# 用于保护企业计算环境中基于RESTful Web服务的自适应安全体系结构

## 摘要

在这个现代企业计算时代，企业应用程序集成（EAI）是一个众所周知的业界公认的体系结构原则，它基于松散耦合的应用程序体系结构构建，其中面向服务的架构（SOA）是EAI的架构实现,其计算元素被称为“服务”。尽管SOA可以在各种技术中实现，但SOA的Web服务实现由于其基本互联网协议的简单性而成为当前的可选选择。 Web服务技术定义了几种支持协议和规范，例如用于与客户端和服务器进行数据交换通信的SOAP和WSDL。近年来SOA中出现了一种新的架构范式，称为REpresentational State Transfer（REST），它也被用于通过系统集成联盟集成低耦合的服务组件，称为RESTful Web服务。这个SOA实现并不具备足够的安全解决方案，其安全性完全依赖于比如Web 2.0及其升级版本Web 3.0等最新Web技术而过时的网络/传输层安全性。供应商安全产品具有主要的实现限制，例如他们需要安全的组织环境而违反SOA规范，因此引入新的漏洞。于是，我们从流行的OWASP评级方法的角度审视RESTful Web服务的安全漏洞，并分析现有安全解决方案中的差距。因此，我们为REST提出了一种自适应安全解决方案，它使用公钥基础技术来增强安全体系架构。所提出的安全体系结构被构建为一种自适应的前向物联网（IoT）友好安全解决方案，其由三个循环部分组成：学习，预测和预防。引入了一种新型的安全组件“智能安全引擎”，该安全组件利用人工神经网络学习算法预测SOA上可能发生的安全威胁，然后根据所研究的理论安全模型得到的结果预测了对SOA的潜在攻击，并且安全解决方案部分的书面算法可以防止SOA攻击。本文旨在介绍其中一种此类算法，以RESTful Web服务中的SOA攻击，并讨论实时SOA环境中进行的概念验证结果的讨论。与其他竞争解决方案进行比较来得出我们提出的系统的优越性。

关键字

SOA Web服务 安全性 REST EAI

## 1 介绍

当企业中的相关应用程序集成到实现小型到大规模计算系统的设计，开发，部署和维护的松散耦合成为当前应用时代的生存和成功的强制要求时，企业计算进行了模式转变,即强制形成了企业应用程序集成（EAI）标准。 但是，由于平台，数据库结构，计算语言以及供应商提供的不同架构解决方案的差异，EAI一直面临着许多挑战。 另外，原有制造商不再支持传统系统[1]。 EAI在实现三个核心目标方面面临着这些急迫的挑战：（1）跨系统的数据集成，（2）独立于供应商架构（3）用于应用程序集成的公共接口。

尽管EAI可以使用许多不同的技术来实现，但基于松耦合的独立业务运营模块（称为“服务”）的面向服务的架构（SOA）是业界最新的战略技术趋势，并由于其简单性成为组织的不错选择。 Web服务是SOA实现，可用于超文本传输协议（HTTP）等基本Internet协议。 有两种Web服务：简单对象访问协议（SOAP）和REpresentational State Transfer（REST）[2，3]。

SOAP是一种行业标准通信协议，因为它使用服务接口来展示客户端和服务器之间的业务逻辑,它建立有状态通信并为Web服务提供可扩展标记语言（XML）格式的数据传输。 然而，REST只是Roy Fielding在他的博士论文“架构风格与基于网络的软件架构设计”[4]中提出的一种架构风格。 REST使用任何Web支持的数据格式（如JavaScript对象表示法（JSON）和客户端和服务器之间的XML）进行无状态数据传输，该服务器基于直接HTTP方法（如GET，PUT，DELETE和通过URI进行POST）。 基于此REST原则的公开业务服务称为RESTful Web服务。

由于REST在通信中的轻量级，它为企业计算带来了诸多优势，例如带宽更少，服务速度更快。 但是，由于没有服务接口，REST不是协议，并且没有元描述。 这导致了不同于SOAP Web服务中建立的特定安全模型。 目前，REST数据传输的安全性取决于网络和传输层安全，这些特性由于最新一代的网络通信及其漏洞[5,6,7,8,9,10]而完全过时，例如传输层安全性 （TLS）信道，这些信道应该经常重置以用于非点对点通信。

作为文献综述的一部分，本文对现有的和相关的REST安全解决方案进行了分析，发现它们都不能做到为整个RESTful Web服务设计提供全面的安全解决方案而不会违反REST和SOA原则。 研究人员提出了几种安全解决方案, 但是，它们可能会导致引入其他漏洞，因为通常安全协议被解释为非常棘手，而供应商安全产品主要运行在需要安装特定产品的受信任和安全的组织环境中。

在本文中，我们提出了一个自适应安全体系结构，在不违反任何REST和SOA原则的情况下，在安全和非安全的企业计算环境中保护RESTful Web服务免受SOA攻击。 所进行的研究强调，这个提出的安全架构上的RESTful Web服务可能是目前业内流行的基于SOAP的Web服务的更好选择。

本文的其余部分安排如下：第二章在验证其重要性方面回顾了研究课题。 所建议的安全体系结构的方法和功能在第三章中进行了说明。 实施和验证分析在第四部分讨论。 相关的作品以及与我们的贡献的讨论和比较在第五部分给出。 第六部分通过展示原始贡献，局限性，影响和潜力来总结本文,为未来发展做铺垫。

## 2 背景和动机

以下小节描述了REST Web服务的体系结构模型，其对实时统计安全漏洞数据的安全需求，进行研究的动机以及对文献中针对REST的可用安全解决方案的回顾。

### 2.1 RESTful Web服务架构

与基于SOAP的Web服务不同，RESTful Web服务不保留单独的注册表来维持服务描述细节。 基于SOAP的Web服务的基本体系结构如图1所示。服务描述由服务提供者以Web服务描述语言（WSDL）的形式发布到服务注册中心，然后，服务请求者从服务提供者基于通过SOAP协议定义的服务接口调用所需的服务。 可以通过SOAP请求/响应的元数据描述来强制保证安全。 在RESTful Web服务的情况下，服务请求者通过HTTP方法直接调用服务提供者，如图2所示。在这种REST架构原则中，不可能对底层Web服务调用的元数据元素应用安全性。 因此，REST风格的Web服务比基于SOAP的服务更容易受到攻击。

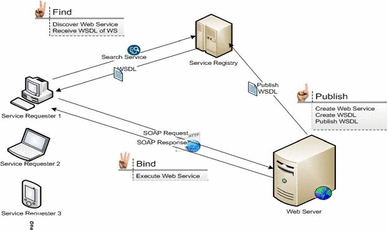


图1 基于SOAP的web服务



图2 基于REST的web服务

### 2.2 RESTful Web服务的安全漏洞

开放Web应用安全项目（OWASP）是一个处理Web应用程序安全问题的国际性非营利组织[11]。 它定期收集和观察与全球级安全事件相关的信息，并提供Web应用程序安全性的最高漏洞列表; 其关于“十大漏洞”的报告在安全行业得到了广泛的认可，随后又有安全产品的主要供应商跟进。 由于RESTful Web服务基本上是一个Web应用程序，因此应该彻底处理OWASP评级方法列出的以下安全漏洞，以提供安全服务。

* 注入攻击和消息改变
* 基于认证的攻击
* 拒绝服务（DoS）和缓冲区溢出
* 跨站点脚本/跨站点请求伪造
* 中间人（MITM）攻击
* 重播攻击和欺骗
* 不安全的直接对象引用
* 敏感数据曝光
* 缺少函数级别的访问控制
* 未经验证的重定向和转发
* 恶意软件故障

由于REST API使用简单的基于Internet的HTTP / HTTPS协议而不是使用诸如CORBA，COM / DCOM和RMI等复杂的特定于应用程序的协议，因此它容易出现所有应用层安全漏洞，如XSS和参数篡改[12,13，14]。 通常REST会处理JSON格式的数据交换; 有可能通过JSON注入颠覆或干扰服务上的应用逻辑，这可能会导致服务异常，数据失窃，资源删除和恶意软件执行。 由于所有基于Web服务的交互都是通过基于文本的数据交换在Web上进行的，但由于HTTP的本质，当然缺乏保护用于实现SOA的Web服务的数据的安全性证明。

惠普企业安全公司[15]的2015年“网络犯罪成本”研究表明，网络犯罪的平均年度成本为1500万美元，每家公司每年在美国的销售额为1.999美元至6500万美元不等。根据Symantec的报告[16]，2015年有超过4.31亿新的恶意软件样本曝光，这意味着每天大约发布一百万个威胁。 WhiteHat Security，Inc.于2016年发布的Web应用程序安全统计报告指出，虽然应用程序安全解决方案已有多年，但漏洞仍然猖獗;开发人员和IT团队在开发，定制或实施应用程序时需要优先考虑安全性和功能性。即使引入了基于硬件的新解决方案，发生拒绝服务（DoS）/分布式拒绝服务（DDoS）攻击的比例也不会降低[18,19]。设计时必须充分考虑安全方面的问题，因为网络服务由于其架构设计而需要高安全性。

### 2.3当前RESTful安全性的限制