文章编号：2096-1472(2022)-05-60-03
DOI:10.19644/j.cnki.issn2096-1472.2022.005.014

基于Docker的蜜罐系统的设计与实现

黄成鑫，康晓凤，王 可，孙 典，茅璋瑞

(徐州工程学院信息工程学院，江苏 徐州 221000)
✉761369543@qq.com; kxfeng07@163.com; admi1n@163.com;
dsil3nce@foxmail.com; 1244529720@qq.com

摘 要：面对日益严峻的网络安全威胁，只有主动发现存在的漏洞和风险，才能最大限度地减少损失。然而主流的杀毒软件、防火墙等都属于被动防御技术，而蜜罐技术的出现弥补了之前技术的不足，变被动为主动。本文基于Docker和Python，设计实现了针对MySQL恶意攻击的蜜罐系统，系统包括MySQL服务器模拟模块、日志模块、捕获模块、管理模块和Docker模块。使用本系统可以捕获攻击者信息，追踪和分析黑客，达到主动防御网络威胁的目的。

关键词：主动防御；蜜罐；MySQL；网络安全
中图分类号：TP315 **文献标识码**：A

Design and Implementation of Honeypot System based on Docker

HUANG Chengxin, KANG Xiaofeng, WANG Ke, SUN Dian, MAO Zhangrui

(*College of Information Engineering*, *Xuzhou Institute of Technology*, *Xuzhou* 221000, *China*)
✉761369543@qq.com; kxfeng07@163.com; admi1n@163.com;
dsil3nce@foxmail.com; 1244529720@qq.com

Abstract: With the increasingly severe network security threats, only by proactively discovering existing vulnerabilities and risks can losses be minimized. However, mainstream anti-virus software, firewalls, etc. are passive defense technologies. The emergence of honeypot technology makes up for the shortcomings of this technology, changing passive defense to active defense. Based on Docker and Python, this paper proposes to design and implement a honeypot system for MySQL malicious attacks. The system includes MySQL server simulation module, log module, capture module, management module and Docker module. The system can capture attacker information, track and analyze hackers, achieving the purpose of actively defending against network threats.

Keywords: active defense; honeypot; MySQL; network security

1 引言(Introduction)

随着计算机技术与信息技术的发展和融合，计算机网络技术覆盖了人们生活的方方面面。计算机网络技术从20 世纪中叶发展至21 世纪，仅仅通过几十年的时间就极大地改变了人们的生活，这一点在信息交换上尤为明显。然而伴随着计算机网络技术的飞速发展，网络空间中愈演愈烈的安全威胁也不可忽视[1−2]。

本蜜罐系统使用Python语言开发，通过解析MySQL的数据包，模拟了正常的MySQL服务[3]，实现了MySQL的TCP/IP套接字连接过程中客户端和服务器的通信模拟。本系统封装在Docker容器中，能够更高效地利用系统资源[4]，一致的运行环境降低了交付和部署的难度，减少了用户对运行系统所需环境、版本的部署成本。同时，Docker特有的隔离性更满足了用户的安全需求。本系统与Web服务相结合，用户可在Web端查看当前服务器捕获到的网络攻击和反制的恶意攻击者的信息。

2 Flask介绍(Introduction to flask)

Flask是一个用Python语言编写的轻量级Web应用程序框

基金项目：2020年徐州工程学院大学生创新创业基金项目(2020047)；2021年徐州工程学院大学生创新训练计划项目(xcx2021322，xcx2021327).

架，比其他类似的框架更灵活，更安全，更易于使用。Flask结合MVC模式进行开发，可以根据自己的需要调用插件增加功能。Flask的基本方式是将view函数赋给URL。在程序中，用户每次访问URL，系统都会在URL上执行指定的视图函数，并获取该函数的返回值显示给浏览器，如图1所示。

图1 Flask流程图

Fig.1 Flask workflow

3 系统设计与实现(System design and implementation)

本系统核心模块基于Python的Socket模块，结合Flask框架模板渲染技术，实现蜜罐诱捕攻击和查看统计功能。服务器与用户进程之间采用TCP/IP套接字进行通信。系统可分为五大功能模块[5]：MySQL服务器模拟模块、日志模块、捕获模块、管理模块、Docker模块[6]。

3.1 MySQL服务器模拟模块

本模块是系统的核心模块，负责伪装成正常的MySQL服务器。使用MySQL数据库时，需要使用TCP/IP协议和MySQL服务器建立通信，在MySQL服务端与客户端通信的过程中，通过以下三个步骤实现认证。

第一步，初始化握手消息。服务端首先发送问候包Greeting，返回服务端的版本信息如Version、salt等[7]。

第二步，客户端向服务器发送登录认证消息。当账户密码正确时，服务器返回成功的身份验证包。在此过程中，蜜罐需要模拟出正常的MySQL服务器发送给客户端Greeting认证包，如图2所示。当攻击者尝试爆破用户密码时，会向服务器发送用户名密码字段。

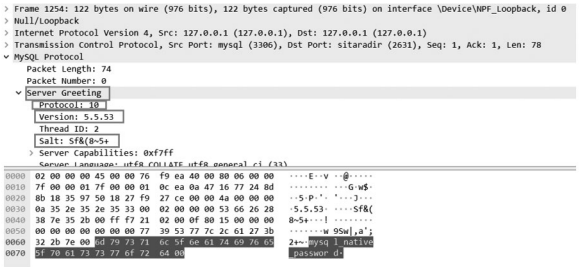

图2 服务器向客户端发送Greeting包

Fig.2 Greeting packet sent to clients by server

第三步，此时MySQL返回认证成功的消息，攻击者就会被误导成功爆破出MySQL服务器的用户名、密码。

客户端收到服务器发送的Greeting包后，会尝试发送账户密码进行验证。MySQL服务器收到发送的账户密码后进行验证，验证成功后向客户端发送认证成功包，如图3所示，发送用户名为root的验证包，服务器返回Response OK的返回包。

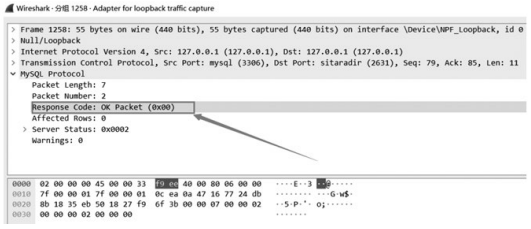

图3 服务器向客户端发送认证成功包

Fig.3 Authentication success packet sent to clients by server

第四步，身份验证成功后，客户端会向服务器发送查询包查看MySQL状态，Info为Request Query的数据包就是客户端向服务器发送的查询包，如图4所示，发送了三个查询包，此时即可准备反制攻击者。

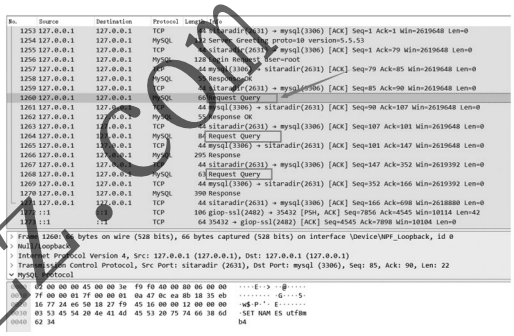

图4 客户端向服务器发送查询包

Fig.4 Query packet sent to clients by server

3.2 日志模块

本模块记录了攻击者的IP和尝试爆破MySQL密码的时间。日志保存在log文件夹下的attacker.log文件中，管理员可根据需要用日志审计工具生成攻击者画像。attacker.log文件格式如图5所示。

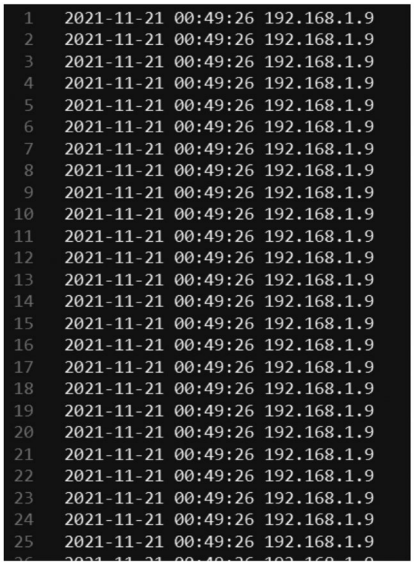

图5 包含时间和IP的攻击者日志

Fig.5 Attacker log with time and IP

3.3 捕获模块

本模块用于连接反制，当攻击者爆破到MySQL服务器的密码时，便会尝试用Navicat等工具进行连接。当攻击者尝试连接后，会发现无法连接。此时会调用MySQL的load data local infile '' into table test fields terminated by '\n'；语句。可以读取客户端本地文件并插进表中，向连接服务器的客户端发送读取文件的payload。读取文件流量包如图6所示。

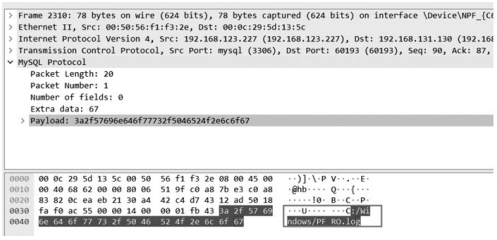

图6 读取文件流量包

Fig.6 Reading file traffic packets

此数据包是伪造的MySQL蜜罐向客户端发送读取C:\Windows\PFRO.log文件的数据请求。在能够实现任意文件读取的情况下，我们就可以读取攻击者PC上的文件，以此来捕获攻击者的画像。首先会读取C:\Windows\PFRO.log这个文件，然后返回结果并以Base64的加密格式保存在log文件夹下对应IP文件夹下的PFRO.log文件里，如图7所示。

图7 储存的PFRO.log文件内容

Fig.7 Stored PFRO.log file contents

当获取到了PFRO.log文件中的用户名之后，系统就会尝试拼接文件目录，读取微信、QQ、Chrome历史记录文件。同样，数据以Base64格式储存。

系统尝试获取C:\Users\username\Documents\WeChat Files\All Users\config\config.data文件内容，此文件中含有攻击者机器上登录过的微信号。C:\Users\username\AppData\Local\Google\Chrome\UserData\Default\History文件下缓存了用户上网的历史记录。通过获取这个文件，我们就有可能从中找到攻击者最常访问的网站或者一些特殊的站点。

3.4 管理模块

本模块是Web管理界面，可以查看捕获到的攻击者的信息。当蜜罐捕获攻击者的本地文件时，会储存到本地相应的路径下，通过正则匹配等方式将本地文件提取出来，从中识别出攻击者的信息，并呈现在Web界面上。Web端由Flask构建，默认端口为5000，这样就能在任意的环境下访问到管理模块。Flask采用模板渲染的方式呈现数据，可以动态地展示已捕获的攻击者信息。

本模块采用会话认证鉴定用户身份，用户需完成身份认证才可以使用本系统。本模块可以前端查看捕获到的微信、QQ、Chrome History等信息，如图8和图9所示。

图8 读取攻击者Chrome浏览器历史

Fig.8 Reading attacker's chrome browser history

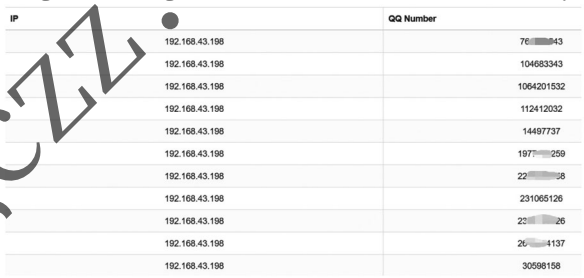

图9 读取攻击者QQ号

Fig.9 Reading attacker's QQ number

3.5 Docker模块

本模块负责整个系统的封装，旨在减少用户使用、移植的成本，方便用户部署，用户只需下载Docker软件并使用简单的命令即可启动Docker容器。运行docker-compose up -d命令，Docker会自行构建容器[8]。docker-compose.yml中记录了Docker的相关信息，包括Docker容器端口和服务器端口的映射情况，前一项是宿主机端口，后一项是Docker端口。如需改变映射端口，可以直接修改前一项的值，配置界面如图10所示。

```
# yaml 配置
version: '3'
services:
  web:
    build: .
    ports:
     - "5000:5000"
     - "3306:3306"

```

图10 Docker配置界面

Fig.10 Docker configuration interface

(下转第59页)