基于SSL VPN接入机制的研究*)

欧阳凯 周敬利 夏 涛 余胜生
(华中科技大学计算机科学与技术学院 武汉430074)

摘 要 SSL VPN是基于SSL安全机制,在不改变现有防火墙构架的基础上,实现的一种安全网络构架。本文提出的PHI(Private Handshake Information)协议是专门为SSL VPN设计的控制协议,通过该协议可以增加SSL VPN的安全保护,建立更为可靠的透明安全隧道。本文详细地论述了PHI-SSL VPN的体系设计以及PHI协议的原理,并对可能的攻击进行探讨。

关键词 PHI,SSL,VPN,隧道

The Research of SSL VPN-Based Access Mechanism

OU Yang-Kai ZHOU Jing-Li XIA Tao YU Sheng-Sheng
(College of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan 430074)

Abstract SSL VPN is a safety network framework implemented on the basis of SSL safety mechanism and the current firewall system. The PHI (Private Handshake Information) Protocol put forth in this paper is the control protocol particularly designed for SSL VPN, by which the safety protection of SSL VPN is strengthened and a more credibly transparent safety tunnel is built. This paper details the system design of PHI-SSL VPN and the principle of PHI. Moreover, it discusses the likelihood of the attack case.

Keywords PHI,SSL,VPN,Tunneling

1 引言

传统VPN(Virtual Private Network)是基于IP安全结构的网络安全体系[1,2]。典型代表是基于网络层实现的IPSec VPN,其主要问题:要求VPN远端服务群的防火墙打开多个端口;不能穿越支持NAT的设备[3]。

随着HTTPS[5]的广泛应用,一种基于SSL/TLS协议[4],依托Web Server的VPN构架(SSL VPN)应运而生,其相对于传统VPN,具有无客户端(Clientless),无需客户端管理的特性。据权威的IT研究与顾问咨询公司Gartner 2003年初统计,未来全球SSL VPN的市场增长速度将达到170%以上[6]。但是,SSL VPN的局限在于对传统网络应用程序(比如:HTTP、FTP)无法做到透明的支持。

因此,本文提出了PHI-SSL VPN,其核心思想是将PHI(Private Handshake Information)协议引入SSL VPN,通过PHI加强SSL VPN的安全控制。PHI-SSL VPN通过引入两项新的技术支持传统应用程序以及提高SSL VPN的安全性:首先,PHI-SSL VPN提供了两种VPN访问模型(Clientless模型和简单客户端模型),简单客户端(Legacy Client)模型使得传统应用协议的客户端能够透明、安全地访问VPN远端的服务群;其次,PHI-SSL VPN实现了独立于Web Server的接入控制、访问列表控制和用户生命周期控制。

2 PHI-SSL VPN结构体系

本节通过在一个现有防火墙体系中部署VPN服务器的案例,引出PHI-SSL的基本结构体系,进而从协议栈的角度探讨了PHI的设计以及PHI-SSL VPN的总体设计。

2.1 PHI-SSL VPN结构体系概述

现有高安全级别的防火墙,通常只允许对某个特定端口的访问开放,一般是443(HTTPS标准端口)或者80(HTTP标准端口)。

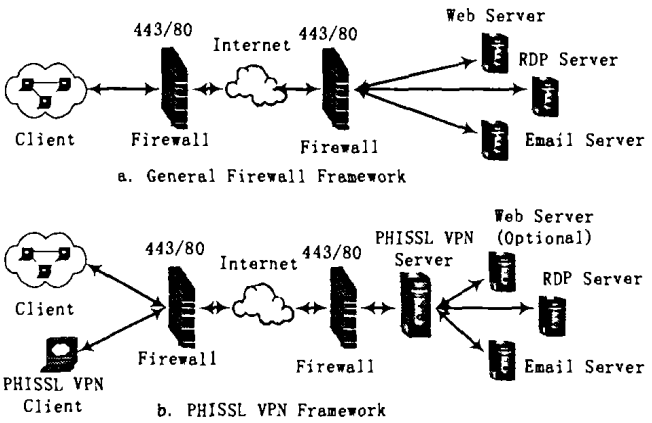


图1 PHI-SSL VPN结构

在图1-a中:当防火墙只允许443/80端口访问时,跨跃公用网的客户端无法访问内部的服务群(比如:RDP Server和Email Server);如果采用传统VPN,防火墙需要打开相应的端口,我们并不能假设对应的内部服务器是可靠的、无安全漏洞的,因此必然带来新的安全隐患;如果采用SSL VPN,意味

*)国家自然科学基金,编号:60373088.欧阳凯 博士研究生,研究方向为网络安全结构及网络存储的安全性.周敬利 教授,博士生导师,研究方向为高性能网络存储技术及网络安全结构.夏 涛 讲师,研究方向为高性能网络存储技术及网络安全结构.余胜生 教授,博士生导师,研究方向为高性能网络存储技术及网络安全结构.

着必须为每一种 TCP 应用层协议提供基于 SSL 的客户端版本，这既是对现有资源的巨大浪费，也需要花费大量人力物力。

本文提出的 PHI-SSL VPN 如图1-b 所示：首先确保现有防火墙结构和策略不改变，在服务群里添加一台 PHI-SSL VPN Server(该 Server 也可以和 Web Server 集成在一起)。其次，PHI-SSL VPN 结构提供了可选择的两种客户端模式：Clientless 模型——对一些典型的 TCP 应用层服务器提供基于 Java Applet 实现的跨平台客户端，只要客户端支持 Java，并有 Web 浏览器就可以随时下载客户端，并访问内部的服务群，同时保证所有的数据在公用网是安全、可靠的；Legacy Client 模型——目的是最大限度地支持已有的应用层服务软件，仅需要用户安装一次，对用户使用而言，和 Clientless 模型接入 VPN 没有任何区别，同样无需用户管理。

2.2 PHI-SSL VPN 协议栈设计

PHI 协议栈的引入是 PHI-SSL VPN 构架核心所在，如图2所示。PHI 协议处于 SSL 层与应用层之间，其目的是为 SSL VPN 提供更为可靠的安全控制和细粒度的访问控制。PHI 协议报文由协议消息头和负载组成，我们将在 PHI 协议一节中详细论述。

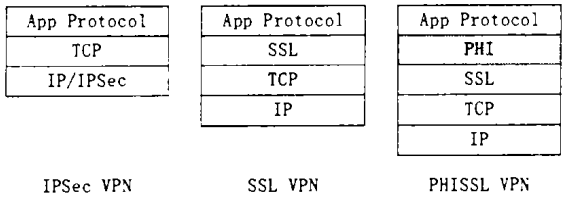


图2 三种 VPN 结构的协议栈

目前，PHI 协议包含了三个方面的设计：

A. 独立的认证接口机制：利用该接口做到 VPN 接入机制与 Web 登录认证机制分离，使得 PHI-SSL VPN 服务端可以根据需求采用不同的认证机制(比如：Radius 和 LDAP)验证用户的合法性；

B. 动态服务群访问列表：客户端只有在认证通过后，才能获得其能够访问的服务器对象描述列表；

C. 上层应用协议引擎：当客户端访问一台具体服务器时，通过该引擎，才可以建立一条能够穿越防火墙的、透明的和安全的隧道。

2.3 PHI-SSL VPN 设计

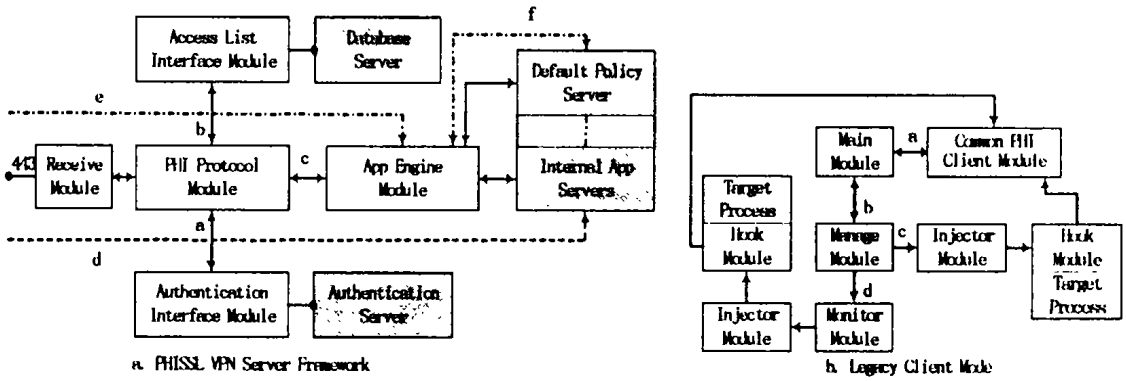


图3 PHI-SSL VPN 设计框架

图3-a 展示了 PHI-SSL VPN 服务端的设计。其中阴影方框代表了 PHI-SSL VPN 的必要部署，但不属于其本身设计范畴。当一次连接请求到达接收模块时，将其交由 PHI Protocol 模块仲裁，其处理主体流程如下：

a. 在 Authentication Interface 模块确认用户的合法性后，产生具有全局唯一性且不可逆的 cookie 值(与 Web cookie 不具有相关性)；

b. 通过 Database Interface 模块获取用户对服务群的访问控制列表，用户成功接入 PHI-SSL VPN；

c. 用户访问一台具体的内部服务器时，通过 App Engine 模块建立安全透明的隧道，其看见的是如虚线(d)所示的连接，而实际的连接由点虚线(e 和 f)组成；对于不属于 PHI 协议连接请求的范畴，PHI 只是盲目地通过 App Engine 模块转向缺省服务器(通常是 Web Server)。

对于两种客户端模型，其共性是实现了 PHI 协议的客户端部分。Clientless 模型还实现了应用协议相应的功能，这和具体协议相关，就不再一一讨论。我们仅论述传统 TCP 应用程序如何通过 Legacy Client 模型安全透明地访问远端服务群。

如图3-b 所示，Target Process 是已有的传统应用程序，程序本身并不支持安全、穿越防火墙访问服务器的功能。当 PHI-SSL VPN 的 Legacy Client 守护进程启动以后：

a. 通过 Common PHI Client 模块，接入 PHI-SSL VPN 构架；

b. 启动管理模块，其包含三个子功能，一是与主模块的互交管理，二是对已启动的应用软件管理，三是启动进程监控模块；

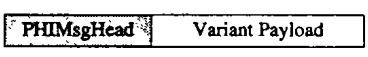
c. 通过管理模块获得当前进程信息，并将钩子模块植入目标进程；

d. 通过管理模块启动进程监控模块，该模块的作用是当系统创建客户需要访问远端服务群的进程时，触发植入模块，将钩子模块植入目前进程。

传统的应用程序通过钩子模块[7]与 Common PHI Client 模块互交以支持 PHI-SSL VPN 的访问，从而使得原本不支持 SSL VPN 的程序安全、可靠地通过公用网访问服务群，而客户端访问远端的服务群如同访问本地局域网一样方便。

3 PHI 协议

正如前面对 PHI-SSL VPN 体系的论述，不难看出 PHI 协议在该体系中的核心枢纽地位，本节详细论述 PHI 协议的结构，并讨论其如何保证 SSL VPN 的安全性。



PHI Protocol Format

图4 PHI 协议格式

如图4所示，PHI 协议由其报头 PHIMsgHead 和相应的负载(Payload)组成，其报头如下所示：

```
typedef struct {
    int32       phiMagic,
    int32       phiCommand,
    int32       phiResponeStatus,
    int32       phiTunnelTimeout,
    u_int32     phiPayloadSize,
} PHIMsgHead, *PPHIMsgHead;
```

phiMagic 域是 PHI 协议的魔术号，指明了该报文是 PHI 协议类型；phiCommand 域指明了 PHI 协议的命令类型；phiResponeStatus 域指明了 PHI 协议的应答消息状态，该域只在 PHI_RESPONSE 中有意义；phiTunnelTimeout 域指明了该次安全隧道建立后的生命周期；phiPayloadSize 域指明了紧随其后的数据负载尺寸。

3.1 PHI 协议状态描述

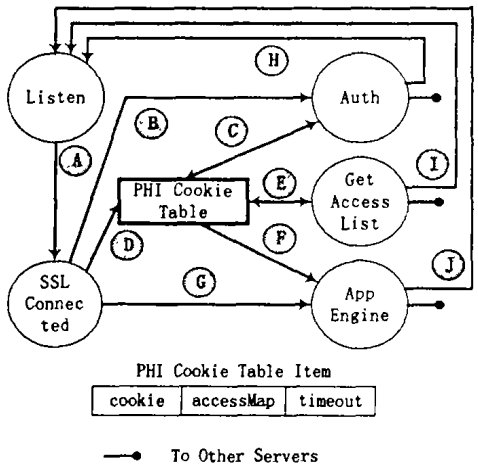


图5 PHI 协议状态

图5显示了 PHI 协议的状态变迁情况。其中 PHI Cookie Table 由〈cookie，accessMap，timeout〉组成，索引值是 cookie。

非 PHI 协议状态变迁 服务端接收到一个连接请求，建立 SSL 连接(A)；通过 App Engine 的缺省处理模式转发给缺省策略服务器处理(G)；返回 Listen 状态(J)。

认证状态变迁 服务端接收到一个连接请求，建立 SSL 连接(A)；由 PHI 消息头的 phiCommand 域值 PHI_CMD_AUTHENTICATION 指明类型，通过认证模块验证用户的合法性(B)；在验证成功后，创建 PHI cookie，更新 PHI cookie table(C)；发送 PHI 应答消息并返回 Listen 状态(H)。

服务器访问列表请求状态变迁 服务端接收到一个连接请求，建立 SSL 连接(A)；由 PHI 消息头的 phiCommand 域值 PHI_CMD_ACCESSLIST 指明类型，基于 cookie 的有效性验证访问的合法性(D)；再由服务器访问列表请求模块获得列表(E)；验证成功后，更新相应 PHI cookie 表项的 accessMap 域(E)，发送 PHI 应答消息并返回 Listen 状态(I)。

应用层协议引擎状态变迁 服务端接收到一个连接请求，建立 SSL 连接(A)；由 PHI 消息头的 phiCommand 域值 PHI_CMD_ENGINE 指明类型，基于 cookie 的有效性和 accessMap 的匹配，验证访问的合法性(D)；再由应用层协议引擎建立隧道(F)；发送 PHI 应答消息并返回 Listen 状态(J)。

3.2 增强 PHI-SSL VPN 的安全性—PHI cookie

在 PHI 协议状态图(图5)中，可以得知 PHI cookie table 在 PHI 协议中的枢纽作用，本节通过分析一般 SSL VPN 提供商的 VPN 接入机制以及相关案例阐述 PHI cookie 存在的必要性。

目前，SSL VPN 提供商都提供基于 Web Server 的 VPN 接入机制，通常是通过 Web Server 保证用户的有效性和请求的合法性，从而保证内部服务群的安全访问。我们认为，Web Server 的安全机制不是为 VPN 设计的，与 VPN 用户的接入没有直接必然的关系；如果 VPN 的接入安全性是由 Web 控制，则会形成 VPN 的单一失效点，并可能造成安全隐患。

我们通过三个案例说明如何通过 PHI cookie 增强 SSL VPN 的安全性。

案例1：若某恶意用户已知远端服务群的配置情况，其完全有可能绕过 Web 的 cookie 机制(比如利用 Web 服务器的安全漏洞)，在没有认证的情况下，接入 VPN 并伪造建立隧道请求，从而为所欲为。

解决：PHI-SSL VPN 在用户接入认证后，PHI 协议服务端都会产生 cookie 值，cookie 可以由服务器选择其独一无二的值(比如本机时间)为种子，通过单向散列函数(比如 MD5)生成全局唯一且不可逆的值，并把该值反馈给用户。以后该用户的每一次连接必须携带该 cookie。由于 PHI cookie 是 PHI 协议产生的，这样就算恶意用户取得 Web Server 的控制，也不可能在没有认证的情况下接入 VPN，攻击整个服务群。

案例2：某部门主管 A 有权限访问内部服务器 S1，而某雇员 B 则没有权限访问该 S1，假设 S1并不是绝对没有安全漏洞，为了 A 能访问，则 S1至少允许来自 VPN 服务器的访问。当 B 已知服务器对象描述，则其可以很容易利用自己的帐号接入 VPN，然后以 VPN 服务器为跳板入侵 S1。

解决：在用户接入认证后，服务端产生 cookie 值，并将其放入〈cookie，accessMap〉对中，其中 accessMap 指明了该用户可以访问的内部服务群映射关系，初始化为空；在用户通过 PHI 服务器访问列表请求成功后，动态填充 accessMap。该案例的 B 接入 VPN 以后，当其想攻击 S1时，PHI-SSL VPN 服务端发现这不属于 B 的访问范畴，是非法访问，直接回绝。

案例3：用户 C 可以访问 VPN 内部 Telnet 服务器 S2和 S3，当 C 使用 S2较长时间，并且没有 Web 访问(Web Server 已经认为超时)，按理应该认为该用户是活跃的，应该可以创建新的连接访问 S3，如果是基于 Web 的失效控制，则会拒绝连接；另一种情况，C 建立 S2连接后，在没有断掉连接的情况下离开，稍后 Web Server 发现超时，其只能断掉 HTTP 的连接，并不能终止 Telnet 服务器 S2与 C 之间的连接。

解决：PHI-SSL VPN 的超时机制由两个方面组成：cookie 的有效时间和 PHI 消息头的 phiTunnelTimeout 域。当用户接入 VPN 以后，产生 cookie，并放入〈cookie，accessMap，timeout〉对中，timeout 一般可由 PHI-SSL VPN 服务端配置获得，其含义是在 timeout 时间内，不存在任何含有 cookie 的连接，则认为该 cookie 超时，将来不再对携带该 cookie 值的任何连接建立安全隧道，除非用户重新接入 VPN，生成新的 cookie；而 PHI 消息头的 phiTunnelTimeout 域则仅控制一条安全隧道在没有数据流传输的情况下最大的有效时间。因此，案例3的情况则不会发生。

3.3 PHI 协议总结

PHI 协议是建立在 SSL 层之上的、专门为 SSL VPN 构架设计的控制协议。出于效率的考虑，该协议本身并没有加密，其传输时的加密是由 SSL 层完成的。

通过对 PHI 的分析和论述，PHI 协议具有如下特性：

A. 建立独立于 Web 登录机制的 VPN 安全接入机制；

B. 建立 cookie 安全控制机制；

C. 通过服务列表请求消息，获得细粒度的用户访问控制；

D. 通过 App Engine 请求消息，透明地访问 VPN 远端的服务群；

E. 良好的模块化设计，有利于与现有的认证机制(如 Radius、LDAP)以及数据库(如 Mysql、Oracle)整合。

4 SSL VPN 结构对比与性能分析

4.1 结构对比

目前，SSL VPN 领域并没有一个完整的标准以及公开的设计方案：一般 SSL VPN 提供商仅实现了基于 Web Server 的认证和接入机制，且没有考虑如何使传统应用软件能够安全地接入 SSL VPN(Netilla[8]、Whale[9]等)；一些 SSL VPN 研究性项目(Stunnel[10]、OpenVPN[11]等)只是实现了基于 SSL 的转发机制，对于 Clientless 模型没有考虑，并且对于用户接入的安全控制则没有涉及。

通过以上对 PHI-SSL VPN 结构体系的分析和论述，不难得出 PHI-SSL VPN 具有更为优越的体系设计思想，并且是一个完整的解决方案：

A. 提供了两种接入 SSL VPN 模型(Clientless 模型和 Legacy Client 模型)；

B. 引入 PHI 协议，增强安全性；

C. 具有良好的可扩展性，可以和现有的认证机制(Radius、LDAP 等)以及数据库(Mysql、Oracle 等)整合；

D. 细粒度的用户访问控制；

E. 完全透明地访问 VPN 远端的服务群。

4.2 性能分析

与基于 Web 的 VPN 接入机制相比，PHI 需要产生自己的 cookie，以及插入 PHI cookie 表的操作。但是用户接入 VPN 相对于用户使用 VPN 而言，是非经常性事件。因此根据访问局部性原理，为了体系的安全性，不经常发生事件(用户接入)产生的一定延迟是可以接受的。

假定：认证为本地用户认证(密码采用 MD5算法，64B/block，B 代表一个字节)；SSL 在握手建立以后，采用对称密钥加密算法保证数据安全性，我们以 RC4(128B/block)为例；Web cookie 采用 MD5算法(64B/block)生成；PHI cookie 采用与 Web cookie 同样的算法生成定网络零延迟(测试平台：CPU - Intel Pentium III 800，Memory - 256MB，Linux-2.4.21，SSL 库为 OpenSSL 0.9.7a)。

基于 Web 接入的 SSL VPN，在用户接入的响应耗时：

$$T_{web}=T_{decrypt}+T_{web\_auth}+T_{web\_cookie}+T_{encrypt}$$

基于 PHI 协议的 SSL VPN，在用户接入的响应耗时：

$$T_{phi}=T_{decrypt}+T_{web\_auth}+T_{web\_cookie}+T_{phi\_auth}+T_{phi\_cookie}+T_{encrypt}$$

$T_{decrypt}$是用户发送给服务器的 Web 请求包经过 RC4解密的耗时(当请求报文的长度不超过128B 时，需要一次 RC4解密)；$T_{encrypt}$是服务器返回给客户端的 Web 页面经过 RC4加密的耗时，是由页面尺寸决定的；T_{web_auth}、T_{phi_auth}、T_{web_cookie}和T_{Phi_cookie}分别是 Web 认证、PHI 接入认证、生成 Web cookie 和生成 PHI cookie 的耗时(由于均采用 MD5算法且明文不超过64B，故耗时相同)。

根据对 OpenSSL 加密算法耗时的测试(采用 speed 测试工具[8])，系统在3秒内分别通过 MD5和 RC4加密处理的字节数(单位：kB)如表1所示。

表1 加密性能测试表

次数	1	2	3	4	5
MD5	45376.06	45404	45292.5	45324.06	45290.5
RC4	85444.41	85430.52	85443.42	85448.11	85374.15

由此计算可得：

采用 MD5(64B/block)的平均耗时：4.14us；

采用 RC4(128B/block)的平均耗时：4.39us。

由于服务器返回给客户端的 Web 页面的尺寸不定，选取 Web 页面的尺寸为1kB、3kB、5kB、10kB、15kB 和25kB 为测试对象。

因此根据上述公式及算法性能测试值的计算，可得两种接入机制的响应时耗，如图6所示。

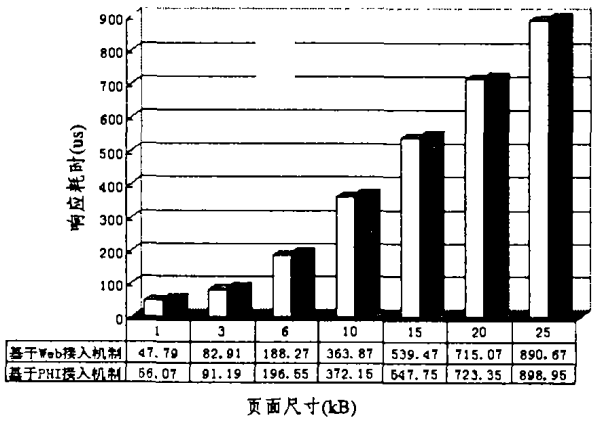


图6 响应时耗对比

根据图6，响应耗时随 Web 页面尺寸线性增长，这是由 Web 页面的加密耗时所决定的；由此，可以得知在 SSL VPN 中引入 PHI 协议所带来的延迟并不是 SSL VPN 的主要延迟，其主要延迟是数据(页面)加密耗时。

5 后续工作

基于安全网络体系结构考虑，我们的后续工作主要是建立入侵规范，并根据该规范实时与 IPS(Intrusion Prevention System)[12~14]通信。

IPS 是传统防火墙与 IDS(Intrusion Detection System)互交产生的一种新型计算机信息安全技术，具有主动防止入侵的特性，一般处于安全网络体系的前端；同样，作为 PHI-SSL VPN 的服务端，PHI 协议拥有自己的安全控制，可以根据该控制设计针对 SSL VPN 的入侵范式，主动向 IPS 报告入侵信息，使得 IPS 能够实时防范入侵攻击。

假设1：当来源同一个地方的客户端携带一个不存在或者超时的 cookie 值在短时间内发起两次或两次以上的 App Engine 请求，PHI 协议可以认为这是一种入侵，因为当发起第一次这样的连接，PHI 协议的服务端就会通知客户端先接入 VPN，而 PHI 协议的正常客户端则会停止这种尝试。

假设2：通过 HPI 协议的访问控制机制保证，基于 PHI 协议实现的 SSL VPN 构架是不可能出现案例2的情况，有理由认为案例2的情况属于入侵。

参 考 文 献

1 Lin A，Heinanen J，Armitage G，Malis A. RFC2764：A Framework

for IP Based Virtual Private Networks. Feb. 2000. http://www.ietf.org/rfc/rfc2764.txt
2 Kent S, Atkinson R. RFC2401: Security Architecture for the Internet Protocol. Nov. 1998. http://www.ietf.org/rfc/rfc2401.txt
3 Aboba B, Dixon W. Draft: IPsec-NAT Compatibility Requirements. Oct. 2003. http://www.ietf.org/internet-drafts/draft-ietf-ipsec-nat-reqts-06.txt
4 Dierks T, Allen C. RFC2246: The TLS Protocol Version 1.0. Jan. 1999. http://www.ietf.org/rfc/rfc2246.txt
5 Rescorla E, Schiffman A. RFC2660: The Secure HyperText Transfer Protocol. Aug. 1999. http://www.ietf.org/rfc/rfc2660.txt
6 Gartner Company Site. http://www3.gartner.com/
7 Hunt G, Brubacher D. Detours: Binary Interception of Win32 Functions. In: Proc. of the Eighth ACM SIGOPS European Workshop. 1999
8 Netilla Company Web Site. http://www.netilla.com/
9 Whale Company Web Site. http://www.whalecommunications.com/
10 Stunnel Project. http://www.stunnel.org/
11 OpenVPN Project. http://sourceforge.net/projects/openvpn/
12 Ghosh A K, Schwatzbard A, Shatz M. Learning Program Behavior Profiles for Intrusion Detection. In: Proc. 1st USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara, California, 1919
13 Snort Project. http://www.snort.org/
14 The Florida Honeynet Project. Know Your Enemy: Gen2. Nov. 2003. http://www.honeynet.org/

(上接第47页)

类别6 Reliability-related

特性 Fault Tolerance 描述了系统性能随着随机的出错节点的增加而发生的变化，我们可以随机选择系统中的节点，进行故障模拟，然后考察系统各方面的性能随着出错节点的增加而变化的情况。

特性 Attack Tolerance 描述了系统性能随着某些特定的出错节点的增加而发生的变化。与上述特性的区别在于，这里选取的出错节点是某些特定节点，如“连接度最大的节点”，“协作发起节点”等。

特性 Recovery 描述了系统从错误中恢复的能力，我们从多次试验中统计系统错误恢复时间，并考察错误恢复后系统能否继续正常工作、系统性能上的变化等。

类别7 Security-related

特性 Encryption Reliability 描述了系统加密方法的可靠性。针对系统对安全级别的不同要求，判断系统中使用的加密算法、Hash 算法、数字签名算法及密钥长度等是否合理。

特性 Access Control 描述了系统中的访问控制策略。我们可以从可能的攻击手段出发，考察系统的访问攻击策略的有效性。常见的攻击手段包括攻击数据源，窜改被请求的文件，拒绝承认非法操作等。

特性 Anonymity 描述了系统防止信息审查、保护用户隐私的能力。类似于 FreeNet 的考察方式，我们可以从匿名性的类型划分、需要考虑的系统攻击类型、匿名性的级别这三方面来综合评价系统的 Anonymity。

类别8 Scalability-related

特性 Discovery Scalability 描述了随着节点数目和负载的变化，系统在发现目标节点方面的性能变化，Pathlength 值（平均值，中数等）、Hit Ratio、Response Time 等的变化情况都是我们关注的对象。

特性 Interoperation Scalability 描述了随着节点数目和负载的变化，系统在协作方面如 Interoperation decision、Interoperation negotiation 等性能变化。

特性 Resource Scalability 描述了随着节点数目和负载的变化，系统在分摊负载方面的表现，包括负载在各个节点上的分布以及系统对资源的消耗等发生的变化。

特性 Capacity Scalability 描述了随着节点数目和负载的变化，系统在吞吐量方面性能的变化。我们可以度量系统的 Transaction Throughput、Byte Throughput 等的变化情况。

由于分析问题的角度不同，一些服务质量特性可以归属多个类别；同时，类别的划分也并非完全互斥的，这一点在 Scalability 这一类别上有所体现。Scalability 这一类别中的服务质量问题并不是一些全新的问题，而是侧重考察了在网络大小或系统负载等因素发生变化时其它类别中的服务质量特性相应的变化。

小结 本文首先对 P2P 系统作了简单的介绍，然后引出了 P2P 系统的服务质量问题，并以 Freenet 系统为实例，介绍了在 Freenet 上的服务质量研究工作，最后，我们提出了一个 P2P 系统的服务质量度量框架，力图系统全面地对 P2P 系统进行评估。这个 P2P 系统的服务质量度量框架将对 P2P 系统的设计和选择起到帮助和指导作用。

参 考 文 献

1 Milojicic D S, et al. Peer-to-Peer Computing: [Tech. Rep. HPL-2002-57]. HP Laboratories Palo Alto. March 2002
2 Campbell A, Aurrecoechea, Hauw. A Review of QoS Architectures. In: Proc. of the 4th IFIP Intl. workshop on Quality of Service (IWQS'96), Paris, March 1996
3 Waddington D G, Coulson G, Hutchison D. Specifying QoS for Multimedia Communications within Distributed Programming Environments. In: COST237 Workshop, 1996
4 Stoica I, Morris R, Karger D, Kaashoek M F, Balakrishnan H. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. SIGCOMM'01, San Diego, California, USA, Aug. 2001
5 Hong T. Performance, in Peer-to-peer: Harnessing the Power of Disruptive Technologies. Oramed A, ed. O'Reilly: Sebastopol, CA, USA, 2001
6 Clake I, Sandberg O, Wiley B, Hong T W. Freenet: A Distributed Anonymous Information Storage and Retrieval System. ICSI Workshop on Design Issues in Anonymity and Unobservability, 2000
7 Zhang Hui, Goel A, Govindan R. Using the Small-World Model to Improve Freenet Performance. IEEE Infocom, 2002
8 Clarke I, Hong T W, Miller S G, Sandberg O, Wiley B. Protecting Free Expression Online with Freenet. IEEE Internet Computing, 2002, 6(1): 40～49
9 Reiter M K, Rubin A D. Anonymous web transactions with Crowds. Communications of the ACM, 1999, 42(2): 32～38
10 Watts D J, Strogatz S H. Collective Dynamics of Small-World Networks. Nature, 1998, 393: 440