTP Basic 两种认证方式。由于 Basic 这种认证方式在消息明文中直接传送密码,因此密码很容易在传送过程中被窃取,存在很大的安全隐患,在 RFC3261 中这种方式被废除。

在 HTTP Digest 认证方式中,服务器采用一种基于挑战响应的方式来验证用户的身份。这种方式虽然不在消息中明文传送密码,但也存在很大的安全隐患。例如,客户端不能验证服务器的身份,如果攻击者伪装成服务器就能获得用户的一些敏感信息。除此之外,这种方式还容易遭受离线密码猜测的攻击。

2　HTTP Digest 认证[5]

2.1　认证流程

认证开始前,客户端和服务器预先共享一个密码。服务器用这个密码用来验证客户端的身份。认证流程如图 2。

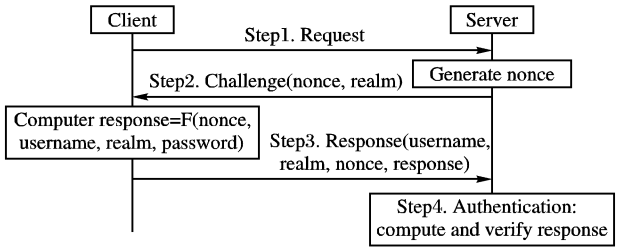

图 2　HTTP Digest 认证流程

Step1　客户端→服务器:

客户端发送一个请求到服务器。

Step2　服务器→客户端:

服务器要验证客户端身份的合法性,向客户端回送一个包含 nonce 和 realm 的摘要盘问消息,即 challenge. nonce 是服务器产生的一个包含时间戳信息的随机数串,realm 通常指服务器所负责管辖的域的域名。

Step3　客户端→服务器:

客户端根据收到的 nonce、realm 及自己的用户名和共享密码通过一个单向哈希函数 F 计算 response 值。response = F(nonce,username,realm, password)。通常 F 被用来产生一个摘要认证信息,多数情况下用的是 MD5 算法。然后客户端将 response 值发送给服务器。

Step4　通过用户名,服务器在数据库中提取出密码,然后检验 nonce 是否正确。如果正确,服务器计算 F(nonce, username,realm, password),然后将结果跟收到的 response 比较,如果匹配,服务器就认为客户端是一个合法的用户。

2.2　认证存在的缺陷

1)离线密码猜测攻击

在步骤 2 和步骤 3 中,攻击者通过截获消息,很容易获得 nonce,username,realm 及 response 的值,接着攻击者猜测密码 password′,并计算 F(nonce,username,realm, password′),将结果值与 response 值进行比较,如果匹配,则 password′便是正确的密码。

2)服务器伪装攻击

如果用户不验证服务器的身份是否合法,当它收到服务器发来的 challenge 消息时,就会回复一个 response。这样一个攻击者可以伪装成服务器的身份向用户发送 challenge 来获得 response,收到 response 值后,它就可以离线猜测用户的正确密码了。具体步骤如下:

Step1　客户端→攻击者:

客户端发送一个请求到服务器,中途被攻击者截获。

Step2　攻击者→客户端:

攻击者产生一个包含 nonce′和客户端所属域 realm 的挑战信息,并发送给客户端。

Step3　客户端→攻击者:

客户端用 F(nonce′, username, password, realm)计算出 response 值,并将其发送给攻击者。

Step4　攻击者收到客户端的回复后,它就获得了 nonce′, username,realm 和 response 的值,然后就可以离线猜测密码,直到猜出用户真正的密码。

3　一个安全的认证策略

3.1　认证流程

由于以上认证方式存在着容易遭受离线密码猜测攻击和服务器伪装攻击的缺陷,我们提出了一种安全的认证机制,这种方式可以实现客户端和服务器的相互认证并有效弥补了离线密码猜测攻击的缺陷。认证流程如图 3(图中⊕表示异或,·表示数串连接符号)。

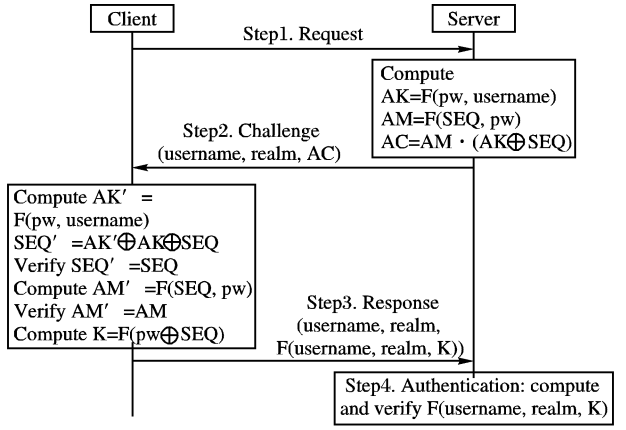

图 3　安全认证流程

认证开始前,服务器端和客户端预先共享一个密码 pw 和一个比较大的认证序号 SEQ(SEQ 以递增的顺序进行),认证流程如下:

Step1　客户端→服务器:

客户端发送一个请求到服务器。

Step2　服务器→客户端:

服务器计算两个参数 AK 和 AM 的值：AK = F(username, pw)，AM = F(SEQ, pw)，AC 为 AK ⊕ SEQ 与 AM 连接后的结果，这种连接方式为服务器和客户端预先设定好的一种方式，比如说 AC 中从密码长度位开始的一部分为 AM，其余字段为 AK ⊕ SEQ。F 仍然是前面所提到的单向哈希函数，然后服务器向客户端发送挑战信息，其中包括 username，realm，AC 三个参数的值。

Step3 客户端→服务器：

客户端计算 AK′ = F(username, pw)，并与收到的 AK ⊕ SEQ 进行异或计算，得出 SEQ′ = AK′⊕ AK ⊕ SEQ，验证 SEQ′ 的正确性。如果收到的 SEQ′≠SEQ，说明服务器方产生的挑战信息不是最新的，停止以下步骤的进行。如果 SEQ′ = SEQ，则继续验证服务器的身份，计算 AM′ = F(SEQ, pw)，比较 AM′与收到的 AM 是否相等，如果不相等说明服务器身份不合法，拒绝向服务器发送 response 信息；否则认为服务器是可信任的，计算 K = F(pw ⊕ SEQ)，response = F(username, realm, K)，接着向服务器发送 Response 消息，包含 username，realm 和 response 的值。

Step4 服务器计算 F(username, realm, K) 的值，并与收到的 response 进行比较，如果匹配，则服务器器验证了客户端的身份。

3.2 安全性分析讨论

1）重放攻击

由于 SEQ 是不断更新的，当服务器进行重放攻击时，客户端能够根据 SEQ 的值判断出服务器方发来的挑战信息是旧值，并予以拒绝。用 AK ⊕ SEQ 传送有效避免了 SEQ 的值在网络中被窃取的可能。因此重放攻击在本算法中行不通。

2）服务器伪装攻击

在步骤 3 中，通过验证 SEQ 的正确性以及 AM 的正确性，两次验证了服务器的身份。显然，攻击者服务器无法伪装成服务器来欺骗用户。

3）离线猜测密码攻击

如果 response 值被截获，攻击者猜测密码 pw′和 SEQ′，并计算 K′ = F(pw′⊕ SEQ′)，response′ = F(username, realm, K′) 以匹配 response 的值。这就要求 K′ = K，如果两个未知参数 pw 和 SEQ 的数值均比较复杂的话，所得解的个数将非常巨大。因此这种方式非常有效地避免了密码猜测的攻击。

表 1 安全性比较

攻击类型	HTTP 摘要	加密密钥交换	我们的方案
重放攻击	无	无	无
离线密码猜测攻击	存在	无	无
服务器伪装攻击	存在	无	无

表 2 代价比较

攻击类型	HTTP 摘要	加密密钥交换	我们的方案
单向哈希函数个数	2	无	8
幂运算次数	无	4	无
对称加密次数	无	9	无
或运算次数	无	无	4
消息流数	3	4	3

4 试验结果分析

为验证本算法的效率，我们在 Linux 下编写了一个试验模拟系统。本系统由两部分构成：注册客户端和注册服务器。函数 F 使用 MD5。注册服务器集成了认证服务和位置服务的功能。注册流程仍然与标准的注册流程一样（注册与认证一块进行）：

Step1 客户端向注册服务器发出一个注册请求。

Step2 注册服务器检验请求信息中无认证信息，向客户端发出挑战信息。

Step3 客户端验证注册服务器身份合法后，向注册服务器返回 Response 信息。

Step4 注册服务器验证了用户身份合法后，将注册地址写入位置服务器，并向客户端返回 200 OK 信息。

注册客户端为多线程程序，运行在硬件配置为奔腾 Ⅳ 3GHz，392M 内存，操作系统为 Linux RedHat 9 的 PC 机上。服务器端运行在硬件配置相同，操作系统为 Linux Fedora 2 的 PC 机上。客户端和服务器之间通过 100M 快速以太网相连。SIP 承载层使用 UDP 协议，服务器为有状态服务器。

在本试验中，客户端的每个线程串行产生一系列注册过程（比如 80 个）。当前一个注册流程完成时，新的注册立即开始，而各个线程之间是并行的。我们通过测量客户端的一个线程在 1 s 内完成的注册流程数（即单线程吞吐量）计算服务器的总吞吐量。若总线程数为 N，则每个线程的吞吐量为服务器总吞吐量的 $1/N$，这样服务器总吞吐量便可计算出来。表 3 显示了注册服务器在不同线程下的处理性能。图中的吞吐量表示服务器在 1 s 内所能处理的认证流程的个数。用同样的方法，在同样的配置环境下，我们测量了在 HTTP 摘要认证策略下的服务器处理性能。

表 3 两种策略下服务器处理性能

线程个数	本策略		HTTP 摘要策略	
	单个线程吞吐量/s^{-1}	总吞吐量/s^{-1}	单个线程吞吐量/s^{-1}	总吞吐量/s^{-1}
1	15.1	15.1	18.9	18.9
2	7.9	15.9	10.0	20.0
3	5.3	16.0	6.7	20.2
4	3.9	15.8	5.1	20.3
5	3.2	15.9	4.1	20.3
6	2.7	16.0	3.4	20.4

我们将这两种策略下服务器总吞吐量对比转化为百分比，并规定 HTTP 摘要认证策略下为 100%。对比关系如表 4。

表 4 对比关系

线程数	性能比	线程数	性能比	线程数	性能比
1	0.799	3	0.792	5	0.783
2	0.795	4	0.778	6	0.784

表 4 中的数据表明，我们提出的这种安全策略在效率上基本上可以达到 HTTP 摘要认证策略的 80%。如果将两者中的最大吞吐量进行对比，对比关系为 0.784，也非常接近 80%。因此本算法不失为一个高效的算法。

参考文献：

[1] SPARKS R, HANDLEY M, SCHOOLER E. SIP: Session Initiation Protocol, RFC 3261[S], 2002-06.

[2] HANDLEY M, SCHULZRINNE H, SCHOOLER E, *et al*. SIP: Session Initiation Protocol, RFC 2543[S], 1999-03.

（下转第 623 页）

信息的大小。然后点击“加密”按钮,弹出要求输入密钥 Key1 的对话框,输入密钥后,程序对隐藏信息进行加密预处理。点击“嵌入”按钮,程序弹出要求输入密钥 Key2 的对话框,根据 Key2 确定在 25 个资源中的哪几个资源间扩展冗余空间来嵌入水印,最后把处理过的信息分块嵌入到 PE 文件中,把隐藏了信息的载体保存为 SingleDoc1. exe。

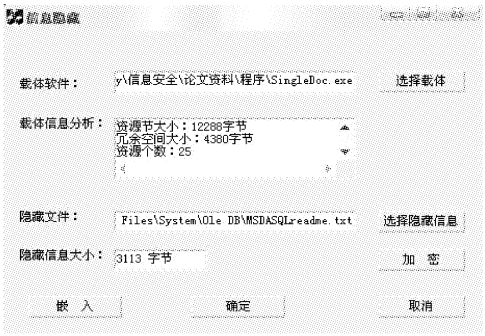

图 4　PE 文件资源节信息隐藏实现界面

执行隐藏了信息的 SingleDoc1. exe,载体软件仍能正常执行且保持了原始功能,表明载体软件并没有受影响,方案具有较好的隐蔽性。提取隐藏信息是嵌入的逆过程。图 5 是“Comparel”软件对提取出来的隐藏信息与原始隐藏信息的比较,分析结果表明两文件一样,方案具有较好的对称性。

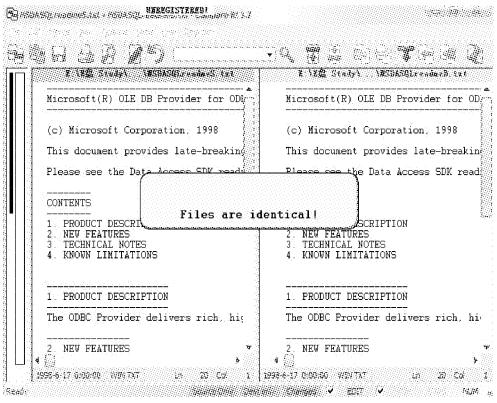

图 5　被隐藏的源信息与提取信息比较

4　方案分析

现有的两种向 PE 文件中插入信息的方法:1)把信息隐藏在添加的新节中; 2)把信息直接隐藏在节末尾的冗余空间。第一种方法会改变原始 PE 文件的总体大小;第二种方法虽然能保持 PE 文件的总体大小不变,但简单把大量信息集中存放在节末尾,也是极易被发现的,两者隐蔽性都不高。

本方案利用了 PE 文件资源节的冗余来隐藏信息,没有改变软件的实际大小,隐藏信息后的 PE 文件仍然可以正常执行和拥有原始功能。方案还根据资源节的特殊结构和资源调用特点,把第五层资源信息两端的冗余搬移到了第五层的资源节点之间,扩展资源节点间的冗余,然后把信息分块隐藏在这些分散的冗余中,克服了已有方法中隐藏信息过于集中、容易被发现受攻击的缺陷,提高了隐藏信息的隐蔽性。

本方案分析了资源节中存在的四种冗余,已有的把信息直接隐藏在节末尾的冗余空间中的方法,只是利用了资源节中的冗余 4。从表 2 中可看出本方案可利用的总冗余比资源节节尾的冗余 4 大很多,提高了资源节空闲可利用率和信息嵌入量。

表 2　已有方案与本方案资源节空闲利用比较(单位:字节)

可执行文件	资源节节尾冗余	本方案总冗余
AcroRd32. exe	3 248	16 370
Visio2002chsSetup. exe	2 764	11 146
dvdplay. exe	384	11 838
freecell. exe	240	1 439
calc. exe	160	1 195

本方案把信息分散在 PE 文件资源节的资源中,使隐藏信息和载体 PE 文件更好地融合在一起,在没有原始 PE 文件和密钥的情况下,查找出信息成功概率很低。如果嵌入信息的 PE 文件遭到剪切、添加攻击,都会使程序不能正常执行。对于可执行文件常用的压缩、加解壳操作,也都不会对隐藏信息造成破坏。本方案还对隐藏信息进行了加密预处理,提高了隐藏信息的抗攻击性。本方案保证了信息隐藏的稳健性。

5　结语

采用目前 Internet 环境下传输比例较高的 PE 文件作为隐藏载体,是近年来新的信息隐藏研究方向。本文分析了 PE 文件资源节的特殊结构和软件资源调用的特点,首次总结了资源节中存在的四种冗余及它们的优缺点,并在此基础上提出了一种新的信息隐藏方案。新方案充分利用了资源节的冗余空间来隐藏信息,没有改变 PE 文件的大小,用转移冗余空间的方法综合了几种冗余空间的优势,和已有信息隐藏方法相比,克服了隐藏信息过于集中的缺陷,提高了信息隐藏的隐蔽性和资源节的空闲可利用率,且方案能有效抵抗多种常见攻击,具有较强的鲁棒性。

参考文献:

[1]　王朔中,张新鹏,张开文. 数字密写和密写分析[M]. 北京:清华大学出版社,2005. 10 - 11.

[2]　胡珊. 向 PE 文件中插入可执行代码的研究[J]. 鞍山科技大学学报,2005,28(2): 119 - 122.

[3]　JIURL. JIURL PE 格式学习总结(四)——PE 文件中的资源[EB/OL]. http://www. blogcn. com/user3/jiurl/index. html, 2006.

(上接第 618 页)

[3]　ARKKO J , TORVINEN V , CAMARILLO G , *et al* . Security mechanism agreement for SIP sessions[R]. IETF Internet draft (draft-ietf-sip-sec-agree-04. txt), 2002.

[4]　SALSANO S, VELTRI L, PAPALILO D. The SIP Authentication Procedure and Its Processing Load [J]. IEEE Network, 2002, 16 (6):38 - 44.

[5]　SI DF, LONG Q, HAN XH, *et al*. Security mechanisms for SIP-based multimedia communication infrastructure [A]. IEEE Conference on Communications, Circuits and Systems (ICCCAS) [C]. New Jersey: IEEE Press, 2004. 575 - 578.

[6]　林霞,董魁松. SIP 认证机制的研究和改进[J]. 计算机工程与科学, 2006,28(4):13 - 18.

[7]　王宇飞,范明钰,王光卫. 一种基于 HTTP 摘要认证的 SIP 安全机制[J]. 重庆邮电学院学报, 2005,17(6):749 - 751.

[8]　杨雅雯. 无线网络与 3G/UMTS 整合环境之认证协定研究[D]. 台中:朝阳科技大学, 2005.

[9]　HTTP authentication: basic and digest access authentication, RFC 2617[S], 1999 - 06.