

基于云计算的欺骗蜜网环境编排机制的研究

梅鹏, 罗守山

(北京邮电大学网络空间安全学院, 北京 100876)

摘要: 随着互联网技术的发展, 网络安全形势也变得日益严峻。针对信息系统易被探测和攻击等问题, 欺骗蜜网作为一种主动防御技术值得关注。然而, 当前欺骗蜜网环境的构建存在业务场景难以动态学习、无法重构复杂的大规模场景等问题。云计算是一种基于分布式计算、网格计算和并行计算的商业计算模型, 以一种按需使用, 按量计费的方式提供计算、存储、网络等资源。云计算技术的发展为欺骗蜜网环境构建提供新的解决思路, 因此本文提出一种基于云计算环境的欺骗蜜网环境编排机制, 并设计与实现可视化的欺骗蜜网环境编排系统。聚焦于网络拓扑拖拽式编辑以及动态学习业务运行环境, 抽象业务资源依赖关系以及基于DAG的编排算法的研究, 利用云环境重构业务运行的欺骗蜜网环境。最后, 使用 OpenStack 开源云计算平台构建实验环境, 并通过实验验证该编排系统的可行性。

关键词: 云计算; 欺骗蜜网; 资源编排机制; 有向无环图

中图分类号: TP399

Research on Orchestration Mechanism of Deception Honeynet Environment Based on Cloud Computing

MEI Peng, LUO Shoushan

(School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: With the development of Internet technology, the network security situation has become increasingly severe. Aiming at the problem that information systems are easily detected and attacked, deceiving honeynets is worthy of attention as an active defense technology. However, the current construction of deceiving honeynet environments has problems such as difficulty in dynamically learning business scenarios and inability to reconstruct complex large-scale scenarios. Cloud computing is a business computing model based on distributed computing, grid computing, and parallel computing. It provides computing, storage, and network resources in an on-demand, volume-based manner. The development of cloud computing technology provides new solutions for the construction of deceptive honeynet environments. Therefore, this paper proposes a mechanism for orchestrating honeynet environment based on cloud computing environment, and designs and implements a visual deception honeynet environment orchestration system. Focus on the research of network topology drag-and-drop editing and dynamic learning of business operating environment, abstract business resource dependencies and DAG-based orchestration algorithms, and use the cloud environment to reconstruct the deceiving honeynet environment of business operations. Finally, the OpenStack open source cloud computing platform is used to build an experimental environment, and the feasibility of the orchestration system is verified through experiments.

Key words: cloud computing; deceiving honeynets; resource orchestration mechanism; directed acyclic graph

0 引言

随着信息化技术的不断深入发展, 网络安全也日益严峻, 主动防御技术是一种可以让防

作者简介: 梅鹏 (1995-), 男, 硕士研究生, 主要研究方向: 信息安全、云计算

通信联系人: 罗守山 (1962-), 男, 教授、博导, 主要研究方向: 密码学、网络安全. E-mail: buptlou@bupt.edu.cn

御方占据主动地位的防御技术，蜜网技术是主动防御中的关键技术。欺骗蜜网是通过虚拟化技术实现的仿真工作网络，可以生成与环境相适应的各种诱饵，包括网络设备、主机系统、应用服务等。然而目前的欺骗蜜网环境编排面临两个问题，一方面是如何动态生成与环境相对应的网络拓扑结构、系统配置、安全防御机制等运行环境，另一方面是基于现有技术的欺骗蜜网环境编排受物理资源的限制以及无法满足大型应用场景。

近年来，云计算以其灵活性、易伸缩扩展性的特点受到学术界和工业界的广泛关注。云计算是一种建立在分布式计算、并行计算、网格计算以及虚拟化技术等基础上按需使用、按量付费的计算模型，用户可以根据需求便捷的使用云所提供的 CPU、内存、存储、网络等多种资源。云计算作为一种新兴的技术，在电子商务、商业计算以及互联网领域拥有巨大的发展潜力。

云计算在不同领域中具有丰富的应用案例，在面对构建欺骗蜜网环境的需求时，云环境下可伸缩、弹性的计算、网络、存储等资源为欺骗蜜网环境的建立提供充足的基础设施资源。因此，基于真实的业务运行环境，利用云计算技术重构欺骗蜜网环境是本文的研究重点。

1 相关技术

1.1 云计算

云计算是一种新兴的商业计算模型，它是一种按需使用，按量付费的资源提供方式，具有易扩展、灵活性、敏捷性、低成本等特点。云计算的核心是新一代数据中心技术，包括并行处理、网格计算、分布式计算以及数据中心虚拟化等。

云计算的体系架构如图 1 所示:最底层是计算、网络、存储等硬件设备，利用虚拟化技术将物理资源进行抽象，将计算、网络、存储等各种资源整合成资源池，提供统一的接口，实现资源的共享、隔离、动态分配，并结合安全、自动化、监报告警、多租户管理、资源管理等不同层面的功能，以网络作为基础访问，向用户提供弹性、自助、简单的服务^[1]。



图 1 云计算体系架构

Fig.1 cloud computing architecture

70 根据服务内容和层次的不同，云计算又可分为三种类型：

(1) SaaS:软件即服务，用户并不直接控制系统，硬件等资源，不必维护软件本身，只需使用软件提供的服务并进行相应的付费。SaaS 服务提供商利用网络将软件交付用户，常见的产品如人脸识别、智能语音、视频服务等。

75 (2) PaaS:平台即服务，云服务商提供给用户开发，运维应用程序的运行环境，用户负责维护自己的应用程序但并不控制底层的基础设施，常见的产品如云数据库、云缓存、对象存储等。

(3) IaaS:基础设施即服务，云服务商以服务的形式提供虚拟硬件资源，如虚拟主机/存储/网络/安全等资源,用户具有较大的自主权,可以在提供的资源上进行个性化的配置与部署。常见的产品如:云服务器、负载均衡、云硬盘等。

80 **1.2 有向无环图**

在图论中，如果一个有向图无法从某个顶点出发经过若干条边回到该点，则这个图是一个有向无环图，简称 DAG 图。有向无环图是一种存储数据的方式，由一组顶点和有向边组成，有向是指所有数据顺着同一个方向存储，无环指数据结构间不构成循环。

85 云计算环境中资源之间具有相互依赖性，各种资源的实例化需要遵循一定的顺序进行，有向无环图是描述系统进行过程的有效方式，DAG 图中包含多个节点，每个节点对应一个任务，每个节点之间的有向边表示任务之间具有一定的关联关系^[2]。如图 2 所示：

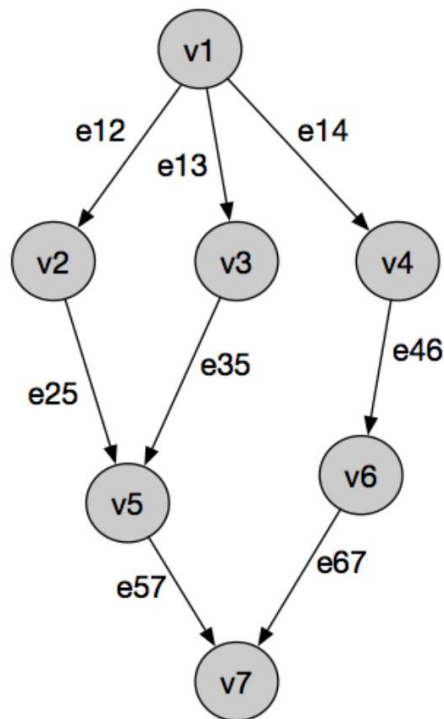


图 2 有向无环图示例

Fig.2 DAG model example diagram

1.3 调度算法

1.3.1 表调度算法

表调度算法的核心思想是赋予每个任务优先级，以此确定一个调度队列，根据队列从第一个节点开始执行任务。在传统的表调度算法中，调度队列需要预先确定，并且在调度的过程中不可更改。这种方式在调度过程中对于需要优先执行的任务但是优先级低而不能执行，因此基于动态表调度的算法也被提出。改进后的表调度算法依然采用优先级作为调度的判断依据，但是每执行一次任务后都会重新计算调度队列，再选取优先级最高的任务进行调度。目前基于表调度的算法有 ETF、LAST、ISH 以及 MCP 算法等。

1.3.2 聚簇调度算法

聚簇调度算法的核心思想是以最多的处理机数目来最大程度减少任务的调度长度，将每个任务都作为一个簇，通过对簇的不断合并直到无法合并为止。聚簇调度算法可以分为线性聚簇和非线性聚簇，如果任务之间具有依赖关系，那么就是线性聚簇，反之如果任务之间互相独立，那么就是非线性聚簇，目前常用的聚簇算法有 MD、DCP、LC、EZ 等^[3]。

1.3.3 基于任务复制算法

基于任务复制的算法核心是在不同的处理机上复制任务，以此减少任务间的通讯延迟，但由于向后搜索和存储更多的任务，任务复制算法的时间以及空间复杂度都相比较同类算法要高。目前基于任务复制的算法有 DSH、PY、LCTD、TCSD、BTDH 等。

1.3.4 遗传算法

遗传算法是学者模拟生物进化和遗传学机理的计算模型，遵循适者生存、优胜劣汰遗传机制的随机化搜索方法，核心是选择、交叉、变异操作。通过对问题的解进行编码，每一个解看作遗传算法中的一条染色体，将整个染色体群一代代不断进化，直到收敛到最优解。遗传算法是一种高度并行、随机、自适应的启发式智能算法，常见的启发式算法有蚁群算法、模拟退火算法、神经网络算法等^[4]。

2 相关工作

蜜网作为一种有效的主动防御技术，打破攻防博弈中防御方的被动地位。而近年来云计算、SDN 以及虚拟化等技术的日益成熟，为蜜网环境的构建带来新的解决思路。围绕着蜜罐与蜜网技术的研究，以及基于虚拟化、云计算技术的系统构建与资源编排方案，国内外已经有了一些研究。

(1) 贺文娟在《蜜罐技术分析与蜜网设计》^[5]中对蜜罐的原理以及技术动态进行了阐述，并以虚拟机为基础构建了一个虚拟蜜网系统，通过防火墙 IPtables 对系统进行监控以及数据捕获。

但是，没有考虑到业务场景的复杂性，利用虚拟机技术搭建的虚拟蜜网，在面对复杂网络环境以及多样的软件应用时无法保证诱饵与周边环境的高仿真度。

(2) 胡毅勋在《基于 Openflow 的主动防御关键技术研究》^[6]中提出一种基于 Openflow 的动态蜜网构建模型，解决现有蜜网系统流量控制困难的问题，引入网络虚拟化和服务虚拟化技术。提出了叠加虚拟蜜网的概念，可以在一个蜜网物理实体中叠加多个虚拟蜜网。设计了一种虚拟蜜罐系统，这种蜜罐系统可以虚拟运行多个主机节点，并通过实验和仿真验证分析其有效性和性能。

但是，其更多的是针对蜜网系统的模型化构建，缺乏对真实业务场景的学习以及资源关系的抽象与实例化过程。

(3) 李祥池在《基于 SDN 的网络攻防平台的研究与实现》^[7]中利用 SDN 和虚拟化等技术，设计开发了一套基于 SDN 的网络攻防平台。网络虚拟化使用开源的企业级虚拟交换机 OpenvSwitch，使用 Floodlight 作为 SDN controller 控制和管理整个网络部分，并且搭建测试环境对重要部分进行测试，验证了平台的可用性和可行性。

但是，没有对三层网络和高级网络服务进行测试与验证，且无法实现业务场景的自动编排功能。

(4) Antonescu A F 等在《Dynamic Topology Orchestration for Distributed Cloud-Based Applications》^[8]中描述了一种规范语言和架构，用于动态地管理分布式软件和映射的计算、存储和网络基础设施服务，将其称为动态应用程序拓扑编排，基于服务水平协议(SLA)中的操作约束来动态调整分布式、互连、相互依赖的应用程序服务和基础设施资源的映射和配置。

但是，该文只关注分布式应用程序的部署，而不关注虚拟资源的可扩展性。对于业务场景的描述以及基于此的网络拓扑在云环境下的重现是关键部分，对于网络设备、应用程序、

主机系统之间的依赖关系也没有进行抽象化处理。

综上所述，前两篇文章分别通过虚拟机以及 Openflow 来实现欺骗蜜网的构建，但是都有着各自的缺陷，受物理资源资源的限制以及所能实现的网络、主机、应用等无法满足大规模的、复杂的、多变的业务场景。第三篇在基于 SDN 和虚拟化实现一套网络攻防平台，但是受所能提供的网络层次限制，并且在资源的编排上有待解决。最后一篇提供了一种规范性语言描述应用程序拓扑，并可动态映射到基础设施服务，但是缺少对资源依赖关系的抽象以及具体的业务场景重现策略阐述。本文提出了一种基于云计算的欺骗蜜网环境编排机制，并基于此实现可视化的欺骗蜜网环境编排系统。

3 欺骗蜜网编排系统设计

3.1 欺骗蜜网编排系统功能组件

欺骗蜜网环境编排系统的功能组件主要包括 Web 子系统、生成拓扑描述文件的模版管理子系统、抽象有向无环图的模版解析子系统、生成执行计划的资源调度子系统、以及具有可扩展性的云基础设施接入子系统。

3.2 欺骗蜜网编排系统总体架构

欺骗蜜网编排系统总体架构共分为三个部分：运行环境学习、欺骗蜜网环境重构以及云接入，如图 3 所示：

运行环境学习主要是 Web 子系统，主要功能是提供 Web UI 界面、网络拓扑编辑以及网络拓扑配置，采用 B/S 模式，通过浏览器在 Web 界面通过拖拽的方式完成网络拓扑的编辑以及网元信息的配置，包括网络设备、主机系统、以及应用服务等多种诱饵。

欺骗蜜网环境重构主要包含模板管理子系统、模板解析子系统、以及资源调度子系统；服务端在接收到来自 web 端编辑生成的业务环境拓扑信息后，经过模板管理子系统生成拓扑描述文件，再经过模板解析子系统生成 DAG 有向无环图来描述资源之间的依赖关系，以 DAG 图为基础，通过资源调度子系统来生成包含若干个可并行调度队列的执行计划。

云接入包含负责不同的云基础设施提供商接入该系统，避免产商锁定等问题，该部分主要包含两个部分：身份认证以及请求的封装。

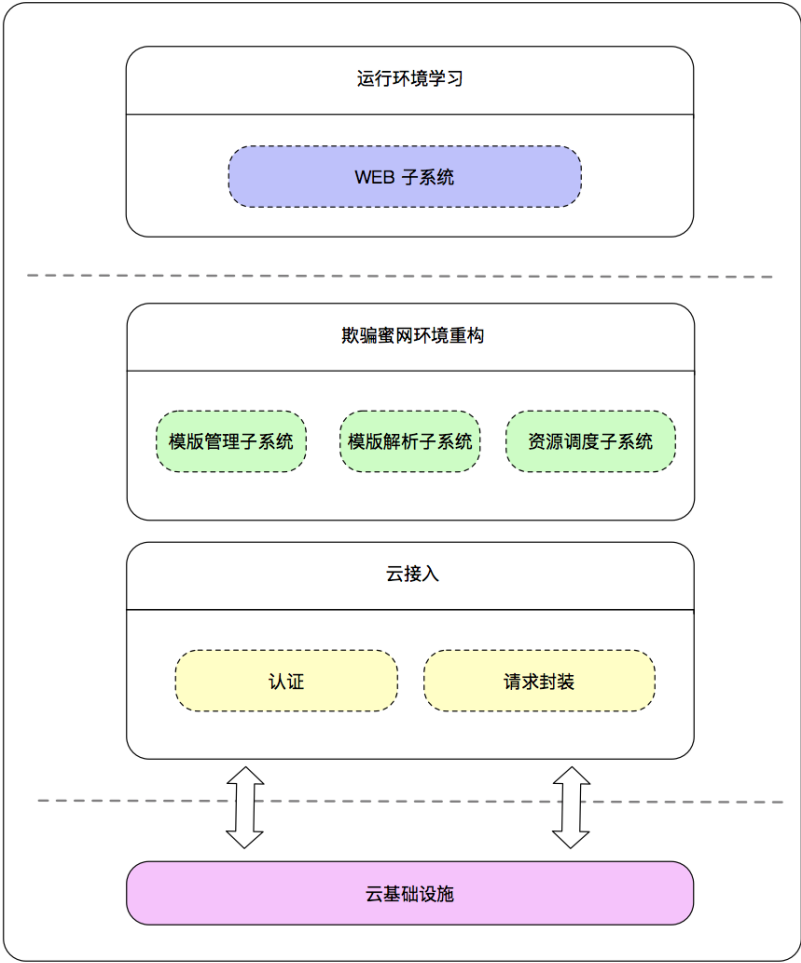


图 3 欺骗蜜网编排系统总体架构图

Fig.3 Deception honeynet orchestration system overall architecture diagram

3.3 欺骗蜜网编排系统详细设计

3.3.1 Web 子系统

Web 子系统主要包括用户 Web UI 界面，网络拓扑编辑，网络拓扑配置三个功能。Web UI 界面是用户和系统交互的接口，用户通过 Web UI 界面向系统发起请求，实现前端和后台的交互。网络拓扑编辑和网络拓扑配置主要完成用户定制化创建和管理网络靶场，即：用户可以通过拖拽不同的网元节点来构建复杂的网络拓扑，配置虚拟网络的属性，如网络类型、网关、虚拟交换机、路由器、安全组的配置等，最终要完成网络拓扑的提交，将网络拓扑的相应参数进行封装并通过 HTTP 协议发送至服务端。

3.3.2 模版管理子系统

在 Web 前端构建网络拓扑结构并完成相应的配置，服务端在接收到相应的网元和属性配置信息后，通过模版管理子系统生成网络拓扑描述文件。模版是一种文本文件，描述网络拓扑中的网元属性、配置信息、网元之间的依赖关系，可以使用 git, svn 等工具来维护不同版本的模版，确保网络拓扑可配置、可演进、可回溯。模版的组成结构如图 4 所示：

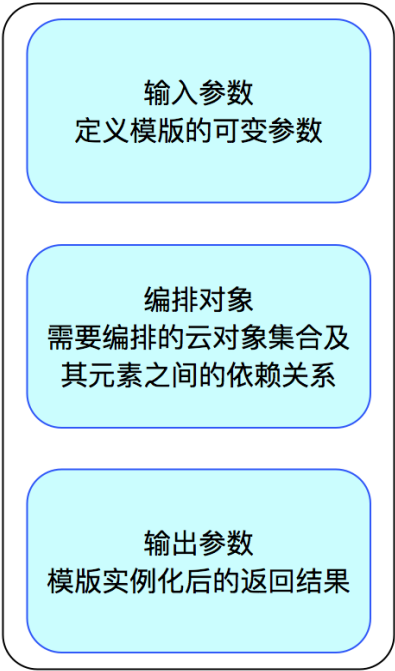


图 4 模板组成

Fig.4 template composition

模版管理子系统根据编排网元对象的不同将模版划分为多种类型，如:网络、子网、浮动IP、安全组等。每种类型的模版又可分为基础模版和实例模版，这种区分在概念上与虚拟化技术中镜像的派生类似，参考从后备镜像中派生出实例镜像的过程，基础模版包含特定网元的描述信息，具有通用性，实例模版是根据用户的个性化配置从基础模版中生成。所有的模版文件保存在模版存储库中，本质上是利用服务端文件系统存储^[9]。如图 5 所示：

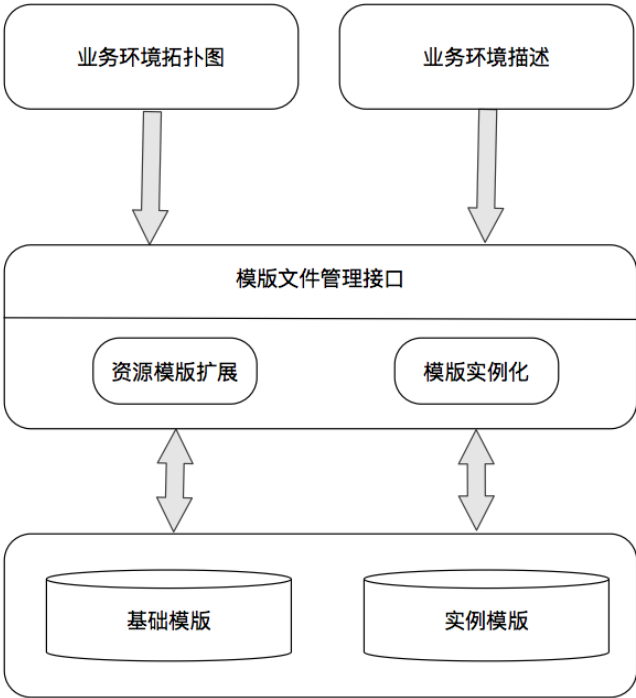


图 5 模板管理子系统

Fig.5 template management subsystem

本文设计的模版管理子系统，主要是对请求的处理与参数的提取，管理存储库中存储的各类对象元素，包括资源对象和关系对象，模版文件的填充与实例化，具体的过程如下：

(1) 处理用户发起的请求:用户通过 HTTP 请求携带编排对象的相应属性，模版管理子系统对请求中的参数进行提取，并进行相应的分类，最终导入模版存储库中进行相应的属性填充、编辑、修改等操作。

(2) 管理对象模版:该系统将模版文件划分为基础模版和实例模版，在实例化模版时会从模版存储库中读取相应的基础模版进行编辑、修改、保存在对应的欺骗蜜网编排项目文件中。

(3) 管理模版类型:可编排资源对象和模版文件具有一对一的映射关系，该系统对编排的资源对象具有可扩展性，通过定义对应的资源基础模版，可以创建、删除、修改可编排资源类型。

3.3.3 模板解析子系统

模版管理子系统负责将前端传递的网络拓扑信息进行整合，最终生成网络拓扑的描述文件。如果将描述文件中的每个资源的实例化看作是一个任务，由于资源之间的相互依赖性，任务之间也具有关联性，通过有向无环图 DAG 来描述这种任务之间的依赖关系^[10]。模版解析子系统的作用就是在拓扑描述文件基础上构建有向无环图 DAG，具体转换流程如下：

(1) 确定网元对象，从拓扑描述文件中提取具体的网元资源对象，每一个对象映射一个资源，整个网络拓扑的编排过程 V 分解为多个网元对象的集合，网元对象集合记为 S ，则：

$$V = \{ v_i | v_i \in S, i=1, 2, \dots, n, n \geq 2 \} \quad (1)$$

其中， v_i 表示 V 中的一个具体对象；

(2) 确定网元对象之间的关联关系，在拓扑描述文件中，不同网元资源之间通过资源属性的引用来标记一种网元资源实例化的先后关系；具体化元素对象之间的依赖关系，根据资源属性之间的引用将关联关系用集合 $E(e_1, e_2, \dots, e_n)$ 来表示：

$$E = \{ e_j | e_j \in S, j=1, 2, \dots, m, m \geq 1 \} \quad (2)$$

其中， e_j 表示 E 中的一个具体的网元依赖关系；

(3) 构建有向无环图，使用确定的网元对象，确定的网元对象关系等要素建立 DAG 图，网元对象集合 S 可以通过 V, E 来表示：

$$S = \{ V, E \} \quad (3)$$

模版管理子系统通过将具体的资源映射成图的顶点，资源对象之间的关系映射为边；整个抽象过程包括业务运行资源依赖关系的抽象、网络资源依赖关系抽象以及主机资源依赖关系抽象等，如图 6 所示：

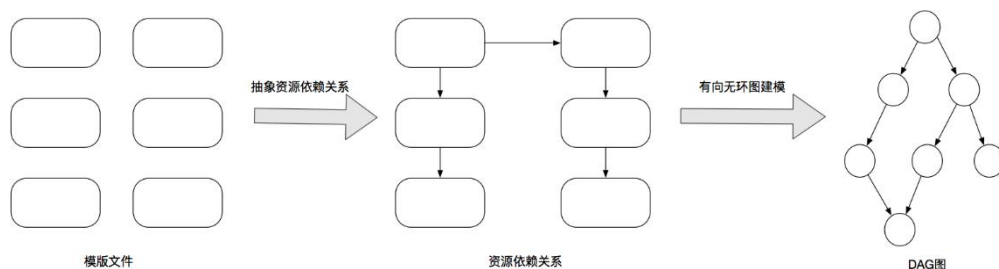


图 6 抽象资源依赖关系图

Fig.6 abstract resource dependency

3.3.4 资源调度子系统

基于云计算的环境下资源并行优势，可以根据得到的 DAG 图来进行任务的调度，将相关联、相依赖的任务划分为多个并行的任务队列。通过工作流任务调度算法来得到在最短时间

230 内，云计算环境下的网元实例化任务最优的策略。目前常见的调度算法按照任务之间是否有依赖关系可以分为两种:独立任务调度和依赖关系任务调度。其中，独立任务调度不需要考虑任务之间的依赖关系，这里的网元实例化任务调度属于具有依赖关系的任务调度。

3.3.5 云接入子系统

云接入子系统主要负责管理身份验证、服务访问规则和服务令牌功能的组件。通过该模块完成接入系统的云基础设施的用户管理、认证与授权^[1]。该子系统是不同云服务提供商接入欺骗蜜网编排系统的入口，系统抽象出对云服务提供商的 API 调用，对于不同云服务提供商只需要提供对应的插件就可以接入系统，插拔式的架构提高系统整体的可扩展性，避免系统转移到不同云环境下的接口锁定问题。统一的接入系统可以提高系统的灵活性，如图 7 所示：

235

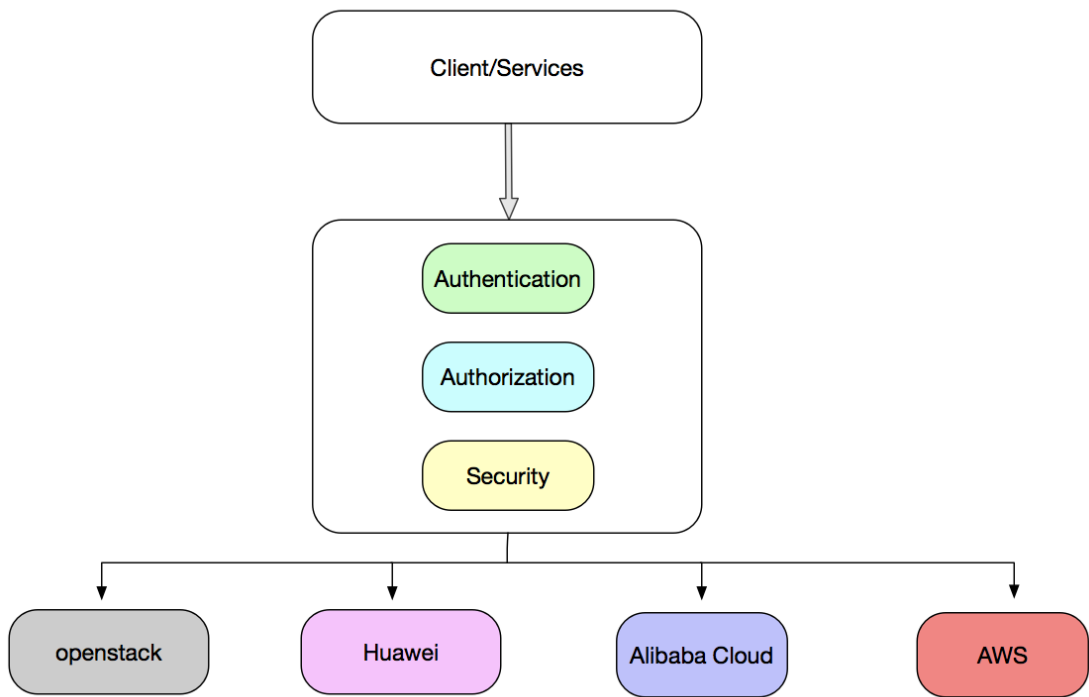


图 7 统一的云接入系统

Fig.7 unified cloud access system

云服务提供商在接入欺骗蜜网编排系统之后，对于不同的云服务商，需要指定相应的插件 plugin，云接入系统的组成如图 8 所示：

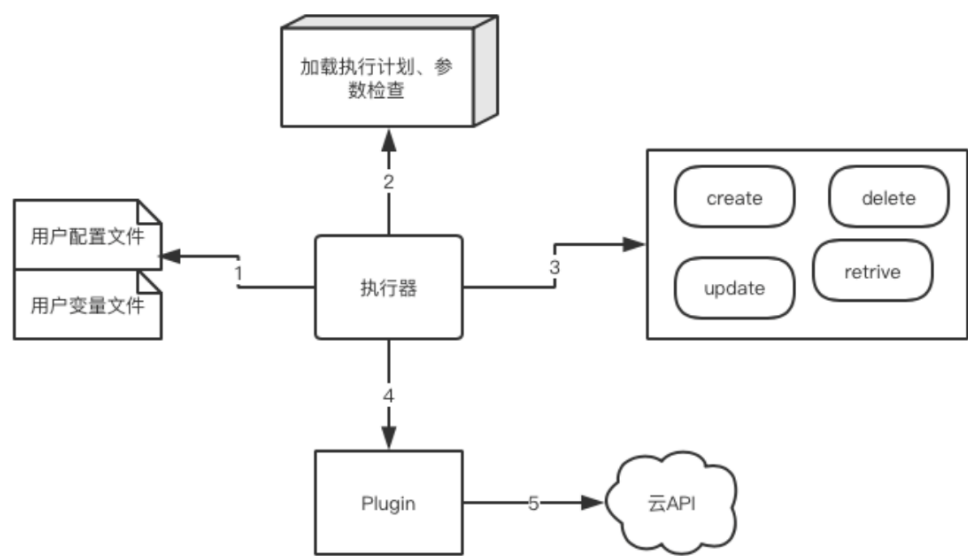


图 8 云接入系统结构

Fig.8 cloud access system structure

3.3.6 系统工作流程

欺骗蜜网环境编排系统的工作流程如图 9 所示：

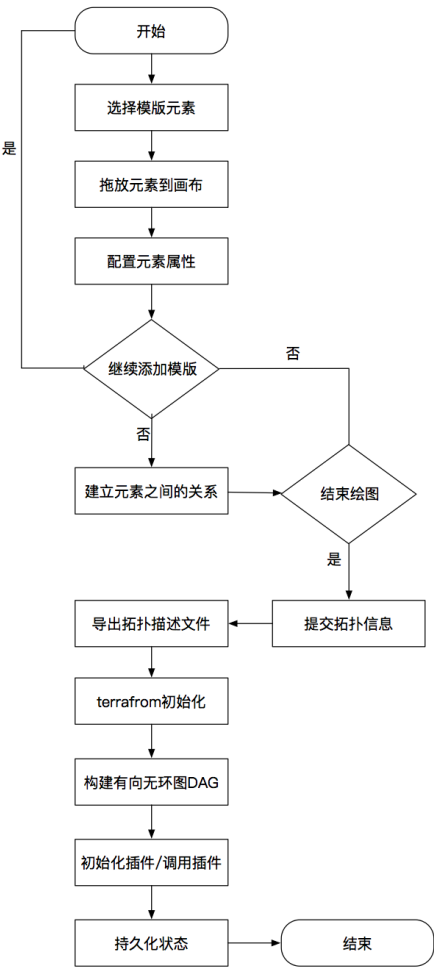


图 9 欺骗蜜网环境编排系统工作流程

Fig.9 deception honeynet environment orchestration system workflow

主要的工作流程可以总结如下：

(1) 网络拓扑编辑:用户在可视化界面上，选择相应的网元对象拖拽到画布上，并建立网元之间的连接关系。网元对象的属性值可以是默认，也可以通过用户自定义配置。设计完网络靶场拓扑后，提交拓扑信息到后端服务器。

(2) 构建拓扑描述文件:服务器根据用户提交的拓扑信息，提取相应的参数根据模版管理器生成拓扑描述文件。

(3) DAG 建模:以拓扑描述文件为基础，以网元实例化作为一个任务，对网元对象以及网元对象之间的依赖关系建立 DAG 模型。

(4) 生成执行计划:将上一步得到的 DAG 模型注入调度器中，通过调度算法得到多个并行的调度队列。

(5) 插件:可插拔式设计使整个系统具有高可扩展性，通过插件调用不同云基础设施的 API，完成网元的实例化。

4 可行性验证

4.1 实验环境

本文提出的基于云环境的欺骗蜜网环境编排系统在设计上将云基础设施与编排系统分离，因此，理论上对于任何云产商只要开发对应的插件接入系统即可。如：OpenStack、CloudStack 以及 Azure、Alibaba Cloud、AWS 等公有云。本次可行性验证试验搭建 OpenStack 开源云计算平台作为测试环境，实验室一共部署 4 台设备，具体的设备参数如表 1 所示：

表 1 服务器硬件参数

Tab. 1 Server hardware parameters

编号	CPU	内存	硬盘	节点类型
01	Intel(R) Xeon(R) CPU E5-2430 0 @ 2.20GHz	64G	500G SSD + 1T HDD	控制节点
02	Intel(R) Xeon(R) CPU E5-2430 0 @ 2.20GHz	64G	500G SSD + 1T HDD	计算节点
03	Intel(R) Xeon(R) CPU E5-2430 0 @ 2.20GHz	64G	500G SSD + 1T HDD	计算节点
04	Intel(R) Xeon(R) CPU E5-2430 0 @ 2.20GHz	64G	500G SSD+1T HDD	计算节点

4.2 功能测试

为了测试在各个系统的功能需求中的系统功能点是否都已完成，并且正确执行。需要对系统设计的主要模块进行单元测试，保证功能模块都能正确处理输入以及返回预期的输出。最终将各模块进行组合测试，保证模块之间交互的正确。

首先在 Web 界面手动拖拽构建并配置网络拓扑图，如图 10 所示：

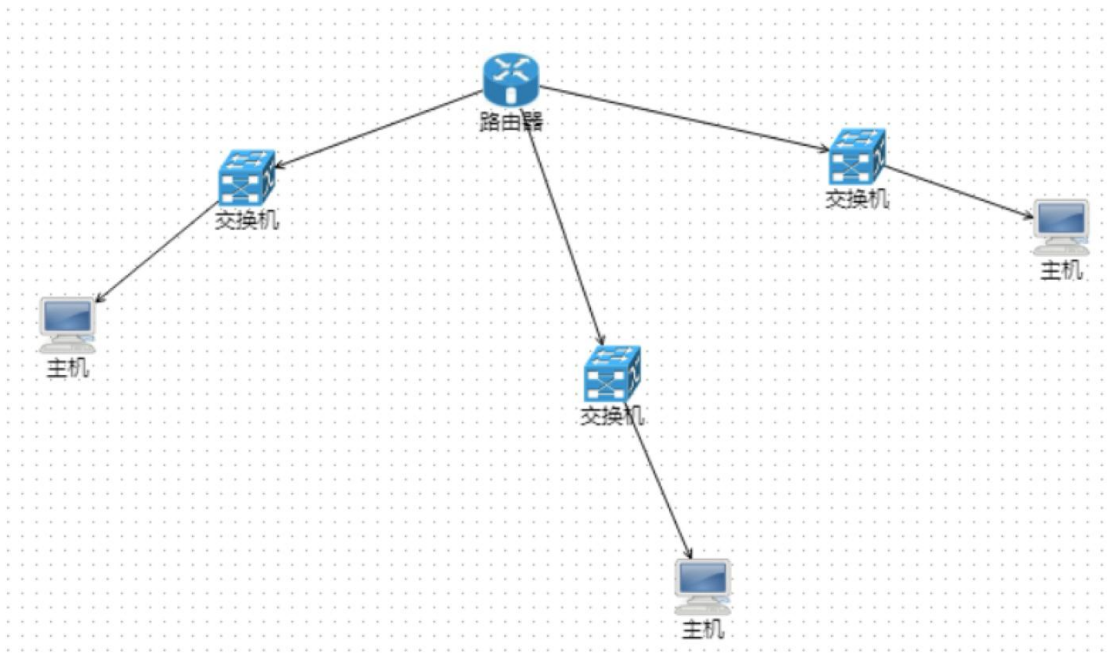


图 10 编辑网络拓扑环境

280

Fig.10 edit network topology environment

其次，判断模版管理器的功能是否正常，具体的判断方式是服务端是否能够生成拓扑描述文件以及相关的配置参数是否正确。测试环境下生成的部分描述文件如下所示：

```
{
  "network": {
    "properties": {
      "cidr": "10.1.1.0/24",
      "name": "test_network_1"
      "admin_state_up": "true"
      "type": "Vxlan"
    }
  }
}
```

图 11 network 资源模板文件

285

Fig.11 network resource template file

如图 11 所示：该模板文件表示的是 network 资源描述信息，包含 network 的网络类型、网络名称、子网 CIDR、状态等。

```
{
  "instance": {
    "properties": {
      "networkId": "test_network_1",
      "imageId": "ubuntu0S",
      "instances": 1,
      "flavor": "flavor_1",
      "nics": {
        "subnetId": "subnet_test"
      }
    }
  }
}
```

图 12 instance 资源模板文件

290

Fig.12 instance resource template file

如图 12 所示：该模板文件表示的是 instance 即虚拟机的资源描述信息，包含 instance 的

所属网络、子网信息、flavor 信息等。

另外，在模板管理器可以从 Web 端获取参数并解析成拓扑描述文件的情况下，判断网元对象在云基础设施中的实例化过程，由于本次测试环境采用的是 OpenStack 平台，可以通过 OpenStack 组件中的 Horizon Web 页面来查看下资源编排是否成功^[12]。在 Horizon 中查看编排后的网络拓扑如图 13 所示：

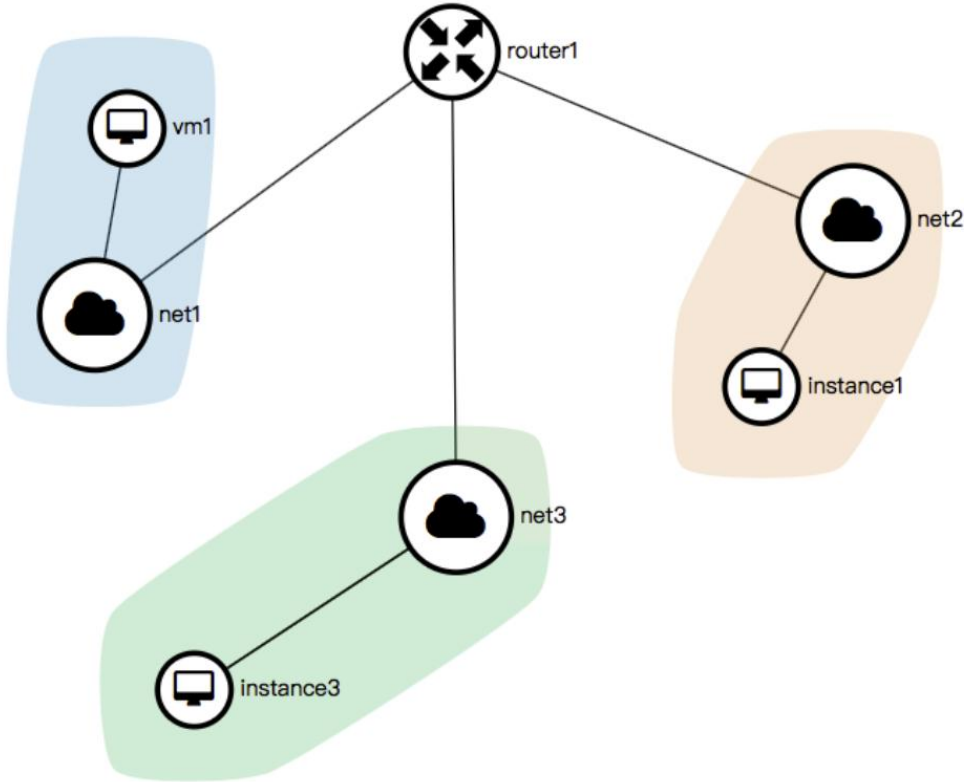


图 13 Horizon 网络拓扑描述
Fig.13 horizon network topology description

5 结论

本文提出了一种云计算环境下的欺骗蜜网环境编排系统，从可视化网络拓扑编辑与配置，到构建网络拓扑描述文件，利用 DAG 建模描述网元依赖关系，再以 DAG 图为基础生成多个并行的执行队列，根据云基础设施提供商的不同开发对应的插件实现对云平台 API 的调用完成网元对象的实例化；最后通过部署 OpenStack 云计算平台作为测试环境对该系统进行可行性验证，整个系统可以完整且正确的运行，证明了本文所提出的云计算环境下的欺骗蜜网环境编排系统是可行且有效的。

未来的工作:该系统可以实现复杂业务场景在云环境下的自动化编排，并且支持不同的云基础设施接入，但是在复杂环境下的编排调度算法、监控告警、网元对象的迁移等方面仍有不足，值得进一步研究。

[参考文献] (References)

[1] 余亚莉. 云计算研究综述[J]. 中国无线电, 2012(04):83-85.
[2] 刘丹琦, 于炯, 英昌甜. 云计算环境下多有向无环图工作流的节能调度算法[J]. 计算机应用, 2013, 33(09):2410-2415.
[3] 李俊雅, 牛思先, 程星. 一种基于 K 均值聚簇的虚拟机分类与部署方法[J]. 计算机应用与软件, 2019(8).
[4] 李建锋, 彭舰. 云计算环境下基于改进遗传算法的任务调度算法[J]. 计算机应用, 2011, 31(1):184-186.

- [5] 贺文娟. 蜜罐技术分析与蜜网设计[D].安徽大学, 2011.
- [6] 胡毅勋. 基于 Openflow 的主动防御关键技术研究[D].北京邮电大学, 2017.
- [7] 李祥池. 基于 SDN 的网络攻防平台的研究与实现[D].山东大学, 2015.
- 320 [8] Antonescu A F , Robinson P , Braun T . Dynamic Topology Orchestration for Distributed Cloud-Based Applications[C]// IEEE Second Symposium on Network Cloud Computing and Applications. IEEE, 2012.
- [9] 郭景元. 网络编排器中 NFV 编排器子系统的设计与实现[D].北京邮电大学.2018.
- [10] 唐一韬, 黄晶, 肖球. 一种基于 DAG 的 MapReduce 任务调度算法[J]. 计算机科学, 2014, 41(S1):42-46.
- [11] Gharbaoui M , Cerutti I , Martini B , et al. An orchestrator of network and cloud resources for dynamic provisioning of mobile virtual network functions.[C]// Netsoft Conference & Workshops. IEEE, 2016.
- 325 [12] Moges F F, Abebe S L. Energy-aware VM placement algorithms for the OpenStack Neat consolidation framework[J]. Journal of Cloud Computing, 2019, 8(1).