学　　号：23620199803
论文密级：公开
中图分类号：TP393.1
学科分类号：520.2020
学校代码：91037

战略支援部队信息工程大学

专业学位论文

主动防御增强的蜜网系统设计与实现

论文作者：路祥雨
指导教师：伊鹏 研究员
申请学位：工程硕士
专业领域：计算机技术
研究方向：网络安全
论文提交日期：2022 年 4 月 9 日
论文答辩日期：2022 年 6 月 9 日

战略支援部队信息工程大学

2022 年 4 月

A Dissertation Submitted to

PLA Strategic Support Force Information Engineering University

for the Degree of Master of Engineering

Design and Implementation of Active Defense Enhanced Honeynet System

Candidate: Lu Xiangyu

Supervisor: Prof. Yi Peng

Apr. 2022

摘 要

网络技术的演进给人们的生活方式带来了一场革命，物联网、智能电网等技术的应用使得网络设备的数量呈指数级增长，也带来了比以往更多的漏洞与后门。同时，攻击者的攻击手段越发智能化，现有的基于特征匹配与边界控制的防御技术在攻防对抗中逐渐处于被动。蜜网即是一种能够扭转这种不利局面的主动防御技术，也因此得到了广泛运用。然而，现有蜜网存在安全性不足、欺骗能力差等缺陷，导致其易被识别而失陷。提升蜜网的安全性与欺骗能力，成为影响蜜网技术发展与应用的关键问题。

本文以提升蜜网安全性与欺骗能力为目标，以主动防御与蜜网的融合为主题开展研究，重点研究了拟态思想与蜜网基础设备蜜罐的结合以及网络欺骗思想对蜜网结构与组成的改进，并基于此实现了主动防御增强的蜜网系统，本文的主要研究成果如下：

1．针对蜜网欺骗节点蜜罐存在的安全性与欺骗能力不足等缺陷，提出了一种基于动态异构冗余（Dynamic Heterogeneous Redundancy，DHR）架构的拟态构造蜜罐设计方案。首先基于蜜罐的特殊用途，将虚拟化蜜罐的执行体划分为虚拟化层与业务层，并对虚拟化层与业务层分别进行 DHR 改造，根据两个层级的功能需求设计了双重裁决机制，为拟态构造蜜罐实现了对运行数据与最终应答的分别裁决，基于此构建了拟态构造蜜罐的整体结构与功能模块。其次，针对拟态构造蜜罐中进行裁决与调度时应考虑的特有参数，对执行体信任度、甜度、历史威胁数据价值进行重新描述，并利用这些参数设计了适用于拟态构造蜜罐的调度与裁决算法。最后，通过实验验证了所提拟态构造蜜罐的可用性及其良好的安全性与欺骗能力。

2．针对因蜜网结构存在动态性不足与配置盲目缺陷导致的易被识别、欺骗环境构建能力不足等问题，提出了一种面向蜜网的网络欺骗机制，该机制首先设计了双频率 IP 地址跳变机制实现了对不同节点设置不同的 IP 地址跳变速率，并为真实业务设置针对性的蜜罐配置方案，增强了蜜网络系统的动态性和欺骗能力。然后，基于博弈论对网络攻防进行建模，证明了网络欺骗机制对蜜网的增益效果，并基于软件定义网络（Software Defined Network，SDN）技术实现了融合网络欺骗机制的蜜网系统设计。最后，通过实验验证了该网络欺骗机制能够提升蜜网的安全性与欺骗能力。

3．基于前述两种主动防御设计，利用虚拟化技术，实现了主动防御增强的蜜网系统，对蜜网的总体结构与模块功能划分进行了设计，为用户提供了态势展示、节点管理与蜜网管理等功能界面。最后通过在实际网络环境中的测试，验证了本文设计实现的蜜网系统有良好的应用价值。

关键词：蜜网，主动防御技术，拟态防御，网络欺骗

Abstract

The evolution of network technology has brought a revolution to people's lifestyle. The emergency of technology such as the Internet of Things and smart grids has led to an exponential increase in the number of network devices, which has also brought more loopholes and backdoors than ever before. At the same time, the attacker's attack methods are gradually becoming intelligent, and the existing defense technologies based on feature matching and boundary control are gradually passive in the attack and defense confrontation. Honeynet is an active defense technology to reverse this unfavorable situation, for which it has been widely used. However, the existing honeynets have shortcomings such as insufficient security and poor deception ability. Improving the security and deception capabilities of honeynets has become an important issue restricting the development of honeynet technology.

This paper aims to improve the security and deception capabilities of honeynets, and conduct research on the theme of the integration of active defense and honeynets. It focuses on the combination of mimic idea with honeypot, and the improvement of honeynet structure and configuration by network deception idea. The main work of this paper are as follows:

1. To make up the defects of honeynet deception node honeypot, such as insufficient security and deception ability, a scheme of mimic honeypot based on Dynamic Heterogeneous Redundancy (DHR) architecture is proposed. Firstly, based on the special purpose of the honeypot, the executive body of the virtualized honeypot is divided into the virtualization layer and the business layer. And the DHR transformation is carried out on the virtualization layer and the business layer respectively. Based on the functional requirements of the two levels, a double adjudication mechanism is designed. The decision of the operation data and the final response is realized for the mimic honeypot, and the overall structure and function modules of the mimic honeypot are constructed based on this. Secondly, according to the unique parameters that should be considered in the adjudication and scheduling in the mimic honeypot, the trust, sweetness, and historical threat data value of the executive body are redescribed, and the scheduling and adjudication algorithm suitable for the mimic honeypot is designed. Finally, the feasibility of the proposed mimetic honeypot and its good security and deception ability are verified by experiments.

2. The honeynet structure has the defects of insufficient dynamic and blind configuration. As a result, it is easy to be identified and the ability to deceive the environment is insufficient. In order to solve this problem , a network deception mechanism oriented to honeynet is proposed. The mechanism firstly designs dual-frequency IP address hopping. The mechanism realizes setting

different IP address hopping rates for different nodes, and sets up a targeted honeypot configuration scheme for real business, which enhances the dynamics and deception capabilities of the honeynet system. Then, the network attack and defense are modeled based on game theory, and the gain effect of the network deception mechanism on the honeynet is proved. Subsequently, the honeynet system design integrating the network deception mechanism is realized based on the Software Defined Network (SDN) technology. Finally, it is verified through experiments that the network deception mechanism can improve the security and deception ability of honeynets.

3. Based on the above active defense designs, a honeynet system enhanced by active defense is realized by using virtualization technology and SDN technology. The overall structure and module function division of the honeynet are designed to provide users with situation display, node management and virtual network management and other functional interfaces. Finally, through the application in the actual network environment, the honeynet system designed and implemented in this paper has good application value.

Key words: Honeynet, Active Defense Technology, Mimic Defense, Network Deception

目 录

图 录

表 录

第一章 绪论

1.1 背景及意义

1.1.1 研究背景

随着信息技术的普及与深度应用，越来越多的软硬件设备进入了人们的生活，同时也给网络中各信息系统带来了更多的漏洞。据美国国家漏洞数据库 2021 年的数据显示，当年全美漏洞数量高达 18378 个，而在 2001 年，这一数字还低于 2000。计算机及网络系统所面临的威胁形势也愈发严峻。网络攻击的形式由零星松散的手动攻击逐渐发展为自动化程度极高的有组织智能化攻击，恶意挖矿、勒索病毒等新型攻击手段更是层出不穷[1]，给人类的生产生活都带来了极大困扰。2021 年 5 月，挪威公司 Volue 遭遇勒索软件攻击，导致境内数百座城市的自来水厂设备工控程序关闭，影响了其全国约 85%的居民；同月，燃油输送管道巨头科洛尼尔公司因网络攻击而暂停其业务，超过三分之一的美国国土面临油料紧缺；2021 年年末，Apache Log4j2 漏洞爆发，因其在全球软件供应链中的重要地位，该漏洞影响了近 60%的企业机构，造成了以十亿计的经济损失，甚至影响到了各国国家安全战略。而根据国家信息安全漏洞共享平台发布的互联网监测数据显示，如图 1.1 所示，仅 2020 年，我国国家信息安全漏洞共享平台就收集了各类网络系统漏洞 20721 个[2]，相比 2019 年增加了 28.0%；全国各级网络举报部门共受理举报 16319.2 万件，相比 2019 年增加了 17.4%。由这些数据不难发现，网络安全事件的量级以及影响都在不断升级，在信息化时代，网络安全事件造成的后果难以估量。安全问题不仅制约着网络的持续健康发展，也决定着其能否得到更广泛的应用。

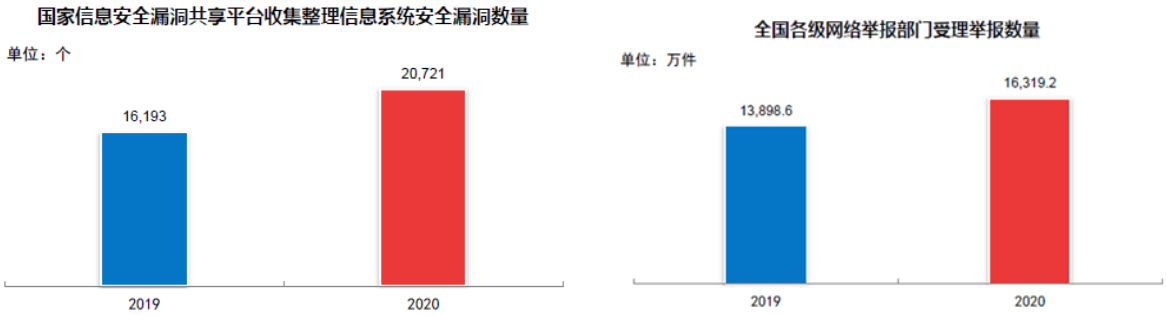

图 1.1 安全漏洞和受理安全举报数量

为应对复杂多变的网络攻击，人们使用基于特征识别的边界检测技术（如杀毒软件，入侵检测技术等）来防止攻击者突破边界防御。然而在有组织攻击者的攻击下，特别是遭遇有明确目的的高级持续威胁（Advanced Persistent Threat，APT）[3]时，上述安全防御体系被绕开或攻破的可能性极大。攻击者利用社会工程学、软硬件漏洞或其他手段总是

能够避开边界防御，获取系统权限，并渗透网络系统。因此在对网络系统进行防御时，必须考虑到边界防御是可以被攻破的，且攻击者在此之后可以在网络中建立据点，并通过据点侦察网络系统的防御策略。而传统的被动式防御技术受到其后验性与特异性的限制，预判性较差，且缺乏对威胁数据的分析机制，无法有效应对复杂的网络安全挑战。因此，研究人员开始将目光转向基于主动的安全防御技术。蜜网[4]即是一种重要的主动防御技术，其配置具有漏洞的蜜罐作为陷阱，将不具备真实价值的资源设置为诱饵以吸引攻击，通过接受攻击者的探测、攻击，甚至被攻击者攻陷，以获得相应的威胁数据，对数据进行分析与研判，从而得到攻击者来源、攻击手段以及攻击目的的刻画，并针对性地设置防御策略，增强网络系统的安全防范能力。如何增强蜜网的诱骗能力，诱使攻击者在蜜网内停留更长时间，与蜜网进行充分交互，并同时确保蜜网自身安全性，是蜜网安全防御体系中亟待解决的难题，对更好发挥蜜网主动防御效果，提升蜜网捕获恶意数据的能力，有着重要意义。

1.1.2 研究意义

蜜网技术仿照真实系统部署无实际业务用途的网络资源集合，诱使攻击者发起非法访问与交互，从而在预先设置的场景中对攻击者的恶意行为进行捕获、统计和分析，获得攻击者的攻击目的、手段、路线以及所使用的工具等信息，并根据获取的知识调整安全策略，保护真实系统。蜜网技术增强了网络系统的威胁检测能力，同时能够有效防范未知类型的恶意行为。

蜜网技术弥补了被动式防御技术依赖先验知识、缺乏预判能力的缺陷，是网络安全防御体系的有力补充。然而任何安全技术都不可能抵御所有的网络攻击，攻击者的攻击手段日益智能化，反蜜罐[5]技术也随之出现。而传统蜜网受到动态性不足，配置盲目单一的限制，存在无法有效欺骗攻击者、网络内部署的蜜罐易被识破等问题。蜜罐被识别后可能会被绕过，导致蜜网无法捕获有效的行为信息。攻击者甚至可以将蜜罐作为其进入内网的通道，将其设置在攻击路径中，对第三方进行攻击[6]。

而在虚拟蜜罐因其管理的便利性与较低的成本被大量使用的今天，上述情况可能导致更加严重的后果。在使用虚拟蜜罐设备的蜜网中，攻击者识别蜜罐设备后，蜜网就面临其利用蜜罐内漏洞发起虚拟机逃逸的风险。虚拟机逃逸[7]是指恶意程序利用漏洞绕过虚拟机的防护限制，到达虚拟机所在的宿主系统中，并能在宿主系统中执行指令。虚拟机逃逸通常需要完成对虚拟机提权进而触发虚拟化平台漏洞，而蜜罐设备的本质就是一种预设漏洞吸引攻击者攻击的网络服务，因此通过虚拟蜜罐完成虚拟机逃逸有着先天的便利。对攻击者而言，在完成虚拟机逃逸后，网络系统的内部几乎是不设防的。

本文设计的主动防御增强的蜜网，将主动防御中的先进技术与蜜网结合，“由点及面”地增强了蜜网防御体系的安全性与欺骗能力。首先着眼于增强蜜网欺骗节点蜜罐的安全性与欺骗能力，将拟态防御思想应用到蜜罐中。针对虚拟化平台存在的未知逃逸隐患，

使用拟态动态异构冗余（Dynamic Heterogeneous Redundancy，DHR）架构进行规避，同时为增强其欺骗能力，设计双重裁决机制将多个响应中诱骗性最高的发送至攻击者。其次面向蜜网设计了一种网络欺骗机制，对蜜网的网络拓扑和软件配置等重要信息进行了混淆，提升了蜜网整体的安全性与欺骗能力，降低了其被识别的概率。最后，综合以上两项研究设计实现了主动防御增强的蜜网，提供了一种在安全性与欺骗能力上都有显著提升的蜜网系统。

1.2 研究现状

1.2.1 蜜罐技术

蜜罐[8]是一种通过吸引攻击完成防御任务的安全工具，通过伪装为有价值的网络资源，对恶意行为进行诱捕。蜜罐技术不依赖任何先验知识，能够捕获未知攻击，弥补了传统安全防御体系的缺陷，因而被广泛应用于各类网络信息系统，如工业控制系统[9]、物联网[10]、智能电网[11]等。蜜罐工具的主要工作就是伪装业务系统、捕获与分析攻击行为，对应其主要工作，蜜罐的核心技术机制包括诱骗场景构建、攻击行为捕获与威胁数据分析。从攻击者的角度看，诱骗场景构建能力也可被称为“甜度”，即蜜罐模拟的业务对攻击者的吸引程度，高甜度的蜜罐能吸引攻击者与其进行充分的交互，从而获取更多威胁数据。

过往的研究侧重于蜜罐使用场景的扩展与数据分析能力的提升，忽视了蜜罐设备的安全性。事实上，在反蜜罐技术逐渐成熟的今天，传统蜜罐由于存在位置固定、配置单一、缺乏感知能力等缺陷，极易暴露自身存在，导致失效甚至被利用。因此如何规避攻击者识别，提升蜜罐安全性，成为近年来蜜罐研究的侧重方向。

Kuwatly 等人[12]设计了一种使用入侵检测技术对抗蜜罐指纹识别的动态蜜罐系统，使用 Nmap 与 Snort 等入侵检测工具的特征库对蜜罐响应进行检测，以规避攻击者对蜜罐的识别。该系统用 Honeyd 搭建虚拟网络，并将恶意流量转发至高交互蜜罐，从而与攻击者进行更多交互，蜜罐配置模块基于系统状态获取虚拟网络配置并进行组网，网络管理人员也可以通过该组件对网络进行手动配置，增强了网络的灵活性。Saeedi 等人[13]提出基于网络环境全局信息对蜜罐进行动态调整的方法，打破了蜜罐位置的静态性，该方法根据从网络系统内的安全设备与软件中获取的网络总体态势调整安全策略，动态配置蜜罐适应网络状态，其中蜜罐的配置调整可以自动化进行，使安全策略始终能与当前的网络状态匹配。

Gerard 等人[14]则在理论层面讨论了自适应蜜罐的策略选择问题，将高交互蜜罐与攻击者的攻防过程进行博弈建模，并讨论高交互蜜罐在交互过程中的策略问题。通过解决博弈论问题，寻找蜜罐自身安全性与更多交互间的平衡，在捕获更多恶意数据的同时转移攻击者精力，保护真实主机不受侵害。

Sochor 等人[15]对比了当前主流高交互蜜罐的开源代码和管理方法，并在搭建的蜜罐系统中应用所选取的蜜罐与管理方法。开发人员选择了一种操作系统蜜罐与一种服务蜜罐并入系统，其中，操作系统蜜罐 Linux Debian 存储大量的蜜标数据，使用 Honeyd 模拟各类路由，以吸引攻击者侦测扫描；服务蜜罐 Web server 内建带有漏洞的 Web 应用，并打开其 80 端口，响应 HTTP 请求。攻击者的恶意行为数据（如暴力破解、运行脚本等）将被存储在数据库中，并在管理界面可视化呈现。

石乐义等人受生物界中生物保护色和警戒色现象的启发，设计了拟态蜜罐设备[16]，借用保护色和警戒色的现象，提取真实业务环境的特征指纹信息作为保护色赋给蜜罐，使攻击者无法识别，而警戒色是由真实业务模仿蜜罐特征，使攻击者误认为其是蜜罐而放弃攻击，但本文提出的拟态构造蜜罐与此概念有所区别。

1.2.2 蜜网技术

仅使用一种蜜罐进行网络安全防御不仅无法获取全面的攻击数据，还存在被攻击者识别的安全问题。因此，Spitnzer 等人提出了第一代蜜网(honeynet)架构[17]，并于 1999 年成立了研究组织 The Honeypot Project[18]，致力于发展与推广蜜网技术，在之后推出了第二代[19]与第三代蜜网[20]。蜜网是一种网络体系结构，如图 1.2 所示，其包含多种蜜罐设备，并集成了边界过滤与控制设备、系统行为记录与数据分析等模块。在蜜网欺骗环境中，攻击者从进入网络开始的所有恶意行为都将被记录，因而安全研究人员可以更为全面地获取攻击者的信息。

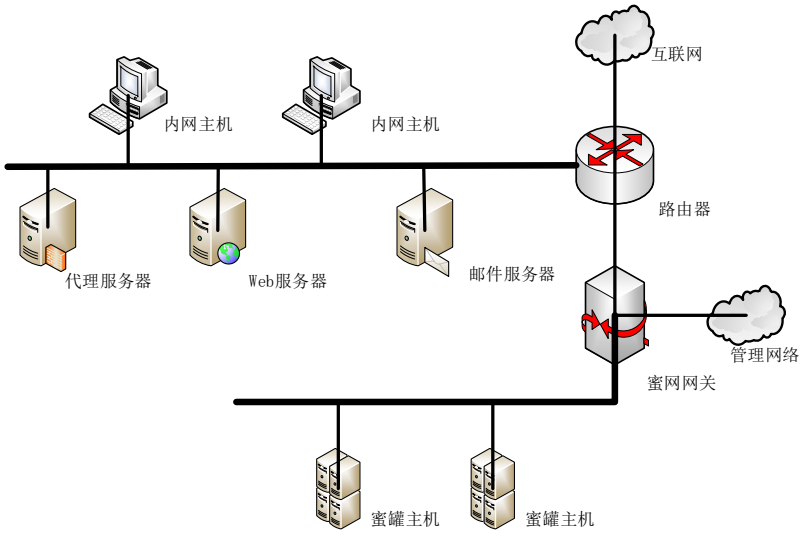

图 1.2 蜜网结构图

然而，传统蜜网结构固定[21]，欺骗环境构建能力较差，攻击者可以轻易获取系统的路由拓扑、软件配置等关键信息。这可能导致攻击者对蜜网进行识别并绕过，甚至将其作为跳板进行内网渗透。同时若攻击者能够轻易获取网络情报信息，则其在蜜网内停留时间会相应减少，导致捕获的恶意行为不全面，对攻击者的刻画不准确，无法有效提升系

统防御能力。为弥补这一缺陷，研究人员提出了诸多改进方案。

Yang[22]等人提出一种动态分配 IP 的虚拟蜜网，系统采用虚拟主机构建蜜网，蜜网中 IP 地址不绑定主机。系统定期扫描可用的全部地址空间，将地址空间中空闲的 IP 分配给虚拟蜜罐，然而这种分配方式中真实主机的 IP 地址是静态的，容易暴露。

Fan[23]等人提出了用于蜜网生成部署的基础平台，该平台包含一种基础蜜网结构，这个结构具有对所处网络环境进行自适应的功能，能够基于所在环境特征生成和构建蜜网，并利用多种蜜罐完成动态部署，然而其生成过程复杂，且生成的蜜网无法随环境动态迁移。

软件定义网络（Software Defined Network，SDN）[24]架构具有良好的流量控制能力，能以更低的成本实现蜜网的改进。Kyung[25]等人基于 SDN 技术开发了包含代理模块的蜜网，代理模块随攻击者与蜜网的交互进程变换其代理模式，透明模式下不进行转发；广播模式下代理模块接收恶意请求，并将其发送到多个蜜罐中，最终通过特征匹配的方式对可能暴露蜜罐存在的应答进行排除；相持模式下代理模块维护一对一通信。Stefan[26]等人为对抗从内部网络发起的扫描探测，提出了一种基于 SDN 的蜜网系统 RDS（Reconnaissance Deception System），该系统在应用层部署了欺骗模块，对内网发出的流量进行检测，获取到有嫌疑的主机地址后，欺骗模块会构设虚假网络信息并发送到该地址。然而使用 SDN 技术的研究大多聚焦于 SDN 北向接口的可编程性，并没有充分利用 SDN 在网络管理上的优势。

1.2.3 网络空间拟态防御

目前，网络空间处在一种易攻难守的困境下。在设计或开发等环节中出现的不可知漏洞与精心构设的后门，导致网络信息系统已经失去了对软硬件设备可信度和系统安全风险的控制[27]，这意味着没有完全消除网络系统或设备中的安全漏洞的可能性。同时，只要攻击者探测到任意一个漏洞并进行利用，信息系统就面临着无法预料的安全风险。而大多数网络信息系统选用的安全技术和设备，如防火墙、漏洞扫描软件、入侵检测系统（intrusion detection system，IDS）等，都基于控制、检测与阻断，在攻防过程中是“后知后觉”的[28]，这些被动防御技术几乎无法应对未知漏洞和后门威胁，导致网络信息系统的防御存在一定的隐患。

随着网络信息系统陷入无法抵御未知漏洞后门的困局，研究者将更多的研究精力投入到主动防御中。主动防御思想是在不具备恶意行为与数据特征的先验知识时，即完成系统防御，其主要包括入侵容忍[29]、网络欺骗[30]和拟态防御[31]等。在恶意行动发起前就完成对系统的防御，不仅降低了系统被攻击的可能，也减小了恶意行为对系统的负面影响。

拟态防御是邬江兴院士为解决网络信息系统中潜在的无可避免的漏洞与后门，提出的一种模块易失效的风险可控、安全可信的方法，其优势已在拟态防御技术应用中得到检

验。

拟态防御通过对同一激励下各异构执行体的输出进行裁决和动态改变任务的执行环境，防止系统因单个漏洞而无法提供预设的服务，提升了攻击者对系统进行侦察和攻击的难度，在现有工程技术条件下，给出了一种能够抵御未知漏洞后门的可行方案。正是因为拟态架构的独特优势，其被广泛运用于各种网络设备。

马海龙等[32]基于拟态防御技术，将多个异构的同功能执行体引入路由器设备，实现路由器设备的主动防御。实验结果表明，基于拟态构造的路由器结构能够破坏攻击链，显著提升攻击者的攻击难度，并能抵御未知类型的攻击。

仝青等[33]对 Web 服务器进行了拟态化的改造，给出了拟态 DHR 架构具有安全性增益的数学证明，并对 Web 服务器拟态化改造的特殊设计进行了介绍，最终通过实验验证了拟态防御模型能够赋能 Web 服务器设备安全性。

1.2.4 网络空间欺骗技术

网络空间欺骗是从蜜罐的诱骗思想演化而来的主动防御技术，该技术通过混淆攻击者对网络资源现状的判断，诱使其做出错误的攻击选择[34]。与传统安全技术侧重于对攻击者行为进行控制和检测不同，网络欺骗聚焦于扰乱攻击者对系统的感知，使攻击者无法得到系统的准确信息，从而做出有利于系统防御的行为选择。

如图 1.3 所示，网络欺骗防御技术主要包括网络地址转换、OS 混淆、拓扑仿真以及蜜标技术等，这些技术针对网络系统的不同层次进行信息伪造与混淆，根据其作用层次不同，大致可以分为数据层、应用层、设备层与网络层。本文所使用的主要是网络层欺骗技术，因此，仅对网络层欺骗技术研究现状作详细介绍。

网络层欺骗的主要研究内容是如何使网络中的业务设备隐身以及发挥欺骗节点最大作用，其往往需要综合使用其它三种层次的欺骗技术，以达到最佳欺骗效果。

OpenFire[35]是一种面向网络探测行为进行欺骗的新型防火墙。OpenFire 与一般防火墙不同，其接收全部流量并将本应丢弃的部分转发给蜜罐设备群。同时，OpenFire 设置其所在网络的 IP 和端口全部开放，以吸引攻击者，转移其攻击目标，消耗其精力，从而保护真实系统。

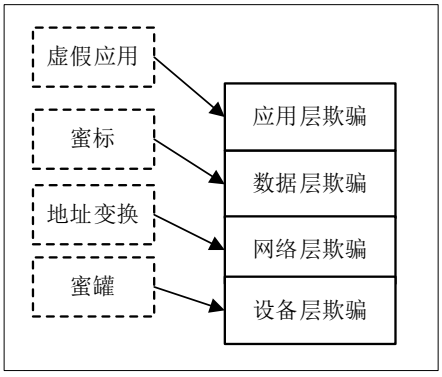

图 1.3 网络欺骗的层次

网络层中，通过将 IP 地址与网络节点解绑，并定期对其进行重新分配，可以动态调整网络的拓扑，使系统设备在探测中保持隐匿。Antonatos 等[37]设计了网络地址随机化的 IP 地址分配机制，通过代码实现为网络内所有节点定时分配 IP 地址的功能，以抵抗蠕虫攻击。MUTE[38]是一种基本思想与文献[37]相近的方案，其使用动态地址跳变与动态指纹跳变对网络结构与组成进行了全方位的动态化设计，使攻击者无法得到网络结构或节点的准确信息。Robertson 等[39]则设计了另一种通过网络拓扑与配置进行欺骗的定制信息网络（CINDAM），该网络中网络拓扑与配置不按照时间进行变换，而是向不同视角呈现不同的网络视图，即从不同的网络节点对网络进行探测，得到的网络情报不同，从而使攻击者无法得到关于网络的准确情报。Wang 等[40]提出了一种使用动态域名的机制，该机制通过将域名进行动态化设置，扩大了原有的地址空间，增加了攻击者进行扫描探测的代价。

1.3 现实需求

本文以提升蜜网安全性与诱骗能力为研究目标，分别从蜜网基础设备与蜜网架构出发，开展基于拟态架构的蜜罐技术研究和面向蜜网的网络欺骗机制研究，并将相关研究成果应用到蜜网系统的设计与开发中，从网络节点与网络结构两个方向出发，“由点及面”地提升蜜网系统的安全性，增强其诱骗能力。具体需求如下：

1. 针对网络攻击日趋智能化、复杂化的问题，设计主动防御增强的蜜网系统

随着万物互联时代的到来，软硬件网络设备的数量呈爆炸式增长，也带来了更多的未知漏洞与后门。虽然蜜网研究已取得大量成果，但网络攻击也在不断升级，呈现出持续时间长、协同性强的特点，且利用各类零日漏洞，导致蜜网存在失陷可能。目前，主动防御技术与其他设备的结合已取得较大进展，将主动防御技术与蜜网系统进行深层次融合，在赋能蜜网安全性的同时，也能够提升蜜网的欺骗能力。因此，在传统蜜网难以应对现有攻击的情况下，需要设计一种主动防御增强的蜜网系统。

2. 针对蜜网基础设备蜜罐安全性缺失、欺骗环境构建能力差的问题，提出基于拟态架构的蜜罐系统

云计算和虚拟化技术的发展，使得从专用蜜罐设备到动态灵活的虚拟蜜罐的转换成为了可能。虚拟蜜罐因其较低的成本和便利的管理得到了广泛的应用。然而虚拟机与宿主机间的屏障并非牢不可破，由虚拟机触发宿主机漏洞导致虚拟机逃逸的事件时有发生。而在蜜罐设备构建的欺骗环境中，往往预设大量漏洞以吸引攻击者，这些漏洞本身是可控的，然而其是否能够触发虚拟化平台的未知漏洞后门，成为了虚拟化蜜罐的安全隐患。同时，蜜罐设备总是对其模拟业务选择一种实现方案进行配置，导致其构建的欺骗环境特征单一，易被攻击者识别。因此，针对现有蜜罐存在的安全性与欺骗能力缺陷，需提出一种能够抵御虚拟化平台未知漏洞后门的蜜罐设备，同时该设备应具有更强的欺骗能

力。

3. 针对蜜网结构存在的动态性、欺骗性与诱导能力不足的问题，提出面向蜜网的网络欺骗机制

蜜网由于其结构存在动态性不足，配置盲目单一等缺陷，在面对攻击者长时间、多路径的渗透时，被识别的可能性极大，蜜罐识别技术的成熟更是加剧了这一局面。蜜网被识别后不仅无法发挥其本身的作用，甚至可能被攻击者用作跳板对第三方发起攻击，造成更为严重的后果。此外，简单固定的蜜网探测难度低，无法与攻击者产生更深层次的交互，也就无法获得更多攻击者的个性化攻击数据。故针对蜜网体系架构中存在的缺陷，需要提出一种面向蜜网的网络欺骗机制改善蜜网的安全性与欺骗能力。

1.4 研究目标与研究内容

为提升蜜网系统的安全性与欺骗能力，本课题面临的主要问题有：一是蜜网基础设备蜜罐高度依赖虚拟化，导致其存在安全隐患，且模拟业务的实现方式单一，欺骗环境构建能力较差，针对此问题提出了基于拟态架构的蜜罐设计方案；二是蜜网结构组成存在动态性不足与配置盲目的缺陷，导致蜜网欺骗能力差，存在失陷风险，针对此问题提出了面向蜜网的网络欺骗机制；三是基于以上的研究成果设计并实现了主动防御增强的蜜网系统。

论文的主要内容如下：

（1）从蜜网节点出发，研究分析了蜜网中重要基础设备蜜罐存在的安全性与欺骗性不足等缺陷，蜜罐仅提供模拟业务的一种实现，极易被识别，且高度依赖虚拟化技术，攻击者识别蜜罐后能够完成虚拟化逃逸，直接侵害真实业务场景。为防止蜜罐设备中潜在的未知漏洞后门导致虚拟化逃逸，同时为改善蜜罐的欺骗能力，基于网络空间拟态防御思想对蜜罐进行了改造，提出拟态构造蜜罐，首先基于蜜罐的特殊用途，将虚拟化蜜罐的执行体划分为虚拟化层与业务层，并对虚拟化层与业务层分别进行改造，并根据两个层级的功能需求设计了双重裁决机制，为拟态构造蜜罐实现了对运行数据与最终应答的分别裁决，基于此构建了拟态构造蜜罐的整体结构与功能模块。其次，针对拟态构造蜜罐中进行裁决与调度时应考虑的特有参数，对执行体信任度、甜度、历史威胁数据价值进行重新描述，并利用这些参数设计了适用于拟态构造蜜罐的调度与裁决算法。

（2）从蜜网结构出发，研究分析了蜜网网络结构中存在的动态性不足、配置盲目单一等问题，面对长时间、多角度的渗透时，蜜网易暴露自身存在，从而导致攻击者的绕过或利用。因此，面向蜜网提出了一种网络欺骗机制，通过在蜜网所处网络系统中部署双频率 IP 地址跳变，并为真实主机配置针对性蜜罐群对网络真实信息进行混淆，以增强蜜网欺骗攻击者的能力。此外，将蜜罐分布式部署在网络各区域，最大程度模拟真实网络，使攻击者难以识别蜜网存在，从而迟滞攻击者进攻，获取更多攻击数据，提高了攻击者攻击的难度并增强了蜜网获取攻击信息的能力。

（3）基于所提出的设计实现了相应的拟态构造蜜罐，并结合提出的网络欺骗机制，设计实现了主动防御增强的蜜网防护系统。该系统主要包括节点管理模块、蜜网管理模块以及态势展示模块。节点管理模块的主要功能是对网络中各类节点的状态进行监控和管理，支持对执行体进行新建、清洗等操作；蜜网管理模块能够完成蜜罐的部署，并管理与初始化网络欺骗机制；态势展示模块采用可视化的方式，将攻击数据按不同分类方式进行展示。

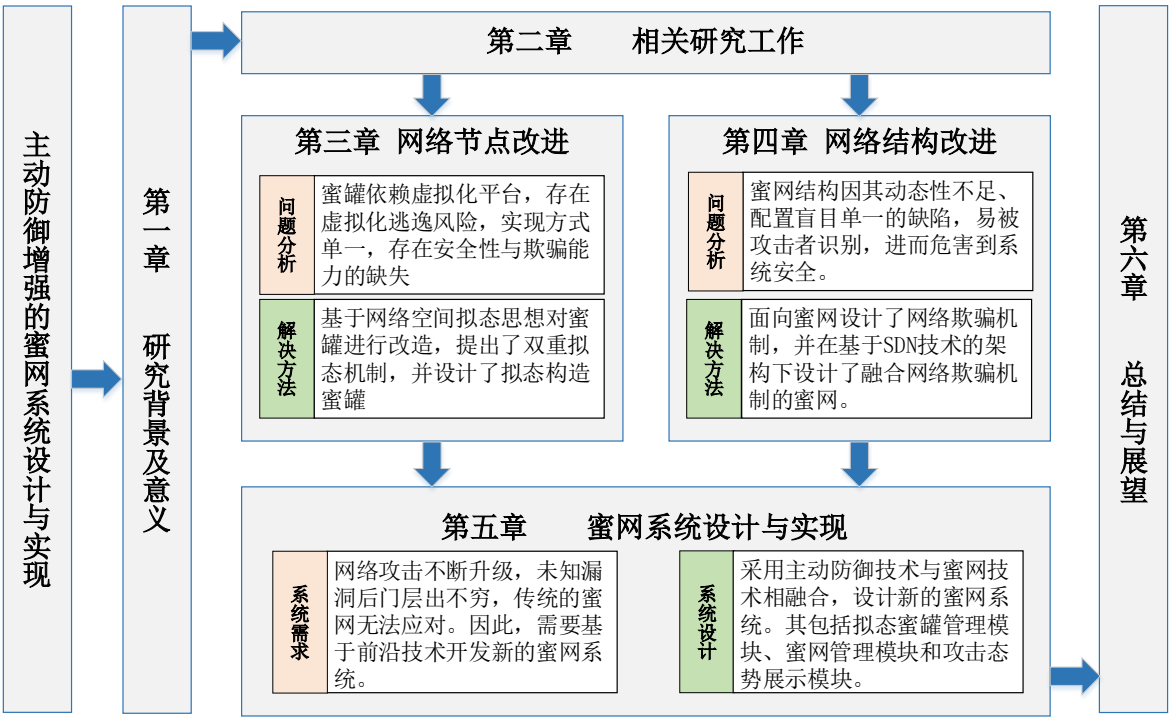

图 1.4 论文组织结构图

1.5 论文的组织结构

本文以主动防御技术与蜜网的融合为主题展开研究,论文的组织结构如图 1.4 所示，章节内容安排总结如下：

第一章是本文的绪论部分，介绍了网络安全面临的形势，引出蜜网的概念。接下来介绍并分析了国内外近年来在相关领域的研究现状，重点分析了蜜网改进的现实需求，并给出了本文的研究目标与主要工作，最后对论文的组织架构进行了规划。

第二章介绍了相关理论和技术。首先介绍了文中涉及到的拟态防御技术的原理及实现方案，并对本文中用到的博弈论与网络攻防博弈进行了概述，为下一步开展应用提供理论支撑。

第三章介绍了基于拟态思想的蜜罐研究。针对蜜罐高度依赖虚拟化技术导致的安全缺陷，与仅提供模拟业务的单种实现导致的欺骗能力不足，提出了基于拟态架构的蜜罐，针对拟态构造蜜罐的特点设计了双重裁决机制，介绍了拟态构造蜜罐的设计方案、总体

结构和各模块功能，并提出了两种适用于拟态构造蜜罐的算法。最后设计实验验证了拟态构造蜜罐的可用性与欺骗性。

第四章提出了面向蜜网的网络欺骗机制。针对蜜网结构的静态性与配置盲目的缺陷导致的欺骗能力差、安全性较低等问题，设计了面向蜜网的网络欺骗机制，通过博弈论验证了结合网络欺骗机制的蜜网的优势。最后基于 SDN 技术设计和实现了使用该机制的蜜网，给出了该机制下的网络通信方法，并通过实验验证了网络欺骗机制在不降低蜜网服务质量的同时可以有效地欺骗攻击者。

第五章介绍了主动防御增强的蜜网系统设计与实现。基于改善蜜网安全性与欺骗能力的研究成果，设计并实现了融合主动防御技术的蜜网防御系统，完成了蜜网系统总体需求分析，对系统中各功能模块进行了划分与设计，并对相关功能的实现进行了介绍，最后展示了所实现的蜜网系统前端界面，对系统进行了测试。

第六章总结与展望。总结了主动防御增强的蜜网系统取得的阶段性成果，并指出当前研究存在的不足与未来的研究愿景。

1.6 本章小结

本章首先对开展本文研究的背景进行了介绍，并就此讨论了研究的意义，简要概述了国内外对蜜罐、蜜网、网络空间拟态防御及网络欺骗的研究现状，并鉴于现有研究的不足指出提出本课题的研究，最后给出了论文的研究内容，并对各章节的组织结构进行了规划。

第二章 相关技术

本文针对蜜网存在的缺陷，提出使用主动防御技术赋能蜜网安全。为此，本章首先对主动防御相关技术原理进行详细介绍，主要包括拟态技术中的 DHR 架构[41]；其次，对本文中用到的博弈论与网络攻防博弈进行介绍，为后续将主动防御技术与蜜网相融合奠定理论基础。

2.1 主动防御相关技术

主动防御技术起源于由容错技术演进而来的入侵容忍技术。容错技术最初提出的目标是解决计算机系统中计算结果不一致的问题，但很快就被扩展至处理漏洞导致的系统异常，使系统在遭受入侵后仍能提供正常服务，故被称为入侵容忍系统（Intrusion Tolerance System，ITS）。入侵容忍系统使用容错技术与思想来实现对外界入侵的容忍，保持系统的可用性，是一种攻击者对某些部件进行了成功的攻击后，仍可以维持正常运行状态的系统[42][45]。

由于冗余成本高昂，对入侵容忍技术的研究逐渐衰落。为扭转网络攻防中的不利局面，一些国家转变思路，将防御方式由对已出现攻击的容忍转移至主动防范未出现的威胁，将工作重点转移到增强网络的灵活性和动态性，增加攻击者发动攻击的成本与风险。2011 年，美国国家科学技术委员会发布了一项名为“可信赖的网络空间：联邦网络安全研发战略”的计划。在该计划中提出的移动目标防御（Moving Target Defense，MTD）被视作一种从本质上改变网络空间攻防现状的颠覆性安全防御技术[46]。MTD 的愿景是设计能够在不可信环境中持续稳定运行的灵活可靠系统，该技术通过部署防御者控制的、时变的、贯穿多个系统层级的变化机制和策略，减少漏洞的出现时长，降低攻击者攻击可行性，并提高攻击需付出的代价[47]。移动目标防御在系统的各个层级动态改变其结构、配置等信息，使攻击者无法获得关于系统的准确信息或使其获得的信息快速失效，导致其难以发现可利用的攻击路径，从而维护系统的安全。

2.1.1 拟态防御的 DHR 架构

尽管网络攻击的种类纷繁复杂，但除 DDoS 之类无差别消耗资源的攻击类型外，其它类型的网络攻击均需要一个“着力点”，即一种特定的漏洞。围绕该漏洞，攻击者会开展攻击策划，并通过多次渗透触发该漏洞，最终达成攻击目标[48]。可以看出，网络攻击的发生和复现都高度依赖所攻击系统的特性。根据这一特点，如果能够提供一种系统特性动态变化的系统实现，就能切断攻击者的攻击路线，降低其攻击成功的可能性。

拟态防御技术从可靠性领域中的非相似余度（Dissimiliar Redundancy Structure，DRS）构造[49]中得到了灵感，提出了实现系统特性动态变化的拟态 DHR 架构。非相似余度构造

是将多个不同构造的执行体进行并联，在给定相同外界激励后各执行体分别独立运行，并从各执行体给出的结果中裁决得到最终输出，减小了系统中相同功能模块同时失效的可能性。“静态、异构、冗余”的构造使得非相似余度构造的系统具备了较强的容侵能力，在不依赖任何先验知识的前提下能够抵挡未知攻击。拟态防御技术继承了非相似余度构造中的“异构、冗余”，同时消除了“静态性”带来的攻击者通过长期侦察得到准确信息的隐患，通过对异构执行体进行基于输出负反馈的动态调度，使得基于拟态 DHR 架构构建的系统始终提供预设的服务[50]。

拟态 DHR 的架构如图 2.1 所示，下面对图中各基本概念进行介绍：

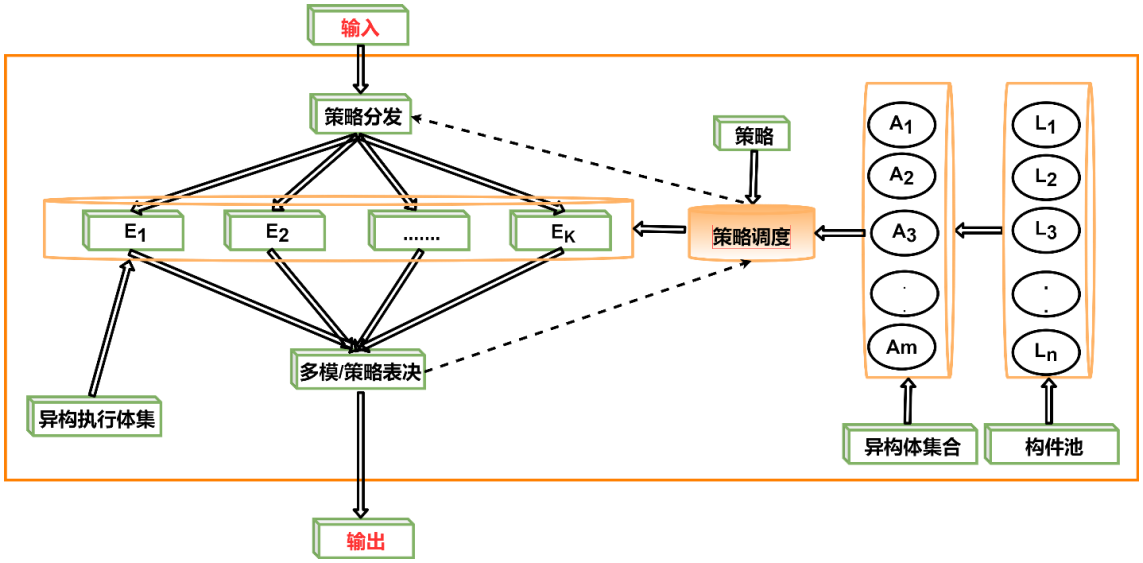

图 2.1 拟态 DHR 架构图

• 输入代理：所有发往执行体的数据都由输入代理接收，并根据冗余度要求复制并分发至各执行体。

• 异构执行体集：包含多个并联的异构执行体，执行体间相互独立，分别执行输入代理分发的任务。

• 多模、策略表决：判断多执行体执行同一任务的冗余结果是否一致，并根据设定的裁决策略选择最终结果。

• 输出代理：将最终结果输出，并可对系统的响应时间进行伪装。

• 策略调度：按照预设的规则，根据各执行体输出结果动态地对执行体进行切换。

图中的各异构执行体功能等价，在输入代理模块分发相同激励后独立进行计算，通常会得到正确的输出，但由于执行体实现方式不同，其输出不会完全相同，因此需要多模表决模块对各执行体输出进行处理后进行裁决。执行体输出的结果存在异常时，多模表决模块则会根据预设的裁决算法剔除异常输出，并触发策略调度模块，对出现异常的执行体进行切换与清洗，同时根据预设的调度算法选择新执行体，保证系统的冗余度。

拟态防御通过对同一激励下各异构执行体的输出进行裁决和动态改变任务的执行环境，防止系统因单个漏洞而无法提供预设的服务，提升了攻击者对系统进行侦察和攻击的难

度，在现有工程技术条件下，给出了一种能够抵御未知漏洞后门的可行方案；与“静态、异构、冗余”架构的防御机制相比，拟态技术动态改变执行环境的设计降低了长期威胁下攻击者获取系统情报的概率，同时使相同的攻击不能在系统中成功复现；而与随时间变化调整攻击面的移动目标防御相比，基于裁决结果调整执行环境是“有的放矢”的，避免了在未遭到攻击时的盲目调整造成的系统性能损失。

2.2 博弈论与网络攻防博弈

2.2.1 博弈论基础

博弈论[51]是一种对竞争过程进行建模的理论，用于对竞争参与者的策略选择进行分析，其中任何一位参与者的行动和选择都会影响所有参与者的收益，并且竞争过程中的参与者被认为是理性的，在博弈过程中将尽其所能最大化其收益。博弈有几种基本的分类方法，合作博弈中相互作用的竞争多方存在彼此均承认的约束，不存在此种约束的即为非合作博弈；根据竞争多方对竞争对手相关信息的掌握程度同样可以对博弈进行分类，完全信息博弈即竞争多方彼此均掌握了对方的精确信息，不掌握或掌握信息不准确的即为不完全信息博弈；最后，博弈还可被分为静态博弈与动态博弈，静态博弈是指竞争各方均在不掌握其他参与者行动信息的情况下选择自己行为的博弈过程，而当竞争各方可以对首先进行选择的参与者行为进行观察时即为动态博弈。

下面对博弈论中的基本概念[52]进行介绍：

• 参与者（Player）：又称玩家或局中人，是指在博弈场景中的策略选择与实施者。

• 策略空间（Strategy Set）：针对博弈中可能出现的情况，决策者可能采取的可选行动计划。

• 收益函数（Payoff Function）：当参与者使博弈达到某个特定结果时，可以取得的收益情况的表达式。

• 均衡（Equilibrium）：存在一组策略的组合，对所有参与者而言，只要其他人不改变策略，其选择的策略将带来最大的收益。均衡点即是指所有理性参与者都没有单独改变策略的想法。

2.2.2 网络攻防博弈

网络空间的安全态势日益复杂，攻击方法与防御手段都在不断升级演进，复杂的现状使得网络安全研究人员无法掌握攻防过程的准确情况，博弈论则能够提供对网络攻防过程的量化描述。将网络攻防过程进行抽象化后可以发现，网络攻防实质上就是攻防双方两种参与者在攻防行动的策略空间中选择策略，期望用最小的代价获得最大收益的过程。因此，利用博弈论对网络攻防进行建模分析，对攻防行为进行描述，对安全结果进行量

化分析，能帮助网络安全研究人员理解所面临的攻防局势，选取网络防御的最佳策略，同时对网络攻防博弈求解均衡点可以得到对安全结果的定量预测，能够证明所选策略的有效性。

当前已有一些学者将博弈论应用于网络安全分析中。其中多数研究基于经典博弈模型[53]3-[55]对网络中攻击者与防御者的攻防过程进行抽象建模描述，将攻击者与防御者的成本与收益都进行定量分析，并对系统应采取的最优安全策略进行均衡求解。但既有的网络攻防博弈研究偏重于理论化，欠缺与现实攻防场景的结合，无法有效描述动态性与智能化程度日益提升的攻防过程。将博弈理论应用于蜜网场景攻防过程，能够量化攻防策略的收益，为系统安全策略的选择提供指导，具有重要的现实价值。

2.3 本章小结

本章主要介绍了蜜网系统相关研究中所涉及的相关技术原理，一是介绍了基于拟态的 DHR 架构的技术原理与优势，二是对文中用到的博弈论和网络攻防博弈模型进行了描述，并介绍了其研究现状。

第三章 基于拟态思想的蜜罐设计

3.1 引言

蜜罐是蜜网的核心基础设备，为蜜网构建欺骗环境提供模拟真实服务的虚假端口、主机和服务。在网络环境中部署蜜罐，能够隐藏真实的网络资产，干扰攻击者的探测并吸引攻击，对恶意行为进行诱捕与分析，掌握攻击者的目的、方法与路径等个性化信息，从而扭转攻防博弈的不对称局面。然而，专用蜜罐设备成本太高，且无法进行业务转换与扩展，严重阻碍了蜜罐技术的发展与应用。而云计算和虚拟化技术的发展，使从笨重的硬件蜜罐到动态灵活的虚拟蜜罐的转换成为了可能。虚拟蜜罐不仅成本较低，且管理便利，拥有良好的扩展性，因而被广泛应用于各类信息系统。

然而，蜜罐作为一种放置在网络内部的设备，为了诱捕攻击者，需保持与外界的连通。这使蜜罐成为了一个可供利用的跳板，若攻击者获得蜜罐节点的权限，就可以通过蜜罐对内网进行横向渗透，甚至攻击第三方，造成巨大危害。

而在虚拟蜜罐被大量使用的今天，上述情况并非天方夜谭。虚拟化技术给蜜罐带来机遇的同时，也带来了安全方面的隐患。虚拟机与宿主机间的屏障并非牢不可破，如图 3.1 所示，近年来，虚拟机逃逸（Virtual Machine Escape）引发的安全事件时有发生。虚拟机逃逸是指攻击者从虚拟机发起攻击，突破虚拟机防御边界进入宿主机，在宿主机中执行指令。而对基于虚拟化技术的蜜罐设备，在攻击者得到蜜罐节点的控制权限后，不仅其所在蜜网的功能作用完全丧失，还将成为攻击者进入内部网络的跳板，危及到整个网络系统。

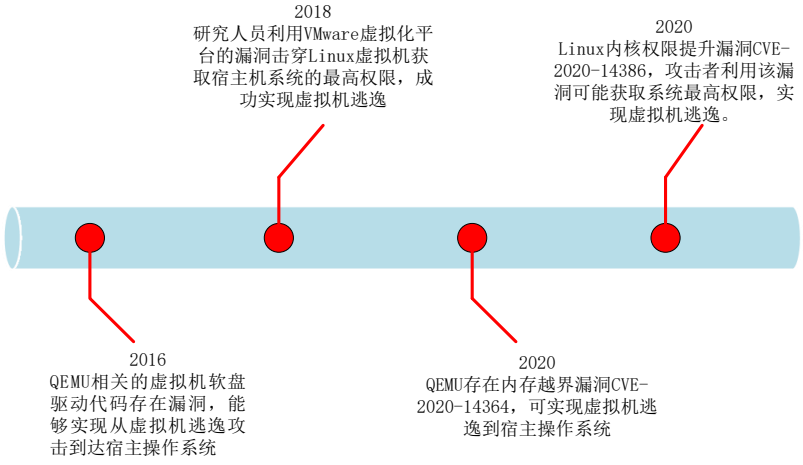

图 3.1 虚拟化逃逸现状

当前蜜网普遍采用虚拟化蜜罐，在同一硬件设备上部署多种类型的蜜罐诱饵。虚拟化平台如果存在安全漏洞，那么蜜网系统的安全基础就如空中楼阁，反而会成为攻击者突破内网安全防护的入口。

同时，现有蜜罐通常是固定配置的，单个蜜罐仅能提供模拟业务的一种实现方案，导致其指纹特征单一，欺骗能力不足，在构建蜜网网络欺骗环境时，所模拟环境的真实程度与甜度欠缺，无法有效吸引攻击者。

为防止攻击者利用漏洞后门对蜜网系统中蜜罐的虚拟化层进行逃逸攻击，并同步提升蜜罐的欺骗环境构建能力，本文在蜜罐中引入网络空间拟态防御思想，提出拟态构造蜜罐。基于 DHR 架构为蜜罐植入“动态，异构，冗余”的思想，为解决上述问题提供了一个有效可行的答案。

3.2 拟态构造蜜罐总体设计

3.2.1 双重裁决机制

对蜜罐进行基于拟态架构的改造，不能简单地遵循其他拟态构造设备的改造方法，这是因为蜜罐设备期望攻击者在蜜罐内部进行可控范围内的攻击尝试，因此，与其它设备不同，当蜜罐出现异常情况时，不能简单地屏蔽，而是需要根据具体情况进行处理。在拟态蜜罐的设计过程中，判断攻击者对蜜罐所进行的攻击能否被限制在蜜罐内部，而不会进一步对所在的宿主机造成侵害，并提升蜜罐诱骗能力，使攻击者能与蜜罐进行最大程度的交互，是设计的核心问题。针对这一问题，提出一种拟态化的蜜罐架构，并设计了双重裁决机制。

为了在确保虚拟蜜罐安全性的同时，提升其诱骗能力，将虚拟蜜罐的执行体划分为两个层次，分别为虚拟化层与业务层，在两个层级分别实现异构冗余架构，并根据两个层级不同的需求，设计了不同的裁决机制。

首先在虚拟化层运用拟态防御思想防止虚拟化逃逸。虚拟化环境中，宿主机上运行的虚拟机监控机是虚拟环境中运行的应用程序访问底层物理资源的接口，然而通过这个接口完成传输的不仅有正常数据，还可能是恶意程序的攻击指令。因此在虚拟化层级中，使用择多裁决机制对虚拟机与宿主机之间的交互数据以及该交互引发的宿主机指令数据进行裁决。通过对上述数据的比对表决，屏蔽出现异常的虚拟化平台，防止攻击者利用未知漏洞后门进行提权，获取宿主机的部分权限，进而将其作为跳板展开下一步攻击。

业务层的裁决机制则与一般的拟态裁决有所不同。拟态思想中，对发至系统内的请求由多个异构的执行体处理，经过择多表决机制屏蔽存在失陷可能的应答，避免攻击者在执行体内部发现可供其利用的漏洞后门并利用其进行攻击。然而蜜罐作为一种设计本意就是被探测、攻击和攻陷的网络资源，在虚拟化层的拟态机制确保了攻击行为的可控性后，做出应答时就应选择可能存在漏洞后门的应答，从而吸引攻击者与蜜罐进行更深层次的交互。故业务层的裁决机制与一般的拟态实现相反，为择少裁决，因此称其为反向裁决机制。反向裁决的拟态机制与一般拟态机制的主要区别就在于裁决依据，业务层的裁决模块收集各执行体运行产生的应答，对其进行反向裁决，最终输出甜度更高的响应。

3.2.2 拟态构造蜜罐的总体结构

与一般的拟态设备相同，拟态构造下蜜罐系统需要将所接收到的流量分发给同时运行的多个执行体，还需要系统对各执行体的输出进行裁决，得到输出结果。同时通过裁决机制的输出结果反馈获得相关执行体的运行情况，据此对执行体进行相应的调度。因此，在对蜜罐进行异构冗余配置之外，拟态构造的蜜罐系统中也设置了用于复制分发流量的代理程序，裁决执行体输出结果的裁决模块，以及对执行体进行调度的调度模块。

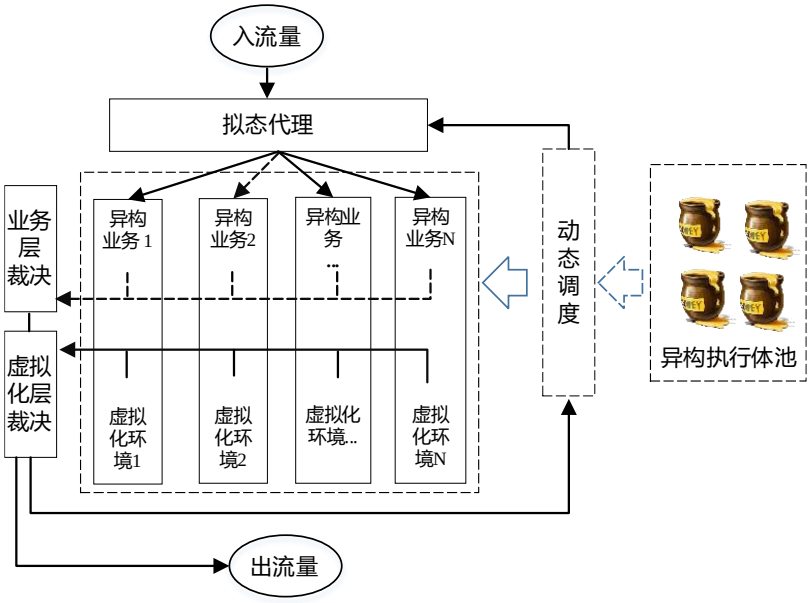

图 3.2 拟态构造蜜罐结构图

蜜罐系统的整体结构如图 3.2 所示，代理程序接收流量后，将其复制并分发至各执行体中，执行体在运行中将会产生不同层级的数据与最终应答，首先对虚拟化层的数据进行裁决，将虚拟化层存在异常情况的执行体的应答丢弃，同时激活动态调度模块，对出现虚拟化逃逸嫌疑的执行体进行下线清洗，并从执行体池中选取执行体上线，以保证蜜罐系统的异构性与冗余度。在第一轮裁决中虚拟化层未出现异常的执行体的应答将进行第二轮反向裁决，此轮裁决中将选择存在异常的应答作为最终输出，若均为正常，则依据预设的裁决规则进行选择。

下面对拟态构造蜜罐的四个功能模块：代理程序、异构执行体、双重裁决、动态调度分别进行介绍。

（1） 代理程序

代理程序完成与攻击者的数据交互。攻击者的访问请求被重定向到代理程序，代理程序收到访问数据后，对其进行复制，并转发给各异构执行体，同时接收并存储来自各执行体的响应数据。裁决模块比对判断后，把选择结果返回给代理程序，代理程序再将相应的输出返回给攻击者。

（2）异构执行体

拟态蜜罐中同时运行的异构执行体数量为三个，这是由于选用双执行体无法解决输出结果不一致的裁决问题，而三模冗余是择多判决的最低要求[56]，同时执行体数量增加势必导致算法复杂度与实现成本的激增，但却对增加表决结果的正确性概率无太大帮助，甚至有可能导致异常执行体多过正常执行体的情况。因此，三模冗余在实际中的可用性更强。

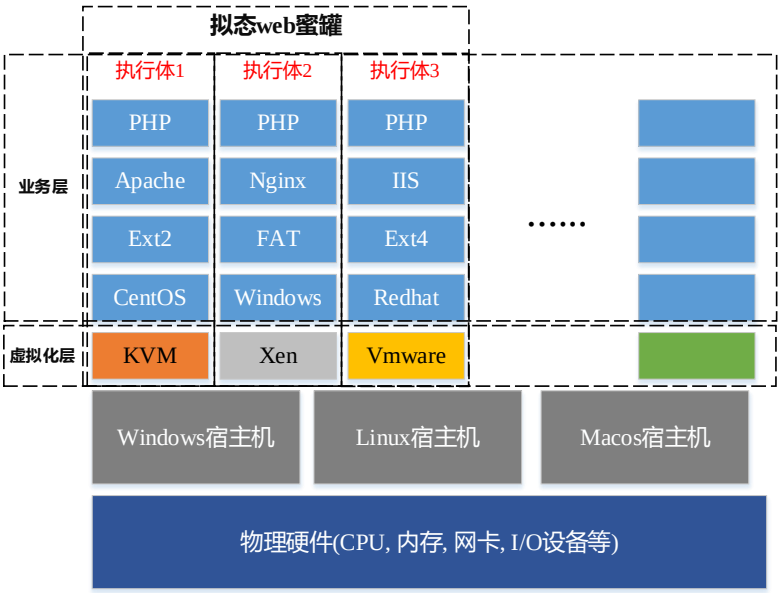

图 3.3 异构执行体实现示例图

为实现拟态构造蜜罐的设计目标，对执行体分为两个层次进行异构化，首先将承载蜜罐的虚拟机进行异构冗余处理，如 KVM、VMware、Virtualbox、XEN 等虚拟机平台。其次对蜜罐的模拟业务层进行异构冗余处理，以提供相同业务的多种实现，其中以拟态 Web 蜜罐为例，如图 3.3 所示，将相同功能不同构造的蜜罐与不同的虚拟化平台组合，组成拟态蜜罐系统中的执行体。当攻击流量进入拟态蜜罐时，流量将同时被分发到基于不同虚拟化平台的相同功能的异构蜜罐执行体中处理，并选择其中一个作为最终应答，这一过程对攻击者而言是透明的，攻击者不会感受到多个拟态蜜罐执行体的存在。

（3）裁决模块

裁决模块是拟态构造蜜罐实现其特殊设计目标的关键功能模块，对应所设计的双重裁决机制，裁决模块分为两个子模块，且两个裁决子模块串联运行。

虚拟化裁决模块对虚拟化层与宿主机间的交互数据与指令进行裁决。攻击者在进行逃逸攻击时，通常需要依赖特定平台的特定漏洞，因此其完成逃逸的方法在其他虚拟化平台成功复现的可能性很小，那么在其只能完成一个拟态执行体的逃逸的情况下，其他执行体仍然处于未逃逸状态，对交互数据与指令进行一致性比对，进行择多裁决，即能屏蔽逃逸执行体的异常扰动。虚拟化裁决的数据存在不一致情况时，则可能存在逃逸攻击，对不一致的执行体进行下线清洗，依据动态调度策略从执行体资源池中调用新的执行体替换失陷的执行体。

经过第一轮次的裁决后，对未出现异常的执行体的应答进行反向裁决：在各执行体应答中寻找不一致的选项，即择少裁决，作为最终输出结果返回给代理模块，同时在选取最终应答的过程中，可进行不同层级的表决。以拟态 web 蜜罐为例，在服务器将请求发送到不同实现的数据库时，对 sql 语句进行多模裁决，若该语句出现异常，那么也将增加该执行体输出结果被选中的概率，目的是最终选择甜度最高的应答并将选择结果通知代理程序。

（4）调度模块

调度模块在接收到裁决模块传输的蜜罐执行体异常信息后，依据所设计的调度策略从异构蜜罐执行体资源池中选择一个蜜罐执行体上线，将存在失陷可能的虚拟化执行体下线，并保存相关数据以还原攻击者的攻击手段，获取攻击者利用的漏洞，同时对虚拟化平台进行更新升级，修补漏洞。

3.3 拟态构造蜜罐策略

在基于拟态的 DHR 架构中，多模裁决与动态调度是对拟态架构运行效果影响极大的两种核心机制。对异构冗余执行体的输出进行裁决可为 DHR 架构带来对系统中未知扰动的感知功能。在拟态架构中对排除语义及语法干扰的多个输出进行一致性判别，可以感知出拟态架构中的非协同攻击或随机失效情况，同时裁决组件可以将相关状态信息发送给拟态系统中的动态调度组件作为系统负反馈调整的依据。动态调度组件同样是 DHR 架构中的重要一环，负责选择正常的执行体以替换失效或出现异常的执行体，能够赋予拟态构造设备多样化且难以预测的特性，对于拟态构造系统的防御效果有着直接的影响。

拟态构造蜜罐作为一种设计目的是被侦察攻击的网络设备，所承担的防御任务与其他拟态设备不同，故其两种关键性算法，即调度算法和裁决算法，也需要进行针对性的调整。基于拟态构造蜜罐的设计目标，为面向拟态构造蜜罐的两种机制引入新的参数，使拟态构造蜜罐得到针对性地增益。下面就引入的参数及其评价方法进行介绍。

1. 执行体信任度

能够稳定运行，并始终输出预设之中的响应是拟态 DHR 架构中执行体的核心属性，也是拟态系统进行调度和裁决的重要依据。因此，根据执行体上线时的运行状态，基于文献[32]中提出的执行体调度机制，设计了执行体信任度属性，作为对执行体运行和应答稳定程度的度量。

在初始状态下，每个执行体具有相同的信任度初值。执行体上线运行期间，如果出现异常状况，则信任度数值将会大幅下降；而随着执行体持续稳定运行与正常工作时间的增加，信任度数值会缓慢增长。其生成算法伪代码如下：

算法 1. 执行体信任度生成算法组.

输入：执行体及其状态

输出：执行体信任度

初始化流程.

输入: 执行体池 EB

FOR EACH eb IN EB

eb.trust=100, eb.state=NORMAL, eb.increaseRate=0

虚拟层裁决中表决发现执行体 ebx 为异常执行体时信任度处理算法.

输入: 异常执行体 ebx

IF ebx Is indanger:

ebx.state = FORBIDDEN

ebx.trust = ebx.trust - 81

IF ebx.trust < 1:

ebx.trust = 1

ebx.increaseRate=0

运行中执行体出现非虚拟化逃逸类型的异常时

输入: 异常执行体 ebx

IF ebx IS abnormal:

ebx.state = SUSPECT

ebx.trust = ebx.trust - 10

IF ebx.trust < 1:

ebx.trust = 1

ebx.increaseRate=ebx.trust/10

#HALF_STEP=20,

IF ebx.trust <20:#信任度低于 20 后，该执行体将不再被调度

ebx.state = FORBIDDEN

ebx.forbiddenTime = now()

执行体信任度的定时更新算法.

输入: 执行体池 EB

WHILE TRUE:

slebp(5)#每 5min 递增一次信用值

FOREACH eb IN EB:

eb.trust = eb. trust + eb.increaseRate

```
IF eb.trust > 100:
eb.trust = 100
eb.state=NORMAL
ELIF eb.trust > 20:
eb.state = SUSPECT
ELIF eb.trust > 0:
eb.state = FORBIDDEN
eb.forbiddenTime = now()
```

2. 甜度

甜度是蜜罐领域一种通俗说法，其主要是指蜜罐所构建的欺骗环境对攻击者的吸引力大小，所谓高甜度蜜罐就是指攻击者与之进行初步交互后，有较强攻击意愿的蜜罐。当蜜罐中内置了对攻击者而言可用性更高的漏洞时，自然会引发其强烈兴趣，从而吸引攻击者与蜜罐进行更深入的交互。因此，甜度是蜜罐产品的一种关键性能指标，是蜜罐欺骗环境构建能力的重要基础。

而在基于拟态 DHR 架构的蜜罐中，由执行体完成与攻击者进行交互的功能，故只有提升执行体的甜度，才能提升拟态构造蜜罐的甜度，增强拟态蜜罐整体的欺骗能力。执行体的甜度主要取决于其内置漏洞之于攻击者的可用性，可用性具体分为漏洞威胁等级与漏洞利用频率，分别表征了漏洞可造成的危害程度与漏洞被利用的频繁程度。更常见的以及能造成更严重后果的漏洞显然更受攻击者的青睐，而当执行体内嵌此类型漏洞，无疑会激发攻击者的攻击意愿。因此，本文对执行体的甜度进行量化评估，作为评价执行体的重要指标。

选用漏洞的 CVSS 评分[57]评估漏洞的威胁等级，对执行体中预设的漏洞，提取其 CVE_ID，得到其 CVSS 评分，以表征攻击者利用该漏洞可造成危害的严重程度。漏洞 CVSS 评分示例如表 3.1 所示。

表 3.1 漏洞 CVSS 评分表

编号	CVE_ID	CVSS 分数
1	CVE-2018-16844	7.8
2	CVE-2017-16064	5.0
3	CVE-2019-13980	6.8
4	CVE-2020-10865	7.5
5	CVE-2021-3349	2.1

如表 3.2 所示，漏洞利用频率分为五种，由漏洞在一定时期内被利用的次数决定，由专家经验给出不同漏洞利用频率的阈值，划分漏洞利用频率的等级，并给出不同利用频率所占权重值。

表 3.2 漏洞利用频次等级表

利用频次	权重	说明
总是	1.0	总是被利用的漏洞
经常	0.7	经常被利用的漏洞
较少	0.5	较少被利用的漏洞
很少	0.3	极少被利用的漏洞
从不	0.1	从未被利用过的漏洞

根据漏洞在上述两种评估中的分数，单个漏洞的甜度数值计算如下：

$$value = w * s \tag{3-1}$$

其中w是根据漏洞利用频次得到的权重，s是漏洞的 CVSS 评分。计算单个漏洞的甜度后，根据执行体内预设漏洞的情况，对执行体甜度计算如下：

$$value_{exc} = \sum_i^n w_i * s_i \tag{3-2}$$

其中n表示执行体中预设漏洞的个数，w_i，s_i分别表示第i个漏洞的利用频次权重与威胁程度。

3. 历史威胁数据价值

蜜罐工具在与攻击者交互的过程中，对攻击者的恶意行为进行记录，以完成对攻击者攻击目的、策略和方法的刻画，并通过对攻击者个性化行为的分析，提升网络系统的防御能力，甚至能够通过这些数据调查、追踪攻击者。可以看出，蜜罐获取数据的价值相当重要，因此，将执行体获取的历史威胁数据的价值作为考量执行体优劣的一个参数。然而，由于威胁数据对系统的安全增益情况需要安全管理人员在实践中进行感知，无法直接给出明确的评价方法，因此，执行体的历史威胁数据价值由安全管理人员在对安全事件进行分析后根据专家经验给出。

引入上述三种参数后，裁决与调度策略增添了更多蜜罐安全性与欺骗性增益的考量，下面对两种策略进行介绍。

（1）裁决模块策略

拟态构造蜜罐的裁决模块分为两个部分，两部分的裁决目标不同，所以其裁决策略也存在较大差异，需要对其分别进行描述。

虚拟化层的裁决目标是保证虚拟化基础平台的稳定运行，其裁决策略的基础是择多判决。然而简单的择多判决不能应对全部情况，虚拟化层的裁决策略需要引入信任度参数完善其功能。具体描述如下：

• 若三个执行体均正常运转，且产生相同的待判数据，则三个执行体均进入下一轮裁决；

• 若三个执行体均正常运转，其中两个产生相同的待判数据，则产生相同数据的执行体进入下一轮裁决；

• 若三个执行体均正常运转，均产生不同的待判数据，则信任度较高的执行体进入下一轮裁决；

• 若只有两个执行体能够正常运转，且产生相同的待判数据，则两个执行体均进入下一轮裁决；

• 若只有两个执行体能够正常运转，且产生不同的待判数据，则信任度较高的执行体进入下一轮裁决；

• 若只有一个执行体能够正常运转，则其进入下一轮裁决。

业务层的裁决目标是向攻击者提供甜度更高的应答，其裁决策略的基础是择少判决，择少判决中能够直接做出裁决的情况不多，更需要引入其他参数来辅助裁决。具体裁决规则如下：

• 若存在三种应答，有一种应答不一致，选择不一致的应答作为最终响应；

• 若存在三种应答，三种应答均一致或均不一致，又或者只存在两种应答时，首先选择历史威胁数据价值最高的执行体的应答作为最终响应，若存在相同的历史威胁数据价值，则选择甜度值更高的执行体的应答，若两者均一致，则随机选择一种作为最终响应；

• 若只存在一种应答，将其作为最终响应。

（2）调度模块策略

拟态构造蜜罐的设计目标是在提升蜜罐自身安全性的前提下，增强其诱骗能力，在考虑调度模块策略时，应当包含这些因素，从而使设计的 DHR 架构为蜜罐设备带来预期的增益。完全基于随机的调度显然无法满足这一要求，故将执行体信任度与甜度引入调度策略中，具体如下：

对执行体池中的执行体进行遍历，计算每个执行体的信任度与甜度，并基于此计算执行体的调度指数（SI，scheduling index），调度指数的计算方法如下：

$$SI = r_1 * tru + r_2 * vul \qquad (3\text{-}3)$$

其中，$r_1 + r_2 = 1$，用于表征信任度与甜度的权重值。根据系统当前状态进行调整，若对稳定性要求较高，则选择较大的r_1值；若对吸引攻击者有较高要求，则选择较大的r_2值。计算完成后，在需要进行执行体调度时，优先调度调度指数较大的执行体。

3.4 实验与评估

基于拟态架构的蜜罐实验环境如图 3.4 所示，使用 1 台服务器作为拟态业务节点，其中部署了拟态代理、多模裁决和动态调度等拟态机制服务。使用 3 台服务器作为拟态执行体节点，各服务器分别运行不同的虚拟化环境，并部署多种蜜罐服务的异构实现，此外设置两台对照服务器，分别用于部署基于不同虚拟化平台的普通蜜罐与真实业务，其中业务类型设置为 web 服务，并在三种测试主机中均配置相同的网站，以便对拟态构造蜜罐进行全面的评估。在局域网外设置一台主机，作为测试数据的发送源，模拟从外网发起的访问。

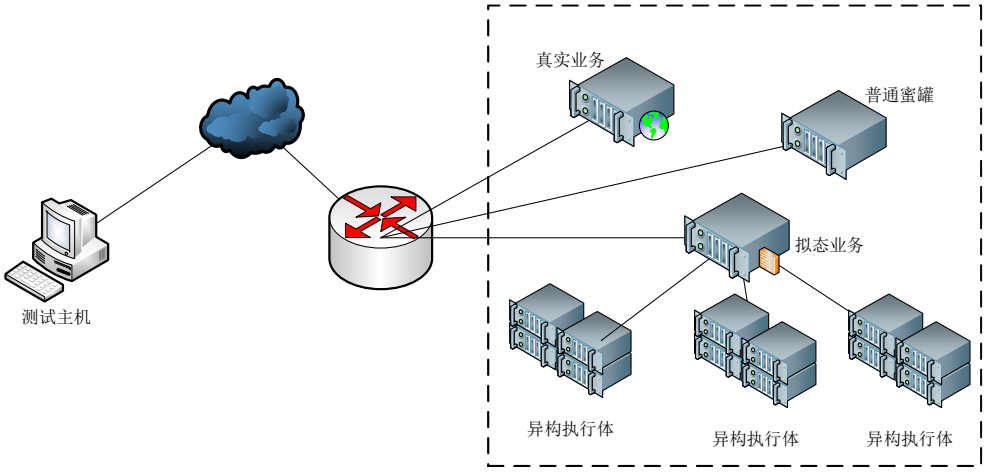

图 3.4 蜜罐实验网络图

3.4.1 性能测试

首先对拟态构造蜜罐的响应速度与数据传输速度进行测试，蜜罐的响应速度与数据传输能力不仅影响蜜罐捕获恶意数据的能力，过长的响应时间还可能引起攻击者的警觉，可能导致蜜罐暴露，故维持与真实业务系统相近的系统性能亦是蜜罐的核心能力。拟态构造蜜罐中数据需要经过较复杂的处理流程，带来一定的性能损耗，下面通过实验对比真实业务、普通蜜罐与拟态构造蜜罐的响应性能。

1. TCP 连接建立的响应速度

对各类型主机中建立一个完整 TCP 连接的时间进行统计，其结果如图 3.5 所示。根据实验结果可发现，真实业务建立 TCP 的速度较快，普通蜜罐中的速度稍慢，这可能是由于蜜罐中的监控程序导致的时间消耗。而本文所提拟态架构蜜罐则由于流量分发与应答裁决，造成了一定的时间消耗。

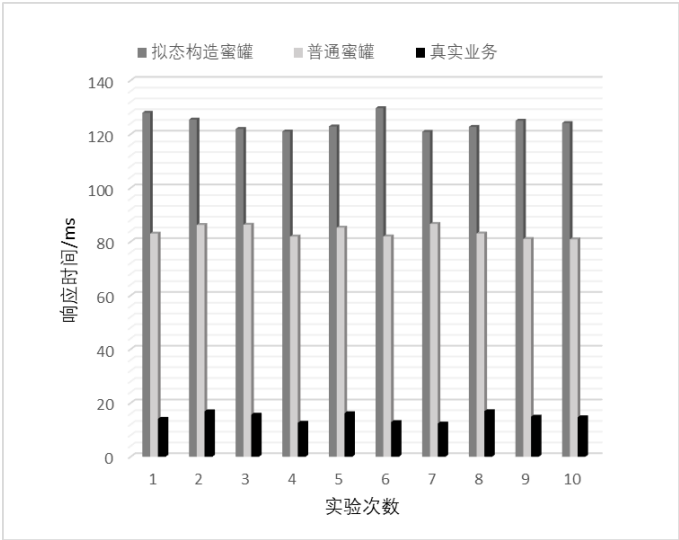

图 3.5 TCP 连接建立响应时间图

2. TCP 数据传输的速率

对各类型主机中基于 TCP 传输不同数据量所用时间进行统计，其结果如图 3.6 所示。由图 3.6 可以看出，三种主机在完成数据传输任务时所需时间几乎一致，这是因为拟态构造蜜罐仅需在建立连接时消耗较多时间，而在传输过程中的损耗较小，与普通蜜罐的损耗大致相同。

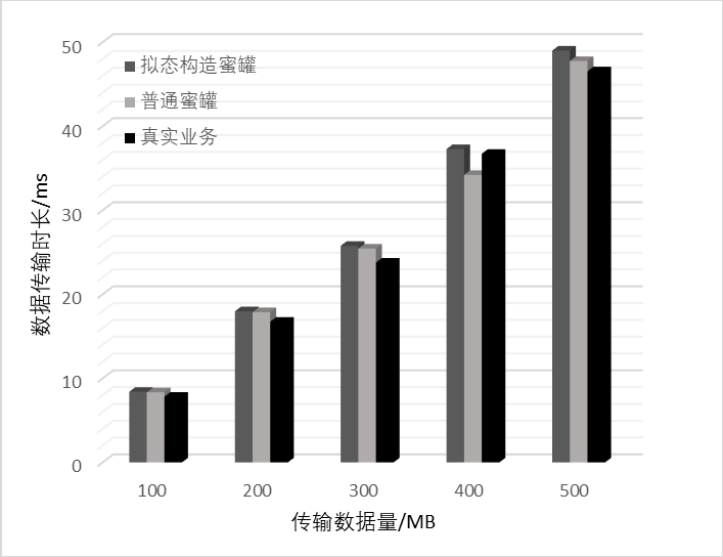

图 3.6 TCP 数据传输时长

根据上述两个实验的结果可发现，尽管拟态构造蜜罐在完成连接建立时响应速度较慢，但就完整的 TCP 数据传输过程而言，拟态构造对蜜罐性能带来的影响不大，不会降低欺骗环境的仿真程度，进而引起攻击者的警觉。

3. HTTP 请求响应速率

HTTP 请求的响应速率同样是 web 服务器的重要性能指标，关乎到用户访问服务器中内置网页的速度，故该响应速率是否在正常范围内，同样会影响到欺骗环境构建的真实程度。在测试主机中安装脚本，脚本内容如图 3.7 所示。

```
const Http = new XMLHttpRequest();
const url='https://jsonplaceholder.typicode.com/posts';
Http.open("GET", url);
Http.send();

Http.onreadystatechange=(e)=>{
    console.log(Http.responseText)
}
```

图 3.7 HTTP 请求脚本

由图 3.8 的实验结果可以看出，拟态构造蜜罐对 HTTP 请求的响应速度与其他两类主机几乎无差别，能较好地模拟 HTTP 服务，不存在暴露蜜罐存在风险。

上述蜜罐性能测试结果表明，拟态构造蜜罐能够较好地完成其模拟仿真的业务。在引入拟态架构后，受到分发机制等的影响，蜜罐的响应速度有一定的降低，但仍在合理的范围内，未对蜜罐所构建欺骗环境的真实程度造成影响，不会引发攻击者的怀疑，也无

法成为蜜罐识别的依据。

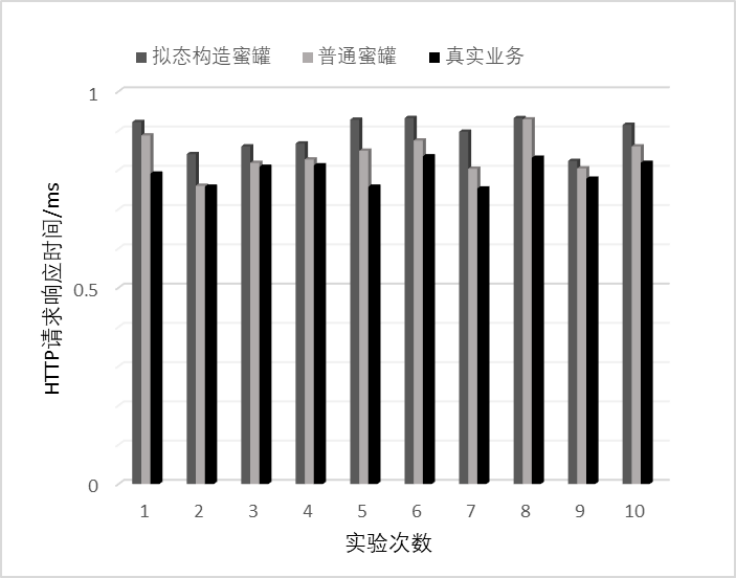

图 3.8 HTTP 请求响应时间

3.4.2 安全性与欺骗性测试

拟态构造蜜罐引入 DHR 架构提升其虚拟化层抵御未知漏洞后门的能力，增强了蜜罐的安全性。设计实验对普通蜜罐与拟态构造蜜罐分别进行测试，验证拟态构造蜜罐具有优于普通构造蜜罐的安全性与欺骗性特性。

测试中，在实验网络内仅同时开启一种蜜罐，从测试主机出发，模拟攻击者对两种蜜罐进行全阶段攻击，观察两种蜜罐的表现。

在扫描侦察阶段，使用网络扫描工具 Nmap 等对拟态构造蜜罐与普通蜜罐进行扫描，探测其内部信息，发现经过多次扫描，仍无法获取拟态构造蜜罐的准确信息，这是因为在攻击的开始阶段拟态构造蜜罐就对攻击进程进行了阻断。而普通蜜罐在多轮扫描后，其信息始终一致，显然无法吸引攻击者采用更加个性化的攻击方法对系统进行侦测。

在漏洞利用阶段，为检测拟态构造蜜罐在抵御虚拟化层未知漏洞后门的能力，为各虚拟化执行体与虚拟化普通蜜罐埋设可实现虚拟化逃逸漏洞的后门，分别测试触发漏洞能否引起虚拟化逃逸。选取 KVM，vbox，docker 作为测试使用的三种虚拟化异构执行体，在虚拟化层中预先埋设 shell.php 文件，在访问过程中使用攻击工具远程调用脚本，从而在虚拟化层中产生异常交互数据，造成蜜罐执行体的异常。

以 vbox 执行体为例，Vbox 的 exp 利用为反弹 shell，在一台能与 Vbox 异构体通信的主机上开启 NC 监听端口，命令为 nc -lvp 12345 ，然后在蚁剑中调用集成脚本，执行命令为 bash ./exp.sh one vbox ip port 会自动反弹 shell 到监听主机上，引发了如所示的虚拟机逃逸示警。

图 3.9　蚁剑攻击界面

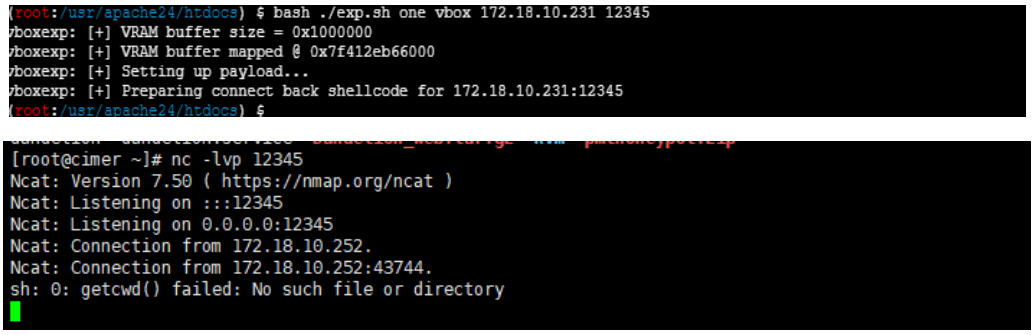

图 3.10　反弹 shell 示意图

图 3.11　虚拟机逃逸示警界面

漏洞利用阶段的实验结果显示，在引发虚拟化层异常后，拟态构造蜜罐对其进行了响应，将出现异常的执行体进行了下线操作，并将其响应丢弃。而普通蜜罐对其虚拟化层的状态并无感知能力，因而对虚拟化层的异常状况未作出任何反应，存在较大的安全隐患。

在攻击维持阶段，首先关闭拟态机制，对选用的执行体虚拟化层进行一次成功的逃逸攻击，随后开启拟态机制，测试能否再次利用该后门进行逃逸，以检测拟态构造蜜罐在攻击维持阶段的防御能力，同时在普通虚拟化蜜罐中进行相同操作。测试结果显示在拟态构造蜜罐中，尽管基于既有的漏洞后门进行过成功攻击，攻击者仍无法再次进行虚拟化逃逸，而在普通蜜罐中，攻击可以进行复现，并就此展开对宿主机的攻击。

上述实验结果表明，拟态构造蜜罐能够对全阶段的攻击进行干扰和阻断，使攻击者无法得到系统内设备的准确信息，同时基于系统漏洞后门的攻击均被阻断，极大地提升了攻击者对蜜罐设备进行识别和利用的难度，保障了蜜网核心基础设备蜜罐的安全性，增强了其欺骗性。

3.5 本章小结

本章针对蜜罐高度依赖虚拟化技术导致的安全缺陷，以及蜜罐服务构建方式单一导致的诱骗能力欠缺等问题，提出了基于拟态思想的蜜罐系统，介绍了拟态构造蜜罐的设计方案、总体结构和各模块功能，针对拟态构造蜜罐的特点设计了双重裁决机制，并设计了两种适用于拟态构造蜜罐的算法。最后通过实验验证了拟态构造蜜罐的可用性、欺骗性与抵御未知漏洞后门的能力。

第四章 面向蜜网的网络欺骗机制

4.1 引言

在现实攻防场景中，攻击者往往会使用多种攻击手段，从网络中的多个主机出发，对网络进行渗透，最终危及整个系统。而传统防御技术大多属于被动式防御，总是落后于攻击方，无法对网络内出现的复杂威胁做出及时响应。主动防御技术主张在攻击对信息系统造成实质破坏前进行主动保护，能更好地应对 APT 攻击与零日漏洞攻击，因此受到了广泛的关注。

蜜网即是一种重要的主动防御技术，通过提供虚假数据、端口、预设漏洞的服务和主机等网络资源，构建欺骗环境，以吸引攻击者的攻击，诱使攻击者与蜜网进行深度交互，从而获取攻击者的个性化数据与未知漏洞的利用信息，利用这些数据对网络防御策略进行调整，能够有效提升网络信息系统的安全性。

然而，蜜网作为一种网络设备，同样存在失陷可能。特别是传统蜜网中网络结构固定、蜜罐配置盲目单一，导致其动态性与欺骗能力不足，构建的欺骗环境真实程度较低，不仅无法有效吸引攻击者进行交互，在交互过程中被识别的可能性也较大，无法获取恶意数据的同时甚至会成为攻击者进入内网的通道及攻击的跳板。

因此，基于主动防御中的网络欺骗技术，从蜜网结构与组成出发，本文面向蜜网提出了一种网络欺骗机制，以改善蜜网静态呆板的网络结构与软件配置，增强蜜网的安全性与欺骗环境构建能力。

4.2 网络欺骗机制设计

攻击通常可分为三个步骤:获取拓扑和端口信息，寻找主机漏洞并利用，发送蠕虫渗透内网。本文针对前两个步骤，提出一种网络欺骗机制，对攻击者想要侦察的网络情报进行隐藏。该机制包括蜜罐设置与双频率 IP 地址跳变两部分，分别用于在设备层中混淆网络中软件、系统的配置信息和在网络层中混淆网络拓扑信息。通过部署该机制，使攻击者无法得到准确信息或得到的信息快速失效，以此迟滞攻击者的攻击进程，捕获更多恶意行为。

4.2.1 双频率 IP 地址跳变机制

IP 地址是主机的重要标识符，也是攻击者进行攻击的必要信息，IP 地址跳变可扰乱攻击者获取拓扑信息的进程，从而中断其攻击。而在部署蜜网的网络中进行 IP 地址跳变就需考虑在不降低真实主机服务质量的同时，如何利用蜜罐实现最大程度的信息混淆。

考虑上述问题，结合文献[58]中的 IP 跳变方法，本文提出一种双频率 IP 地址跳变机

制，该机制采用多个虚拟 IP（virtual IP，vIP）与单个主机的真实 IP（real IP，rIP）相对应，并随时间改变主机对应 vIP 的方法。其中真实主机和蜜罐主机的 IP 地址跳变频率不同，真实主机以较低的频率进行跳变，蜜罐主机则以真实主机 n 倍的频率进行跳变(跳变参数 n≥1)。同时，在蜜罐主机与攻击者进行交互时，会降低其地址跳变频率，以迷惑攻击者。双频率跳变机制在真实主机跳变频率较低时即可完成对网络信息的混淆，既保证了真实主机的服务质量，又最大化了网络内 IP 地址的总体变化频率。跳变的具体过程如下：地址空间内的可用 IP 地址存储在虚拟 IP 地址池内，将 IP 地址进行分组。一个地址跳变周期内，真实主机分别分配一组地址，蜜罐主机分别分配 n 组地址，作为该主机在跳变周期中的虚拟地址组。进行跳变时，各主机在其虚拟地址组内选择一个地址作为其 vIP。分组分配跳变地址的方式避免了跳变过程中一个 vIP 可能对应多个 rIP 的问题。

4.2.2 特殊蜜罐群配置

为实现更好欺骗效果，蜜网中的蜜罐往往模拟网络中的真实主机。本文中蜜罐部署在内网的各业务区域，为增强蜜罐诱骗攻击者的能力，使其提供与所在区域真实主机相同的服务。不同业务的蜜罐均配置少量高交互蜜罐和适量低交互蜜罐，这样就以较低的成本在网络中形成了一个与真实业务系统总体结构相似但分支结构更为复杂的蜜网系统，提升了系统结构的复杂程度，真实主机与蜜罐混合的网络结构也让攻击者无法轻易识别并绕过蜜网。

此外，对服务类型相同的蜜罐进行异构配置。若服务类型相同的所有主机的系统与软件配置均一致，则会为攻击者利用漏洞提供便利：攻击者可以针对特定配置选取特定的攻击方法与手段。因此，机制为服务类型相同的主机进行异构化处理，例如对应使用 Windows+IIS 的 web 真实主机，系统将配置具有 Linux+Apache 特征的虚拟蜜罐作为系统和软件配置多样性的补充。多样化的系统与软件配置信息隐藏了业务系统的真实配置，使攻击者无法准确发现主机漏洞并利用。本文第三章所提出的拟态构造蜜罐可为特殊蜜罐群配置提供简便的实现方法。

4.3 网络欺骗机制下的 SDN 蜜网系统设计

为实现前述的网络欺骗机制，蜜网系统需完成以下任务：分配每台主机的 vIP 地址范围；确定主机在当前跳变周期中的 vIP；完成网络通信并保证跳变周期的结束不会造成通信中断；统计网络内的通信数据并做出分析。而 SDN 技术具有可编程的特点，且控制器可对蜜网内的流量进行精细化的控制，为在蜜网中部署所提出的网络欺骗机制提供了灵活的实现方式。

4.3.1 蜜网总体架构

蜜网总体架构如图 4.1 所示，蜜罐宿主机分布式地部署在各业务区域内，宿主机中可

挂载多个虚拟蜜罐，其挂载的虚拟蜜罐模拟与区域内真实主机相同的业务，真实主机与虚拟蜜罐的对应关系由控制器进行存储。控制层中，SDN 控制器(如 NOX[59])通过南向接口监视和控制整个网络[60]，收集交换机的转发行为并同步地进行地址转换，同时完成网络流量统计、流表下发等功能；通过北向接口与应用层交互，接收跳变策略并上传应用层所需的数据。

应用层由三个模块构成:地址跳变模块、态势分析模块与策略下发模块。地址跳变模块对可用地址进行分组与分配，并维护虚实地址间的对应关系。态势分析模块通过分析网络中的 packet_in 事件监测攻击者的攻击动向，并将相关的信息提供给地址跳变模块，作为分配地址的重要依据。策略下发模块则将地址跳变模块制定的策略以流表的形式下发给控制器。策略下发模块负责接收地址跳变模块制定的策略，以流表的形式传输给控制器。

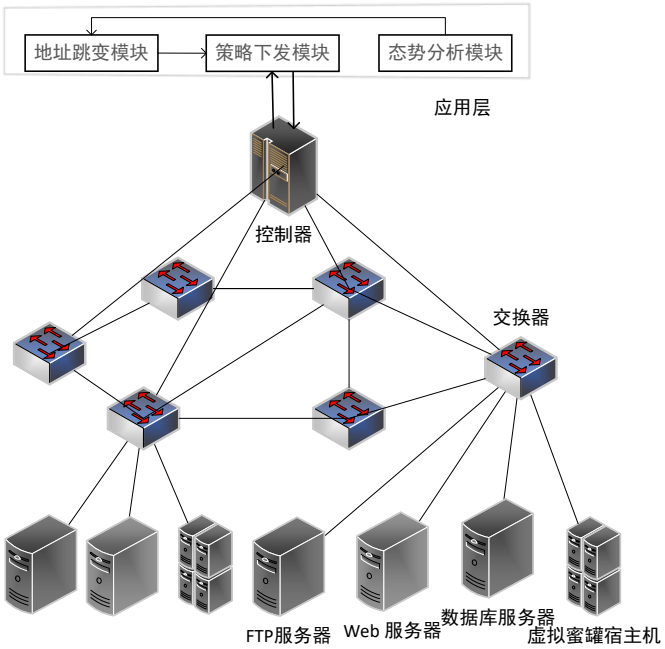

图 4.1 蜜网总体架构

4.3.2 网络通信方法

由于网络中真实主机的 vIP 地址会定期更改，因此访问真实主机需通过 DNS 查询获得其当前 IP 地址。而对基于 SDN 的蜜网，转换是主机所属的物理子网的交换机执行的。其具体流程如图 4. 2 所示：

STEP1 源主机发送 DNS 查询请求，该请求需通过控制器;

STEP2 DNS 服务器将响应发送到控制器；

STEP3 控制器通过查询地址关系对应表，将 DNS 响应中的 rIP 用其对应 vIP 替换；

STEP4 源主机接收目标主机的 vIP 并启动连接；

STEP5 以此 vIP 作为目标地址，交换机询问控制器，控制器将流表下发到相应交换机中；

STEP6 交换机按流表将数据包转发至目的主机子网的交换机；

STEP7 数据包到达目标子网，控制器将目标 vIP 用其相应的 rIP 替换，数据包到达终点。

未完成的通信进程中将始终执行此地址转换。这样的机制可以保证即使主机的 vIP 地址发生变化，已建立的通信进程不会中断。

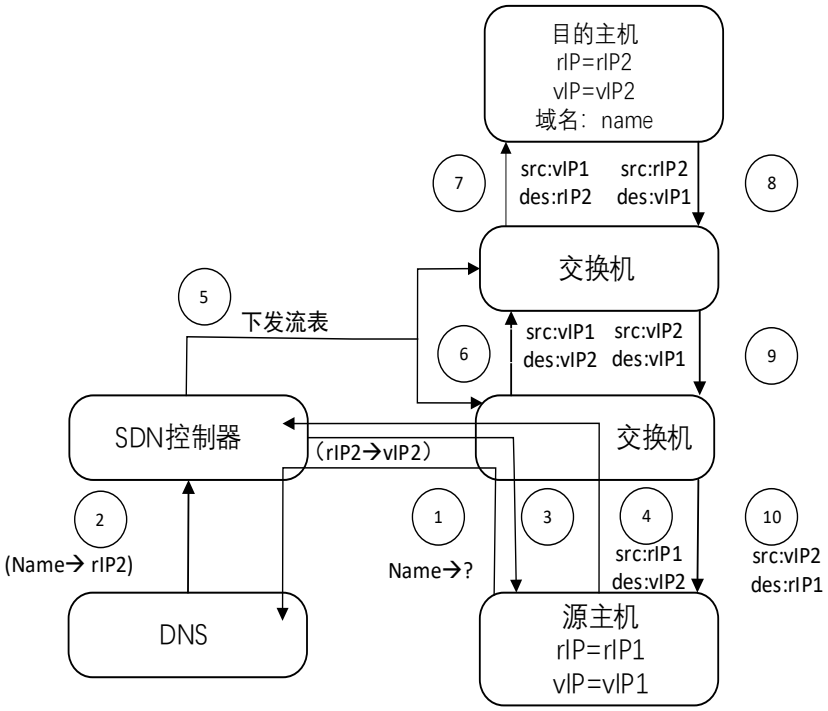

图 4.2 通信流程图

4.3.3 模块功能

地址跳变模块维护可用虚拟地址池，并在每个跳变间隔计算所需地址组数量并生成地址组。在完成虚拟地址组的生成后，将其与主机对应，并传输给策略下发模块。

虚拟地址的选择与分配考虑两个因素：首先，为了提高攻击效率，攻击者往往遵循一定的策略来发现攻击目标，发现攻击目标后，下一个攻击点将主要在初次攻击地址的附近进行选择，因此被扫描过的地址可认为其暂时安全，而其附近的地址接下来被探测的几率很大。地址跳变模块接收态势分析模块发送的攻击态势，并将攻击者扫描过的地址范围打上标记，将标记地址附近的地址分配给高交互蜜罐，以期获取其更多攻击信息。而对已标记的地址范围，攻击者往往不会重复扫描，可认为其较为安全，因此将这些地址分配给真实主机；其次，在当前轮次未分配给主机的 IP 地址，在下轮地址跳变中将被优先选择。

跳变周期结束时，某些 vIP 可能因所属主机的通信进程仍未完成而不被释放，地址跳变模块会将这些 vIP 暂时排除。针对被占用的 vIP，策略下发模块将特定条目添加到交换器流表中，目的地址为被占用 vIP 的数据包被路由到其对应主机。这种机制保证了通信进程不被中断，也避免了攻击者通过建立许多长期连接耗尽可用地址范围[61]。

态势分析模块通过统计连接尝试中指向的空地址范围收集攻击者的攻击倾向，并将相应数据传输给地址跳变模块，作为其分配地址的依据。只统计指向空地址的连接是因为普通访问者通过 DNS 查询等正常流程对真实主机进行访问，且其请求通常不会指向未分配给主机的空地址，因此指向空地址的连接数据中一般不会包含普通访问者的请求，能更准确地体现攻击者的攻击倾向。

此外，态势分析模块会对蜜罐是否处于交互状态进行判断，并将处于交互状态的蜜罐信息传输给地址跳变模块。地址跳变模块将调整给定蜜罐的跳变频率以迷惑攻击者。

策略下发模块负责从地址跳变模块接收策略并将其实施。策略下发模块通过北向接口与控制器进行交互，以虚实地址间的映射关系为载体向控制器下发策略，控制器则将相应 vIP 与 rIP 的转换关系以流的形式下发到交换机内。

4.4 蜜网攻防博弈分析

蜜网与攻击者间的攻防过程可以简化为一种策略选择的博弈。为了验证本文提出网络欺骗机制在网络防御中的有效性，使用博弈论对蜜网与攻击者之间的攻防进行建模。网络攻防是攻击者与网络管理者之间的一种对抗，故攻防双方都将隐藏自己的策略，也不存在达成协议的可能性，因此双方处于非合作状态。此外，管理者对攻击者的行为特征、工具方法等信息没有详尽的数据，攻击者也无法获取蜜网的全面信息。因此管理者与攻击者之间的博弈可以被认为是目的上对立，战略上依存的非合作不完全信息动态博弈。此外，尽管蜜网与攻击者的对抗过程是动态的，但在对抗的不同轮次中，对抗双方的策略选择并无先后顺序，因此可将攻防过程进行拆解，即得到静态博弈模型并单独分析，以证明网络欺骗机制对蜜网的增益。

4.4.1 攻防博弈模型

本文所提融合网络欺骗机制的蜜网在攻防中呈现出不同于普通蜜网的特点：从攻击者的角度来看，网络内的主机类型不仅有真实主机，还有与真实主机指纹特征相同的蜜罐以及用于异构的异构蜜罐，同时蜜罐中存在着不同交互程度的个体；而从蜜网防守者的角度来说，蜜网的访问者中将可能出现攻击者和正常用户两种。

对攻防博弈过程的参与者给出以下合理性假设：

1. 理性人假设

攻防过程中的玩家均是理性的个体，在攻防过程中为求取自身的最大利益选取绝对理

性的行为策略，这是攻防博弈过程完成的基础。

2. 最大利益假设

攻击者所谋求的最大利益是在以可接受的代价获取网络信息系统的真实资源。管理者求取的最大利益是在保护真实主机提供给正常用户预设服务的同时，尽可能地对攻击者恶意行为进行监测与记录。

同时对本文中的蜜网攻防过程用四元组$\{N,\Omega,\Theta,E\}$形式化描述如下：

与攻防博弈相关的局中人集合N=$\{N_s$,N$_v\}$，其中N_s代表服务提供者，N_v代表系统访问者。

局中人的类型空间中$N_s=\{S_r$,S$_{hh}$,S$_{lh}\}$，其中S_r代表真实主机服务，S_{hh},S_{lh}分别代表高交互和低交互蜜罐服务；$N_v=\{V_n,V_a\}$，其中V_n代表正常访问者，V_a代表攻击者。

局中人的策略集合分为服务提供者与系统访问者两种。服务提供者的策略集合为$\Omega=\{\Omega_0,\Omega_1\}$,其中$\Omega_0$代表蜜网主机不提供服务，$\Omega_1$表示主机提供服务。访问者的策略集合为$\Theta=\{\theta_0,\theta_1\}$，其中$\theta_0$表示访问者不访问蜜网主机或攻击者不发动攻击，$\theta_1$表示访问者访问蜜网主机或攻击者发动攻击。

局中人的收益分别为$E=\{E_s,E_n,E_a\}$，其中E_s为蜜网系统的整体收益，这其中既包括真实主机提供服务带来的收益，也包括蜜罐系统捕捉攻击者获取的信息收益；E_n为正常访问者的收益情况，E_a为攻击者的收益情况。

4.4.2 攻防决策收益量化

对各局中人的收益进行量化，可以发现，当蜜网系统提供的服务类型不同时，各局中人的收益情况也不相同，因此需要分情况计算：

当蜜网系统提供的是真实主机服务时，此时若访问系统的为正常访问者，则真实主机向正常用户提供了服务，蜜网系统与正常访问者的收益均为α；而若访问系统的为攻击者，则攻击者在获取系统信息的同时还占用了真实主机的服务资源，故蜜网系统的收益为$-\mu\alpha$（μ为攻击者的不同攻击消耗真实主机服务资源的程度，$0<\mu<1$），相对应地，攻击者在访问中获得的收益为$\mu\alpha-\gamma$（γ为攻击者的攻击成本）。

当由于系统跳变导致访问者无法访问而无法提供服务时，蜜网系统和正常访问者的收益均为$-\alpha$，蜜网系统和攻击者的收益则分别为 0 和$-\gamma$。

而当蜜网提供的是蜜罐服务时，若访问系统的是正常访问者，则蜜罐主机是无法给用户提供可用服务的，那么蜜网系统和用户的收益均为$-\alpha$；若访问系统的是攻击者，则蜜罐主机进行诱骗，其收益为$\eta\lambda_h+(1-\eta)\lambda_l$（$\lambda_h$，$\lambda_l$分别为高低交互蜜罐可获得的攻击者信息价值，一般地，$\lambda_h>\lambda_l$，η为与攻击者进行交互的是高交互蜜罐的概率），攻击者被蜜罐主机诱骗的收益则为$(\eta-1)\lambda_l-\eta\lambda_h-\gamma$。上文中各符号的含义如

表 4.1 所示。

蜜网攻防博弈过程中的双方是战略依存的，攻防者的策略选择都会受到另一方的影响。

本文中，蜜网系统服务提供者的纯策略组合为$\{\Omega_0,\Omega_0\}$，$\{\Omega_0,\Omega_1\}$，$\{\Omega_1,\Omega_0\}$，$\{\Omega_1,\Omega_1\}$其中前者表示真实主机是否提供服务，后者表示蜜罐是否提供服务。访问者的纯策略组合用二元组表示为：$\{\theta_0,\theta_0\}$，$\{\theta_0,\theta_1\}$，$\{\theta_1,\theta_0\}$，$\{\theta_1,\theta_1\}$，前者表示正常访问者是否访问系统，后者表示攻击者是否攻击系统。

表 4.1 符号含义表

符号	含义
α	正常访问者获得预设服务后正常访问者与蜜网获得的收益
μ	攻击者不同攻击消耗系统资源的程度
γ	攻击为单次攻击付出的代价
λ_h，λ_l	分别为高低交互蜜罐获取恶意数据的收益
η	与攻击者进行交互的是高交互蜜罐的概率

在进行蜜网攻防博弈的过程中，本文引入海萨尼局中人来处理不完全信息博弈，由虚拟的局中人“自然”决定服务提供者和访问者的类型。蜜网系统无法预知访问者的类型，但对访问者的类型具有一个先验的判断：

$$p(V_a)=p，\ p(V_n)=1-p，\tag{4-1}$$

同样地，访问者对所要访问的服务类型也有一个先验判断：

$$p(S_{hh})=q_1，\ p(S_{lh})=q_2，\ p(S_r)=1-q_1-q_2\tag{4-2}$$

首先对服务提供者的角度计算其最优策略，在实际的蜜网系统中，正常访问者通常会选择访问系统，因此访问者的策略组合仅考虑$\{\theta_1,\theta_0\}$，$\{\theta_1,\theta_1\}$两种。

在访问者策略为$\{\theta_1,\theta_1\}$的情况下，分别计算两种服务提供者的收益：

由图 4.3 中的博弈树可知，在访问者采取$\{\theta_1,\theta_1\}$策略时，蜜罐类型的服务提供者的绝对占优策略为Ω_1即提供服务。

真实主机提供或拒绝服务的收益计算如下：

$$\begin{aligned}E_s(\Omega_o)&=p(V_a|\theta_1)\times 0+p(V_u|\theta_1)\times(-\alpha)\\&=p\times 0+(1-p)\times(-\alpha)=-\alpha(1-p)\end{aligned}\tag{4-3}$$

真实主机类型的服务提供者的最优策略为Ω_1，则需要满足前提条件$E_s(\Omega_1)>E_s(\Omega_o)$。

化简得到关于p的不等式$p<2/2+\mu$，即在约束条件$p<2/2+\mu$下，服务提供者的最优策略集合为$\{\Omega_1,\Omega_1\}$。又已知$0<\mu<1$，那么只有在$2/3<p<1$的极端情况：即攻击者的访问占所有访问数的绝大多数时，此时蜜网系统应关闭真实主机，不再提供服务，而这也符合实际中蜜网博弈的规律。

在访问者策略为$\{\theta_1,\theta_0\}$的情况下，分别计算两种服务提供者的收益：

同样地，由图 4.3 中的博弈树可知，在访问者采取$\{\theta_1,\theta_0\}$策略时，蜜罐类型的服务提供者的绝对最优策略为Ω_1，即提供服务。

$$E_s(\Omega_1) = p(V_a|\theta_0) \times 0 + p(V_n|\theta_1) \times \alpha = 0 + (1-p)\alpha = (1-p)\alpha \quad (4\text{-}4)$$

$$E_s(\Omega_0) = p(V_a|\theta_0) \times 0 + p(V_n|\theta_1) \times (-\alpha)$$
$$= 0 - \alpha(1-p) = (p-1)\alpha \quad (4\text{-}5)$$

p作为概率值，取值范围为[0,1]，在此条件下，$E_s(\Omega_1) > E_s(\Omega_0)$恒成立，即服务提供者存在绝对优势战略组合 。

综合上述，蜜网系统服务提供者的优势策略为提供服务。

基于蜜网服务提供者的最优策略集，对系统访问者的收益进行计算：

$$E_{V_a}(\theta_1) = p(V_s|\Omega_1) \times (\mu\alpha - \gamma) + p(V_h|\Omega_1) \times (-\eta\lambda_h - (1-\eta)\lambda_l - \gamma)$$
$$= \mu\alpha - \gamma - q[\mu\alpha + \eta\lambda_h + (1-\eta)\lambda_l] \quad (4\text{-}6)$$

$$E_{V_a}(\theta_0) = p(V_s|\Omega_1) \times 0 + p(V_h|\Omega_1) \times 0 = 0 \quad (4\text{-}7)$$

由上式可以得到当$q < 1/2$时，$E_{V_u}(\theta_1) > E_{V_u}(\theta_0)$，即当正常访问者选择访问系统时所访问的主要为真实服务时，其收益大于选择不访问的收益，此时正常访问者选择访问系统是较优策略。而本蜜网中，正常访问者访问时通过域名进行访问，对其而言蜜罐的分布概率接近于 0，符合其占优策略，不会影响正常用户的使用。

$$E_{V_u}(\theta_1) = p(V_s|\Omega_1) \times \alpha + p(V_h|\Omega_1) \times (-\alpha) = (1-2q)\alpha \quad (4\text{-}8)$$

$$E_{V_u}(\theta_0) = p(V_s|\Omega_1) \times 0 + p(V_h|\Omega_1) \times 0 = 0 \quad (4\text{-}9)$$

由上式可以得到，当$q > \mu\alpha - \gamma/\mu\alpha + \eta\lambda_h + (1-\eta)\lambda_l$时，$E_{Va}(\theta_1) < E_{V_a}(\theta_0)$，即攻击者若选择不攻击系统，则其收益将大于其选择攻击系统，此时理智攻击者的最优策略是选择不发起攻击。对不等式进行分析，其左侧为攻击者成功攻击的收益表达式，右侧为真实主机的损失与蜜罐诱骗攻击者的收益的和。在真实主机和蜜罐均提供服务时，实质上决定攻击者占优策略的是蜜罐的分布概率，在攻击者攻击收益较小而蜜罐捕获收益较大时，认为蜜罐的分布概率满足使攻击者放弃攻击的均衡条件。而本文提出的蜜网系统通过双频率 IP 地址跳变提升了蜜罐的分布概率，高交互蜜罐地址的特殊设置提高了蜜罐诱骗攻击者的收益，满足了不等式成立的条件，从而控制了攻击者的占优策略。

通过对蜜网攻防过程攻防双方的收益进行建模，计算分析得到在这一博弈过程中存在贝叶斯纳什均衡，并得到了满足均衡的条件，证明了所提蜜网系统在不降低服务质量的前提下提供了更强的防护能力。

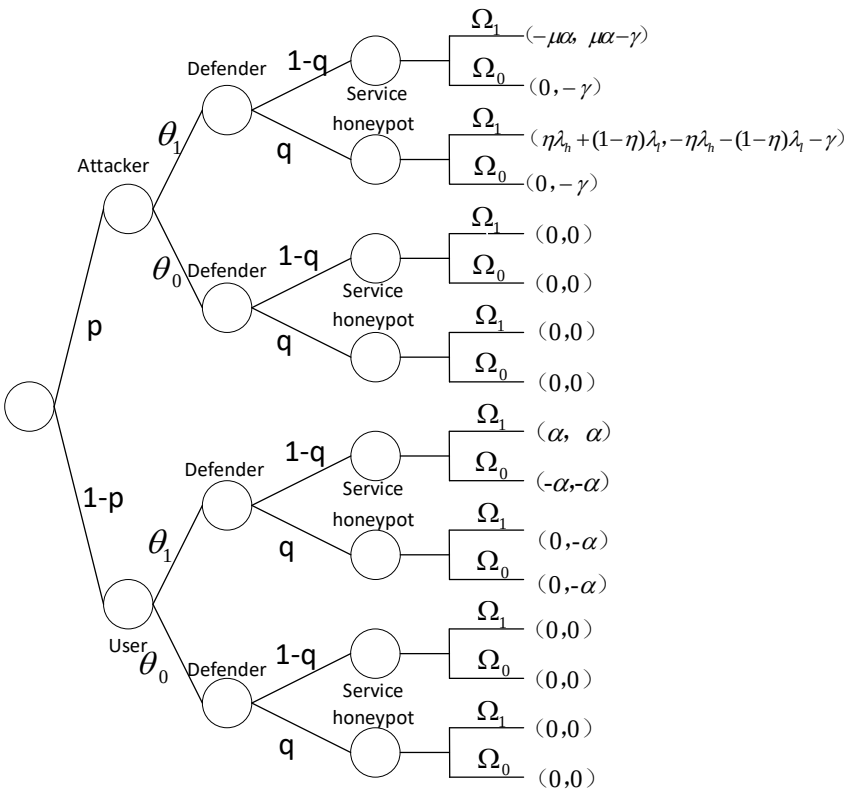

图 4.3 博弈树

4.5 实验与评估

本章中利用 Mininet 网络模拟器模拟融合网络欺骗机制的蜜网，使用 OpenvSwitch 交换机创建 OpenFlow 交换机网络，实验拓扑如图 4.4 所示。网内的真实主机设置域名，蜜罐主机仅设置 IP，路由由 ryu 控制器处理。为了模拟外部主机对蜜网内部进行访问，指定一个子网为外部子网(即设置外部网络地址)。Ryu 控制器中的功能模块使用 python 并基于 Ryu API 进行实现。

Mininet 运行在 VMware Station16 的 ubuntu16.04 虚拟机上，虚拟机内存为 8GB，核心处理器数量为 1。

测试的目的首先是验证融合网络欺骗机制的 SDN 蜜网的可行性，因此在搭建蜜网后，对其中真实主机的可达性进行测试，并与未实施网络欺骗机制的 SDN 蜜网进行服务质量的对比，验证其可行性；其次对基于网络欺骗机制的 SDN 蜜网进行扫描，将得到的网络拓扑与实际网络拓扑进行对比，以检测蜜网系统对攻击者的欺骗效果；最后进行重放攻击实验来检测蜜网系统对攻击行为的捕获能力。

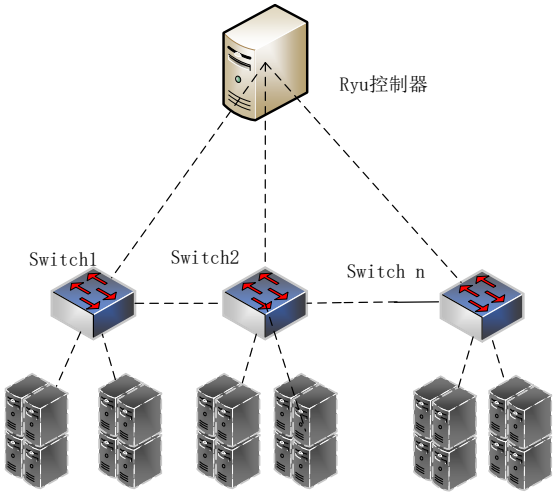

图 4.4 实验环境拓扑

4.5.1 开销测试

分别搭建一个普通 SDN 蜜网和融合网络欺骗机制的 SDN 蜜网，通过设置的外网主机对蜜网内部的真实主机通过域名进行访问，首先验证了在结合网络欺骗机制的 SDN 蜜网中真实服务是可用的。然后在数量不同的交换机下，对两种不同蜜网访问域名的平均时延进行记录。图 4.5 展示了跳变参数设置为 2 时，以 5s、10s、20s 为跳变周期的网络欺骗蜜网和普通 SDN 蜜网完成域名访问响应的平均时延数据。统计数据中剔除了在首次访问域名时因流表下发产生的时延，其数值过大，会影响到对数据的理解。

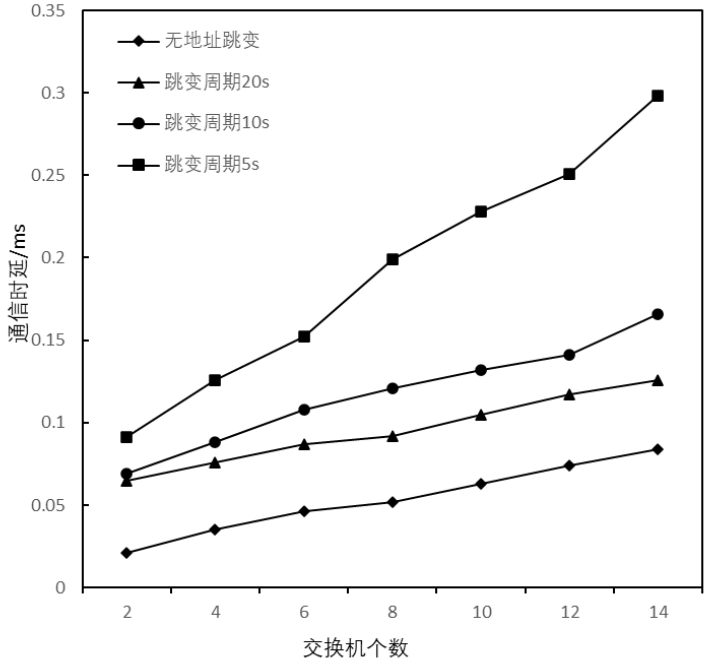

图 4.5 服务质量测试

由图 4.5 服务质量测试可以看出，随着交换机个数的增加，通信时延也随之增长，

且跳变频率越高，时延增长的速度也就越快。这是因为无地址跳变时，交换机中流表固定，在转发数据流时根据流表即可完成。而跳变频率变高时，虚拟 IP 地址与真实 IP 地址对应关系的变换速率也就增高，控制器需要高频率地更新地址映射关系并下发新的流表，更新越频繁，生成与下发流表所造成的时延也就越高。但同时从图 4.5　服务质量测试中可以看出，跳变周期为 5s 时，通过 14 台交换机转发的时延也未超过 0.3ms，对真实服务的质量未造成太大影响。

高频率生成下发地址组以及维护虚实地址映射同样会来更高的处理开销。图 4.6 展示了在单个 CPU 核下，相同的通信任务对蜜网部署网络欺骗机制前后造成的 CPU 负载情况。

从图 4.6 中可以看出在未部署网络欺骗机制的 SDN 蜜网中，通信进程带来的 CPU 负载率可以稳定在 17%左右，而随着网络欺骗机制中跳变频率的逐渐增大，CPU 负载率也随之增大。由此可见跳变频率的提高导致的计算任务增多，给 CPU 带来了更多的负载。

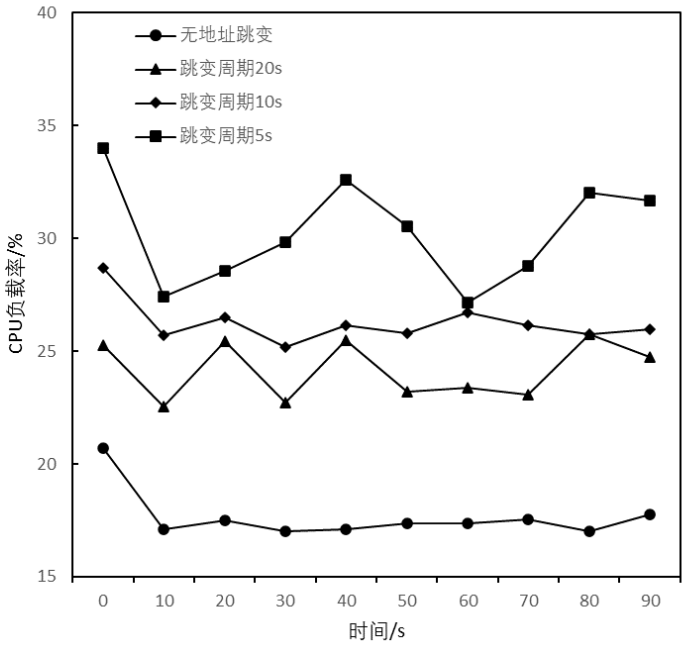

图 4.6 CPU 负载测试

4.5.2 欺骗性测试

为测试网络欺骗机制对攻击者的欺骗效果，在蜜网网络内设置 120 台主机，网络欺骗机制进行周期为 10s 的跳变，跳变参数设置为 2。在每轮跳变后使用 Nmap 对蜜网进行扫描并记录下所得到的网络拓扑，与网络的实际网络拓扑情况进行对比，通过每台主机所对应的 IP 地址的一致性来衡量蜜网的欺骗效果，其结果如图 4.7 所示。

由图 4.7 可以看出，在 100 轮次的扫描中，Nmap 对网络进行扫描的结果与真实情况的一致性最高不高于 29.17%。这说明在启用网络欺骗机制的情况下，攻击者无法获取系统

的有效网络拓扑，证明了欺骗机制的有效性。

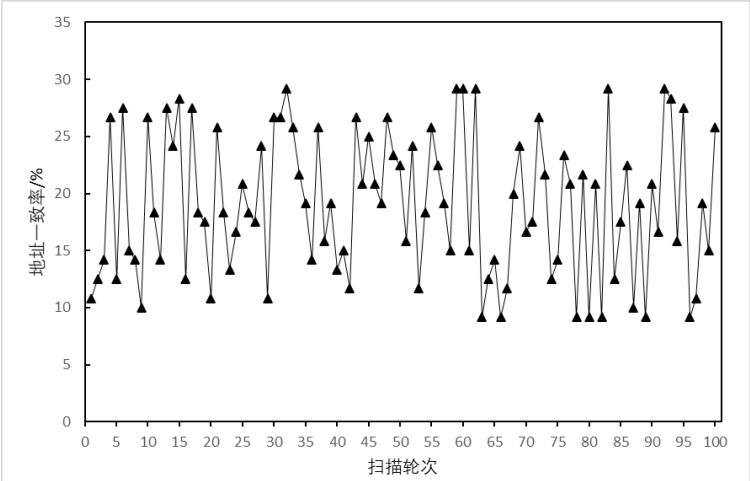

图 4.7 地址一致率

攻击重放实验中使用麻省理工大学林肯实验室发布的 DARPA2000 攻击数据集，该数据集极为全面，包括了五大类共计五十八种经典攻击模式。如 U 2 R 攻击，DoS 攻击等，因此被广泛应用于网络安全领域的测试评价中。实验中排除了其 DoS 攻击的部分，使用其中生成于内网的数据，部分数据如图 4.8 所示。

52	86.828482	Dell_a3:57:db	Cisco_38:46:33	ARP	60 Who has 172.16.0.1? Tell 172.16.112.20
53	86.828603	Dell_a3:57:db	Oracle_83:4a:82	ARP	60 Who has 172.16.112.10? Tell 172.16.112.20
54	86.828656	Oracle_83:4a:82	Dell_a3:57:db	ARP	42 172.16.112.10 is at 08:00:20:83:4a:82
55	86.829982	Cisco_38:46:33	Dell_a3:57:db	ARP	60 172.16.0.1 is at 00:10:7b:38:46:33
56	93.755960	172.16.112.100	172.16.112.10	NTP	90 NTP Version 1, client
57	93.756400	172.16.112.10	172.16.112.100	NTP	90 NTP Version 1, server
58	93.791395	Cisco_38:46:33	Cisco_38:46:33	LOOP	60 Reply
59	103.789244	Cisco_38:46:33	Cisco_38:46:33	LOOP	60 Reply
60	106.285185	172.16.112.20	172.16.112.10	NTP	90 NTP Version 3, client
61	106.285624	172.16.112.10	172.16.112.20	NTP	90 NTP Version 3, server
62	108.035211	WesternD_17:79:5a	Oracle_83:4a:82	ARP	60 Who has 172.16.112.10? Tell 172.16.114.50
63	108.035269	Oracle_83:4a:82	WesternD_17:79:5a	ARP	42 172.16.112.10 is at 08:00:20:83:4a:82
64	108.402917	192.168.1.10	172.16.112.10	NTP	90 NTP Version 3, symmetric active

图 4.8 数据集展示

除本文提出的融合网络欺骗机制的蜜网外，设置两种不同的蜜网作为对照实验：1）完全静态配置的蜜网，蜜网在完成初始配置后，所有网络信息固定；2）仅部署普通 IP 跳变机制的蜜网，三种蜜网中蜜罐与主机数量均一致，其中，网络欺骗机制与普通 IP 跳变机制的跳变周期均设为 10s，网络欺骗机制的跳变参数设置为 2。并对三种方案下的蜜网在使用数据集重放的方式进行攻击。其实验结果如图所示。

首先重放内网攻击数据集，对三种蜜网进行了持续 24 小时的模拟网络攻击，并记录各蜜网中捕获的攻击个数。其结果如图 4.9 所示。可以看出，三种蜜网在最初所捕获的攻击数量差距不大，这是由于 DARPA2000 数据集中的前一部分主要是扫描形式的攻击，因此各蜜网表现差异不大，但在攻击形式转换为以特定漏洞利用为主后，有针对性策略的网络欺骗机制下的蜜网的优势逐渐展现，捕获攻击数快速增长，而其他两种蜜网则表现不佳。

为避免单次实验可能存在的偶然性，分析单次实验中出现的攻击类型占比，并更改实验中攻击类型的占比进行重复实验以模拟真实网络攻击的随机性。重复实验根据前述的

网络威胁模型开展攻击，即扫描后利用漏洞攻击。实验结果如图 4.10 所示。通过对比可知，本文所提方案不仅在攻击捕获数量上占有优势，同时还具有良好的稳定性，对随机化的网络攻击具有较强的适应性。

根据本节中的实验结果，可发现本文提出的网络欺骗机制相比普通 IP 跳变机制能够使蜜网捕获更多攻击，表现出更强的欺骗性，证明了面向蜜网的网络欺骗机制的有效性与可行性。

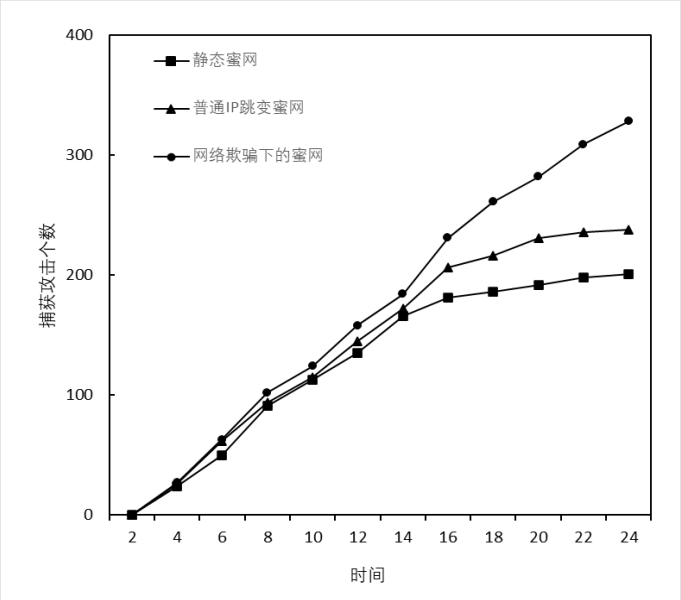

图 4.9 24 小时捕获攻击数量

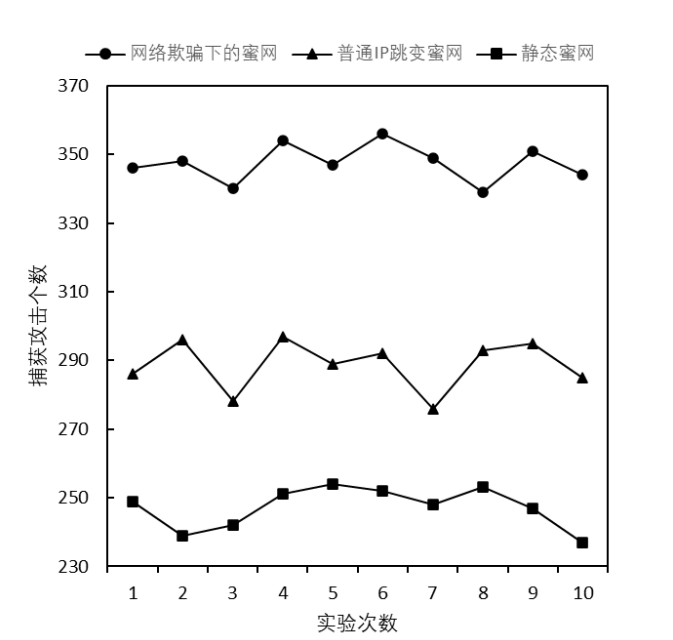

图 4.10 多次实验捕获攻击个数

4.6 本章小结

本章针对蜜网结构动态性不足与配置盲目缺陷导致的欺骗能力差、安全性较低等问题，提出了面向蜜网的网络欺骗机制，该机制设计了适用于蜜网的 IP 地址跳变方法和针对性的蜜罐配置方案，以改进蜜网结构与配置方面的缺陷，并通过博弈论验证了该网络欺骗机制对蜜网系统的增益。最后基于 SDN 技术设计了网络欺骗机制的具体实施方案，给出了该机制下的具体网络通信方法，实现了融合该机制的蜜网，并通过性能测试与欺骗性能测试验证了蜜网在不降低服务质量的同时可以有效地欺骗攻击者，提升自身安全性。

第五章 主动防御增强的蜜网系统设计与实现

网络安全形势日益复杂，漏洞的数量呈爆炸式增长，攻击者利用系统中的未知漏洞后门对网络信息系统发起渗透攻击，导致使用传统架构与设备的蜜网无法准确感知系统内威胁，甚至存在失陷风险。为解决这些问题，本章根据现实需求并结合第三章、第四章的研究内容，设计并实现了主动防御增强的蜜网系统。

5.1 需求分析

主动防御增强的蜜网系统通过引入拟态构造蜜罐与网络欺骗机制，从网络基础设备与网络结构组成出发，提升了蜜网的安全性，增强了其诱骗能力。为实现所提出设备与机制的运用与管理，完成一款具有较强可用性的蜜网工具，有以下业务需求：

（1） 实现对攻击流量的采集与存储，并能将攻击流量重定向至蜜罐服务集群中；

（2） 实现管理员自定义构建蜜网拓扑、设置蜜罐配置，并设置网络欺骗机制的相关参数；

（3） 实现拟态构造蜜罐中将流量分发至各执行体的功能；

（4） 实现对拟态构造蜜罐的各执行体进行管理的功能，包括上下线、清洗重置等；

（5） 实现拟态构造蜜罐中的裁决与调度功能；

（6） 实现对恶意数据的预处理，完成攻击态势的可视化展示，展示攻击类型、源 IP 地址信息等。

5.2 系统概述

5.2.1 系统总体设计

基于上述需求，对蜜网进行总体设计。系统设计的总体流程如图 5.1 所示，主要包括拟态构造蜜罐实现、节点资源管理、蜜网网络管理、数据采集及攻击态势展示。

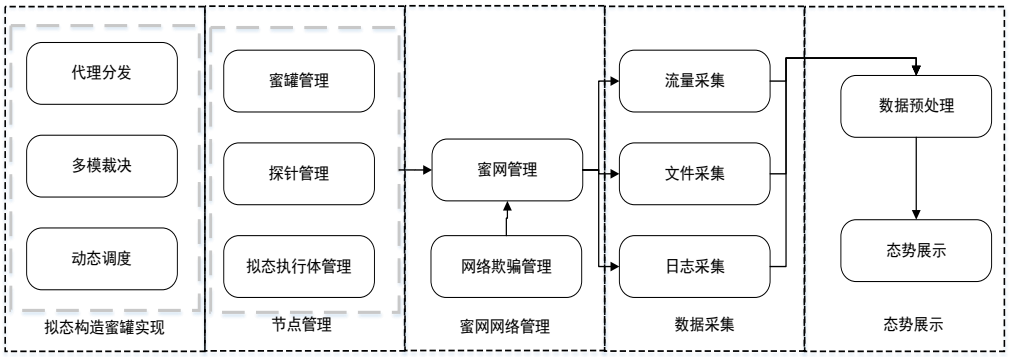

图 5.1 蜜网总体流程图

（1）节点管理。对蜜网中的各节点进行具体管理。包括蜜罐节点、探针节点与拟态蜜罐的异构执行体节点。

（2）蜜网网络管理。对蜜网的网络拓扑、蜜罐配置进行管理，并对网络欺骗机制的具体参数进行设置。

（3）数据采集。对包括外网流量、蜜罐主机内日志文件等在内的数据进行采集。

（4）攻击态势展示。对攻击数据进行预处理，进行存储并完成可视化展示，帮助安全研究人员理解网络形势。

（5）拟态蜜罐实现。对拟态构造蜜罐所用到的代理分发机制、多模裁决机制和动态调度机制进行实现。

5.2.2 模块设计与实现

根据系统的总体流程设计，本小节对系统的各模块进行了划分与设计，主要可分为数据采集模块、节点管理模块、网络管理模块、拟态蜜罐模块与态势展示模块。具体的系统功能模块如图 5.2 所示。

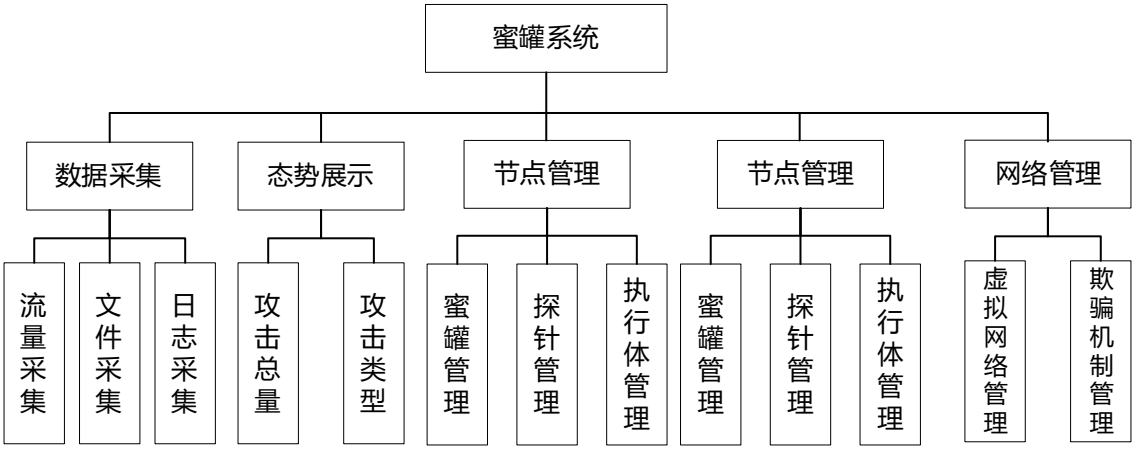

图 5.2 蜜网功能模块划分

1. 数据采集模块

数据采集模块的主要功能是对系统内的威胁数据进行全面的收集与存储。数据采集子系统分为日志采集、文件采集和流量采集。数据采集模块采用 C/S 模式，其工作流程如图 5.3 数据采集过程所示，数据采集子系统中按数据采集来源分为蜜罐节点和探针节点，蜜罐节点中通过系统日志采集登录行为，通过服务日志采集蜜罐欺骗环境服务数据，通过文件采集日志采集攻击者在环境中的文件操作，如添加文件、删除文件、修改文件等，通过网络日志采集攻击者在网络层的操作，将上述采集的日志发送给数据仓库。探针节点通过服务日志采集真实业务主机中 Web 服务、应用服务、系统服务出现的异常数据发送给数据仓库。

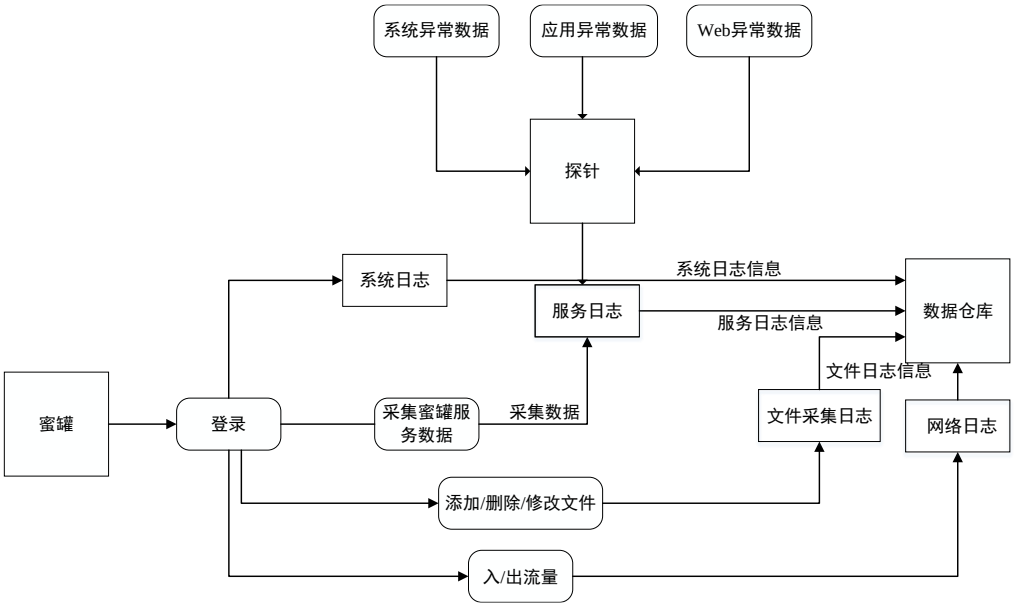

图 5.3 数据采集过程

日志文件主要通过两种方式采集，分别为监听程序主动发送和日志源文件监听：

1.监听程序主动发送：探针程序中包含了数据发送逻辑，使用 http 标准接口把产生的日志数据主动发送到采集子系统中完成采集。

2.日志源文件监听：使用 log_pilot 工具去监听指定蜜罐环境的指定日志源文件路径，当日志源文件发生写入操作，则 log_pilot 工具会将写入的内容发送到数据采集模块的 http 接口当中以完成采集。

日志模块的主要处理流程如图 5.4 所示。当蜜罐受到入侵或访问会产生系统日志和服务日志，经由执行体节点上的 log-pilot 服务集传输到主节点的 Logstash 进行预处理，选择 Elasticsearch 存储所有数据，以便于实现攻击数据的可视化。当部署在真实业务端的探针服务受到入侵或访问时会产生探针日志，探针程序发送至主节点后，同样由 Logstash 进行预处理并存储到 Elasticsearch 中。

流量采集是通过监听蜜罐子节点的出网卡流量实现的，使用开源的 pyshark 流量处理库去监听所有蜜罐的出入流量，然后将捕获的流量转发到拟态代理节点上的 Elasticsearch 中。

文件采集模块采用被动采集模式，系统将会监控蜜罐文件系统的目录以及文件的增删改操作，当蜜罐中文件发生变动时，系统将会进行监控并触发采集，保存蜜罐内发生变化的文件。首先获取蜜罐内的所有变化文件与上一次获取结果比对，比对结果中如有添加、修改和已删除文件则进行处理，处理后将当前获取结果替换到上一次结果中，等待下一次比对。利用添加和已删除文件集获取添加、已删除、修改文件集，获取后去除系统自增文件且为.swp 类型的文件。修改和添加文件需要判断该文件是否存在且不为空，已

删除文件则不进行处理，再将符合条件的文件复制到子节点中，下发复制文件任务，将文件上传到主节点对应的目录下，文件上传后则写入数据库。

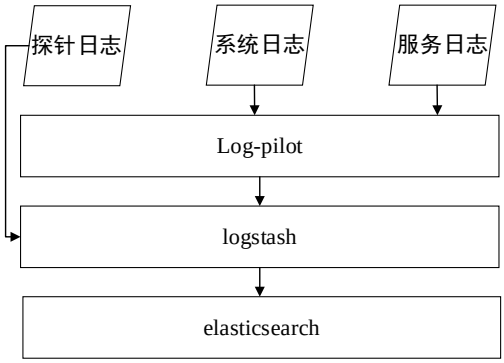

图 5.4 日志模块处理流程

2. 态势展示模块

态势展示模块直观动态地展示攻击源、攻击类型占比、蜜罐捕获流量趋势、最新捕获行为和告警等信息，多维度、多视角地实时监控攻击态势。

如图 5.5 所示，用户访问攻击态势展示页面后，系统的主要处理流程为查询数据库中 statistics 表中的 monitor 类型数据，并进行判断、筛选、去重、排序、分类处理，将其统一为 json 格式返回，最终展示在前端。通过 monitor 表和 attack_event 表分别获得展示页的被攻击服务、攻击源 IP 排行等数据展示信息以及报警事件和入侵事件信息，在页面上通过各种类别的图表进行展示，实现数据可视化。

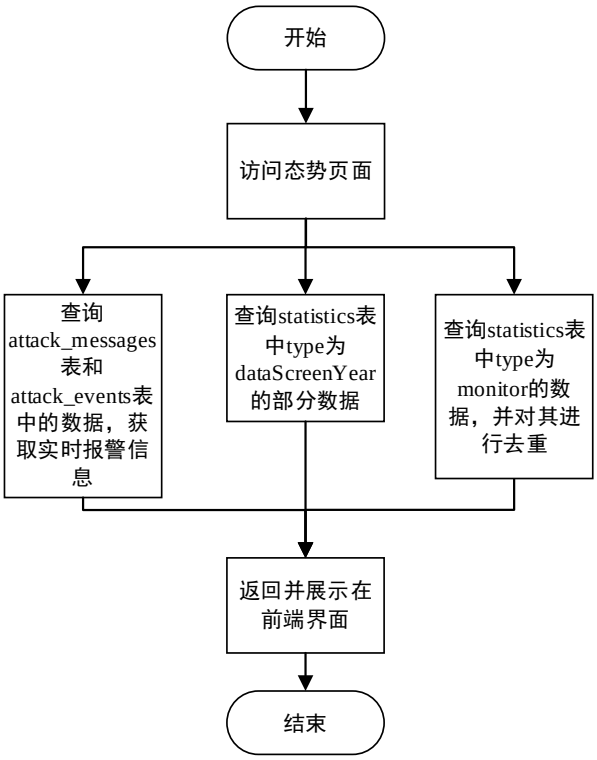

图 5.5 态势展示流程图

3. 拟态蜜罐模块

拟态蜜罐模块包括代理分发、多模裁决和动态调度三部分，其分别对应了拟态机制中的三种核心机制，三部分协作完成了拟态构造蜜罐的工作。

拟态代理完成与攻击者的数据交互，攻击者的访问请求被重定向至代理程序，代理程序收到请求数据后，对其进行复制，并分别转发给各异构执行体。同时接收并存储来自各执行体的响应数据。经过裁决模块比对后，把结果返回给代理程序，代理程序再将输出返回给攻击者，从而保证拟态构造蜜罐多执行体的框架对攻击者而言是透明的。

代理模块采用第三方库 Twisted 完成代理工作。对 Twisted 源码进行调整，使其能够完成一对多的代理模式。恶意流量进入该模块后即由 Twisted 对其进行复制，并分别转发给各异构执行体，代理模块还负责接收各异构执行体的响应内容，并在裁决模块作出裁决后输出最终响应，将其反馈至前端。

拟态裁决模块完成对执行体中相关数据和应答的比对与裁决。各异构执行体是对相同业务的异构实现，对于相同的请求，各异构执行体中产生的运行数据、响应数据在语义上应当是相同的，但所基于的语法可能不同，从而导致拟态裁决处理的数据形式可能存在不同，无法直接进行对比，因此需要首先对数据进行处理，提取其关键要素。

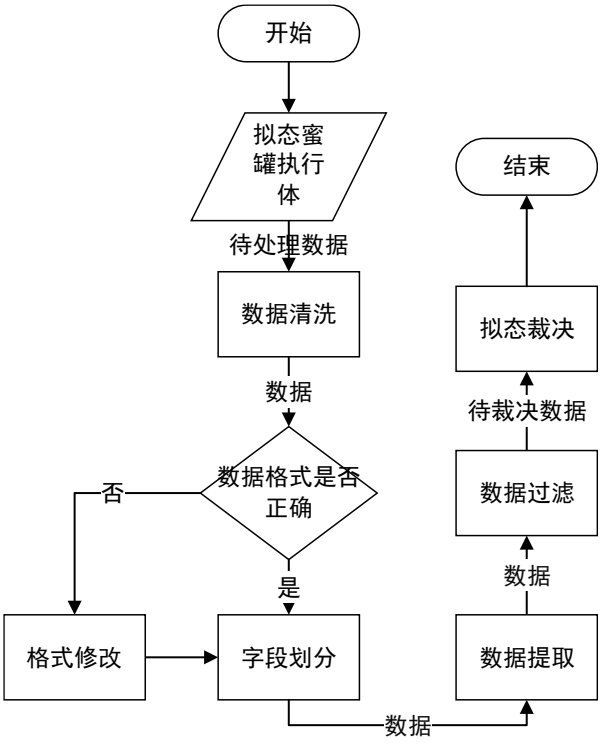

图 5.6 数据处理流程图

数据处理的流程如图 5.6 所示，主要包括数据清洗、字段划分、数据提取和数据过滤。从数据采集模块接收到数据后首先进行数据清洗，按照数据格式标准对不合法数据进行

格式修改，然后将清洗后的数据按照字段划分整理成系统可以自动提取的数据，最后再对数据进行过滤，进行最终的裁决对比。数据清洗主要使用 logstash 中的 filter 模块对数据进行关键字的提取。

字段划分模块中首先针对不同格式的数据编写出不同格式的正则表达式，通过正则表达式提取出不同的字段，之后对一些字段做出重命名，并删除无用字段，将保留字段交给数据提取模块，其部分代码如下所示。

```
#通过正则匹配提取字段
……
grok{
match=>{"message"=>"%{MONTH}\s+%{INT}\s%{TIME}\s([A-Fa-f0-9]{12})\sshd\[%{DATA:pid}\]:\sDisconnected\sfrom\suser\s%{WORD:username}\s%{IPV4:client_ip}\sport\s%{INT:client_port}"}
}
mutate{
rename=>{
"[output_fields][proc.cmdline]"=>"[output_fields][command]"
"[output_fields][user.name]"=>"[output_fields][username]"
"[output_fields][fd.name]"=>"[output_fields][filename]"
}
remove_field=>["[output_fields][container.image.repository]","[output_fields][container.name]","message"]
}
```

字段划分以后，数据已转换为 json 格式，在数据提取中，还需要对数据做出分段，如 HTTP 响应中的包头各字段以及数据，并根据数据中所包含的状态码或关键字信息，对关键数据进行提取。

数据过滤是将无用的数据进行过滤，ELK 采用 json 字典的方式存储数据。对于裁决中不需要的字段，可以通过使用 remove_field 来删除。例如若要丢弃解析出来的 port 和 path 等字段，则可以通过使用 remove_filed 删除字段操作，最终提取的数据将不会出现这几个字段，其代码示例如下所示：

```
……
elseif[type]=="dandelion-probe"{
mutate{
remove_field=>["port","path","logger_name","host"]
}
}
……
```

完成对数据的处理后拟态裁决模块调用函数对数据进行比对，并根据多模裁决算法得出裁决结果。

动态调度模块接收裁决模块发送的调度请求，根据预设的动态调度算法进行相应的操

作。动态调度模块主要处理流程如图 5.7 所示，在接收到裁决模块的调度请求后，根据调度算法，调度模块选择将调用的执行体，并下发到异构体模块进行相应的处理，异构体模块的处理过程示例如下：

（1）docker 异构体：docker 异构体通过调用 docker 的 api，根据收到的调度请求进行相应的操作（创建、删除、关闭等操作）。

（2）kvm 异构体和 vbox 异构体：根据收到的调度请求，通过异构体管理服务类中对应的服务进行相应的操作（部署、开启、关闭、重启等操作）。

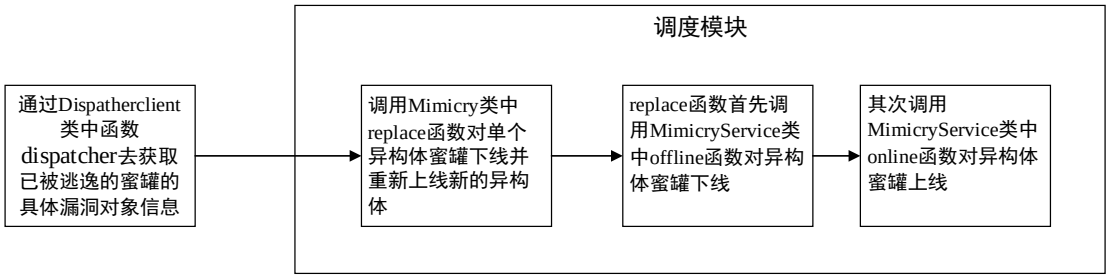

图 5.7 调度模块主要流程图

4. 节点管理

节点管理模块分为蜜罐管理、探针管理、拟态节点管理三个子模块。蜜罐管理主要是管理蜜罐的展示和基本操作，蜜罐的开启、重启、关闭、删除、Console 连接等；探针管理主要监控探针运行状态和对其进行基本操作；拟态节点管理主要是管理拟态业务节点与执行体节点的展示和基本操作，如执行体的重启、删除、部署、初始化及上下线等。

5. 蜜网管理

蜜网网络处理流程如图 5.8 所示，用户在管理端使用虚拟网络管理功能时，可以进行添加、删除、重启操作，在添加操作中针对节点可以选择 MACVLAN 和 NAT 模式创建不同模式的网络。填写参数后，参数将会存储在 MySQL 数据库中，并且前端会发送虚拟网络操作任务，后台接收到此任务以后，会将任务转发给此任务的关联节点，关联节点根据任务参数执行网络操作，此过程中后台系统通过调用不同的网络接口实现对网络的操作。

在网络欺骗模块中，用户在管理端填写相关参数，前端将参数数据存储到 MySQL 数据库当中，虚拟网络管理模块将从数据库读取这些参数，并在所关联的网络上定义相应的欺骗机制。

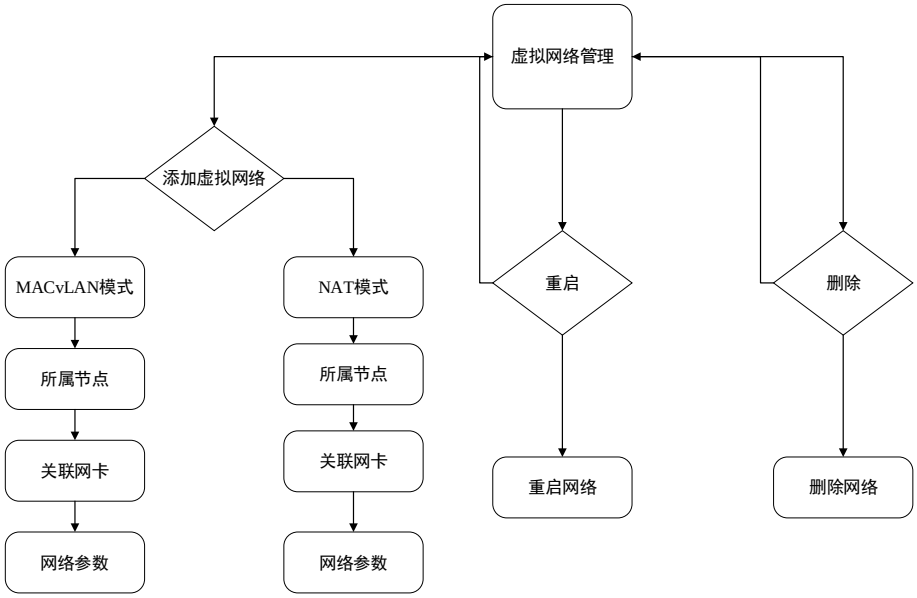

图 5.8 蜜网管理流程图

5.3 系统展示

蜜网系统基于 B/S 架构实现，系统在前端界面的导航栏中设置了三个模块，分别为态势展示，节点管理与蜜网管理，具体界面如图 5.9 所示。

图 5.9 功能模块展示

态势展示界面如图 5.10、图 5.11 所示，当管理员进入态势展示界面后，可按照本月、近半年、全年或自定义时间区间对指定时间段内的攻击情况进行展示。态势展示界面对攻击进行分类统计，将其分为高危、中危和低危三种类别分别计数，将攻击总数与各类别攻击数量分别进行展示，并根据攻击的发生时间绘制攻击事件趋势图。此外，对攻击还根据其攻击类型、攻击指向 IP、攻击来源 IP 与攻击指向的服务进行分类统计，并分别以扇形图、趋势图或计数排名的方式进行展示。同时对最新出现的攻击，界面在最下方进行滚动示警，便于管理人员查看。

态势展示界面将抽象、复杂的系统安全态势从多个维度分解，直观详尽地展现在系统管理人员面前，便于管理人员理解与掌握系统安全形势，并做出相应的调整。

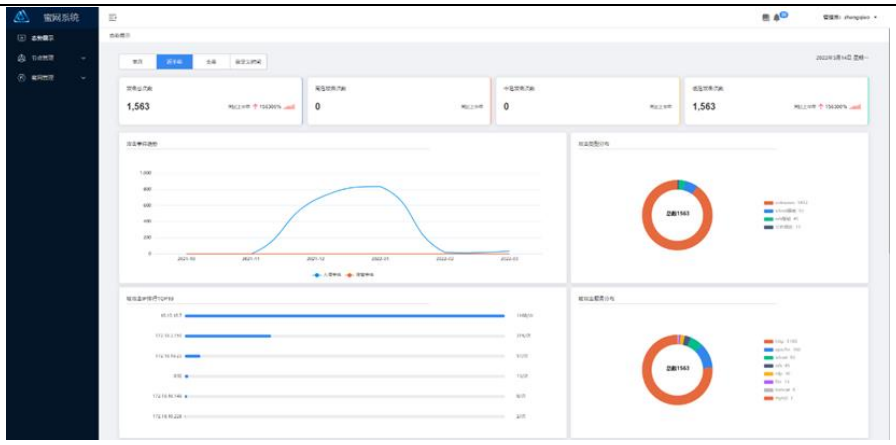

图 5.10 态势展示界面（a）

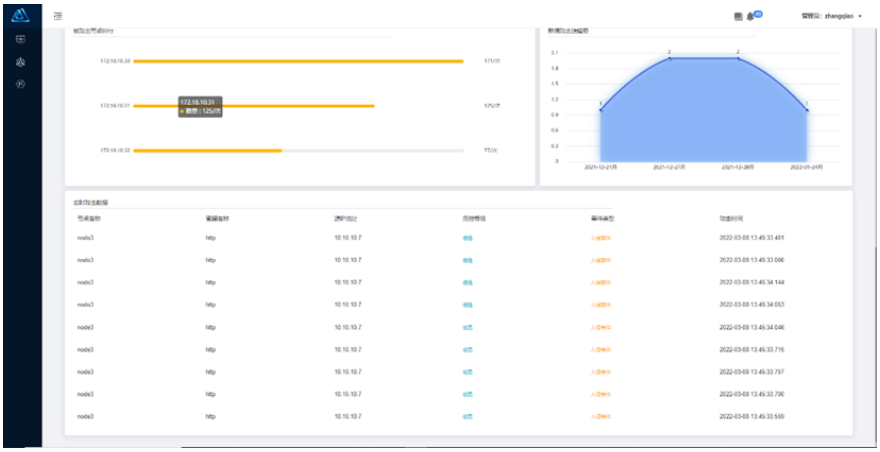

图 5.11 态势展示界面（b）

节点管理对蜜网中的各类节点进行统一管理，包括普通蜜罐、拟态节点、探针节点。节点管理中的普通蜜罐管理界面如图 5.12 所示。在普通蜜罐管理界面，管理员能够对蜜罐进行删除、重启、部署和初始化等操作，同时还能将进入该蜜罐的全部流量进行下载，以便进行分析和利用。

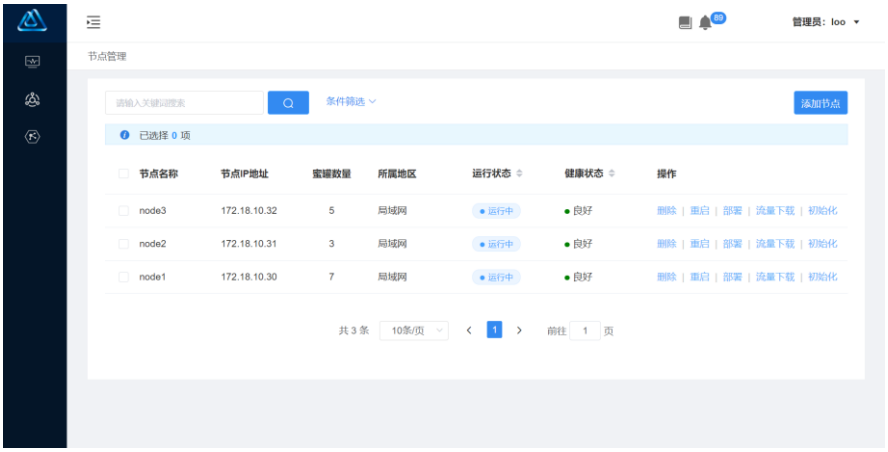

图 5.12 普通蜜罐管理界面

节点管理中的拟态节点管理如图 5.13 所示，为便于对拟态构造蜜罐进行管理，系统将拟态代理与拟态执行体均作为节点进行管理，界面显示各拟态节点的 IP 地址、运行状态与健康状态，管理人员在此界面还能够对拟态执行体进行清洗与部署操作，并能进入节点内部进行相关设置与调整。

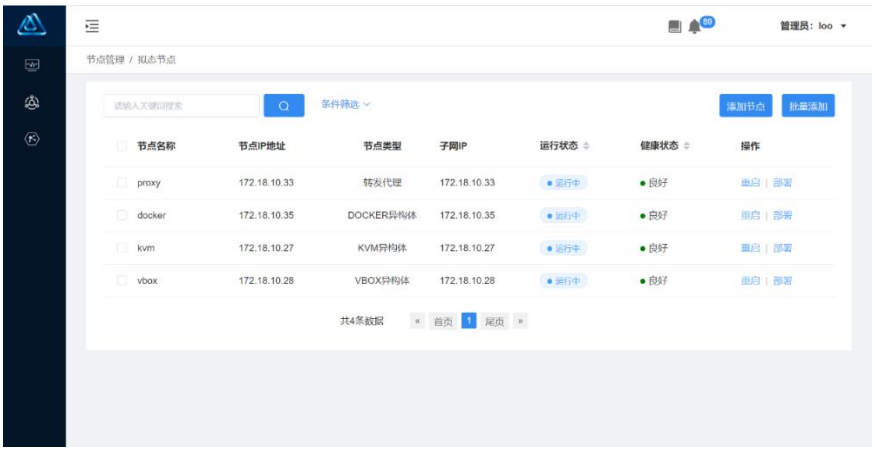

图 5.13 拟态节点管理界面

节点管理中的探针节点管理如图 5.14 所示，系统内置了三种探针，分别支持不同的系统与服务，能够提供对多种环境的探测，管理员可在本页面对探针进行管理与部署。

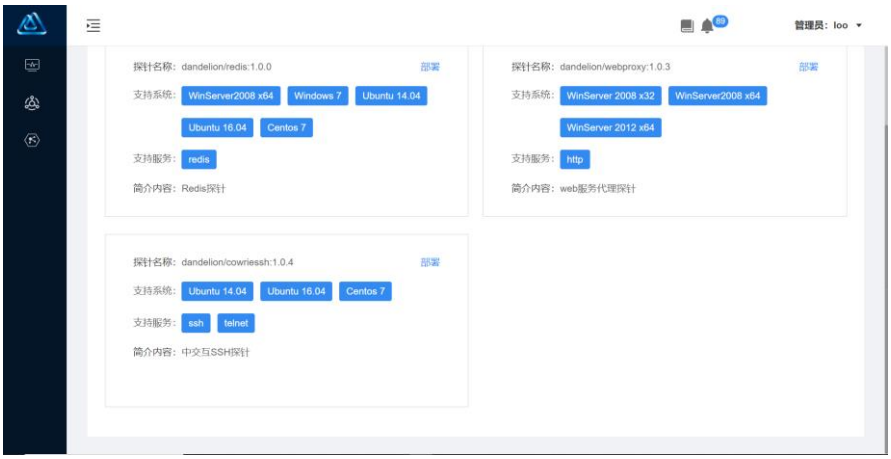

图 5.14 探针管理界面

蜜网管理中蜜网拓扑界面如图 5.15 所示，该页面为各类主机分配了不同的图标，并对当前蜜网拓扑进行了展示，以便管理人员直观掌握蜜网拓扑。

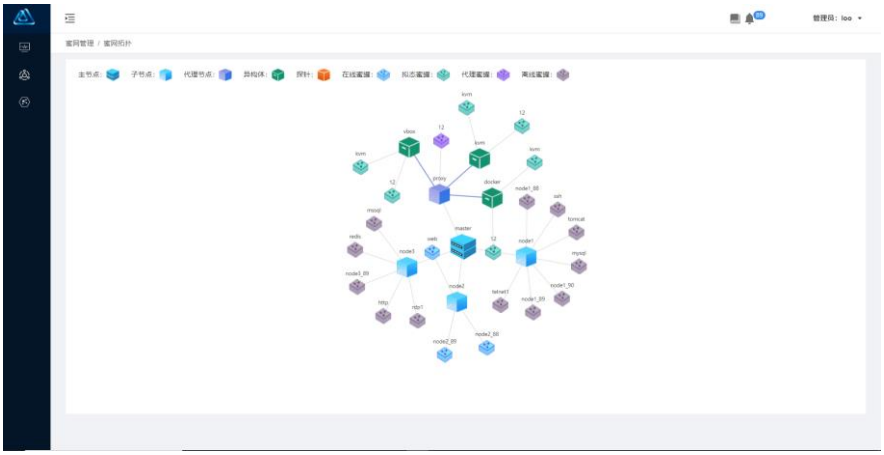

图 5.15 蜜网拓扑展示界面

网络欺骗管理界面如图 5.16 所示，管理员可以通过设置基本参数配置多个不同的欺骗方案，并将其与已有的蜜网网络相关联，系统将自动在指定网络中部署欺骗方案。

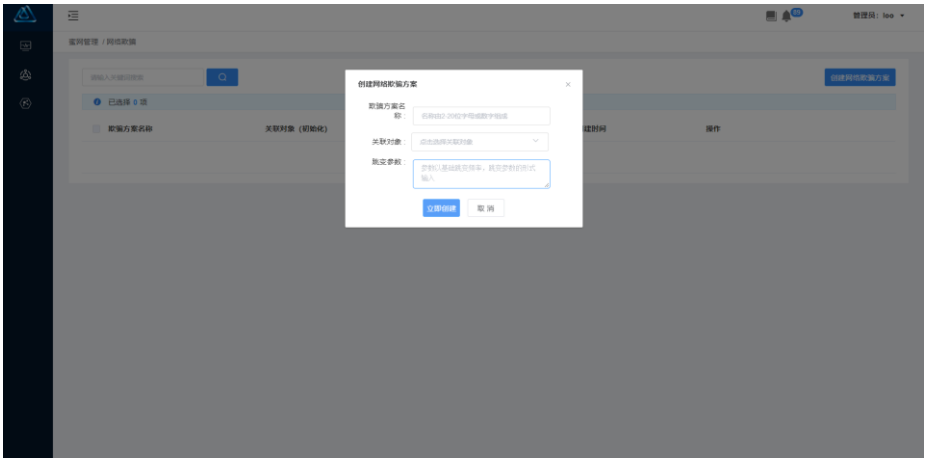

图 5.16 网络欺骗方案配置界面

虚拟网络管理模块如图 5.17 所示，管理员可以对已有网络进行修改或者配置新的网络，包括其所用网卡、所占用 IP 地址段与子网网关等。

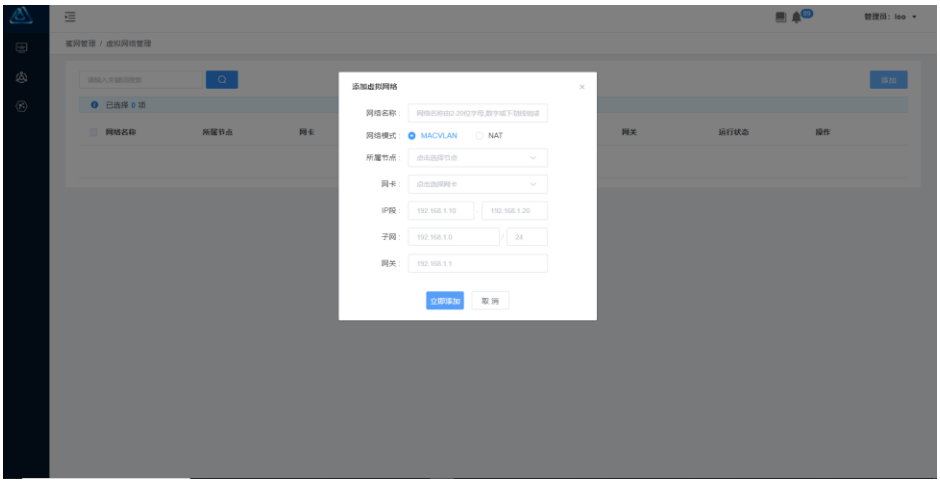

图 5.17 虚拟网络配置界面

5.4 系统测试

本节从功能与性能两方面出发，对蜜网系统进行测试，验证所实现系统的实用价值。

5.4.1 功能测试

功能测试中，蜜网内设置了五台主机，从局域网外的主机出发，使用 Nmap 扫描工具对蜜网进行了嗅探攻击，以检测蜜网的信息混淆与欺骗能力。

首先对未开启欺骗机制的蜜网进行拓扑扫描，其结果如图 5.18 所示，攻击者通过扫描即可轻易地发现蜜网内的主机及其拓扑关系，并据此展开下一步攻击。在此情况下，仅使用一种常见工具，攻击者就获得了关于网络的准确信息，而此时蜜网所能得到的恶意数据显然无法支撑对攻击者的刻画。

```
eve@kali:~$ sudo nmap -sn 192.168.20.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-01 21:49 CST
Nmap scan report for 192.168.20.2
Host is up (0.0034s latency).
MAC Address: 50:00:00:01:00:00 (Unknown)
Nmap scan report for 192.168.20.3
Host is up (0.0018s latency).
MAC Address: 50:00:00:07:00:00 (Unknown)
Nmap scan report for 192.168.20.4
Host is up (0.0033s latency).
MAC Address: 50:00:00:0D:00:01 (Unknown)
Nmap scan report for 192.168.20.254
Host is up (0.011s latency).
MAC Address: AA:BB:CC:00:60:10 (Unknown)
Nmap scan report for 192.168.20.1
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 28.00 seconds
```

图 5.18 普通蜜网主机发现扫描结果

在获取网络内主机列表后，攻击者可对任意主机继续发起侦察，获取主机开放端口、所提供服务等指纹信息。扫描结果如图 5. 19 所示，至此，攻击者通过两次 Nmap 扫描，即对系统完成了侦察，系统牺牲重要信息能够换取的恶意数据过于简单，对后续系统安全策略的动态调整帮助不大。

```
eve@kali:~$ sudo nmap 192.168.20.3 -sV
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-01 21:52 CST
Nmap scan report for 192.168.20.3
Host is up (0.0022s latency).
Not shown: 997 filtered ports
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup
: WORKGROUP)
MAC Address: 50:00:00:07:00:00 (Unknown)
Service Info: Host: ADMIN-PC; OS: Windows; CPE: cpe:/o:microsoft:windows
```

图 5.19 普通蜜网指纹信息扫描结果

接下来选取网络中的三台主机，对其应用网络欺骗机制，并再次对网络进行扫描，其结果如图 5. 20 所示，可以发现应用欺骗机制的主机 IP 已动态调整，无法通过简单的扫描进行获取，阻断了攻击者的攻击进程。此时若攻击者采用更为智能化的手段对网络进行

侦察，蜜网系统就得到了更具价值的恶意数据。

```
eve@kali:~$ sudo nmap -sn 192.168.20.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-01 21:54 CST
Nmap scan report for 192.168.20.254
Host is up (0.0035s latency).
MAC Address: AA:BB:CC:00:60:10 (Unknown)
Nmap scan report for 192.168.20.1
Host is up.
Nmap done: 256 IP addresses (2 hosts up) scanned in 32.51 seconds
```

图 5.20 本文蜜网扫描结果

其次对拟态构造蜜罐进行嗅探，分别从两个不同操作系统的攻击主机出发，对拟态构造蜜罐进行扫描，结果如图 5.21 与图 5.22 所示，对不同终端发出的扫描请求，拟态构造蜜罐给出了不同的回应，丰富了蜜网中虚假服务的种类，同时混淆了系统中的配置信息。

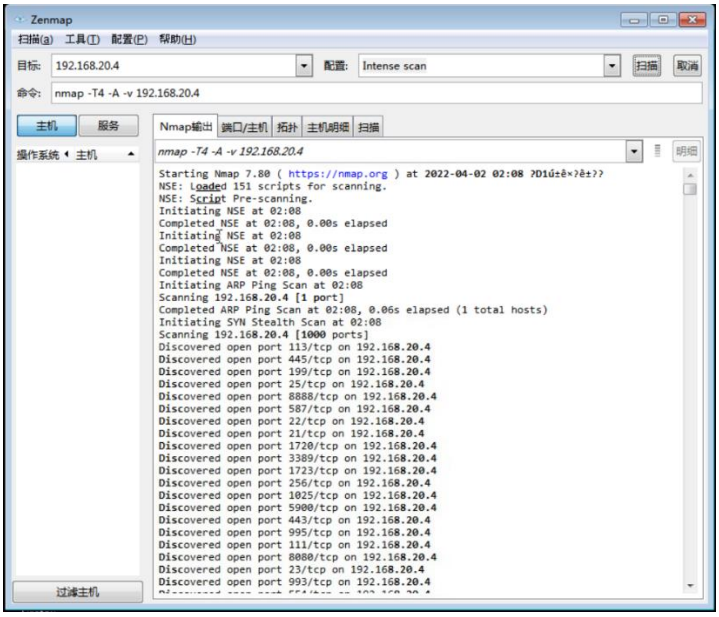

图 5.21 Windows 攻击机扫描结果

```
eve@kali:~$ sudo nmap 192.168.20.4
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-02 10:03 CST
Nmap scan report for 192.168.20.4
Host is up (0.0040s latency).
Not shown: 997 filtered ports
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 50:00:00:0D:00:01 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 17.73 seconds
```

图 5.22 Linux 攻击机扫描结果

5.4.2 性能测试

利用 SprientAvalanche 31000b 测试仪仿真大量 HTTP 请求分别发送至蜜网中各蜜罐服务器与真实业务网络中的各服务器，测试其新建速率、并发数、吞吐量和响应时间等性

能指标，结果见表 5.1。

表 5.1 HTTP 连接测试结果

测试对象	新建速率(/s)	并发数	吞吐量	平均响应时间
真实网络系统	4729	6784	104319	0.007
蜜网系统	3236	6627	102173	0.142

通过与真实业务网络的 HTTP 请求响应性能进行对比，可看出本文所实现的蜜网系统较好地完成了网络性能的伪装，其中，新建连接的速率降低了 29.44%，吞吐量和并发数仅降低了不到 1%，而平均响应时间尽管增幅较大，但仍属于正常范围，综合上述，本文所构建系统与真实业务网络网络性能的差距在可接受的范围内，能够较好地欺骗攻击者。同时该系统具有较强的网络请求处理能力，能够应对现实网络环境中的攻击。

5.5 本章小结

本章结合前述研究，对主动防御增强的蜜网系统进行了设计与实现。首先基于蜜网实现的要求对系统进行了需求分析。其次，对系统的总体结构与各功能模块进行了详细设计，并对其中较为关键机制的实现进行了详细介绍。最后，对各功能模块的界面进行了展示，并对系统进行了测试。

第六章 总结与展望

6.1 论文工作总结

网络技术的演进引发了一场生活方式的革命。网络设备的数量也随之呈指数级增长。然而，网络设备中或由于设计缺陷、或由于供应链问题，存在数量众多的未知漏洞后门，带来了诸多安全隐患。传统安全防御中基于特征的边界阻断与控制技术在网络攻防中处于被动状态，无法应对日益智能化、复杂化的攻击形势，推动了安全研究人员对主动防御技术的研究。蜜网就是为应对这种复杂多变的网络形势而提出的一种主动防御关键技术。蜜网通过配置虚假的数据、端口、服务和主机等网络资源构建欺骗环境，诱使攻击者与之进行交互，获取恶意数据以感知系统安全态势，提升系统防护能力。本文针对传统蜜网中基础设备与网络架构存在的缺陷，设计了拟态构造蜜罐与面向蜜网的网络欺骗机制。

本文的主要工作如下：

（1）首先对蜜网技术研究的背景进行了介绍，阐述了对蜜网设备进行改进的意义；然后对相关研究的现状进行了分析，针对当前存在的缺陷进行探讨，给出了高效可行的解决方案，最后对所涉及的技术与概念了介绍。

（2）提出了基于拟态思想的蜜罐，以提升蜜网系统中单个欺骗节点的安全性与诱骗能力。首先分析了当前蜜罐存在的两个问题，一是蜜罐高度依赖虚拟化技术，可能造成虚拟化逃逸，进而危及到整个系统，二是蜜罐仅能提供模拟服务的一种实现，导致其诱骗能力不足的问题；然后提出基于拟态 DHR 架构对蜜罐进行改进的方案，根据蜜罐设备的特点提出了双层裁决机制，随后阐述了拟态架构蜜罐的架构设计与功能模块的划分，并提出蜜罐场景下裁决与调度中需要考虑的特殊因素，据此分别提出了适用于拟态蜜罐的调度与裁决算法；最后，设计实验对拟态构造蜜罐进行了性能与安全性测试，测试结果表明，拟态构造蜜罐能够有效提升蜜网中欺骗节点的安全性与诱骗能力。

（3）提出了面向蜜网的网络欺骗机制，以提升蜜网系统整体的安全性与欺骗环境构建能力。首先分析了蜜网缺少动态性与配置不合理导致的欺骗环境吸引程度不足及易被攻击者识别的问题；针对此问题提出了面向蜜网的网络欺骗机制，根据蜜网结构与组成特点设计了不同类型节点进行不同速率 IP 地址跳变、基于真实主机进行蜜罐群配置的机制，并通过对蜜网攻防过程进行博弈建模证明了所提机制对蜜网的增益；然后基于 SDN 技术，设计了融合网络欺骗机制的蜜网，对蜜网结构与功能设计进行了详细描述；最后，该蜜网进行了可用性与欺骗性测试，证明了该机制能够增强蜜网系统的诱骗性与防御能力。

（4）在蜜网体系架构的基础上，基于前述两种主动防御设计，利用虚拟化技术，设

计并开发了主动防御增强的蜜网系统，给出蜜网的结构设计与模块划分，并对各模块功能设计与关键机制实现进行了详细说明，通过在实际网络环境中的测试验证，本文设计实现的蜜网系统有良好的应用价值。

6.2 未来工作展望

本文针对改进蜜网缺乏安全性与诱骗能力的缺陷展开研究，分别提出了基于拟态构造的蜜罐设计与面向蜜网整体结构的网络欺骗机制，在一定程度解决了这一问题，但仍存在许多问题需要进行优化与改进：

（1）本文设计的蜜网从基础节点到网络架构对蜜网进行了改造，实现了“由点及面”的安全性与诱骗能力提升，但对网络内其他安全设备未进行充分利用，后续考虑实现网络内全体安全设备的攻击联动与数据共享，实现对蜜网“由点及线到体”的全方位提升。

（2）本文所提网络欺骗机制中，各节点 IP 跳变频率与跳变参数的选择需要通过人工基于对攻击态势的理解进行手动配置，无法对网络安全状态做出全天候即时的应变，同时也难以取得安全性与成本间的最优平衡，后续考虑使用对攻防过程进行建模，求解获得各参数设置的最优解。

（3）本文中对恶意数据的运用尚有欠缺。由于蜜网特殊的设计机制，其所捕获的恶意数据往往不含有正常用户的访问数据，使用这些数据进行攻击溯源、攻击场景关联、攻击预判都有较好的效果。深度学习等人工智能技术在这些方面有着较好的表现，后续考虑将蜜网捕获数据使用深度学习技术进行分析与处理，并将分析结果用于增强系统安全性中。

致 谢

行笔至此，窗外又是落英缤纷的季节了。第一次来到这里时，同样是一个春天，怀揣着对研究生生活的憧憬与对军校的好奇，我踏入了这个美丽的校园，三年的时间如白驹过隙，转眼而逝，回头望去，所有的迷茫、挫折与艰难都在这一刻化为了美好。而这一切，都是因为这一路走来，得到了太多的指引、帮助与陪伴。在这告别的时刻，我想对大家道一声衷心的感谢！

首先，我要感谢我的恩师伊鹏研究员，伊老师不仅学术造诣深厚、治学态度严谨，且待人接物温厚宽和。学高为师，身正为范，在与伊老师的交流中，我学习到的远不止专业知识，更学到了为人处世的道理和脚踏实地的人生态度。与我而言，伊老师不仅是一位良师，更是一位益友，在人生中最关键的路口能够有这样一位老师指引我的方向，我将终生受益。一朝沐杏雨，终生念师恩！我会牢记您的教诲，更加努力奋进。

我还要感谢实验室的卜佑军老师。从进入项目组以来，卜老师就以高屋建瓴的视野引领我学习最前沿的技术，带领我开展工程实践。卜老师非常负责，每当我遇到各种各样的问题时，他总是不厌其烦地帮助我，是我前进路上的引航明灯。同时也要感谢陈博老师，从入学给我鼓励，到开题给我建议，再到平日里的指导，每一个环节您都给予我很大的帮助，正是这些帮助才让我克服了学习中的许多困难同时。最后，我也要对部门其他老师说一声谢谢，每一次组会时的指导，每一次生活上的关心，都温暖着我，也使我受益匪浅。

感谢胡涛、毛明、谢记超、任权、陆杰、李子勇、孙鹏浩、刘迪洋、崔子熙、张桥等各位师兄，优秀的人应当是什么样子，我想我从各位师兄身上学到了很多。感谢与我同窗十三年的老友李鑫，三座城市，辗转两千里，是缘分匪浅，更是心意相通。感谢孙嘉、张稣荣、赵扬、王亮、孙重鑫、陈仲磊、张德升等实验室伙伴，实验室里共同流过的汗水，学习之余的闲聊都将是我一生的回忆。感谢高远、白松浩、王兆辉、张明权、李柯、崔致远、黄赟、刘硕等同学，我们共同构建的美好回忆是我研究生生活的吉光片羽。

感谢我的父母，为了我的成长，你们付出了太多太多，是你们无私的爱培育了我，激励了我。感谢我的女朋友柴同学默默的付出，你的支持与鼓励陪伴着我度过了许多困难。

最后，感谢各位专家教授在百忙之中参加论文的评审，感谢您们提出的宝贵意见！

路祥雨

二〇二二年三月十五日于郑州

参考文献

[1] 刘剑,苏璞睿,杨珉,和亮,张源,朱雪阳,林惠民.软件与网络安全研究综述[J].软件学报,2018,29(01):42-68.DOI:10.13328/j.cnki.jos.005320.

[2] 中国互联网络信息中心《第 47 次中国互联网络发展状况统计报告》[EB/OL]. http://cnnic.cn/gywm/xwzx/rdxw/20172017_7084/202102/t20210203_71364.htm.

[3] 付钰, 李洪成, 吴晓平,等. 基于大数据分析的 APT 攻击检测研究综述[J]. 通信学报, 2015, 36(11):14.

[4] Spitzner L.The Honeynet Project:trapping the hackers[J].IEEE Security Privacy,2003,1（2）：15-23

[5] Uitto J , Rauti S , S Laurén, et al. A Survey on Anti-honeypot and Anti-introspection Methods[C]// World Conference on Information Systems & Technologies. Springer, Cham, 2017.

[6] 王航. 动态蜜网关键技术研究与实现[D]. 电子科技大学.

[7] 范伟. 虚拟机逃逸安全研究[J]. 保密科学技术, 2020(10):6.

[8] 程杰仁, 殷建平, 刘运,等 蜜罐及蜜网技术研究进展[J]. 计算机研究与发展, 2008(z1):4.

[9] Ramachandruni, Ram Sandesh, and Prabaharan Poornachandran."Detecting the network attack vectors on SCADA systems."[C]//2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2015

[10] M. A. Hakim, H. Aksu, A. S. Uluagac, and K. Akkaya, “U-pot: A honeypot framework for upnp-based iot devices,” [C]. 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC) IEEE, 2018, pp. 1–8.

[11] Li, X., Liang, X., Lu, R., Shen, X., Lin, X., Zhu, H.: Securing smart grid: cyber attacks, countermeasures, and challenges. IEEE Commun. Mag. 50(8), 38–45 (2012)

[12] KUWATLY I, SRAJ M, AL MASRI Z, et al. A dynamic honeypot design for intrusion detection[C]. The IEEE/ACS International Conference on Pervasive Services, Beirut, Lebanon, 2004: 95–104. doi: 10.1109/PERSER.2004. 1356776

[13] SAEEDI A, KHOTANLOU H, and NASSIRI M. A dynamic approach for honeypot management[J]. International Journal of Information, Security and Systems Management, 2012, 1(2): 104–109

[14] Gerard, Yang X , Qu L . On the Offense and Defense Game in the Network Honeypot[C]// 0.…

[15] SOCHOR T and ZUZCAK M. High-interaction linux honeypot architecture in recent perspective[C]. International Conference on Computer Networks, Brunow, Poland, 2016:118–131

[16] 石乐义,姜蓝蓝,刘昕,等. 拟态式蜜罐诱骗特性的博弈理论分析[J]. 电子与信息学报, 2013, 35(5): 1063–1068.

[17] The Honeynet Project[OL].http://www.honeynet.org/

[18] The HoneynetProject. Know Your Enemy: Honeynets [OL].http://old.honeynet.org/papers/honey-net/

[19] The Honeynet Project. Know Your Enemy: Genll Honeynets [OL].http://old.honeynet.org/papers/gen2/

[20] The Honeynet Project.Know Your Enemy:Honeywall CDROM Roo [OL].

[21] Spitzner L. Honeypot farms. [OL] 2012. http://www.symantec.com/connect/articles/honeypot-farms

[22] Yang HS. Honeypot Using Dynamic Allocation Technique with IP Scan[J]. Lecture Notes in Electrical Engineering, 2013, 2(15): 197-204

[23] Fan W，Fernández D，Du Z.Versatile virtual honeynet management framework[J].IET Information Security，2017，11（1）：38-45.

[24] 左青云, 陈鸣, 赵广松,等. 基于 OpenFlow 的 SDN 技术研究[J]. 软件学报, 2013, 24(5):20.

[25] Kyung S , Han W , Tiwari N , et al. HoneyProxy: Design and implementation of next-generation honeynet via SDN[C]// 2017 IEEE Conference on Communications and Network Security (CNS). IEEE, 2017.

[26] Achleitner S , Porta T F L , Mcdaniel P , et al. Deceiving Network Reconnaissance using SDN-based Virtual Topologies[J]. IEEE Transactions on Network & Service Management, 2017, PP(4):1-1.

[27] 罗兴国, 仝青, 张铮,等. 拟态防御技术[J]. 中国工程科学, 2016.

[28] Kenkre P S, Pai A, Colaco L. Real time intrusion detection and prevention system[C] // Proceedings of the 3rd international conference on frontiers of intelligent computing: Theory and applications (FICTA)2014. Switzerland: Springer International Publishing, 2015 (1): 405–411

[29] 张险峰, 张峰, 秦志光,等. 入侵容忍技术现状与发展[J]. 计算机科学, 2004, 31(10):5.

[30] 贾召鹏, 方滨兴, 刘潮歌,等. 网络欺骗技术综述[J]. 通信学报, 2017, 38(12):16.

[31] Wu J X. Mimic security defense in cyber space [J]. Secrecy Science and Technology, 2014, 10(1): 4–9.

[32] 马海龙, 伊鹏, 江逸茗,等. 基于动态异构冗余机制的路由器拟态防御体系结构[J]. 信息安全学报, 2017, 2(1):14.

[33] 仝青, 张铮, 张为华, 邬江兴. 拟态防御 Web 服务器设计与实现[J]. 软件学报, 2017, 28(4): 883-897.

[34] Ma D , Lei C . 网络空间欺骗：构筑欺骗防御的科学基石[M]. 2017.：1-16.

[35] BORDERS K, FALK L, PRAKASH A. OpenFire: using deception to reduce network attacks[C]//3rd International Conference o Security and Privacy in Communication Networks and Workshops. 2007: 224-233

[36] RRUSHI J L. NIC displays to thwart malware attacks mounted from within the OS[C]//Computers & Security. 2016: 6159-6171

[37] ANTONATOS S, AKRITIDIS P, MARKATOS E P, et al. Defending against hitlist worms using network address space randomization[J]. Computer Networks, 2007, 51(12): 3471-3490

[38] AL-SHAER E. Toward network configuration randomization for moving target defense[M]//Moving Target Defense. 2011: 153-159.

[39] ROBERTSON S, ALEXANDER S, MICALLEF J, et al. CINDAM: customized information networks for deception and attack mitigation[C]//IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems Workshops, Massachusetts Inst Technol. 2015: 114-119

[40] WANG K, CHEN X, ZHU Y. Random domain name and address mutation (RDAM) for thwarting reconnaissance attacks[J]. Plos One, 2017, 12(5): e0177111

[41] 扈红超, 陈福才, 王禛鹏. 拟态防御 DHR 模型若干问题探讨和性能评估[J]. 信息安全学报, 2016(4):12.

[42] Jajodia S, Ghosh A K, Swarup V, et al. Moving target defense: Creating asymmetric uncertainty for cyber threats [M]. New York: Springer,2011.

[43] Gupta V, Lam V, Ramasamy HG V, et al. Dependability and performance evaluation of intrusion-tolerant server architectures [M]. Berlin: Springer, 2003.

[44] Wang F, Jou F, Gong F, et al. SITAR: A scalable intrusion-tolerant architecture for distributed services [C]// Proceedings of the 2001 IEEE— Workshop on information assurance and security. New York: United States Military Academy, 2003.

[45] Malkhi D, Reiter M. Byzantine quorum systems [J]. Distributed Computing, 1998, 11(4): 203–213

[46] Kewley D L, Bouchard J F. DARPA information assurance program dynamic defense experiment summary [J]. IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans, 2001,31(4): 331–336

[47] Okhravi H, Hobson T, Bigelow D, et al. Finding focus in the blur of moving-target techniques [J]. IEEE Security & Privacy, 2014, 12(2): 16–26.

[48] Start McClure , Joel Scambray , George Kurtz .黑客大曝光[M]. 北京:清华大学出版社, 2013.

[49] Chen Z J, Qin X D, Gao J Y. Dissimilar redundant flight control computer system[J]. Acta Aeronautica Et Astronautica Sinica, 2005, 26(3): 320-327.

[50] 邬江兴. 网络空间拟态防御导论[M]. 北京: 科学出版社, 2017.

[51] Morctti S,Vasilakos A V. An overvicw of recent applications of Game Theory to bioinformatics[J].Information Seienees,20 10,180(22):43 12-4322

[52] Lyc K W,Wing J M.Game strategics in network sccurity[J].International Journal of Information Sccurity,2005,4(1-2):71-86

[53] Sankardas Roy D D,Game Theory for Cyber SccuritylC].Workshop on Cyber Sccurity &Information Intelligence Rescarch.ACM,2010,30-31

[54] Wang K,Du M,Maharjan S,ct al.Strategic Honcypot Game Model for Distributed Denial of Servi-ce Attacks in the Smant GridJ],IEEE Transactions on Smart Grid,2017:1-1.

[55] Lyc K W,Wing JM.Game stratcgics in network sccurity[J].International Journal of Information Sccurity,2005,4(1-2):71-86.

[56] Wang ZY, Yang XJ, Zhou Y. Scalable triple modular redundancy fault tolerance mechanism for MPI-oriented large scale parallel computing. Ruan Jian Xue Bao/Journal of Software, 2012,23(4):1022−1035 (in Chinese with English abstract)

[57] Scarfone K , Mell P . An analysis of CVSS version 2 vulnerability scoring[C]// Proceedings of the Third

International Symposium on Empirical Software Engineering and Measurement, ESEM 2009, October 15-16, 2009, Lake Buena Vista, Florida, USA. IEEE, 2009.

[58] Jafar Haadi Jafarian,Ehab Al-Shaer,Qi Duan. An Effective Address Mutation Approach for Disrupting Reconnaissance Attacks.[J]. IEEE Trans. Information Forensics and Security,2015,10(12).

[59] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. Nox: towards an operating system for networks. SIGCOMM Comput. Commun. Rev., 38(3):105–110, July 2008.

[60] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: enabling innovation in campus networks,” ACM SIGCOMM Computer Communication Review,vol. 38, no. 2, pp. 69–74, 2008.3

[61] J. H. Jafarian, E. Al-Shaer, and Q. Duan. Openflow random host mutation: transparent moving target defense using software defined networking. In Proceedings of ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN’12), pages 127–132. ACM, 2012