安全分析中网络节点属性挖掘技术研究

摘　要

计算机网络中所有计算机、设备均表现为一个节点，这个节点都有哪些安全分析相关的属性，如何通过主动探测的方法检测这个节点的安全性，是本文主要回答的问题。具体地，对网络空间中的节点进行画像——也就是安全分析中重要信息的搜集，以及安全性检测两个方面的原理和技术是本文研究的主要方向。基于以上原理与技术，本文设计并实现了主要应用于自动化渗透测试中的前渗透阶段的系统，以自动化主动探测的方式，实现了对计算机网络中给定节点的信息安全领域的信息搜集和安全性检测。

关键词：网络节点画像；网络节点安全性评估；自动化属性探测；主动探测

**A study of network node attribute mining techniques in security analysis**

Abstract

All computers and devices in a computer network are represented as a node. What are the attributes of this node related to security analysis and how to detect the security of this node by active detection are the main questions answered in this paper. Specifically, the principles and techniques of portraiture of nodes in cyberspace - that is, the collection of important information in security analysis - and security detection are the main directions of research in this paper. Based on the above principles and techniques, this paper designs and implements a system mainly applied in the pre-infiltration phase of automated penetration testing to automate active probing and achieve information collection and security detection in the field of information security of a given node in a computer network.

**Key words:**  Network node profiling; network node security assessment; automated attribute detection; proactive detection

目　　录

[摘　要 I](#_Toc74754263)

[Abstract II](#_Toc74754264)

[1.　绪论 1](#_Toc74754265)

[1.1　课题研究背景与意义 2](#_Toc74754266)

[1.2　论文的主要研究内容 4](#_Toc74754267)

[1.3　论文的组织结构 4](#_Toc74754268)

[2.　对网络空间指定节点的地理位置信息探测 6](#_Toc74754269)

[2.1　IP地理位置信息探测原理 6](#_Toc74754270)

[2.2　IP地理位置信息探测算法 6](#_Toc74754271)

[2.3　IP地理位置信息探测具体实现 6](#_Toc74754272)

[3.　对网络空间指定节点的开放端口探测 8](#_Toc74754273)

[3.1　开放端口探测原理 8](#_Toc74754274)

[3.1.1　TCP全连接扫描 8](#_Toc74754275)

[3.1.2　半连接扫描 8](#_Toc74754276)

[3.1.3　FIN扫描 8](#_Toc74754277)

[3.1.4　利用可能会导致系统错误的数据包扫描 8](#_Toc74754278)

[3.1.5　TCP ACK扫描 9](#_Toc74754279)

[3.1.6　TCP窗口扫描 9](#_Toc74754280)

[3.1.7　TCP Maimon扫描 9](#_Toc74754281)

[3.2　开放端口探测扫描具体实现 9](#_Toc74754282)

[4.　对网络空间指定节点的子域名扫描 10](#_Toc74754283)

[4.1　子域名扫描原理 10](#_Toc74754284)

[4.1.1　谷歌语法 10](#_Toc74754285)

[4.1.2　暴力枚举 10](#_Toc74754286)

[4.1.3　证书搜索 10](#_Toc74754287)

[4.1.4　DNS搜索 10](#_Toc74754288)

[4.2　子域名扫描自动化实现 11](#_Toc74754289)

[4.2.1　谷歌语法 11](#_Toc74754290)

[4.2.2　证书搜索 11](#_Toc74754291)

[4.2.3　子域名扫描工具实现 11](#_Toc74754292)

[5.　对网络空间指定节点域名的whois扫描 12](#_Toc74754293)

[5.1　定节点域名的whois扫描原理 12](#_Toc74754294)

[5.2　定节点域名的whois扫描实现 13](#_Toc74754295)

[6.　对网络空间指定节点的安全性检测 15](#_Toc74754296)

[6.1　基于特征值比对的漏扫底层原理 15](#_Toc74754297)

[6.2　基于特征值比对的漏扫实现 15](#_Toc74754298)

[7.　系统理论部分设计 18](#_Toc74754299)

[7.1　项目系统功能与实现目标 18](#_Toc74754300)

[7.2　项目节点画像部分 18](#_Toc74754301)

[7.2.1　描述画像类设计依据 18](#_Toc74754302)

[7.2.2　描述画像类设计 19](#_Toc74754303)

[7.3　项目节点的安全性分析部分 22](#_Toc74754304)

[7.3.1　描述画像类设计依据 22](#_Toc74754305)

[7.3.2　安全评估类定义 23](#_Toc74754306)

[8.　系统实现部分设计 27](#_Toc74754307)

[8.1　系统的总体架构 27](#_Toc74754308)

[8.2　系统的节点画像功能 27](#_Toc74754309)

[8.3　系统的节点安全性分析功能 28](#_Toc74754310)

[8.4　数据库实现功能 29](#_Toc74754311)

[8.4.1　数据库种类 29](#_Toc74754312)

[8.4.2　数据库使用情况 29](#_Toc74754313)

[8.4.3　数据库和表单设计 29](#_Toc74754314)

[8.5　命令行交互设计 33](#_Toc74754315)

[8.6　项目运行环境与所需工具接口 34](#_Toc74754316)

[8.6.1　项目开发与运行环境 34](#_Toc74754317)

[8.6.2　项目用到的辅助设计的开发者工具 34](#_Toc74754318)

[8.6.3　所需的python库 34](#_Toc74754319)

[8.7　结果展示与命令行交互 35](#_Toc74754320)

[9.　系统测试 37](#_Toc74754321)

[9.1　简单命令行交互式测试 37](#_Toc74754322)

[9.1.1　启动 37](#_Toc74754323)

[9.1.2 执行全部功能命令 37](#_Toc74754324)

[9.1.3　组合命令测试 38](#_Toc74754325)

[9.1.4　组合命令测试 39](#_Toc74754326)

[9.2　对节点的画像功能测试 40](#_Toc74754327)

[9.3 对节点的安全性分析功能测试 45](#_Toc74754328)

[10.　总结与展望 47](#_Toc74754329)

[10.1　总结 47](#_Toc74754330)

[10.2　展望 47](#_Toc74754331)

[参考文献 49](#_Toc74754332)

外文原文 50

外文翻译 80

# 1.　绪论

计算机网络上，所有的计算机、设备均表现为一个节点。该节点包含计算机、集线器、交换机或路由器等。至于这个节点到底是什么，具体通过哪些属性能够刻画这个节点的安全性，以及如何通过主动探测挖掘出这些属性，对于攻击方至关重要。该论文重点在于定义出一个属性分类，并提供相应的属性挖掘方法。

俗语言，“知己知彼，百战百胜”。计算机网络由节点和连接节点的链路组成，对于攻击型防御，攻击方主要对节点感兴趣。而面向节点的一系列研究都不得不基于对该节点的网络空间测绘中的画像。因此，对节点进行画像和安全性分析显得至关重要。

对于攻击型防御，目前，包括科研院所和各大公司在内的安全人员大多数是凭借经验，手工进行节点的主动探测，对节点的属性挖掘和安全分析并没有形成一个系统的方法论，因而可能会导致挖掘效率并不如人意，而且结果不够全面细致，安全性分析方面也没有比较明确的理论进行支撑。而本课题，主要致力于定义出一个属性分类，并提供相应的属性挖掘方法，为其提供较为明确的方法论支撑。

国外在该领域的研究，较为著名的有号称黑暗搜索引擎的美国J.Matherly首次发布的Shadan[1]网络设备搜索引擎。随后密西根 v大学的安全团队推出了Censys[2]搜索引擎。Censys结合设备识别工具Ztag来识别网络空间中的工业控制系统以及运行Modbus工业控制协议的设备[3]。而国内最早启动这一领域的科研项目并且做得比较出色的有钟馗之眼（Zoomeye）（https://www.zoomeye.org）项目，据该项目方称，该项目可以搜索包括控工设备在内的全球设备情况。

网络空间中节点属性挖掘的相关技术研究已有十几年的发展历史。网络空间中节点属性挖掘主要涉及两个关键技术，探测技术和识别技术。研究工作[4]总结了近20年探测技术的发展历史，分析了探测技术面临的问题和挑战。Nmap[5]是最早的网络探测工具，通过发送探测包和收集响应包，完成网络空间探测任务。Nmap的探测范围比较全面，但探测速度较慢。近年来，密西根大学研究团队提出的Zmap[6]极大地提高了网络探测速度，可以在数十分钟内完成40亿的IPv4地址空间的探测。Zmap的探测速率是Nmap探测速率的数千倍。Xuan[7]提出了ARE原型系统，在Zmap搜索的结果上，自动生成物联网信息。J.Richard[8]的综述工作总结了探测技术在网络安全检测方面的应用和发展。识别技术是基于探测技术收集的数据，提取物联网相关信息，例如操作系统、应用服务、物理设备种类和标识信息。研究人员Zain[9]提出了利用TCP数据包重传超时时延来识别网络空间中的操作系统版本信息，T.Kohno[10]提出了利用设备的时钟偏移提取物联网设备的标识信息。典型的标识方法包括：基于IP地址的标识，基于MAC地址的标识和基于Cookie的标识。标识信息可以用来发现和检测网络空间的异常行为。研究人员D.Formby[11]发现真实的和伪造的工业控制设备的响应包在时间上具有差异性。目前国内外对网络空间中节点属性挖掘的相关技术基本成熟，但是将网络中节点属性挖掘和渗透测试相结合的自动化实现却鲜有成果。网络空间资产测绘相关平台都只是简单实现了对网络空间资产的大量扫描，然而在渗透测试中，需要的不是大量扫描而是敏感而精确的信息，过量的垃圾信息或者是大量的冗余信息只会给渗透测试人员带来不必要的烦恼。本论文将二者结合，使其成为渗透测试中前渗透信息搜集的一个自动化辅助工具。

## 1.1　课题研究背景与意义

国际自动计划和调度会议(ICAPS)早在2015年就提出了未来的渗透测试将会实现自动化。信息安全领域将会成为自动化为主，手工为辅的领域。国际上一些比较领先的科技公司已经初步实现了安全岗位的部分自动化，甚至有公司已经在测试全自动化渗透测试，可见自动化渗透测试是大势所趋。

境外高强度有组织持续性黑客攻击，即apt攻击，对中国的威胁渐增，攻击手法日渐复杂而成熟，使得中国的信息安全领域朝着国际大趋势的进一步发展成为必要。近十年，对中国采取定向有组织高强度持续性的黑客攻击（简称APT攻击）的网络间谍组织日渐猖獗。2015年12月31日，国内首次曝光某国外黑客组织“DarkHotel”针对中国企业的APT攻击。2018年360追日团队首次披露境外网络间谍组织“蓝宝菇”，其对中国大陆进行了长达8年的持续性高强度攻击。“蓝宝菇”组织主要窃取核科研和科研等相关信息，据目前所获情报的统计，该组织对中国教育科研机构攻击目标占比最大，高达59.1%；政府机构位居第二，其占比为18.2%；而后是占比9.1%的国防机构；随后是其他事业单位、金融机构和制造业等。据目前所获情报，“蓝宝菇”第一个木马于2011年被打入中国大陆。2019年，国家安全机关发现并且处置的境外apt组织数量竟然高达近百个。其中某特定组织针对我国“两会”、“一带一路”等经济政治重要峰会以及“中国成立70周年”等重大活动全年累计进行了4000多次的定向攻击。像这样将中国的重点科研机构、政府、国防、企业乃至个人的网络空间视为刀俎上的鱼肉的国际黑客组织不在少数。这些例子无不说明国内信息安全渗透测试领域研究的重要性和迫切性。

而放眼国内，信息安全领域的研究起步比较晚，第一批信息安全专业也是近10年左右才开始开设，而且目前国内大部分信安从业者都并不是有着比较完整计算机知识理论体系，有着较为扎实的代码工程能力的高校“院系派”，而是中专等职校培养出来的或者是靠自学无师自通的对理论和程序驾驭能力欠缺的“野生派”。导致国内大部分黑客停留在脱离原理的工具利用和、手工检测和简单工具开发层面，实现更加贴合国际发展趋势的自动化渗透测试路漫漫其修远兮。实现国内的渗透测试领域从手工向自动化转型，与境外针对中国的网络间谍组织抗衡迫在眉睫。

在大数据信息爆炸的时代，近十年漏洞的数量也逐渐成爆炸式增长，而且常见漏洞类型逐渐增加，漏洞在逐渐实现计算机整个体系结构的全方位覆盖。近十年，随着国内外黑客圈子的扩展和黑客技术的提升，爆出来的漏洞数量以一种令人咋舌的增长趋势日渐增加，漏洞覆盖面也越来越广。从数量上分析，光是分析由美国国土安全资助的MITRE公司负责维护的通用漏洞披露库（CVE），国家信息安全漏洞共享平台（CNVD）这两个知名漏洞库近几年的收录情况，就会发现漏洞库逐渐呈现信息爆炸趋势。更不用说，还有安全行业相关从业者向各大公司自己的应急响应平台（src）定向提交的漏洞，和一些类似于面向全世界黑客的漏洞提交平台（exploit-db）、国家信息安全漏洞库（CNNVD）等大型漏洞库平台。据统计，CNVD2019 年收录安全漏洞数量成为历史最高点，此年收录的安全漏洞数量同比增长了 14.0%，总数高达16,193 个，自2013 年以来，每年平均增长率为 12.7%。然而更令人担忧的是，2020年，CNVD共收录通用软硬件漏洞19964个，与2019 年16,193 个相比是有过之而无不及。而且，近五年来，没有可用的补丁程序被业界视为“能够造成最大的风险”的负责应用程序的程序员或供应商所未知的软件缺陷安全漏洞（0day漏洞）如CNVD等各大知名漏洞平台收录数目不断上涨，CNVD的0day漏洞年平均增长率达46.9%。据360最新年报显示，一些曾经比较罕见的漏洞如操作系统级别和协议层级别的漏洞数量也开始恣意增长。在这样一个大环境下，光凭渗透测试人员的经验和手工探测，仅仅是完成对现有漏洞的特征值的比对，都是不可能的任务。因此，利用代码实现渗透测试的自动化是未来渗透测试的必要环节，也是大势所趋。

## 1.2　论文的主要研究内容

根据PTES-渗透测试执行标准，渗透测试一般包括前期交互、情报搜集、威胁建模、漏洞分析、漏洞攻击、后渗透和报告撰写七个部分。

前期交互阶段的主要任务有，确定范围，目标规划和建立通讯渠道等。其中确定范围环节包括对整个项目的时间估计，确定IP和域名的范围等。情报搜集，包括目标的选择，以及开放渠道情报搜集，白盒搜集，踩点和识别防御机制。威胁建模阶段是利用上一阶段获取的信息进行威胁建模和攻击规划。漏洞分析是发现系统和应用程序中漏洞的过程。渗透攻击是利用发现的验证存在的漏洞进行攻击的过程。后渗透顾名思义就是渗透攻击成功拿到系统的权限后的后续操作，包括线权的维持和内网渗透。报告阶段是渗透测试最后一步，一份好的报告要向客户传达测试的目标方法和结果。

本文主要研究在前渗透阶段，具体而言，是前渗透的情报搜集阶段的自动化实现。以主动探测中的节点基本画像信息，即能说明该节点的作为该实现的切入点，以点带动面，通过保留其扩展性，让日后通过完善实现渗透测试全自动化成为可能。系统的设计一直考虑以后的扩展性和扩展以后的安全性。

## 1.3　论文的组织结构

第一章为绪论，主要介绍了国内外相关的研究背景，本文研究内容和所设计的系统的作用，重要性和意义，要研究的重点内容及论文的组织结构。

第二章为对网络空间指定节点的地理位置信息探测原理，最后给出了本系统对于该功能的python自动化实现。

第三章为对网络空间指定节点的开放端口探测原理，最后给出该功能的自动化实现。

第四章为网络空间指定节点的子域名扫描原理，最后给出该功能的自动化实现。

第五章为对网络空间指定节点的域名whois扫描原理，最后给出该功能的python自动化实现。

第六章为对网络空间指定节点的安全性扫描原理，和相关技术介绍，最后给出该功能的python自动化实现。

第七章为整个系统的理论部分的设计和理论分析。

第八章给出了本系统的具体实现，包括设计方法和流程图，设计语言为python3。

第九章为系统测试，本章对此系统进行了所有已开发功能的测试，并结合测试进行了简单的分析。

第十章为总结与展望，本章对系统的功能进行了总结，并分析了目前系统的不足，提出了展望。

# 2.　对网络空间指定节点的地理位置信息探测

对网络空间指定节点的地理位置信息探测在本文中主要研究IP定位，IP定位的基本过程就是通过设备的IP地址测量获得其属性信息,在分析属性信息的基础上获得IP设备的地理位置。这个属性信息既可以是主机的名字、IP地址本身，也可以是IP设备与已知定位位置IP设备的时延以及连接关系等[13]。

## 2.1　IP地理位置信息探测原理

IP 定位的基本原理是，利用IP设备的名字、注册信息或时延信息等来估计其地理位置.定位算法设计的基本原则是：在保证定位精度的前提下,尽量减少测量开销，同时兼具良好的扩展性，并能保护用户隐私。最初的定位算法通过向DNS服务器查询或者挖掘隐含在主机名中的信息来推测IP设备的地理位置。之后，一些定位算法根据时延与地理距离之间的线性关系来估测主机位置，并通过拓扑信息来减小定位误差。

## 2.2　IP地理位置信息探测算法

IP地理位置信息探测常见算法可以分为独立于客户端的定位算法，基于客户端的定位算法两种。

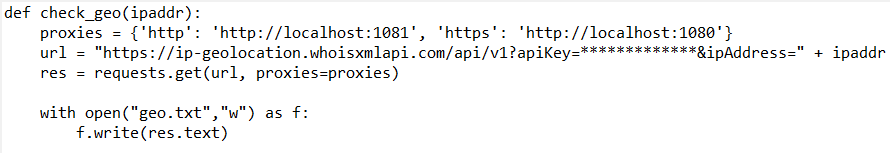
其中独立于客户端的定位算法又包括基于推测的定位算法、基于时延的定位算法和综合定位算法三种。基于推测的定位算法一般通过查询Whois数据库等来获取该IP地址的主机名,所在街道或者通过IP地址段地理位置来推测该IP设备的位置。基于空间理论的定位算法利用时延与地理距离的线性映射关系来估计地理距离的远近。综合定位算法是结合多种定位算法而对主机进行准确定位的算法。

基于客户端的定位算法需要在主机或者设备上安装额外的定位装置，将定位装置提供的位置信息作为主机或设备的位置信息报告给定位系统。一般地，这些系统将GPS、WiFi、蜂窝基站、Zigbee、射频识别、线性调频扩频技术(chirp spread spectrum,简称 CSS)等无线定位系统作为信源。

## 2.3　IP地理位置信息探测具体实现

本系统借用了whoisXMLAPI中给出的https接口，对指定的IP进行地理位置定位查询，具体的实现是程序主动向指定的网址带上指定的apiKey参数和其他包括返回数据格式类型等可选参数发出get请求。

**代码2-3 python实现自动化IP定位**



# 3.　对网络空间指定节点的开放端口探测

在渗透测试中，每一个开放的端口都是一个潜在的攻击通道。倘若能知道目标的开放的端口，就好比找到了一个房子的门窗，闯入这个房子的难度系数书瞬间从穿墙挖洞的难度变成了打破门窗的难度。

## 3.1　开放端口探测原理

尝试与目标主机建立连接，如果目标主机有回复则说明端口开放。以下是几个常用的扫描方式的原理介绍。

### 3.1.1　TCP全连接扫描

建立完整的TCP连接，从而判断该端口是否开放。因为是经历完整的三次握手的连接，准确度是所有扫描方式里面最高的一种。但是这也是最不隐蔽的一种，目标通过审计等方式可以简单判断出来这个端口被扫描过，甚至在扫描的时候就会立刻被发现。而且，在防火墙开启的情况下，会有一些不可靠的回复报文，妨碍对扫描结果的判断。这个扫描不需要root权限。

### 3.1.2　半连接扫描

这种扫描方式称为半开放扫描，不需要建立完整的tcp连接，只发一个syn数据包，也就是第一个tcp连接握手的包，如果端口开放，则会返回syn和ack的数据包，而如果端口未开放，则会返回rst数据包，连接的建立到此为止。这样我们就不用继续进行tcp的第三步握手就能知道端口目前的状态了。这种扫描需要构建raw socket，所以需要root权限。

### 3.1.3　FIN扫描

有些时候防火墙会过滤syn数据包，这时候可以使用fin扫描的方法，也就是发送fin数据包到待检测端口，如果返回rst数据包，则说明端口关闭，如果无返回则说明端口开放，这是tcp协议的一个bug，比一定完全可靠，比如window，但是大部分UNIX系统都适用。

### 3.1.4　利用可能会导致系统错误的数据包扫描

当系统接收到，可能会导致错误的TCP数据包的时候，关闭的端口与开放的端口响应情况不太一样。一个开放的端口接收到可能导致系统错误的数据包的时候，会选择丢弃或不做处理，但是如果是一个关闭的端口，作为返回一个rst。此类扫描的实现一共有三种方式，第一种方式是TCP flag headers全部设置为0，还有一种方式是设置tcp FIN标志，最后一种方式是同时设置fin、tsh和urg标志位。

### 3.1.5　TCP ACK扫描

这种扫描主要是用于判断端口有没有被防火墙屏蔽。扫描过程是向段都发送ACK数据包，如果防火墙没有把该端口屏蔽则会返回一个RST数据包，如果被防火墙端口被防火墙屏蔽则会不作任何响应。

### 3.1.6　TCP窗口扫描

在某些操作系统，开放的端口会返回一个正数的tcp窗口值，倘若收到一个非正数的窗口值，可以判定该端口关闭。这个方法不太具有通用性，因此不常用。

### 3.1.7　TCP Maimon扫描

构造一个同时设置FIN/ACK的数据包并发送，通过判断返回RST还是无响应来判断端口开放情况，前者是端口开放的情况，后者是端口关闭的情况。

## 3.2　开放端口探测扫描具体实现

该设计调用了nmap的python接口。Nmap可以直接根据各种规则和原理构造数据包，然后对端口进行基于不同扫描类型的端口探测。以下是几个常用的扫描类型：TCP全连接扫描命令行用法为使用“-sT”参数，或者不带参数；UDP扫描则使用“-sU”参数；利用可能会导致系统错误的数据包扫描则使用“-sF”、“-sN”和“-sX”参数；TCP ACK扫描则是使用“-sA”参数；TCP窗口扫描则是使用“-sW”参数；TCP Maimon扫描则是使用“-sM”参数。

在本设计中，扫描的实现是直接调用了nmap的扫描python接口。在本设计中的源码中直接引入“nmap”库，使用该库中的PortScanner（）方法生成扫描实例，然后进行指定ip和扫描方法的扫描。

# 4.　对网络空间指定节点的子域名扫描

子域名收集是渗透测试中信息搜集步骤常用的一种方法，可以帮助我们发现更多的服务尤其是在挖是SRC漏洞的时候，因为是针对某一个组织的漏洞挖掘，所以子域名信息相当重要。

## 4.1　子域名扫描原理

### 4.1.1　谷歌语法

谷歌语法搜索之所以会在百度、必应、雅虎等一系列搜索引擎面前成为这个搜索引擎的一个独有的特点，不是因为它独有这个功能，是因为独独只有它这个功能的影响力和效果最强大。道其原因主要有四：支持上百种常用语言搜索；这个一个将近1万台服务器，几百条T3级别的带宽撑起来的高搜索速度；部分功能实现自动化，可以提供符合度比较高的网站；拥有专利级别pagerank算法，使得搜索结果符合度很高。

谷歌搜索语法分为基本语法和高级语法，在指定节点的子域名扫描中主要用到的是高级语法，比如想要搜素百度的子域名，可以通过“site:baidu.com”进行，搜索结果是包括所有包含“baidu.com”域名的有关内容，不管是顶级域名还是二级、三级等其他域名都尽收罗网。

### 4.1.2　暴力枚举

暴力枚举扫描子域名一般是对于有可能包含通配符的给定的字典进行可能的子域名的枚举，然后尝试对该域名进行解析，最后对域名进行去重的一个过程。

### 4.1.3　证书搜索

根据证书透明的证书签发流程，证书必须要记录到可公开验证不可篡改的地方——这个地方被称为CT日志，否则用户的网络浏览器将其视为无效，任何感兴趣相关方可以查看由授权中心签发的所有证书。因为SSL证书支持证书透明度，而SSL里包含子域名。所以可以从SSL证书收集子域名。

### 4.1.4　DNS搜索

这是一种基于DNS记录查询子域名的方法，枚举DNS 记录和子域。

## 4.2　子域名扫描自动化实现

### 4.2.1　谷歌语法

在自动化中很少使用谷歌语法的方法实现，但是如果对这个方法情有独钟，可以结合爬虫的实现原理，通过修改request包中对应的url中的参数，然后抓取返回结果来实现。

### 4.2.2　证书搜索

本设计的python实现中调用了urllib.request、urllib.parse、re和ssl四个库，然后基于python的伪代码如下：

def 基于证书子域名查询(子域名参数):

    with 以爬虫的形式打开('https://crt.sh/?q=' + 子域名参数) as f:

        for 证书, 子域名 in 正则查找全部('<tr>(?:\s|\S)\*?href="\?id=([0-9]+?)"(?:\s|\S)\*?<td>([\*\_a-zA-Z0-9.-]+?\.' + re.escape(domains) + ')</td>(?:\s|\S)\*?</tr>', code, re.IGNORECASE):

            子域名 = 子域名.split('@')[-1]

            with 打开一个被追加的保存结果的文件 as f:

                f.write(子域名)

### 4.2.3　子域名扫描工具实现

子域名扫描的工具有sublist3r，子域名挖掘机layer，fierce。程序可以选择直接调用站长之家的API和whoisxmlAPI工具提供的API，该API是通过http请求的方式进行交互，所以对于开发语言几乎没有限制，要说有限制，那就是该语言一定要有网络功能。

def check\_subdomains(ipaddr):

    根据文档构造url

    res = requests.get(url, 正确设置代理的参数)

    #以下假设返回结果以json形式显示

    res\_js = 以json形式加载(返回结果res)

    num = 取出返回结果数

    lis = 取出返回的子域名并存储成列表

    return lis

# 5.　对网络空间指定节点域名的whois扫描

在渗透测试中，whois扫描是从网站域名入手获取网站拥有者信息的一种方式，通过它能知道域名所有者的个人信息，在渗透测试中，和网络安全的溯源领域用处非常大。

## 5.1　定节点域名的whois扫描原理

WHOIS协议是一个互联网信息查询协议。其基本内容是，先向服务器建立一个TCP连接，发送内容为查询关键字并加上回车换行，然后接收服务器的查询结果。每个网站在正式发布之前，都必须向有关机构申请自己的域名，申请到的域名将会保存到域名管理机构的专用服务器中，同时，域名服务器内的这些域名服务器信息是公开的，任何人都可以访问查询。因此，这个域名服务器内的信息就会暴露给攻击者很多敏感信息，如注册人姓名、注册人姓名、注册机构、通讯地址、邮编、注册有效时间、失效时间。这些敏感信息可以从域名管理机构的专用服务器中获取，常用的重要管理机构及whois服务器域名如表4-1所示。

表5-1　重要Internet管理机构和常用whois服务器

|  |  |  |  |
| --- | --- | --- | --- |
| **机构缩写** | **Whois服务器地址** | **机构全名及地点** | **提供查询内容** |
| CERNIC | whois.edu.cn | 中国教育与科研计算机网网络信息中心  (清华大学·中国北京) | 中国教育网内的IP地址和.edu.cn域名信息 |
| CNNIC | whois.cnnic.net.cn | 中国互联网络信息中心  (中国科学院计算机网络信息中心·中国北京) | .cn域名(除.edu.cn)信息 |
| INTERNIC | whois.internic.net | [互联网络信息中心](http://www.internic.net/)  (美国洛杉矶市Marina del Rey镇) | .com,.net,.org,.biz,.info,.name  域名的注册信息(只给出注册代理公司) |
| ARIN | whois.arin.net | [美国Internet号码注册中心](http://www.arin.net/)  (美国弗吉尼亚州Chantilly市) | 全世界早期[网络](http://net.it168.com/" \t "_blank" \o "网络)及现在的美国、加拿大、撒哈拉沙漠以南非洲的IP地址信息 |
| APNIC | whois.apnic.net | [亚洲与太平洋地区网络信息中心](http://www.apnic.net/)  (澳大利亚昆士兰州密尔顿镇) | 东亚(包括中国大陆和台湾)、南亚、大洋洲IP地址注信息 |
| RIPE | whois.ripe.net | [欧州IP地址注册中心](http://www.ripe.net/)  (荷兰阿姆斯特丹) | 欧洲、北非、西亚地区的IP地址信息 |
| TWNIC | whois.twnic.net | [台湾互联网络信息中心](http://www.twnic.net/)  (中国台湾台北) | .tw域名和部分台湾岛内IP地址信息 |
| JPNIC | whois.nic.ad.jp | [日本互联网络信息中心](http://www.nic.ad.jp/)  (日本东京) | .jp域名和日本境内的IP地址信息 |
| KRNIC | whois.krnic.net | [韩国互联网络信息中心](http://www.krnic.net/)  (韩国汉城) | .kr域名和韩国境内的IP地址信息 |
| LACNIC | whois.lacnic.net | [拉丁美洲及加勒比互联网络信息中心](http://lacnic.net/)(巴西圣保罗) | 拉丁美洲及加勒比海诸岛IP地址信息 |

## 5.2　定节点域名的whois扫描实现

定节点域名的whois扫描底层实现，即用户通过internet访问该服务器的过程主要分为下面三步：

1. 通过tcp服务连接目标whois服务器的43端口；
2. 向服务器发送一个命令，以回车和换行符结尾；
3. 接收命令的返回信息，输出结束后，服务器将关闭连接。

程序实现流程为，先声明socket套接字，然后选择一个whois服务器并连接43端口，向服务器发送需要查询的域名，并在语句后面添加回车和换行符，最后就监听并获取返回的信息。通过伪代码实现whois的查询过程如下：

s = 新建一个套接字实例

s.连接('whois.internic.net',43)

s.发送('需要进行whois查询的网站\r\n')

while 返回数据不为空:

    v = s.接收数据()

s. close()

此外，还可以通过在线的whois查询网站，whois查询API，需要下载到本地的whois工具查询。

# 6.　对网络空间指定节点的安全性检测

本设计使用的网络空间指定节点的安全性检测的方法是基于特征值比对的漏扫。

## 6.1　基于特征值比对的漏扫底层原理

根据一个集成大量扫描插件的插件包，该插件包可以理解成一一映射的特征值集合和漏洞结果，通过发包和查看返回包状态得到端口相应包，这些相应包经过提炼处理成为一个待比对的特征值，通过特征值比对等方式进行漏洞结果的判断，从而完成漏洞扫描，扫描结果则是安全性检测的结果。

该扫描结果的依据就是特征值比对，误报可能性还是有的，但是总体误差在可容忍范围内，但是实际应用中一般会选择对高危和中危进行验证。结果的准确度和插件包有非常大的关系，官方网站会定时更新并挂出安装包，用户需要手动更新。结果的准确度还可扫描策略有关，如果涉及漏洞的对应插件都不在指定策略的扫描范围内，也会造成漏网之鱼的出现，但是如果把所有插件都盲目地用于所有目标的比对，这将会是一个非常庞大耗时的工程。

## 6.2　基于特征值比对的漏扫实现

本设计使用的是Tenable公司旗下的Nessus扫描工具。本人在乙方公司和科研机构实习的过程中都有用到这个软件，可见官方的说辞“目前全世界最多人使用的系统漏洞扫描与分析软件”可信度是非常高的。Nessus起源于该公司“Nessus”远端系统安全扫描程序计划。该计划成果是实现了一个漏洞扫描全面，漏洞数据库更新频繁，可以随着系统资源自动调整如CPU和内存使用情况的资源配置，可以同时本机和远端遥控的，可以自定义插件同时完整支持SSL的系统漏洞分析扫描软件，并将其命名为Nessus。

该工具的使用需要在官网注册得到注册号，根据系统类型下载对应的安装包，然后在本地搭建起一个“完整的web服务”，默认端口是8834，用户交互界面用的就是这个web页面，默认地址是“https://localhost:8834/#/”。打开一般个人使用免费版足够了，如果是商业用户则需要使用收费版本。

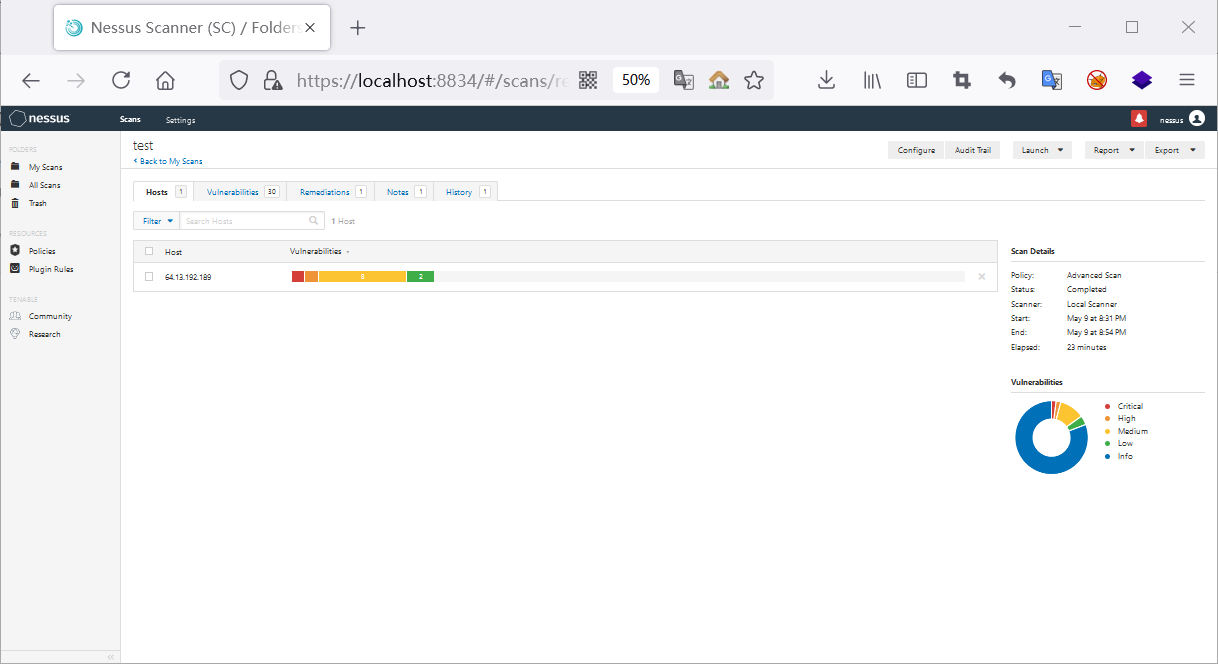


图6-1　手动Nessus扫描的结果图

在该网站的“My Account”页面下可以找到“Access Key”和“Secret Key”，再根据《API documentation》实现自动化实现的设计，算法流程图如下。

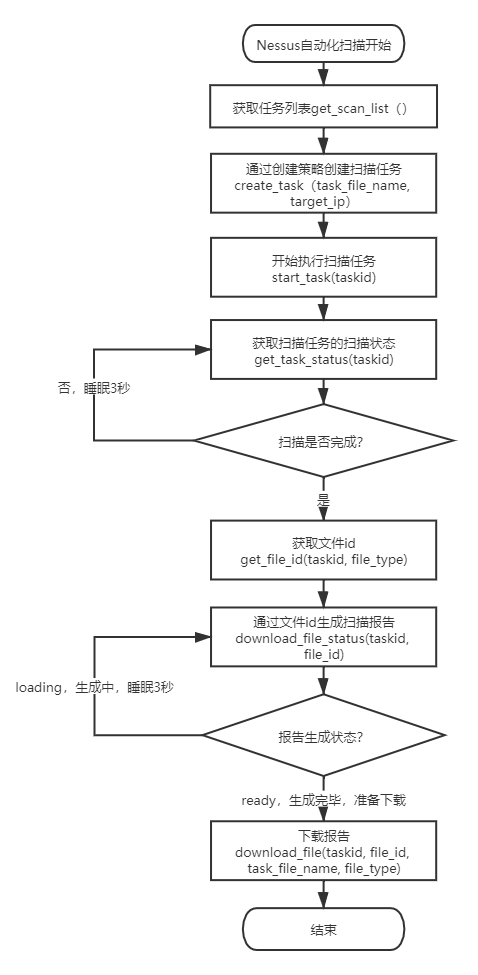


图6-2　本系统的Nessus自动化部分设计流程图

# 7.　系统理论部分设计

## 7.1　项目系统功能与实现目标

该项目理首先会给出一个属性分类的设计，然后系统设计部分提供相应的属性挖掘技术的自动化实现。主要功能是对网络中指定节点的信息进行搜集同时初步判断出其安全性，基本完成前渗透中信息搜集的功能。该系统的设计基于该设计对网络节点的一个属性分类的定义。本设计从宏观上，对安全分析中的网络节点一共定义了两大类，第一类是描述画像类，第二类是安全评估类。

## 7.2　项目节点画像部分

### 7.2.1　描述画像类设计依据

在安全分析中，尤其对于渗透测试人员而言，对于网络中的节点，首先需要有一个空间测绘中画像的认识。如果有一个自动化工具，可以直接将该节点的渗透测试人员所需的信息直接一键收集。对于安全领域中渗透人员而言，是一件非常方便的事情。该项目的出发点即此。

但是要判断出哪一些节点的属性符合安全测试中渗透测试人员的需求，并且帮助渗透测试人员对该节点的安全性作出初步的评估不是一件容易的事情。首先需要考虑的问题是，哪些属性值得被搜集。本系统考虑到的值得搜集的属性一直围绕着该IP的地理位置与归属地，与该IP的固有属性，该IP上的操作系统软件服务，该IP的属主以及属主的信息进行。

将IP所属的地区纳入到收集的信息范畴中理由如下：首先，用户应该先知道该IP地址所属国家，从而知道该国的网络相关法律才可以对该IP地址进行进一步的安全相关操作。而且每一个国家，在网络方面的各方面习惯都并不一致。国内绝大部分的网站，目前实现跨域用的是JsonP的方式。然而在国外目前实现跨域已经很少再用JSP的方式，因为国外的网站大部分都已经实现了前后端的分离，而且后端的交互是用request请求进行，并不存在跨域这个问题，而从渗透测试人员的角度来考虑，国内的网站可以多考虑csrf攻击的危害性，而国外的网站则并不那么适用。而且知道了IP地址的具体地理位置，还可以对该IP进行物理攻击。又比如某一些网络的中继设备，可以通过直接物理连接的方式，获取到连接到同一个设备上的其他设备的发包情况，这就需要对该节点进行地理定位。

将IP的属主信息纳入到收集的信息范畴中理由如下：对该ip进行攻击型防御测试，首先要清楚该IP是不是目的IP，倘若根本不知道IP的属主，就连确认身份都做不到，谈何安全性测试呢？而从另一面分析，知道该IP属主的一些个人信息也有利于对该节点的进一步渗透测试，例如该IP对应的主机上搭建的网站的管理员用户，有可能包含该数组的一些个人信息。

将IP的子域名信息纳入到收集的信息范畴中理由如下：对于一些大型的，常见的常用的，正在使用的站点，想要成功的进行渗透测试并不简单。因为这样的站点一般都处于，专业运维人员的长期运维之下。而且会有团队进行网站的开发更新与维护。然而对子域名的扫描可以找到一些比较不常用的，公司有可能会疏于维护，或者甚至是遗忘了该站点的存在，更谈不上维护了。这样该站点就有可能成为整个服务器的一个脆弱点，相比起一些常用常见正在使用正在被维护的大型网站，它被成功渗透的可能性会大大增加。这就是子域名，对于渗透测试人员最大的作用。

将该节点操作系统纳入到收集的信息范畴中理由如下：是因为会存在一些操作系统级别的漏洞，这也将会是安全性检测中的一个重要方向。检测所开放的端口号以及端口号对应的服务，是因为每一个端口都有可能有对应的漏洞，每一个对应的服务以及每一个存在于该节点上的软件附件，都有可能有自身本身就存在的漏洞。再者，识别出操作系统有利于进一步判断该节点的功能，比如，windows Server操作系统一般对应服务器等。

节点进行简单的识别与画像，画像尽可能包括节点的端口开放状态，端口运行服务，使用的操作系统，节点的地理位置，节点所处的大型网络自治系统或者运营商，节点所属公司、组织或者个人，节点的子域名，域名服务器相关信息，注册时间，注册人或公司组织，注册人或公司组织联系方式等节点信息。在以上提到的信息中，搜集到的越多，节点的画像越成功，也越有利于下一步操作，也就是节点的安全性分析。

### 7.2.2　描述画像类设计

一个描述画像类的基本组成包含的数据对象有，目标ip，目标ip的预测操作系统，目标ip的子域名，目标ip的开放端口，目标ip的地址位置，目标ip的属主信息，目标ip的注册信息。

表7-1　描述画像类的属性分类定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述画像类的7个组成部分 | 该组成部分伪代码描述 | 该组成部分的含义 | 该组成部分的出处 | 该组成部分的数据结构 |
| 目标ip | string ip\_addr; | 每一个描述画像类的实例化对象都是围绕某个ip地址进行的一次信息搜集与整理，该ip地址是整个描述画像类的实例化对象的描述核心与主题；  该ip地址不可能存在且不允许存在模糊度，必须的明确且正确的。 | 用户或者系统给定，作为整个节点画像和安全性检测的目标，必须明确给出，不可以存在模糊度。 | 一个点分10进制ipv4地址，以字符串形式表示和存储。 |
| 目标ip的地址位置 | typedef struct{  string country;  string province;  string country;  float lat;  float lng;  string time\_zone;  string asn\_id;  string asn\_name;  } port[NUM]; | 目标节点所处的国家，省份，城市，时区，经纬度和大型网络自治系统的全球唯一编号和该系统名称。 | 一般是通过网络测量和大数据挖掘。 | 该部分除了经纬度为浮点型，大型网络自制系统的全球唯一编号为整形，别的皆为字符串。 |
| 描述画像类的7个组成部分 | 该组成部分伪代码描述 | 该组成部分的含义 | 该组成部分的出处 | 该组成部分的数据结构 |
| 目标ip的预测操作系统 | string system\_name; | 预测该节点使用的操作系统;  存在一定的误判率，在插件返回的原生数据中包含多个可能的操作系统，每个对应给出相应的正确概率，该系统只取概率最大者。 | 主要通过ping命令每跳间隔数来判断，由nmap插件返回该结果；  存在一定的误判率和模糊度，最终结果取精确的最高的一个； | 该操作系统名字，以字符串形式表示和存储。 |
| 目标ip的子域名 | typedef struct{  int num;  string sub;  } Subdomains; | 对目标节点能探测出来的所有子域名，以及个数；  一般不存在误判，但是很有可能返回结果不全，这与搜集原理有关。 | nslookup查询法直接查询DNS服务器的A记录和CNAME记录；  通过搜索引擎site语法；  字典暴力枚举爆破法。 | 一个自定义数据结构，包括一个整形数字，和一个以逗号分隔的子域名字符串。 |
| 描述画像类的7个组成部分 | 该组成部分伪代码描述 | 该组成部分的含义 | 该组成部分的出处 | 该组成部分的数据结构 |
| 目标ip的开放端口 | typedef struct{  int port\_id;  stirng state;  string service;  } port[NUM]; | 所有能探测出来的开放的端口和探测时的端口状态。 | TCP扫描；  UDP扫描；  特定TCP标志位的扫描。 | 一个数组，数组每个元素是一组数据，包括整型端口号，字符串端口状态和字符串对应的服务。 |

## 7.3　项目节点的安全性分析部分

### 7.3.1　描述画像类设计依据

在安全分析中，对于渗透测试人员的前渗透阶段而言，快速判断目标节点的安全性、脆弱性，有可能存在的脆弱点，可能存在的风险，对于渗透测试而言是非常有帮助的。在对比了目前，比较常用的一些探测工具，漏扫工具以后，本项目的开发最终选择了Tenable旗下的Nessus API作为该项目的插件使用的接口之一。这是因为nurses几乎满足了本项目的在安全性测试方面的所有需求。首先最大的需求肯定是以一种比较温和而快速的方式，从不同的方面，比如操作系统级别中间线级别又或者是一些其他方面的漏洞，对该节点提炼出来的特征（该步骤与本项目中的节点画像部分目标较为一致）进行漏洞的特征指标比对，从而初步判定该节点的安全性。倘若直接比对出来有CP的漏洞，或者是一些别的高危漏洞，则渗透测试安全人员可以考虑直接从这些脆弱点着手。倘若。扫描的结果并没有出现高危中危等漏洞，则说明该节点相对安全，也许并不值得进行渗透测试。

节点的安全性一般从两个角度考虑，物理安全性，和非物理安全性。物理安全性是指该节点的物理载体是否暴露在危险场合，比如雷暴天气频发场合的室外下、易燃物旁，或者对攻击者破坏者易得。而该系统主要考虑非物理安全，也就是节点对非物理攻击方式面前的脆弱程度的一种探测。其中，包括在搜集到的节点信息基础上的CVE漏洞库的特征值比对，开放的端口对应服务的一些简单探测等。

### 7.3.2　安全评估类定义

第二类是基于第一类的对该网络节点的安全性分析结果概况类，简称为安全评估类。第二类包括，所有特征识别后判断出来的漏洞。通过特征识别比对出来的一个漏洞信息的集合，为该分类一个信息基本数据单位。扫描结果有多少个漏洞就会有多少个这样的基本数据单位，该系统为了设计方便内给那个条目一个唯一的id作为主键，但是事实上，如果不添加一个编号，该条目的符合主键应该是“ip地址”，“端口号”和“扫描插件编号”。

该类的定义目的是回答用户对该节点提出的“该节点是否安全”的问题，通过特征比对得出的漏洞越多，说明该节点越不安全，但是这些漏洞都有误报的可能。还有可能出现漏洞全都是误报的可能，因此这些漏洞，尤其是高危和中危险这种，都很需要用户接下来通过其他方式进行验证。该系统日后也有加入漏洞验证插件的打算，不过该部分再法律上需要处理对应节点属主的授权问题，开发的过程因为需要处理授权的问题会比较繁琐，因此该构思一直处于酝酿状态，是否会进行这方面的进一步开发，怎么开发也会进行慎重考虑。

表6-2　安全评估类的属性分类定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述安全评估类的7个组成部分 | 该组成部分伪代码描述 | 该组成部分的含义 | 该组成部分的出处 | 该组成部分的数据结构 |
| 目标ip | string ip\_addr; | 每一个描述画像类的实例化对象都是围绕某个ip地址进行的一次信息搜集与整理，该ip地址是整个描述画像类的实例化对象的描述核心与主题；  该ip地址不可能存在且不允许存在模糊度，必须的明确且正确的。 | 用户或者系统给定，作为整个节点画像和安全性检测的目标，必须明确给出，不可以存在模糊度。 | 一个点分10进制ipv4地址，以字符串形式表示和存储。 |
| 端口号 | int port; | 该漏洞对应的端口号 | 对所有开放端口找到对应的插件进行扫描，如果扫描出有漏洞则纪录下来。 | 是一个整数。 |
| 插件编号 | int id; | 扫描出该漏洞对应的插件号 | Nessus插件包中自带。 | 是一个整数。 |
| 描述安全评估类的7个组成部分 | 该组成部分伪代码描述 | 该组成部分的含义 | 该组成部分的出处 | 该组成部分的数据结构 |
| 插件名称 | string name; | 插件的名称。 | Nessus定期更新插件包中自带插件编号。 | 是一个字符串。 |
| CVE编号 | string cve; | 如果该漏洞对应某个cve，则显示该cve 编号。 | 和cve漏洞特征值比对，如果比对成功则得出的CVE编号。 | 是一个字符串。 |
| CVSS | float cvss; | 通用漏洞评分系统 | 如果存在，给出该漏洞对应的评分。 | 是一个浮点数。 |
| 解决办法 | string solution; | 如果nessus漏洞库中存在，则给出该漏洞的解决办法。 | Nessus给出。 | 是一段文本。 |
| 协议 | string protocol; | 给出该漏洞对应的服务基于tcp还是udp. | Nessu扫描给出。 | 是一个字符串，只可能是“tcp”或者“udp”（不区分大小写） |
| 插件输出 | string plugin\_output; | 给出该插件扫描到的详细结果。 | Nessu扫描给出。 | 是一段文本。 |
| 插件相关信息 | string info; | Nessus给出的插件相关信息。 | Nessu插件库给出。 | 是一段文本。 |
| 描述安全评估类的7个组成部分 | 该组成部分伪代码描述 | 该组成部分的含义 | 该组成部分的出处 | 该组成部分的数据结构 |
| 插件简述 | string synopsis; | Nessus给出的插件的相关信息。 | Nessu插件库给出。 | 是一小段文本。 |
| 风险等级 | string risk; | 描述该漏洞的风险等级。 | Nessus扫描结果给出。 | 是一个单词（字符串）。 |

# 8.　系统实现部分设计

## 8.1　系统的总体架构

主程序是整个项目的入口，使用getopt库和python3中的sys模块定义一个命令行交互的函数。在用户启动该项目时给出一个较为友好美观的界面，并且给出该系统用法的提示，便于用户轻松使用。然后接收用户命令，后台取回必要参数，通过对参数的判断实现对用户所需要调用的模块的判断，然后调用对应模块。

主程序调用的模块除了较为简单的查询模块别的都以插件的形式，分别写在对应命名的文件中。Tenable Nessus接口调用写在“aboutnessus.py”这个文件中，关于WhoisXMLAPI产品中的地理位置相关接口调用写在“aboutgeo.py”中，，关于WhoisXMLAPI产品中的子域名相关接口调用写在“aboutsubdomains.py”文件中，关于WhoisXMLAPI产品中的whois相关接口调用写在“aboutswhois.py”文件中，关于nmap相关接口调用写在“aboutnmap.py”中。数据库相关的操作，包括对数据库的创建，数据库中表的创建，对ness、info、port、subdomains和whois表的插入，查询等操作都写在“mymysql.py”文件中。

项目运行过程中生产的中间数据，例如每个接口调用原始结果以txt或者是csv的形式存储在同一目录下。

## 8.2　系统的节点画像功能

只需要给出网络中节点的IP地址，系统会自动分析该IP地址的相关信息，对该节点“是什么”做出一个比较详细的回答。方便信息安全相关工作者，尤其是渗透测试安全人员对该节点的进一步探测或者是安全性分析。

其中该设计能够给出的相关信息包括：该节点的较为具体的地理位置一般可以精确到国家以及城市以及该节点所位于的经纬度，甚至会给出该节点所在的时区。同时该系统还会在后台扫描并分析出该节点最有可能使用的操作系统，用户可以根据该操作系统比较直观的判断出该节点最有可能是个人计算机路由器还是大型服务器。

同时该系统还可以分析出该节点所位于的大型网络自制系统的全球唯一asn（ Autonomous system number）以及该自治系统的名称和所属的公司。在庞大的互联网之中，一个自治系统，通常情况下包括一个又或者是多个实体管辖下的所有IP网络和路由器的全体，一般情况下他们拥有共同的路由策略。一些比较大的互联网组织例如阿里巴巴集团，可以使用其私有的自制系统编号。多个大型网络组织或者是公司集团，在拥有该编号的情况下，可以与同一个将他们连接到互联网的ISP之间运行GDP协议。根据该原理，倘若可以扫描获得该IP地址所属的自制系统的全球唯一编号，在极大程度上就可以确定该节点所属于的大型网络组织或者是互联网服务提供商，例如在本测试样例中，该节点属于阿里巴巴集团，这是显而易见的。

该系统还可以探测扫描并分析得到该IP所处的地区的邮政编码，该功能算不上是该节点在安全领域的非常敏感的一个信息被探测出来，但是使用者可以从社会工程学等其他方面去考虑该信息的利用。

说到安全领域比较敏感的节点信息。在该系统的画像功能中，比较敏感的应该算得上是子域名的查询。根据实验得出的结果，该查询可以得到比较全面的节点相关的子域名信息。对于安全领域的渗透测试人员而言，子域名的充分搜集，可以使得攻击面得以打开，由原先的攻击点扩充为多个攻击点，使得渗透测试成功的概率得到很大的提升。而且该提升并不是与被扫描出来的子域名数量呈现线性关系。因为一些常见的域名，安全性相对会比较高，因其常见与常用，开发人员运维人员和安全人员都知道其暴露在大众的视野中，所以对其安全性要求相对严格。然而对于另外的一些不常用的子域名，甚至是被荒废的，但是一直没有被处理的子域名，其安全性一般得不到保障，他相对于最常用最普遍被使用的域名而言，更容易成为一个攻击脆弱点。

而对于whois查询的插件，该系统的查询思路是，先对该IP所开放的端口以及端口对应的情况和该端口猜测正在运行的服务进行一定程度上的探测。然后将与web服务相关，比如说HTTP或者https的服务端口，连同IP也就是IP加端口号的形式，放入whose查询中进行进一步的探测，该探测的返回结果，经过提炼总结之后，大概分为如下几个部分：首先是该站点的注册时间以及该网站的过期时间，然后可可以得到该站点的一些注册信息，例如注册的集团，或者组织名称。

## 8.3　系统的节点安全性分析功能

完成对节点的画像以后，安全人员最为关心的莫过于该节点的安全性。对于节点的安全性判断，到目前为止常用的方式有两种，一种是特征值比对的方式，还有一种是攻击型防御的方式。该系统采用的是较为温和的特征值比对方式，该方式会存在一定的误判率。误判率的存在是因为，该方式仅仅是通过对该系统的探测，也就是该系统的节点画像部分所得到的信息进行筛选，综合，提炼，最终得到一个特征值集合。然后根据该特征值集合，与目前大部分的漏洞进行特征值比对，但是并没有对该漏洞进行暴力攻击验证。因此它更像是，对安全人员给出的一种可能存在漏洞的警示。

该系统对于节点安全性的分析，主要着眼于两个方向。一个是该结点所开放的端口以及该端口对应的服务，另一个则是该系统的一些特征，比如说操作系统，或者是所搭建的网站的一些特征，所运行的数据库，软件，中间件等与目前已经被挖掘出来的漏洞复现所需要的前提是否高度吻合。扫描结果主要给出以下信息：倘若匹配出CVE则会给出CV的编号以及对应漏洞评分，会给出该漏洞对应的端口以及该端口基于TCP亦或是udp以及该对该端口扫描的方向，和扫描得到的结果或者信息。有少部分的比较常见的漏洞，还会给出具体避免或者修复建议。

## 8.4　数据库实现功能

### 8.4.1　数据库种类

数据库采用的是Mysql，开发过程中为了操作方便数据库管理采用Navicat数据库管理软件。

### 8.4.2　数据库使用情况

项目启动的第一件事情就是检查是否有创建名为“yyq”的数据库，如果没有则创建一个，如果有则继续沿用该数据库。在数据库创建完成后立刻创建五个表，分别为存储从“aboutness.py”的原始输出结果“test.csv”中提炼出来的信息的ness表，存储节点开放端口信息的port表，存储节点的web服务对于whois查询探测得到的结果的whois表，存储节点的子域名的subdomains表，存储节点预测操作系统等其他信息的info表。这些表都创建成功后，才是和用户的交互和其他功能的实现。

### 8.4.3　数据库和表单设计

本项目一共创建一个数据库——名为“nxq”，一共创建创建如下五个表单。

表8-1　info表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 非空 | 是否主键 |
| ip | varchar | 20 | 是 | 是 |
| system | tinytext | 无设置 | 否 | 否 |
| country | tinytext | 无设置 | 否 | 否 |
| city | tinytext | 无设置 | 否 | 否 |
| 列名 | 类型 | 长度 | 非空 | 是否主键 |
| lat | float | 无设置 | 否 | 否 |
| lng | float | 无设置 | 否 | 否 |
| timezone | tinytext | 无设置 | 否 | 否 |
| asn\_id | Int | 无设置 | 否 | 否 |
| asn\_name | tinytext | 无设置 | 否 | 否 |

其中，“ip”列记录被探测的ip地址。“system”存储的是nmap自动识别出来的匹配率最高的操作系统，由于操作系统版本识别误差比较大，所以在开发过程中决定直接丢弃而没有存进数据库，“country”存储的是探测出来IP地址所属国，“city”是探测出来ip地址所属的城市，“lng”是探测出来IP地址所在经度，“lat”是探测出来P地址所属的纬度，“timezone”是IP地址所处位置的时区，“asn\_id”是IP地址所处于的大型网络自制系统的全球唯一编号，“Asn\_name”是该系统的名称。该大型网络自制系统有可能是运营商或者是大型的网络服务公司或组织。

表4-2　ness表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 非空 | 是否主键 |
| uid | int | 无设置 | 是 | 是 |
| id | int | 无设置 | 是 | 否 |
| cve | tinytext | 无设置 | 否 | 否 |
| cvss | float | 无设置 | 否 | 否 |
| risk | tinytext | 无设置 | 否 | 否 |
| host | tinytext | 无设置 | 是 | 否 |
| port | int | 无设置 | 是 | 否 |
| protocol | tinytext | 无设置 | 否 | 否 |
| name | text | 无设置 | 否 | 否 |
| synopsis | text | 无设置 | 否 | 否 |
| info | text | 无设置 | 否 | 否 |
| solution | text | 无设置 | 否 | 否 |
| plugin\_output | text | 无设置 | 否 | 否 |

“uid”是一个自动递增的列，用于给存入的记录进行标号，是整个表的主键，但是除了编号和主键没有其它实际意义。“id”是Nessus扫描插件的编号。如果扫出有漏洞，这个漏洞如果存在于CVE中，将会给出CVE的编号。“CVSS”是对该漏洞的危险性评分。“Risk”是该漏洞的风险。“Host”是被探测的主机ip。“Port”是该记录对应的扫描端口。“Protocol”是该记录对应端口基于的协议。“Name”是该插件的名字。“Synopsis”是该插件的简单介绍，“infer”是该插件的详细描述，“solution”是如果有建议则提出对该漏洞的修复或者是避免建议，“plugin\_output”是该插件扫描的输出结果。

表4-3　ness表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 非空 | 是否主键 |
| ip | char | 20 | 是 | 是 |
| port | int | 无设置 | 是 | 是 |
| state | tinytext | 无设置 | 否 | 否 |
| servive | tinytext | 无设置 | 否 | 否 |

这个表中的“ip”存的是被探测主机的IP，“port”指的是被探测出开放的端口号，“state”是指该端口对应的状态，“service”是该端口号正在运行的服务。

表4-4　subdomains表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 非空 | 是否主键 |
| ip | char | 20 | 是 | 是 |
| count | int | 无设置 | 否 | 否 |
| domains | tinytext | 无设置 | 是 | 否 |

该表中的“ip”是被探测主机的IP,“count”是被探测的网站被探测出来的子域名个数，“domains”是所有被扫出来的子域名，其中逗号是分隔符，但是所有子域名加上逗号作为一个字符串存储整体。这里是考虑到读取的时候一般是作为一个整体进行读取，所有没有分开存储。

表4-5　whois表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 非空 | 是否主键 |
| ip | char | 20 | 是 | 是 |
| audit | tinytext | 无设置 | 否 | 否 |
| contactEmail | tinytext | 无设置 | 否 | 否 |
| domainName | tinytext | 无设置 | 否 | 否 |
| estimatedDomainAge | int | 无设置 | 否 | 否 |
| parseCode | int | 无设置 | 否 | 否 |
| registrarIANAID | tinytext | 无设置 | 否 | 否 |
| registrarName | tinytext | 无设置 | 否 | 否 |
| registryData | text | 无设置 | 否 | 否 |

“ip”是被探测的域名的IP地址，“audit”是该域名创建时间等，“ContactEmail”是该域名拥有者的邮箱，“domainName”是该域名名字，“estimatedDomainAge”是估计该域名使用年限，“parseCode”是解析码，“registrarlANAID”是注册机构的编号，“registrarName”是注册机构名称，“RegistryData”是注册的时间对应组织与组织地址等信息。

## 8.5　命令行交互设计

定义了一个函数负责命令行启动界面的字符画输出和使用规则提示。然后在“mymysql.py”文件中定义了get\_sql\_list、get\_a\_table函数用于命令行对结果的查询。该命令行一定程度上简化了用户的直接命令行操作。

对用户操作的参数设置如下：

表4-6　该系统命令行参数说明表

|  |  |  |
| --- | --- | --- |
| 完整参数 | 简略版参数 | 对应操作 |
| --IP | -p | 设置ip |
| --al | -a | 运行系统所有插件 |
| --whois | -w | 运行whois插件 |
| --subdomain | -s | 运行子域名扫描插件 |
| --nmap | -n | 运行nmap插件 |
| --geography | -g | 运行地理位置识别插件 |
| --nessus | -e | 运行nessus插件 |
| --data | -d | 查看该表的前五条信息或者五条以下所有信息 |
| --quit | -q | 退出系统 |

该系统支持组合命令，也支持简略版组合命令。

## 8.6　项目运行环境与所需工具接口

### 8.6.1　项目开发与运行环境

Windows 10，python 3.9版本，Mysql数据库，Nessus环境，Nmap工具，WhoisXML API接口。

### 8.6.2　项目用到的辅助设计的开发者工具

VScode编译器，Navicat数据库管理工具，pip3或者anaconda。

### 8.6.3　所需的python库

1.requests

requests是使用Apache2 licensed 许可证的HTTP库，可以自动实现持久连接keep-alive和连接池，支持使用cookie，支持文件上传，支持自动响应内容的编码，支持URL和POST数据自动编码。

2.json

Python标准库中的json模块提供了JSON数据的处理功能。

3.pprint

print()和pprint()都是python的打印模块，功能基本一样，唯一的区别就是pprint()模块打印出来的数据结构更加美观，更加方便阅读打印输出结果。特别是对于特别长的数据打印， pprint()采用分行打印输出，适合打印数据结构比较复杂、数据长度较长的数据。

4.pymysql

pymysql是在 Python3中用于连接 MySQL 服务器的一个库。

5.csv

python内置了csv模块，用于处理对cvs文件的操作，在该系统中，nessus API的返回结果就是以csv的形式被暂时保存下来，然后再通过读取csv取出需要存入数据库的信息。

6.getopt

Getopt是一个专门设计来减轻命令行处理负担的库函数，主要用来分析命令行参数。

7.sys

sys模块用于实现计算机系统与python的交互。

8.prettytable

Python通过prettytable模块将输出内容如表格方式整齐输出，python本身并不内置，需要独立安装该第三方库。

## 8.7　结果展示与命令行交互

实现命令行调用系统中的部分或者全部功能，以及查看结果，界面对用户简洁友好，主要使用到的库是getopt。

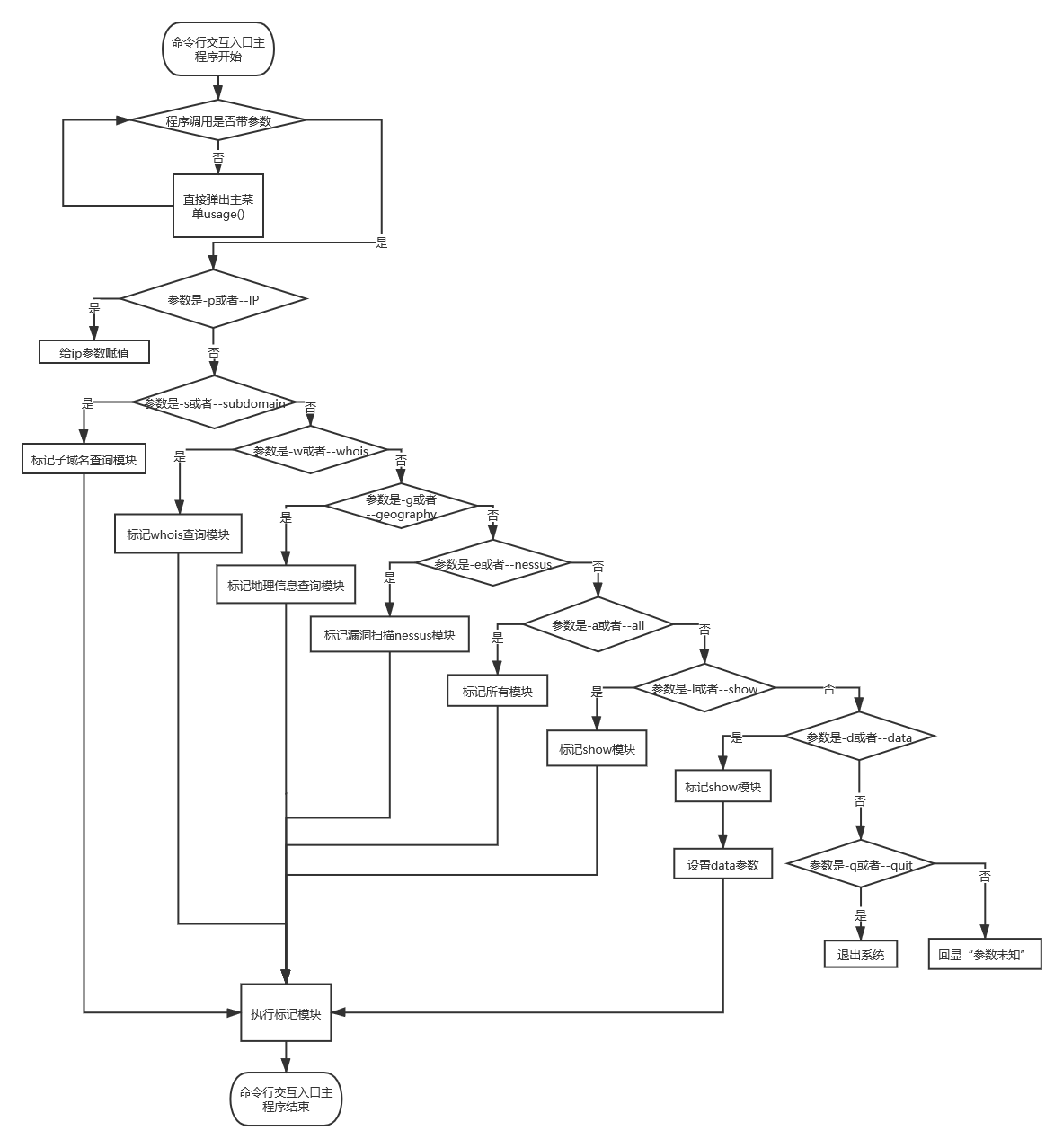


图8-7本系统的Nessus自动化部分设计流程图

# 9.　系统测试

## 9.1　简单命令行交互式测试

### 9.1.1　启动

启动命令为python yyq.py，需要在文件所在路径下。测试结果如下，弹出有字符画的主菜单界面，并且有参数使用说明和例子。

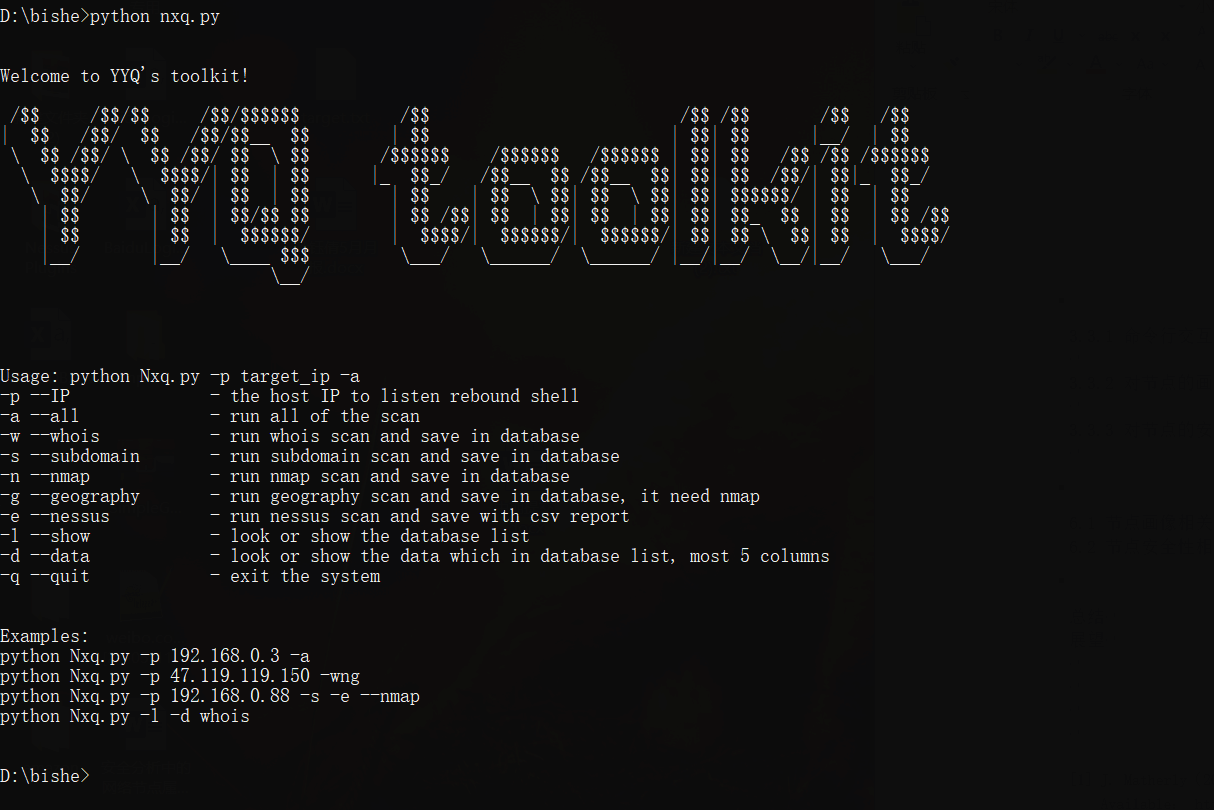


图9-1　系统主菜单界面

### 9.1.2 执行全部功能命令

执行全部功能命令为python yyq.py -p [ip] -a,其中方括号中的是被测ip的地址。结果如下，可见系统判断处理该节点使用linux操作系统，端口开放情况和对应服务也有输出，还有地理位置和whois查询等信息，测试成功。

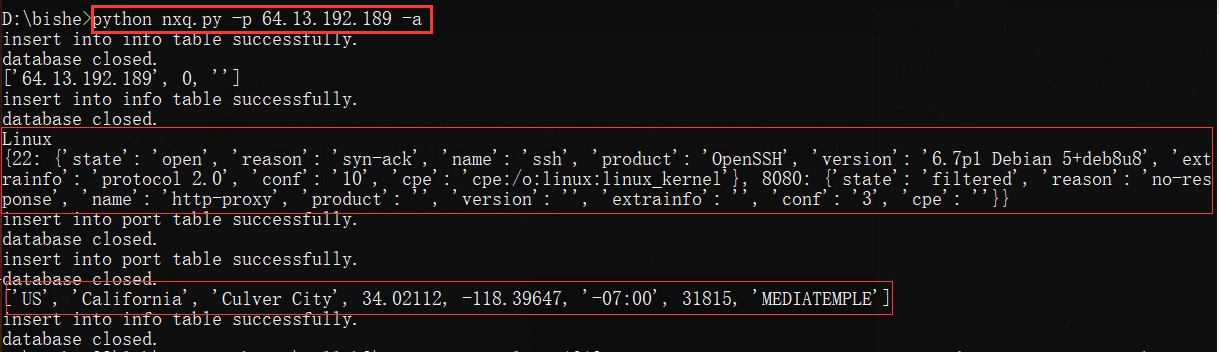


图9-2　执行-a指令结果a

下图可见不仅功能实现成功，而且代码回显的数据库插入成功。

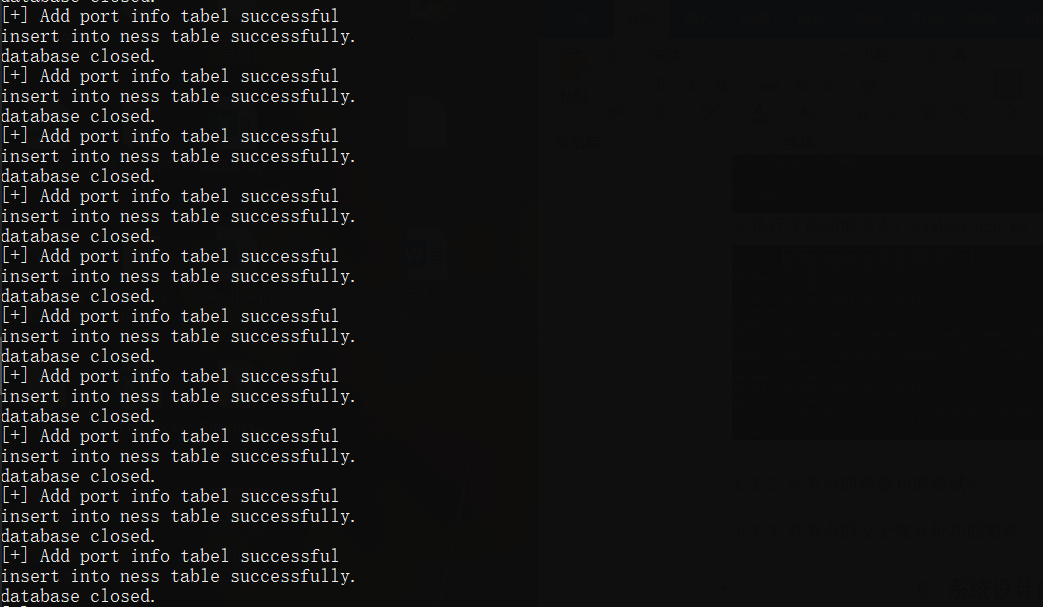


图9-3　执行-a指令结果b

### 9.1.3　组合命令测试

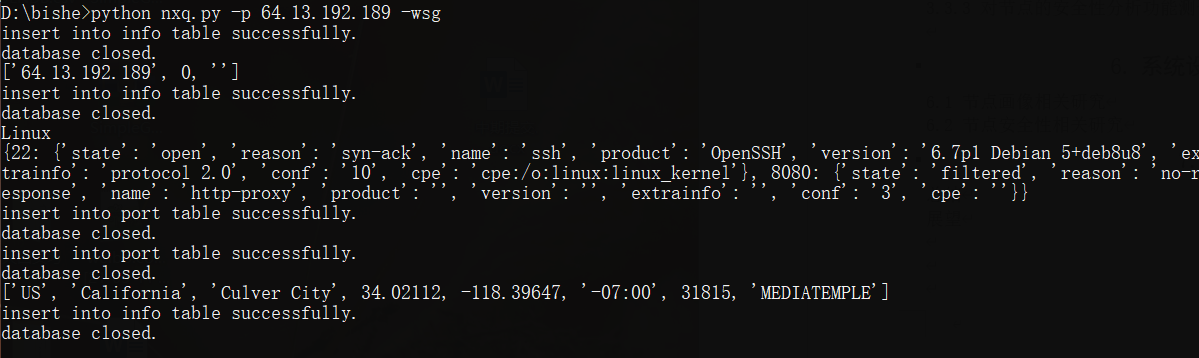
选取一个组合命令测试，命令选择如下python yyq.py -p 64.13.192.189 -wsg。该命令带参数“-p”和“-wsg”。

图9-4　执行组合命令结果

### 9.1.4　组合命令测试

python yyq.py -p 47.119.119.150 -snge



图9-5　组合命令测试a

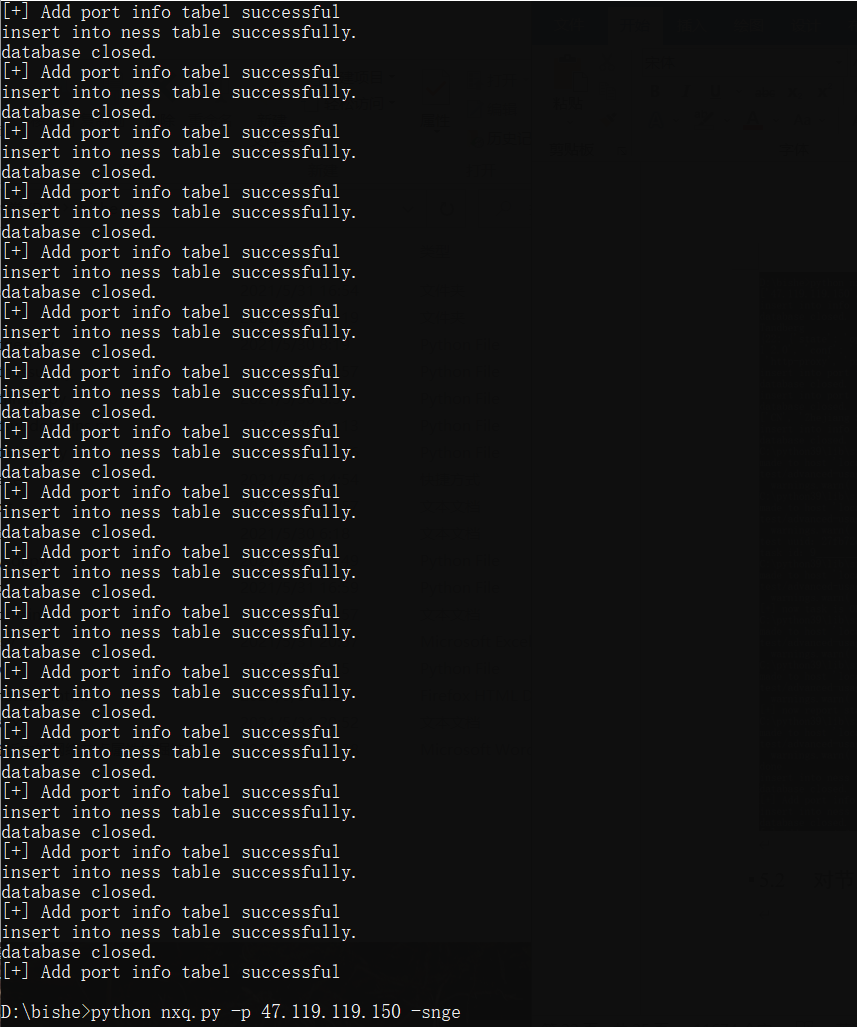


图9-6　组合命令测试b

## 9.2　对节点的画像功能测试

最终写入数据库的数据展示：

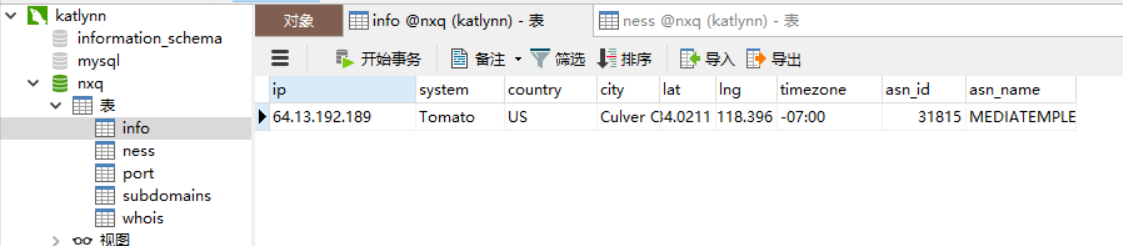


图9-7　数据库info表单

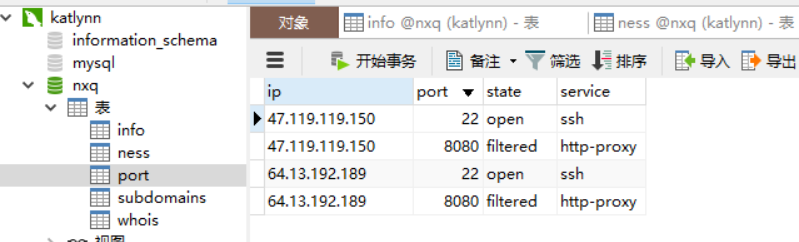


图9-8　数据库port表单



图9-9　数据库info表单

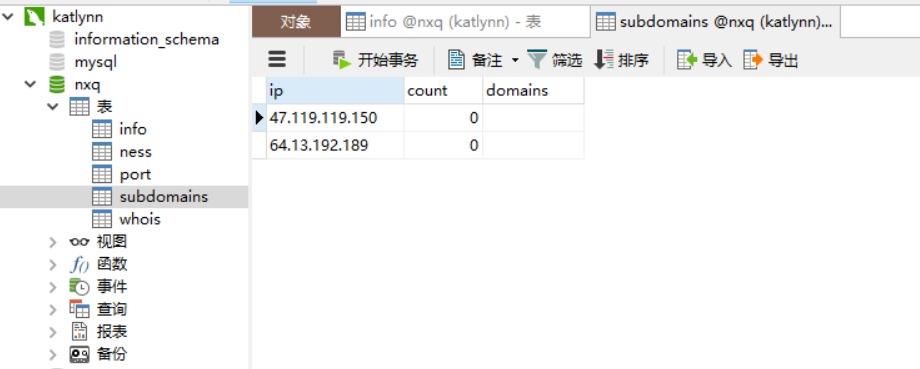


图9-10　数据库subdomains表单

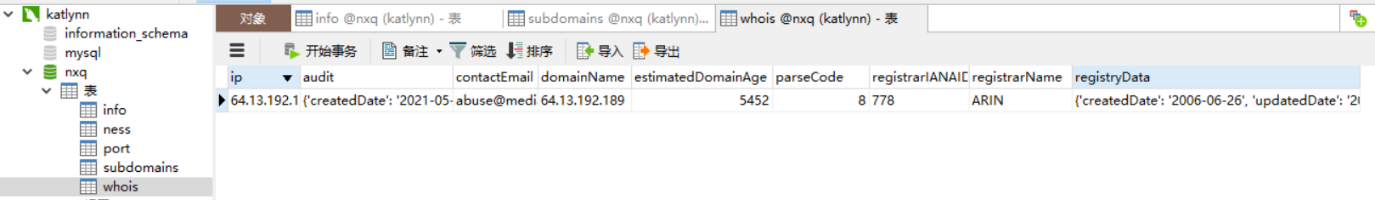


图9-11　数据库whois表单

写入数据库前，项目插件探测直接返回数据展示：



图9-12　数据库geo表单对应的原始数据

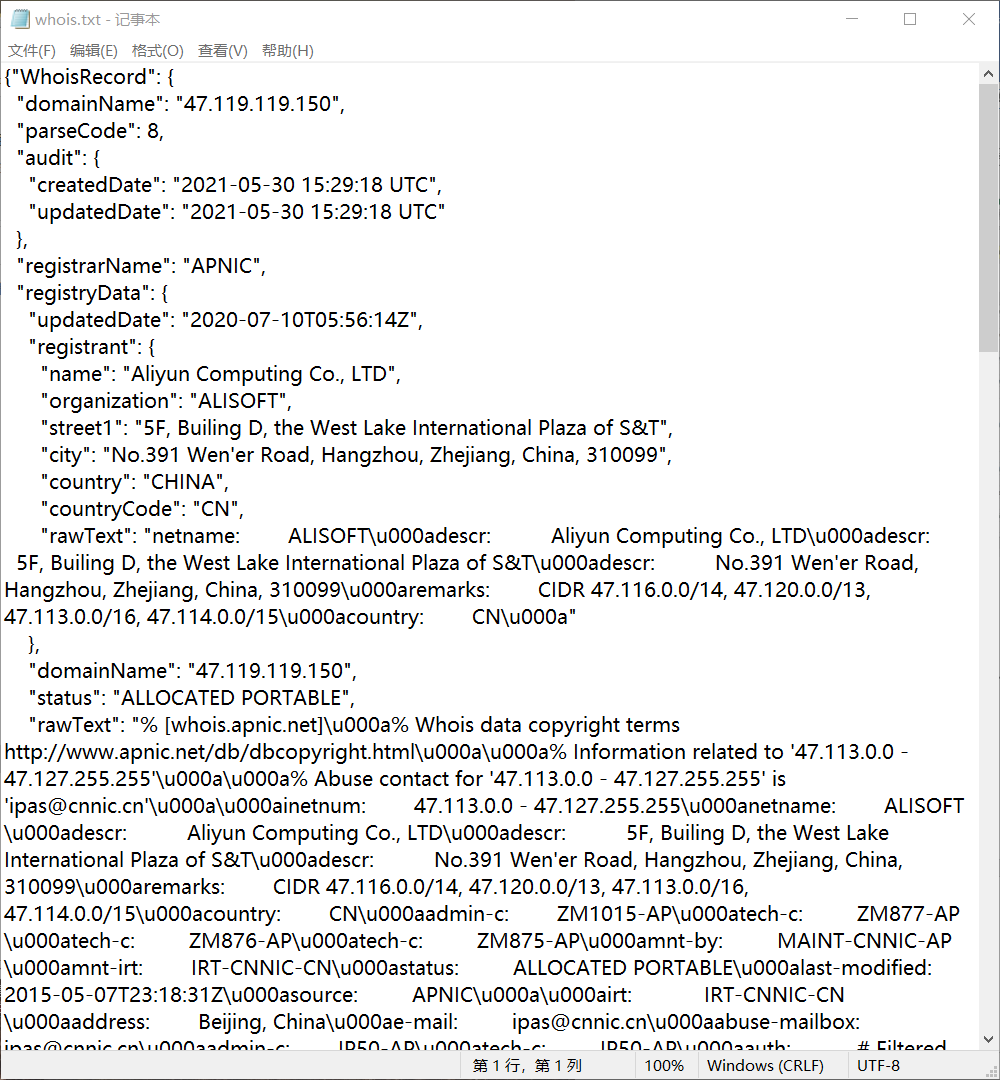


图9-13　数据库whois表单对应的原始数据



图9-14　数据库subdomains表单对应的原始数据

## 9.3 对节点的安全性分析功能测试

最终写入数据库的数据展示：



图9-15　数据库ness表单

写入数据库前，项目插件探测直接返回数据展示：

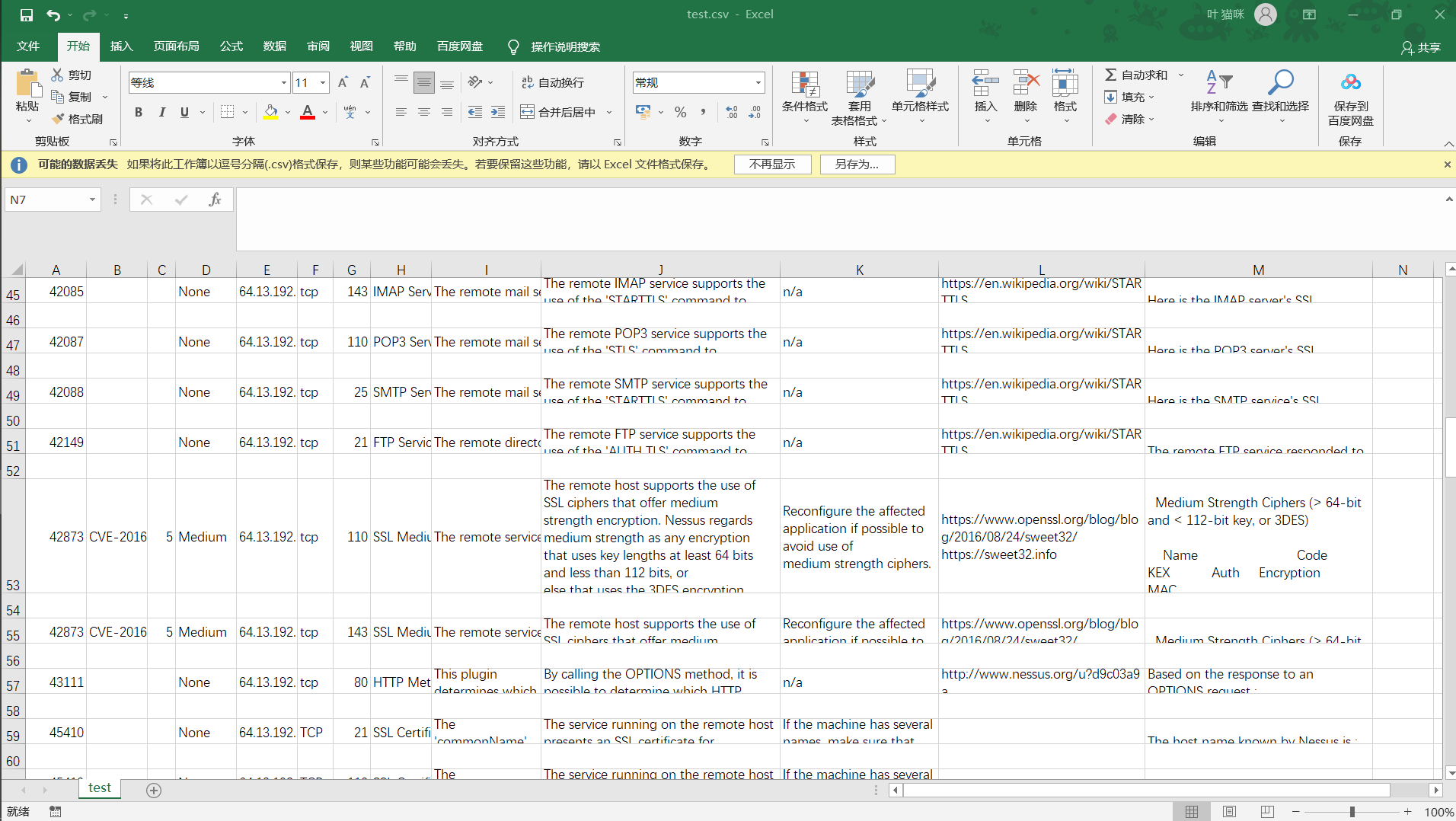


图9-16　数据库ness表单对应的原始数据

# 10.　总结与展望

## 10.1　总结

该系统的设计主要围绕着节点的画像和节点的安全性分析进行。

节点的画像所搜集的信息考虑的出发点，为此本论文为节点画像定义了一个数据结构。该数据结果主要考虑的是在安全领域中渗透测试人员对一个网络中节点的信息的需求。这一节点的画像，收集到的所有信息包括节点的IP，开放的端口，开放端口对应的服务，IP所在的地理位置，精确到城市，经纬度，时区，以及该IP的所在自治系统，该IP或者IP上的某一个站点的拥有者，该拥有者的相关信息如邮件电话，该IP的子域名数量以及子域名列表，该IP的操作系统。这些信息都是。安全领域中渗透测试人员会关心到的。

对于节点的安全性考虑的出发点是，安全领域中渗透测试人员对于一个节点是否脆弱的评估，以及攻击型防御中可能会用到的一些手法或者是评估方式。在不采取非法攻击手段的前提下，在尽可能采用温和的探测方式的前提下，本项目对节点安全性的评估，主要是基于tenable Nessus的API。

在安全分析中，一切节点属性分析都必须围绕整个节点是否安全，有无漏洞或者是脆弱点这一主题。而目前安全领域中最流行的渗透测试，是一个最容易找到漏洞或者是脆弱点的方式。本设计本系统相比起fofa这种产品的网络空间测绘，它更偏向于小范围小部分节点的测绘和渗透测试中的前渗透环节。

第二部分的安全性检测初步告诉渗透测试人员它是否安全，找到漏洞的难易程度，与此同时，它也给渗透测试人员提供了找脆弱点的思路。第一部分则是直接把搜集到的提炼过的信息呈现给渗透测试人员，让渗透测试人员可以根据自身的经验结合这些信息得到一些渗透思路上的启发。

## 10.2　展望

该项目目前信息搜集部分并不是很完善，日后将会将更多的前渗透步骤需要的信息以插件的形式集成到该系统中，例如C段查询。

该项目在开发的过程中，一直在考虑日后的扩展问题。首先是数据库的选择方面：一开始本系统的开发选择的是轻便、移植性好的sqlite3数据库，后来考虑到日后会参考CobaltStrike团队合作的架构，在服务器上进行数据的存储，同时参考fofa的架构，在后台一直扫描，以此更方便该系统的团队协作，同时也提高系统的扫描效率因此使用了MySQL数据库。其次是系统的用户交互方面，现在该系统是通过命令行的形式和用户进行交互，现在正在对其进行进一步的设计，对其进行改进，加入前端，然后数据的搜集可以下载插件在本地运行，也可以在服务器端进行，然后结果的存储主要是靠服务器，结果的展示可以在前端或者是命令行。因为考虑到以后可能会加入前端，在开发的源码中对Mysql数据库的操作也是使用了预编译的方式，这样可以预防sql注入。

然后，是整个系统的数据进一步提炼和利用的问题，这个问题也就是该系统研发的下一个目标，也就是渗透测试的中期渗透问题。当系统的信息搜集到了一定的程度以后，就可以考虑对信息进行进一步的整合。例如，对每个端口的服务进行总结，再进一步对这些服务进行试探，结合节点上的一些别的特征推断这个节点是路由器等中继节点还是终端节点，该节点主要用于什么服务比如邮箱服务。但是怎么对搜集到的信息进行利用，是否应该加入机器学习，深度学习例如卷积神经网络这些技术，如何运用好这些技术，日后有待研究。

目前思路比较清晰的是，可以将扫描出来开放的网页服务，寻找是否存在登录环节，可以对登录进行脚本抓包，然后对返回包进行自动分析，找到是否存在csrf漏洞，该过程分析可以抽象成与或图数学模型，具体实现参考与或图中的AO\*算法。然后还可以分析是否存在页面sql注入，加入sql注入自动化探测插件等。

该系统的最终研发目标是，在法律允许的范围内，在不违反道德的前提下，实现安全领域中，一套合理的适用的属性分类设计，以及基于该属性分类的大范围网络的空间测绘，以及小范围网络的包括前中后其的渗透测试辅助工具的系统开发。

参考文献

1. J. Matherly（2009）, “The Search Engine for Internet Connected Devices,” Available: http://www.shodanhq.com/
2. Censys,A Search Engine for Internet-Wide Scanning, https://censys.io/
3. Durumeric Z, Adrian D, Mirian A, et al, “A Search Engine Backed by Internet-Wide Scanning,” Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015: 542-553.
4. llman M, Paxson V, Terrell J, “A Brief History of Scanning,” Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. ACM, 2007: 77-82.
5. Lyon G, “The Art of Port Scanning,” Phrack Magazine, 1997, 7（52）.
6. Durumeric Z, Wustrow E, Halderman J A, “ZMap: Fast Internet-Wide Scanning and Its Security Applications,” USENIX Security Symposium. 2013, 8: 47-53.
7. Xuan Feng, Qiang Li, Haining Wang, Limin Sun, “Acquisitional Rule-based Engine for Discovering Internet-of-Thing Devices”, 27th {USENIX} Security Symposium ({USENIX} Security 18) 2018, Baltimore, MD.
8. Barnett R J, Irwin B, “Towards A Taxonomy of Network Scanning Techniques, ” Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology. ACM, 2008: 1-7.
9. Shamsi Z, Nandwani A, Leonard D, et al, “Hershel: Single-Packet OS Fingerprinting,” ACM SIGMETRICS Performance Evaluation Review. ACM, 2014, 42(1): 195-206.
10. Kohno T, Broido A, Claffy K C, “Remote Physical Device Fingerprinting,” IEEE Transactions on Dependable and Secure
11. Computing, 2005, 2（2）: 93-108.
12. Formby D, Srinivasan P, Leonard A, et al, “Who's in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems,” In Proceedings of the 2016 Network and Distributed System Security Symposium (NDSS 2016).
13. 王占丰， 冯径， 邢长友， 张国敏， 许博．IP 定位技术的研究[N]．软件学报，2014，25（7）：1527-1540．DOI：10.13328/j.cnki.jos.004621NaN

致谢

作为一名在大学前有幸接触过信息学的学子，我深知信息与计算机的强大，一直坚信技术真的可以改变世界。因而，虽然初临大学的我虽然对计算机的认识宛如荒漠中零零星星的几株仙人掌，但是我亦曾经抱着对计算机领域的满怀好奇与求知欲望，在大学，正式地，闯进入计算机的世界，踏上对技术的无尽求知之路。

在太理的四年，我遇到了许多与我一同陪着星星敲代码，看着月亮调bug的同学与朋友，亦遇到了许多对我关怀备至的舍友和学长学姐。但是最让心怀感激的，是我遇到了许许多多在学术领域有些自己的一技之长，而诲人不倦的老师。他们在我逐渐成熟迈向校园外的大千世界的这四年中，为我答疑解惑，指点迷津，犹如夜行的小船舶眼中一座又一座泛着暖黄色灯光的灯塔。其中我最想感谢的是指导我四年程序竞赛的林福平老师。而该论文的拟题想法源于中科院信息工程研究所六室的龚晓锐老师给我的想法和方向，后来的成功完稿也离不开各位老师与朋友的指点和鼓励。当然，哪怕行过万里路，我的家人所在之处，才是我永远的家，他们对我的关心就像每顿一顿菜肴上飘过的热腾腾的蒸汽，无声无息却常见而亲切。

**Analysis of IoT devices’ Vulnerability Attack Using a Honeypot**

**Abstract**

IoT (Internet-of-Things) attacks have been accompanied by attempts to exploit inherent threats leading to large scale destruction and corroborated fears about the safety of devices connected to the IoT. The emergence of any new technology will always accompanies with a variety of vulnerabilities that are ready to be exploited. In this paper,

we have proposed a method of analyzing vulnerability attacks by deploying honeypot log data to identify vulnerability attack patterns. Honeypot on an AWS cloud was used to collect cyber incident log data. The log data is analyzed by using Amazon Elasticsearch Service, Amazon Kinesis, Kibana, and POT (EKK-POT). We also aim to find new trends of threats to IoT devices by extracting logs from the EKK stack and enhance current mechanisms to face these alarming trends.

**Keywords:** IoT, Honeypot, Malware attacks, Cloud, Security, Botnet.

1. INTRODUCTION

Internet of Things (IoT) exploits networking, sensing, big data, and computer science technology to deliver complete systems for a product or service. However, several challenges face the widespread usage of IoT, like security, privacy, standards, interoperability, and emerging economies and development. During this paper, we perform a vulnerability scanning of IoT devices using the IoT Shodan and a honeypot for threat and vulnerability analysis.

This permits the analysis of IoT devices' vulnerability attacks using Honeypot by analyzing the vulnerability scanning results and showing how IoT devices could be easily attacked and be exposed by hackers.

A honeypot system could be a program made to operate as an actual attraction system to trick and capture the intruders who try to gain entry into the system. During this process, the invader is keenly tracked and observed, in his or her oblivion. This paper proposes using this technique to know the IoT trends and threat landscape and the way to counter them employing a honeypot.

Honeypots produce an enormous amount of data. It is not easy for general-purpose data analysis tools to analyze such amount of data. During this paper, we invoke Amazon Elasticsearch Service, Amazon Kinesis, Kibana, and POT (EKK-POT) technology to analyze honeypots data as it gives the pliability of searching on any size of data set. With the good advancements of technology, society has become greatly connected through the various devices that currently have a direct connection to the Internet. These devices have become vulnerable to attacks as indicated Matheu-García et al (2019).

There are many forms of attack that can be performed on the IoT system to make it more vulnerable, including Man-In-The-Middle (MITM), Sniffing, Denial of Service (DoS), Crypto graphical invasion, Botnet invasion, Dos attack (Denial of Service) and Distributed DoS (DDoS). The main problem with the growing technology is the fact that a huge number of devices are being joined to the global network. Likewise, many of these devices are used by individuals with very small knowledge of the technology they are using, making them extremely vulnerable to attacks and threats (Fahrnberger, Gopinathan, & Parida, 2019). Besides, it has been noticed that the security measures of systems are not expanding proportionally with the growth of the systems itself. Currently, the Internet is considered as a laboratory for hackers and exploiters to use their knowledge to their benefit, which most of the time result in harm to others (Kamesh & Sakthi, 2016). This calls for a deﬁnition of a cyber-security certiﬁcation framework Matheu-García et al (2019).

The main function of the Intrusion Detection System (IDS) system mechanism is to detect and correct malicious or irregular activities in a network.

This tool runs constantly in the background and does not cause great interference in the normal functioning of the network. More recently, a new alternative began to be used involving the use of honeypots and honeynets (Spitzner, 2003).

In 1999, Spitzner linked a computer to the Internet to execute applications with clearly known exposures. The idea was to make this system act as a honey pot (from the name) to attract the attackers.

It was surprising to see that in less than 15 minutes the host had already been compromised (Spitzner, 2003).

Then the honeypot idea emerged: a network of resources to be attacked and compromised (invaded).

It meant a honeypot could be tried and invaded. By permitting such attacks, it is possible to register the events taking place in fact and then provide the necessary information on the plans used by the attackers (Spitzner, 2003).

We examined the EKK-POT over a few weeks, and data were recorded. Afterward, the logs were collected in an elastic search index, and Kibana was used to visualize the analysis-results of these data.

This Paper gives a significant amount of contribution to the IoT security landscape to invite more work in this field. Furthermore, it used EKK stack tools and makes the deployment steps clear and concise to study the data in an integrating environment with a suite of great tools for cybersecurity.

1. RELATED WORK

Cyberspace search engines such as Shodan, FOFA, and other public platforms are introduced to collect honeypots servers.

Only a section of the attacking devices of the honeypots was used to estimate the number of devices used globally and of a given type. Devoted search services such as Shodan, Censys, and ZoomEye help scan IP spectra for related services, polling them and indexing the outcomes. The search carried out for the most frequent headers, routers, and DVRs using Shodan, Censys, and ZoomEye.

Shodan was once the main search tool of the IoT before 2013 when Censys came up as the free competing tool. It also served as an IoT search engine, being dependent on the same fundamental principles. However, as the creator states, it was more accurate in searching for potential IoT vulnerabilities.

Censys provides a collection of devices that share a common vulnerability, such as the ones vulnerable to Heartbleed.

This review is available on all existing honeypot methods that scientists have improvised with time.

* 1. Honeypot Attacks

One vital condition for honeypots is their capacity to maintain undetected attacks: hiding the details of the honeypot. Shodan is a special search tool, which crawls on the Internet and tries to find linked devices such as IP web-cameras and routers (Roland Bodenheim, et al, 2014). In the progressive procedure of crawl and index, the Shodan system updates the database of all systems as well as services connected to or exposed to the Internet. Recently execution and deployment of Shodan detection mechanisms were done for the identification of honeypots. It did several probes and monitoring, and then created scores for every checked device. From the value of this score, Shodan decides if the system is a honeypot system or not.

* 1. Security of IoT application protocols

This sub-section discusses the existing circumstances and problems associated with the current communication protocols used in IoT applications and devices.

**2.1.1. Present circumstances**

Although the attention for the security of IoT technology is increasing, it is agreed that it still has not matured enough. The security problems for the application protocols are normally considered from three distinct aspects (Al-Shaer, Wei, & Hamlen, 2019) . Flaws in the protocol itself, troubles during the protocol implementation and vulnerabilities during the implementation itself. **Table 1** shows the security mechanisms applied in each protocol.

Table (1): Security Mechanism of IoT communication protocols

|  |  |
| --- | --- |
| MQTT | Simple Username/password Authentication, Transport Layer Security/ Secure Sockets Layer (TLS/SSL) for data  encryption |
| XMPP | Simple Authentication and Security Layer (SASL)  authentication, TLS/SSL for data encryption |
| AMQP | SASL authentication, TLS/SSL for data encryption |
| CoAP | Datagram Transport Layer Security (DTLS) / Internet  Protocol Security (IPsec) |
| JMS | Vendor speciﬁc but typically based on TLS/SSL. Commonly used with Java Authentication and  Authorization Service (JAAS) API |
| Simple Object Access  Protocol (SOAP) | Address by WS-Security |

It can be observed that Transport Layer Security cryptographic protocol is used in MQTT, XMPP, AMQP, and JMS. Datagram Transport Layer Security is used in CoAP while WS-Security is used in SOAP protocol. Simple Authentication and Security Layer (SASL) is the most used authentication framework by these protocols.

* + 1. Present Problems with Security of IoT

As previously stated, IoT security problems are taken from three-point of views. The Exposures of the connection protocols:

There are some security issues common with all protocols and others which are specific to a specific protocol.

First, common issues are introduced, and then the specific issues are discussed.

* + 1. Common Problems with protocols:

The common problems include security of booting, ﬁrewall system, and security in the update and patch problems in all the protocols. For example, only authorized software applications ought to be allowed to download in the linked device. That is why precautions must be taken, such as digital signatures or keys for encryption are needed (Fahrnberger, Gopinathan, & Parida, 2019). Therefore, it is considered a security challenge with large deployment and limited resources (Maarof, Senhadji, Labbi, & Belkasmi, 2018).

CoAP:

According to Nastase (2017a) and Nastase (2017b), DTLS has a problem because its design does not match CoAP. First, DTLS cannot enable multicast. Among the weaknesses of using DTLS in CoAP or other proposals for solving the problems(Görmüş et al., 2018). Nevertheless, the plans may not be proper for CoAP and can cause security issues. More research is required.

CoAP: Issues with CoAP’s “NoSec Mode"Four safety modes are brought in (Görmüş et al., 2018), with the initial one being "NoSec Mode". During this mode, there is the transmission of a CoAP message with no security methods implemented.

MQTT: Issues with encryption as well as authentication of MQTT, (Salman & Jain, 2017) discusses a new protocol referred to as Secure MQTT (Singh, Rajan, Shivraj, & Balamuralidhar, 2015), which applies encryption based on "lightweight attribute-based encryption". SMQTT is different from MQTT due to the weakness of MQTT.

XMPP: Issues with SASL authentication There may be possible problems with SASL authentication as explained by Melo (2013) , concerning the security factor that several SASL methods cannot provide enough safety.

Additional protective services may be required to safeguard against the attacks. In addition, XMPP uses SASL for authentication (Lehto & Neittaanmäki, 2018) .Therefore, any form of weakness in SASL is also a weakness in XMPP.

UPnP: Known UPnP weakness.The problem with the UPnP is not a new subject. It has had lots of discussions concerning it on various publications online (Michael Mimoso, 2015)(Brewster, 2013) and in the academic area (Michael Mimoso, 2015) (Al Hasib & Mottalib, 2010). Michael Mimoso (2015) points to security weaknesses in UPnP that put millions of domestic networking tools at the danger of being compromised. (Brewster, 2013) also gives explanations on the use of vulnerabilities of the UPnP to run the attacks.

Protocol Execution Problems:Security issues related to development could occur during the implementation of the protocol, which includes building and installing the server. These issues could vary from bugs, weaknesses, and lack of validation in addition to other issues. Unfortunately, these issues can compromise the full execution. The weaknesses can be mapped to Common Vulnerabilities and Exposures (CVE), a database with a list of known exposures to software applications, including Operating Systems, archives, and frameworks.

The implementations can be from open or closed sources. Various databases share a common collection of CVEs identifiers describe all the present information concerning the vulnerability, including vendor, product, time, vulnerable forms, the kind of vulnerability and description (Russell & Duren, 2018) . This makes it easy to identify exploitation directions for applications on the IoT or IoT platforms with the protocols.

Vulnerabilities When integrating with IoT:Since IoT platforms integrate various IoT techniques and protocols to operate as a single unit, therefore, the IoT is viewed as an immature platform and is prone to several security attacks. Some specific approaches can be taken to put together, analyze, and finally identify the pattern of threats on the IoT platforms.

* 1. Types of Attacks

The different forms of invasion can be conducted by malicious users such as Port scan, MITM attack (Man-In-The-Middle) and Denial of Service (DoS). A brief overview of these attacks is given on these attacks.

* + 1. Port scan attack

This is among the passive attacks, which means it does not harm the system or the server. However, it can store all the details related to the machine or the server of the victim. A port scan aims to find any active port among a spectrum of port addresses for the server by sending a request to the client (Cruz-Cunha, 2014) . The different port scans include Connect Synchronization (SYN) scanning; UDP scan, XMAS scan, Acknowledgement (ACK) scanning, and FINISH (FIN) scan are commonly used by the popularity of the scans applied by the attacker.

* + 1. Man-In-The-Middle (MITM)

In this kind of attack, the invader tries to intercept communication between two partners. The attacker emulates a reliable connection and transfers messages from one party to the other to establish the belief that this is a private conversation. Figure 1 shows how this attack is made; instead of the network transferring the information between the two users, the attacker does this part without the victims knowing; in exchange, all the transferred data is accessible by the attacker.

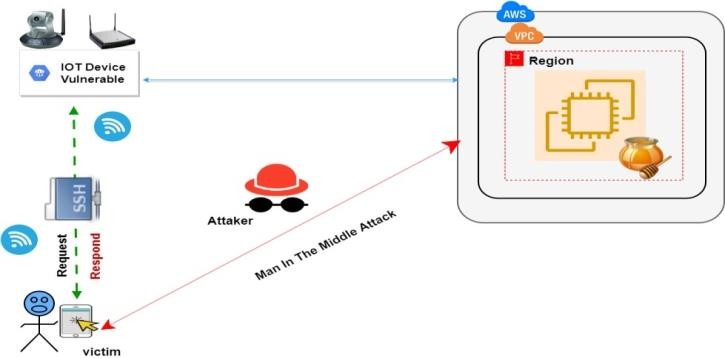


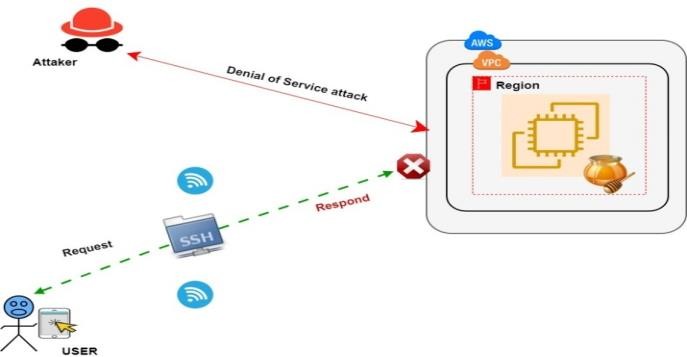
Figure (1): Man in the middle atta

* + 1. Denial of Service attack

This attack aims to close the machine, network, or software for some time to deny the use of the service. In this attack, the system/network is pinged with heavy spam requests that the system/network cannot handle, so it crashes. Governments and banks are usually the main targets for this attack, as it causes them to lose time and money to re-run the system/network.

This type of attack is shown in Figure 2

Figure (2): Denial of service attack



* + 1. Distributed Denial of Service attack

This is the same as denial of service attack, but it differs in the use of many systems by the attacker to take down the system/network. A typical DDoS attack has a master and a zombie.

The master refers to the invaders who begin the attack by exploring all the vulnerabilities in the system and finds the weak systems and attains control over them by applying infection on the systems via malware or by bypassing.

* + 1. Sniffing attack

One of the most common types of attacks is the sniffing attach, carried out over connections and wireless links. It enables the attacker to gain access to and collect, and change data from the user’s system by gaining access over it(Salman & Jain, 2017). The attacker uses two most important methods in the sniffing, including ARP poisoning and methods of theft of TCP sessions. ARP poisoning as a sniffing method is used for attacking the network using packet-spoofing invasion and router-based weaknesses.

On the other hand, the TCP session theft approach is applied in catching the IP address of the packets at the source and destination in promiscuous mode. The attack is illustrated.

* 1. Honeypot technology

After showing that the previously explained protocols determine a standard for normal behavior, the processes of collecting, analyzing, and identifying infrequent activities are done in a dynamic protocol analysis, which includes payload (content) handling, context, and overall pattern examination. In the case of IoT, protocols do not define only the way IoT tools can exchange data but also their inclusion into the IoT platform.

Honeypot technology is considered an approach from many that processes data and recognizes the pattern(Salman & Jain, 2017) Mainly, Honeypot technology is a mechanism that is used to detect malicious activity through simulation of an actual system in a safe environment. Therefore, the invader is lured into the system, thinking it is an actual one; however, it does not affect the system since it is protected, and all activities the invader is being recorded.

The major idea about the Honeypot is to deploy the communication protocols previously mentioned on software that the hacker attacks while recording their activities.

* + 1. Advantages of Honeypot

One advantage of a honeypot is that several security solutions exist in the market and anybody can explore the choices on the internet to find the most appropriate solution to fit their needs. According to Mokube and Adams (2007), honeypots capture invasions and provide information concerning the kind of attacks and where necessary due to the logs, one can see the added information on the attack. New threats can be noticed, and newer security options can be formed by having a look at them. Further analysis can be done by having a look at the kind of misbehaviors. It aids in understanding more attacks, which may take place. Honeypots are not heavy when it comes it comes to capturing data. They are mainly handling the inflow of malicious traffic. Consequently, the data that caught is not the whole traffic. Concentrating only on the malicious traffic simplifies the investigation. For that reason, honeypots become very beneficial. For the few malicious attacks, there is no need for large data storage and new technology to utilize.

Any type of computer can be configured for use as a honeypot system. Therefore, a honeypot system does not increase the cost of the security system. They are easy to understand, configure and deploy. They have no complicated algorithms and need no need for updates and modifications. Since honeypots can detect any malicious attackers, it provides more ideas and does deepen the security solutions.

* + 1. Disadvantages of Honeypot

Since there are several vital advantages of the application of honeypots, they also have disadvantages. As Mokube & Adams (2007) state, they can only pick data while the malicious intruder attacks the system live, otherwise, catching the information becomes difficult. Any attack taking place on another system cannot be detected by the honeypot on a different IoT system.

Consequently, attacks on the external honeypot system can damage other IoT systems and lead to huge problems. The second disadvantage of a honeypot is the fingerprinting of honeypots by an experienced attacker. It is very easy for the attacker to study whether he attacks the honeypot system, or an actual system and this identification can determine his or her activities.

Fingerprinting enables the distinguishing between the two but is not a desirable outcome of the experiment. The honeypot can be utilized as a zombie to reach external systems and weaken their security systems, and this is hazardous.

Related work, as much we have done the research, no study has been done where the complete methodology of honeypot deployment had been explained. Moreover, simulating devices and collecting data of attacks with the help of the EKK stack has not been used to find the unknown attacks by IoT honeypots.

The study focused on the installation of IoT honeypots. They proposed a procedure to install necessary honeypots on IoT systems, given the rank of popular chosen tools and the request to avoid detecting the trap nodes as honeypots among some of the most used IoT scanners. They found that these IoT tools are not designed with adequate security standards, and they are doing vital and sensible roles in our life's specific to our privacy and security (Acien, Nieto, Fernandez, & Lopez, 2018).

To attain the objective of this research, the steps below are followed.Carry out a study on shared IoT based protocols including MQTT, XMPP, AMQP, and UPnP to access and comprehend their design. In this survey, we provide a platform for focusing on precise protocols and possible research sub-topics.

In answering the question of whether the honeypot techniques are appropriate for identifying security problems on the IoT platforms,A Proof of Concept (POC) of the ThingPot applied. By mainly executing the ThingPot POC runs for 46 days on the four nodes (thee of the nodes are for REST API while one is for XMPP), it catches a massive amount of valued information. By conducting detailed data analysis, they found and summarized hackers' actions besides the IoT system. Thus, ThingPot has presented the ability for identifying security problems on the IoT Platforms and demonstrates that honeypot equipment is suitable for identifying safety problems on the IoT system (Wang, 2017).

Intelligent interaction was presented as a method to build IoT honeypot automatically and intelligently. This leverages multiple heuristics as well as an interactive machine learning mechanism for customizing the scan of process and enhances the replying logic to extend to capture the exploit code.

They proposed a general framework to build an intelligent-interaction honeypot for the IoT tools. They explained why and how we require intelligent-interactive honeypot (Wang, 2017).

The aim of interactive intelligence is for learning the right behaviors for interaction with the clients from the zero-knowledge on the IoT tools. The appropriate reactions to the clients ought to be fit for the expansion of the session with potential hackers, tricking them to pass the inspection, while sending the exploit needs as described by Maarof et al (2018). For this goal to be accomplished it needs the system to collect the valid responses as the candidates automatically, via interaction with the hackers; the learning procedure aids the honeypot to enhance the right behaviors for each request (Ramirez et al., 2017) . Any release of the source code has to lead to 2 important outcomes:The principal and most observable that it has provided cybercriminals and hackers not only a functional process for the creation of an active botnet but also the creation of a customized one.

The second outcome is the formation of demand for IoT safety following its demonstrated the weakness of IoT tools and related operating systems for simple malware infection and enslavement trials. The devices are not only weak, but some are also infected.

From the publication of the source code, there is an increased need for IoT safety and a race to enslave, increasingly Internet of Things devices by attackers and bot-herders. However, about 6 billion IoT devices are connected to the internet and the number is expected to reach 20 billion during the year 2020. Hackers compete aggressively, prompting the initiation of malware to erase other malware from manipulating the device, which simplifies espionage, data stealing, or launching of lots of DDoS attacks and looking around for other weak devices and adding them to the botnet (Antonakakis et al., 2017).

IoTPOT proposed a unique honeypot to imitate Telnet services of many IoT tools to monitor and record continuous active attacks on the IoT devices in detail. IoTPOT has distinguished low-interaction responses to cooperation with the backend high-interaction computer-generated environments called IoTBOX. IoTBOX controls many virtual environments usually applied through fixed systems for various Central Processing Unit (CPU) designs (Minn et al., 2015). In the 39 days of operation, there was an observation of 76,605 attempted downloads of malware, an increase from 16,934 visits to the Internet Protocol (IP). (Ramirez et al., 2017) confirm that present honeypots could not capture any of these binaries. Telnet protocol that as telnet password honeypot and honey because they are unable to manage different inflowing commands sent via the hackers(Shodan, 2019) .

To analyze the captured malware binaries; they proposed IoTBOX as the first environment for malware analysis for the IoT devices. IoTBOX runs 8 CPU architectures, that spa ARM, PPC, and MIPS.

The sandbox exploration of 17 samples through IoTBOX shows that the samples were used for performing ten distinct types of DDoS attacks as well as scans on port 23. Ultimately, merging the outcomes of the observations of IoTPOT and the analysis of sandbox IoTBOX, we confirm that:

At least 4 distinctive malware families are spreading by Telnet

Their typical activity is performing additional broadcast over Telnet and DDoS

Some families develop frequently, update quickly, and ship binaries for the varieties of CPU designs, even within the restricted observation period of 39 days (Minn et al., 2015). An automatic framework for detection and identification of compromised devices of botnets and analysis of malware was developed. They proposed some essential research questions like “What are the characteristics of compromised machines belonging to botnets?”

Dionaea a low interaction, server-side honeypot which emulates a vulnerable system or device was installed on the Amazon Elastic Compute Cloud (Amazon EC2) to record attack data and malicious software binaries. Virus Total was utilized in malware grouping while and Shodan was used in device identification as described by Thangavel et al (2014). The top malware variations, types of devices, products, countries, organizations and open ports were examined to find compromised devices, with the continued existence of critical router weakness(Chinn, 2015) .

Thus, we proposed a deception tool based on IoT Honeypot. The deception idea is an emergent class of cyber-security protection. Deception equipment can analyze and detect weaknesses and protect the system and devices from zero-day and radical attacks frequently. They are automated, provide, and accurate awareness of malicious motion in the internal networks, which could be hiding among other forms of cyber guards. The deception concept enables a security system with a more proactive position to seek to trick the invaders, detect them, and conquer them,Permitting the innovativeness to yield to normal processes.

1. EXPERIMENT SETUP

This section describes the experimental setup using the central logs and observing server based on ): the EKK stack (Amazon Elasticsearch Service, Amazon Kinesis, and Kibana) which helps to present data, create visualization and a dashboard for any size of data. The research is based on the EKK stack (Amazon Kinesis, Amazon Elasticsearch Service and Kibana). Elasticsearch refers to a search engine with analytics. Logstash refers to a server-side pipeline for processing of data that takes in data from numerous sources concurrently, transforms the data before sending it to a "stash" like Elasticsearch. Kibana allows users to visualize data with analysis charts and graphs in the Elasticsearch.

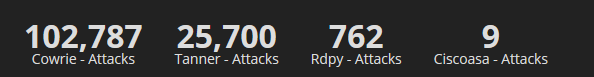
We used the alternative to the known log combination solution, the ELK stack (Elasticsearch, Logstash, and Kibana): the EKK stack (Amazon Elasticsearch Service, Amazon Kinesis, and Kibana). The EKK solution removes the simple heavy lifting of deployment, management, and scaling of the log aggregating resolution. Through the EKK stack, we emphasize an analysis of logs and debugging of the application, instead of management and up-scaling the system that groups the logs.

* 1. Data Analysis using Kibana

In Kibana, we performed several keyword searches using Elasticsearch. The main goal was to find attack events in our honeypots. We have identified several events from the log data analysis. Kibana was used as an open-source tool for visualizing and exploring data. Also, it was applicable in log and time-series analytics, application checking, and operating intelligence use cases. Kibana was considered a very useful tool helpful feature such as bar graphs, trend lines, histograms, line graphs, and other visual figures. Kibana allowed tight integration with Amazon Elastic search, making it our default choice for visualization.

The EKK implementation eradicated the homogeneous heavy lifts of deploying, management, and scaling of the log aggregate resolution. With the focus on the EKK stack, the emphasis was on analyzing logs and correcting the application, instead of management and up-scaling the system to aggregate these logs.

* 1. The most common attack

Figure (3): Most common attacks

The Cowrie honeypot experienced more than 100,000 attacks. It was not surprising that the majority of the Telnet & SSH honeypot had the most attacks. [Cowrie](https://github.com/cowrie/cowrie) is a standard high-interaction honeypot planned to entice and log physical force occurrences and any shell dealings by the attacker. Its main purpose is to interact with an attacker whilst monitoring how they behave when they think they’ve breached a system.

It was no surprise that of the 102,787 attacks, 32,607 came from China, with the United States of America coming second with a modest 14,115 attacks.

* 1. What they did once they logged in

Understanding what attackers are doing once they gain access is key to building a defense. Not every attack is going to come from an unknown user from an unknown IP. With most SIEMs, it is possible to trigger alarms if a user uses any of the listed commands. This way, should a user on the network ever lose their credentials, if any of those commands are used, we can tag it as suspicious behavior and set up rules to block the account till there’s confirmation it’s not a breach.

We noted that Cyber-attacks are more than just the Big Four (Russia, China, USA & Iran). For security teams, understanding this is essential for building your strategy. Regularly reviewing where the service attacks occur can help determine if there is any internal sabotage going on.

We did notice however that a few of the honeypot Dockers went on knocking over after a small brute force and we set up an Elastic cluster separately to monitor the server to see if it was being attacked. We propose using the latest version of Elastic to use their SIEM function to monitor some of the test environments. We also recommend for future research, to declutter the honeypots and deploy them individually for better analysis in addition to making use of machine learning from Elastic.

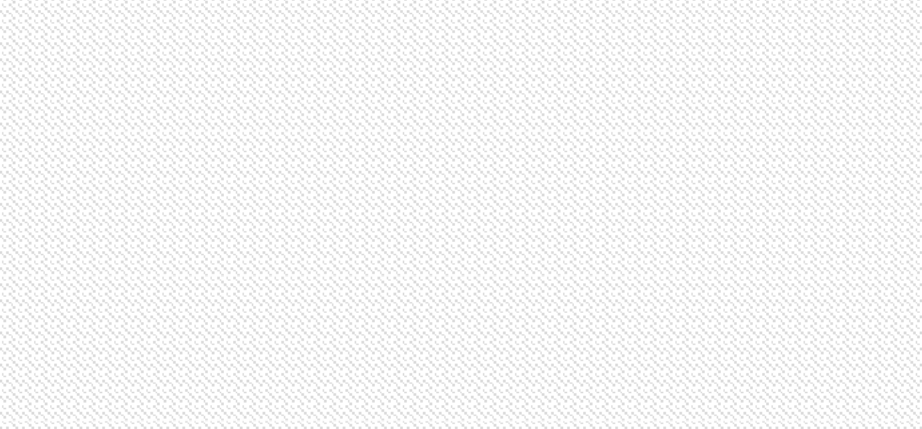
* 1. Statistics

Statistical analysis of the results provided a better insight to the readers about what we observed in the EKK experiment. The first section of this analysis includes the whole structure of attacks on honeypots. It carries the wider idea of attack vector as well as that of common IPs that link different honeypot systems. It also carries the IPs that have had the greatest number of attacks in the connection of all the procedures.

Analysis of this information from REST logs shows that the invaders noticed the IoT environment and all devices. In general terms, all the attackers look for devices such as Philips Hue, Belkin Wemo and TPlink. Particularly, they are interested in getting information regarding smart devices and more importantly the taking over of control. The methodology that the attackers seemed to prefer is first a general scanning to look for vulnerabilities, followed by a more targeted and specific attack via brute force or fuzzing. We set up the T-honeypot (traps) that imitated various devices and connected to the Internet to see what happened to them ‘in the wild’. Through

the entire 24-hour time span, the attempts of connection attacks were in the tens of thousands from various unique IP addresses.

**Count**



200

180

160

190

188

170173

164

150152

160

148

140

120

100

80

60

40

20

0

123 123

114115

122

120

103

93

101

100

83

84

72

54

58

56

54

44

36

44

**Time stamp for 12 Hours by the day**

29-Feb-20

28-Feb-20

27-Feb-20

26-Feb-20

25-Feb-20

24-Feb-20

23-Feb-20

22-Feb-20

21-Feb-20

20-Feb-20

19-Feb-20

18-Feb-20

17-Feb-20

16-Feb-20

15-Feb-20

14-Feb-20

13-Feb-20

12-Feb-20

11-Feb-20

10-Feb-20

09-Feb-20

08-Feb-20

07-Feb-20

06-Feb-20

05-Feb-20

04-Feb-20

03-Feb-20

02-Feb-20

01-Feb-20

Figure (4): The number of unique hits on honeypots from unique IP addresses. Feb 01-Feb 2020.

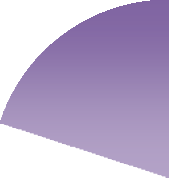
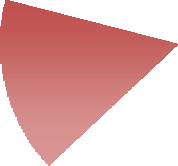
In many instances, the attempts to do connections applied to the telnet protocol;the others applied SSH.

TABLE (3): DISTRIBUTION OF ATTEMPTED ATTACKS BY TYPE OF CONNECTION

|  |  |
| --- | --- |
| Type of connection | Number |
| Telnet | 6700 |
| SSH | 1182 |
| Total | 7882 |

We analyzed the types of devices from which the attacks originated. More than 60% of the attack could be seen in the form of DVR services and IP-based cameras, but approximately 17% were various forms of network tools as well as routers from several main manufacturers. About 1% were Wi-Fi repeaters while other network hardware, TV tuners, VOIP devices, exiting nodes at Tor, printers and all the devices that make up the ‘smart-home’. Other devices could not be identifying and were simply registered as unidentifiable devices, forming 20%.

Figure (5): Distribution of attack sources by device type Feb 10-Feb29 2020



Unrecognized Devices

20%

TV/IP Telephony

1%

Network Devices

16%

DVR or IP Cameras

63%

**DVR or IP Cameras**

**Network Devices**

**TV/IP Telephony**

**Unrecognized Devices**

Several connections attempted to get into some IP addresses at the honeypots system

we constructed, and they responded to the HTTP requests. Naturally, so many devices used each IP address through the NAT technology application. It was common to find that the device, which responded to the HTTP server request was not the ones that caused attacks on the honeypot, as is usually expected.

There were web pages and device controlling panel in response to and in the maintenance of the request and carrying out monitoring through the camera. With the returned page, the devices can be identified. The table below shows the most often used headers for web pages detected by the invading devices:

TABLE (4): NUMBER OF ATTACKS ON DEVICES, CLASSIFIED BY TITLE

|  |  |
| --- | --- |
| Devices | Number of IP addresses |
| DVR or IP Cameras | 4966 |
| Network Devices | 1261 |
| TV/IP Telephony | 79 |
| Unrecognized Devices | 1576 |
| Total | 7882 |

TABLE (5): NUMBER OF ATTACKS ON DEVICES, CLASSIFIED BY HTTP TITLE

|  |  |
| --- | --- |
| HTTP Title | Devices |
| WEB SERVICE | 2569 |
| NETSurveillance WEB | 1432 |
| Dvrdvs | 621 |
| DVR Components Download | 777 |
| NetDvrV3 | 960 |
| IVSWeb | 1523 |
| Total | 7882 |

Several IP addresses are a potential vulnerability. The vulnerable devices are shown below.

TABLE (6): NUMBERS OF IP ADDRESSES OF VULNERABLE DEVICES: IP CAMERAS AND DVRS

|  |  |
| --- | --- |
| HTTP Title | Devices |
| Eltex NTP | 653 |
| RouterOS router | 823 |
| GPON Home Gateway | 1878 |
| TL-WR841N | 1969 |
| ZXHN H208N | 1038 |
| TD-W8968 | 642 |
| iGate GW040 GPON ONT | 879 |
| Total | 7882 |

He honeytraps we setup not only recorded attacks coming from network hardware classed as home devices but also enterprise-class hardware.

Amid all IP addresses that released the attacks, a number of them checked and managed the devices with initiative and safety links, such as:Point-of-sale devices at stores, restaurants and filling stations;Digital TV broadcasting systems;Physical security and access control systems;Environmental monitoring devices;Monitoring at a seismic station in Bangkok;Industry-grade programmable microcontrollers;Power management systems.

It is not easy to confirm that these devices have an infection. However, some attacks on the honeypots arrived from the IP addresses used by these devices, which means at least one or more devices were infected on the network where they reside.

* 1. Geography of infected devices

As already stated, the majority of the infected tools are IP-based cameras and also DVRs. A large number of them are prevalent in China and Vietnam. Others are found in Russia and Brazil among many other countries. The geographic distribution of the devices with the IP addresses that attacked the honeypots is represented in the pie chart below:



**4.485%.7; 34%%; 6%**

**4.13%; 4% 5.97%; 6%**

**6.21%; 6%**

**3.95%; 4%**

**12.26%; 12%**

**6.92%; 7%**

**1..6895%;; 2%**

**3.24%; 3%**

**3.90%; 4%**

**11..5357%;; 21%**

**1.65%; 2%**

**17.44%; 17%13.95%; 14%**

**01..9095%%;;11%%**

China

vietnam Russia Brazil Turkey Tawan Iran

South Korea United States India Argentina Pakistan

Italy Australia Thailand Romania

United Kinddom Colombia

Figure (6): Breakdown of attacking device IP addresses by country Feb 02-Feb 29 2020

* 1. Geographical distribution of server IP addresses

We recorded over 100000 hacking attempts and more than 8000 unique IP addresses from which malware for IoT devices was downloaded. The breakdown of the IP addresses is shown below by countries.

Table 7 Geographical distribution of server IP addresses from which malware was downloaded to devices

|  |  |
| --- | --- |
| Country | Unique IPs |
| Vietnam | 2136 |
| Taiwan, Province of China | 1356 |
| Brazil | 1124 |
| Turkey | 696 |
| Korea, Republic of | 620 |
| India | 504 |
| United States | 429 |
| Russian Federation | 373 |
| China | 361 |
| Romania | 283 |
| Total | 7882 |

It can be deduced that the difference is due to the presence in some of these countries of bulletproof servers, meaning it’s much faster and easier to spread malware than it is to infect IoT devices.

1. CONCLUSION

Honeypot technology is helpful and utmost important part of security strategies in a network. Although honeypot technology is constantly improving, attackers are constantly searching for vulnerabilities in IoT devices and in honeypots to identify them. We have analyzed IoT devices' vulnerability using honeypot data collected from AWS. The data was analyzed using an EKK stack for log data analysis and visualization. It is worth noting that EKK uses elasticsearch, which helped to identify various types of vulnerabilities. It has become apparent that honeypots are constantly being targeted by attackers.

Throughout the paper, we have outlined how to set up a honeypot using the EKK technology and to emphasize the threat to the community of IoT devices’ vulnerabilities and their users. The average Internet daily user is not aware of the dangers as these devices have not been designed with enough security measures. Also, they are increasingly performing important and sensible roles and impacting privacy and security. The experimental result provides a reference for the improvement of honeypots and promotes the development of honeypot technology using EKK technology.

Unlike the old-style security discovery approaches, the honeypot structure, particularly on the IoT research area, is meant for attacks and habitually monitor possible attacks by exploring network packages or log files. Using the EKK technology we were able to extract the exact IoT devices’ vulnerabilities and threats, effective actor tactics, techniques, and procedures from these data. This provides a powerful tool that can be used to generate more defense strategies.

* 1. Recommendations

In future work, we aim to extend the analysis of IoT devices’ vulnerability attack using Machine Learning in AWS EKK POT. Recommendations for further research include but are not limited to the following:

Improvement of the honeypot mechanism to prevent recognition by attackers, and silently capture their behaviors. This calls for intelligent techniques to automatically check whether a remote server is running a honeypot service or not.

Furthermore, research should extend the Honeypot with capabilities to stimulate even more architectures and environments that are common on IoT devices.

We, therefore, recommend further research on Honeypots with Machine Learning based Detection Framework for defending IoT based attacks.

As future work, we can deploy the Honeypot in a larger public network and evaluate its average session time and data capture performance.

1. References

Acien, A., Nieto, A., Fernandez, G., & Lopez, J. (2018). A comprehensive methodology for deploying IoT honeypots. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*,*11033LNCS*, 229–243. <https://doi.org/10.1007/978-3-319-98385-1_16>

Al Hasib, A., & Mottalib, M. A. (2010). Vulnerability analysis and protection schemes of Universal Plug and Play protocol. *Proceedings - 2010 13th IEEE International Conference on Computational Science and Engineering, CSE 2010*, 222–228. <https://doi.org/10.1109/CSE.2010.37>

Anagnostakis, K. G., et al. "Detecting targeted attacks using shadow honeypots." Proceedings of the 14th conference on USENIX Security Symposium. ACM, 2005. 129-144.

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Arbor, A., Bursztein, E., … Zhou, Y. (2017). Understanding the Mirai Botnet. *USENIX Security*, 1093–1110. https://doi.org/10.1016/j.religion.2008.12.001

Al-Shaer, E., Wei, J., & Hamlen, K. W. (2019). *Autonomous Cyber Deception: Reasoning, Adaptive Planning, and Evaluation of HoneyThings.* NY: Springer.

Brewster, T. (2013). How Attackers Can And Will Exploit UPnP Weaknesses[20]. Retrieved from

https:/[/www.silicon.co.uk/workspa](http://www.silicon.co.uk/workspace/how-attackers-will-exploit-upnp-105868)c[e/how-attackers-will-exploit-upnp-105868](http://www.silicon.co.uk/workspace/how-attackers-will-exploit-upnp-105868) Chinn, R. (2015). Botnet Detection: Honeypots and the Internet of Things, *I*. Retrieved

from https://msmis.eller.arizona.edu/sites/msmis/files/documents/sfs\_papers/ryan\_chin n\_sfs\_masters\_paper\_0.pdf

Christian Seifert, Ian Welch, and Peter Komisarczuk. Taxonomy of honeypots, 2006.

Cruz-Cunha, M. M. (2014). *Handbook of Research on Digital Crime, Cyberspace Security, and Information Assurance.* NY: IGI Global.

CVE Details. (2019). CVE security vulnerability database. Retrieved from https:/[/www.cvedetails.com/](http://www.cvedetails.com/)

ElasticSearch-Contributors. (2013). ElasticSearch Documentation. Retrieved from <http://www.elasticsearch.org/guide/>

Fahrnberger, G., Gopinathan, S., & Parida, L. (2019). *Distributed Computing and Internet Technology: 15th International Conference, ICDCIT 2019, Bhubaneswar, India, January 10–13, 2019, Proceedings.* Los Angeles: Springer.

Görmüş, S., Aydın, H., & Ulutaş, G. (2018). Security for the internet of things: A survey of existing mechanisms, protocols, and open research issues. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *33*(4), 1247–1272. https://doi.org/10.17341/gazimmfd.416406

Guide, U. (2012). Amazon Elastic Compute Cloud User Guide for Linux Instances. *Copyright © 2016 Amazon Web Services*. Retrieved from <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-ug.pdf>

Mokube, I. and Adams, M. (2007). “Honeypots: concepts, approaches, and challenges,” in Proceedings of the 45th Annual Southeast Regional Conference, pp. 321–326, ACM, Winston-Salem, NC, USA, March 2007. View at Publisher

Kamesh, & Sakthi Priya, N. (2016). A survey of cybercrimes Yanping. *Security and Communication Networks*, *5*(February), 422–437. https://doi.org/10.1002/sec

Kemmerer, R.A. and G. Vigna. "Intrusion detection: a brief history and overview." Computer 2002: 27-30.

Lehto, M., & Neittaanmäki, P. (2018). *Cyber Security: Power and Technology.* NY: Springer.

L. Spitzner, (2003). Honeypots: Tracking Hackers. Addison-Wesley, [Online]. Available:

<http://www.tracking-hackers.com/book/>

L. Spitzner, (2003) “The honeynet project: trapping the hackers,” IEEE Security & Privacy, vol. 1, no. 2, pp. 15–23, 2003.

Maarof, A., Senhadji, M., Labbi, Z., & Belkasmi, M. (2018). Security on the internet of things. *Security and Privacy in Smart Sensor Networks*, 105–121. https://doi.org/10.4018/978-1-5225-5736-4.ch006

Matheu-García, S. N., Hernández-Ramos, J. L., Skarmeta, A. F., & Baldini, G. (2019). Risk-based automated assessment and testing for the cybersecurity certification and labeling of IoT devices. *Computer Standards & Interfaces, 62,* 64–83.

Mokube, I. & Adams M.,2007.Honeypots:Concepts,Approaches,and Challenges. ACMSE 2007, March 23-24, 2007, Winston-Salem, North Carolina, USA ,pp.321-325.

Melo, A. C. M. (2013). Simple Authentication and Security Layer (SASL). *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699. https://doi.org/10.1017/CBO9781107415324.004

Michael Mimoso. (2015). Filet-o-Firewall UPnP Security Vulnerability in Home RoutersThreatpost[19].Retrieved from https://threatpost.com/upnp-trouble-puts-devices-behind-firewall-at-risk/114493/

Minn, Y., Pa, P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., & Rossow, C. (2015). IoTPOT: analyzing the rise of IoT compromises. *Emu*, *9*, 1.

Mohammed, M., & Rehman, H.-u. (2015). *Honeypots and Routers: Collecting Internet Attack.* Washington: CRC Press.

Nastase, L. (2017a). Security in the Internet of Things: A Survey on Application Layer Protocols. *Proceedings - 2017 21st International Conference on Control Systems and Computer, CSCS 2017*, (July 2016), 659–666. https://doi.org/10.1109/CSCS.2017.101

Nastase, L. (2017b). Security in the Internet of Things: A Survey on Application Layer

Protocols. *Proceedings - 2017 21st International Conference on Control Systems and Computer, CSCS 2017*, (July 2016), 659–666. https://doi.org/10.1109/CSCS.2017.101

Nicholson, D. (2017). *Advances in Human Factors in Cybersecurity: Proceedings of the AHFE 2017 International Conference on Human Factors in Cybersecurity, July 17−21, 2017, The Westin Bonaventure Hotel.* Los Angeles, California, USA: Springer.

Ramirez, D., Uribe, J. I., Francaviglia, L., Romero-Gomez, P., Fontcuberta I Morral, A., & Jaramillo, F. (2017). IoTCandyJar: Towards an Intelligent-Interaction Honeypot for IoT Devices. *Journal of Materials Chemistry C*, *6*(23), 6216–6221. https://doi.org/10.1039/C8TC01582A

Russell, B., & Duren, D. (2018). *Practical Internet of Things Security: Design a security framework for an Internet-connected ecosystem, 2nd Edition.* Chicago: Packt Publishing Ltd.

Salman, T., & Jain, R. (2017). Networking protocols and standards for the internet of things. *Internet of Things and Data Analytics Handbook*, 215–238. https://doi.org/10.1002/9781119173601.ch13

Shodan. (2019). Shodan Exploits. *Shodan*. Retrieved from https://exploits.shodan.io/welcome

Singh, M., Rajan, M. A., Shivraj, V. L., & Balamuralidhar, P. (2015). *Simple Authentication and Security Layer (SASL)*. *Proceedings - 2015 5th International Conference on Communication Systems and Network Technologies, CSNT 2015*. https://doi.org/10.1109/CSNT.2015.16

Matheu-Garc´ıa, S. N., Hern´andez-Ramos, J. L., Skarmeta, A. F., and Baldini, G. (2019). “Risk-based automated assessment and testing for the cyber-security certiﬁcation and labeling of IoT devices,” Computer Standards& Interfaces, vol. 62, pp.64–83.

Roland Bodenheim, Jonathan Butts, Stephen Dunlap, and Barry Mullins. (2007). Evaluation of the ability of the Shodan search engine to identify Internet-facing industrial control devices. International Journal of Critical Infrastructure Protection, 7(2):114–123, 2014.

Talabis, Ryan. "Honeypots 101: A Honeypot By Any Other Name.".

Thangavel, D., Ma, X., Valera, A., Tan, H. X., & Tan, C. K. Y. (2014). Performance evaluation of MQTT and CoAP via a common middleware. *IEEE ISSNIP 2014 - 2014 IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Conference Proceedings*. https://doi.org/10.1109/ISSNIP.2014.6827678

Wang, M. (2017). *Understanding the security flaws of IoT protocols through honeypot technologiess Meng*. *Journal of the Optical Society of America*.

使用蜜罐分析IoT 设备的漏洞攻击

摘要

物联网（物联网）攻击的同时，还试图利用固有的威胁，导致大规模破坏，并证实了对连接到物联网的设备安全性的担忧。任何新技术的出现总是伴随着各种漏洞，准备被利用。在本文中，

我们提出了一种通过部署蜜罐日志数据来识别漏洞攻击模式来分析漏洞攻击的方法。AWS 云上的蜜罐用于收集网络事件日志数据。日志数据使用亚马逊弹性搜索服务、亚马逊基尼斯、基巴纳和 POT （EKK-POT） 进行分析。我们还旨在通过从 EKK 堆栈中提取日志，并增强当前机制以应对这些令人震惊的趋势，从而发现物美网设备面临的新威胁。

关键字：物联网、蜜罐、恶意软件攻击、云、安全、僵尸网络。

# 介绍

物联网 （IoT） 利用网络、传感、大数据和计算机科学技术为产品或服务提供完整的系统。然而，IoT的广泛应用面临着一些挑战，如安全、隐私、标准、互操作性以及新兴经济体和发展。在本文中，我们使用 IoT Shodan 和蜜罐对 IoT 设备进行漏洞扫描，以进行威胁和漏洞分析。

这允许使用 Honeypot 分析 IoT 设备的漏洞攻击，通过分析漏洞扫描结果并显示 IoT 设备如何很容易受到黑客攻击和暴露。

蜜罐系统可以是一个程序，作为一个实际的吸引力系统来欺骗和捕捉入侵者谁试图进入系统。在这个过程中，入侵者被敏锐地跟踪和观察，在他或她的遗忘。这个 paper 建议使用这种技术来了解物科网趋势和威胁格局， 以及使用蜜罐来对付他们的方法。

蜜罐会产生大量的数据。通用数据分析工具分析此类数据 a 的数量并不容易。在本文中，我们调用亚马逊弹性搜索服务、亚马逊 Kinesis、Kibana 和 POT （EKK-POT） 技术来分析蜜罐数据，因为它提供了对任何大小的数据集进行搜索的可操作性。随着技术的进步，社会通过目前与互联网直接相连的各种设备得到了极大的联系。这些设备已变得容易受到攻击，如马图-加西亚等人所示（2019年）。

物联网系统可以进行多种形式的攻击，使其更加脆弱，包括中间人 （MITM）、嗅探、拒绝服务 （DoS）、加密图形入侵、僵尸网络入侵、Dos攻击（拒绝服务）和分布式 DoS （DDoS）。不断增长的技术的主要问题是，大量设备正在加入全球网络。同样，许多此类设备被对所使用的技术知之甚少的个人使用，这使得它们极易受到攻击和威胁（法恩贝格尔、戈皮纳坦和帕里达，2019 年）。此外，人们注意到，系统的安全措施并没有随着系统本身的增长而按比例扩展。目前，互联网被认为是黑客和剥削者利用他们的知识为自己谋利的实验室，这大多数情况下会导致对他人的伤害（Kamesh &sakthi，2016年）。这就要求对网络安全认证框架进行定义，马图-加西亚等人（2019年）。

入侵检测系统 （IDS） 系统机制的主要功能是检测和纠正网络中的恶意或不规则活动。

此工具在后台持续运行，不会对网络的正常运行造成很大干扰。最近，一种新的替代品开始使用涉及使用蜜罐和蜜网（斯皮茨纳，2003年）。

1999年，斯皮茨纳将一台电脑印到互联网上，以执行已知暴露的应用程序。这个想法是使这个系统作为一个蜜罐（从名称）来吸引攻击者。

令人惊讶的是，在不到15分钟，东道主已经妥协（斯皮茨纳，2003年）。

然后，蜜罐的想法出现了：一个资源网络受到攻击和破坏 （入侵）。

这意味着一个蜜罐可以尝试和入侵。通过允许此类攻击，可以登记实际发生的事件，然后提供攻击者使用的计划的必要信息（Spitzner，2003年）。

我们检查了几个星期的EKK-POT，并记录了数据。之后，这些日志被收集到弹性搜索索引中，Kibana 用于可视化这些数据的分析结果。

本文为 IoT 安全格局做出了巨大贡献，邀请了更多这一领域的工作。此外，它使用 EKK 堆栈工具，使部署步骤清晰简洁，以便在集成环境中与一套出色的网络安全工具一起研究数据。

# 相关工作

引入网络空间搜索引擎，如肖丹、FOFA和其他公共平台来收集蜜罐服务器。只有蜜罐攻击装置的一部分用于估计全球和特定类型的设备数量。专门的搜索服务，如肖丹，森斯，和ZoomEye帮助扫描相关服务的IP光谱，对它们进行投票，并索引结果。使用肖丹、森斯和 ZoomEye搜索最常见的头、路由器和Dvr。

在 2013 年之前，Shodan 曾是物科网的主要搜索工具，当时 Censys成为免费竞争工具。它还作为一个 IoT 搜索引擎，依赖于相同的基本原则。但是，正如创建者所指出的，它在搜索潜在的物合物安全漏洞时更加准确。

Censys 提供一系列具有共同漏洞的设备，例如 易 受心出血影响的设备。

这项审查适用于科学家与时间同步改进的所有现有蜜罐方法。

## 蜜罐攻击

蜜罐的一个重要条件是它们能够保持未被发现的攻击：隐藏 蜜罐的细节。Shodan 是一种 special 搜索工具，可在互联网上爬行 ，并尝试查找 IP 网络摄像头和路由器等链接设备（罗兰·博登海姆 等人，2014 年）。在爬行和索引的渐进过程中，Shodan 系统更新 了所有系统的数据库以及 连接到或暴露于 Internet 的服务。 最近执行和部署了肖丹检测机制，以 鉴定蜜罐。它进行了几次探头和监控，然后 为每个检查设备创建分数。From 这个分数的价值， 肖丹决定系统是否是蜜 罐 系统。

## 物科工作协议的安全性

本子节讨论了物联网应用程序和设备中使用的当前通信协议的现有情况和问题。

**2.1.1. 目前的情况**

虽然对物超技术安全性的关注在增加，但人们一致认为，物超技术还不够成熟。应用协议的安全问题通常从三个不同的方面来考虑（Al-Shaer、Wei 和 Hamlen，2019 年）。协议本身的缺陷、协议期间的麻烦以及实施过程中的漏洞。表1显示了每个协议中应用的安全机制。

表（1）：物联信息通信协议 的安全机制

|  |  |
| --- | --- |
| MQTT | 简单的用户名/密码身份验证、传输层安全/安全插座层（TLS/SSL） 数据加密 |
| XMPP | 简单的身份验证和安全层 （SASL）  身份验证，TLS/SSL 数据加密 |
| AMQP | SASL身份验证，用于数据加密的 TLS/SSL |
| CoAP | 数据图传输层安全 （DTLS） / 互联网  协议 安全 （IPsec） |
| JMS | 供应商特定 但 通常基于TLS/SSL。通常 用于身验证和授权服务 （JAAS） API |
| Simple Object Access  Protocol (SOAP) | WS-安全部门的地址 |

可以观察到，运输层安全加密协议用于MQTT、XMPP、AMQP 和 JMS。数据图传输层安全用于 CoAP，而 WS 安全用于 SOAP 协议。简单的身份验证和安全层 （SASL）是这些协议最常用的身份验证框架。

## 目前 IoT 安全性存在问题

如前所述，IoT 安全问题从三点观点出发。连接协议的曝光：

所有协议和其他特定协议都有一些常见的安全问题。

首先，引入共同问题，然后讨论具体问题。

* + 1. 协议的常见问题：

常见问题包括启动安全、防火墙系统以及更新中的安全性以及所有协议中的修补问题。例如，仅应允许在链接设备中下载授权软件应用程序。这就是为什么必须采取预防措施，如数字签名或加密密钥（法恩贝格尔，戈皮纳坦，和帕里达，2019年）。因此，它被认为是一个安全挑战与大规模部署和有限的资源（马鲁夫，森哈德吉，拉比，和贝尔卡斯米，2018年）。

合作：

根据纳斯塔塞 （2017a） 和纳斯塔塞 （2017b）， DTLS 有一个问题， 因为它的设计不符合 CoAP.首先，DTLS 无法启用多广播。在 CoAP 中使用 DTLS 或其他解决问题的建议的弱点之一（Gürmé 等人，2018 年）。然而，这些计划可能不适合 CoAP，并可能导致安全问题。需要更多的研究。

* 合作**：** CoAP 的"无秒模式"问题

引入了四种安全y 模式（Gürmé 等人，2018 年），最初的模式是"NoSec 模式"。在此模式下，将传输 CoAP 消息，但未执行安全方法。

* MQTT： MQTT 的加密和身份验证问题（Salman & Jain，2017 年）讨论了一项称为"安全 MQTT"的新协议（辛格、拉詹、希夫拉杰和巴拉穆拉利达尔，2015 年），该协议基于"基于轻量级属性的加密"应用加密。由于MQTT的弱点，SMQTT与MQTT不同。

XMPP： SASL 认证存在问题，正如 Melo （2013） 所解释的，SASL身份验证可能会出现问题，涉及几个SASL 方法无法提供足够安全的安全性的安全因素。

可能需要额外的保护服务来防范攻击。此外，XMPP 使用 SASL 进行身份验证（莱托和奈塔安梅基，2018 年）。因此，SASL 中任何形式的弱点也是 XMPP 中的弱点。

* + 乌普恩普：已知的 Upnp 弱点

UPnP的问题并不是一个新课题。它在网上的各种出版物（迈克尔·米莫索，2015年）（布鲁斯特，2013年）和学术领域（迈克尔·米莫索，2015年）（哈西卜和莫塔利布，2010年）就这一问题进行了大量讨论。Michael Mimoso（2015）指出了Upnp 的安全漏洞，这些弱点使数百万家用网络工具面临被入侵的危险。（布鲁斯特，2013年）还解释了使用UPnP的漏洞来运行攻击。

* 协议 执行 问题：

在实施协议期间可能会出现与开发相关的安全问题，包括构建和安装服务器。这些问题可能因错误、弱点和其他问题而异，以及缺乏验证。不幸的是，这些问题可能会危及全部执行。这些弱点可以映射到"常见漏洞和暴露"（CVE），这是一个数据库，其中列出了已知的软件应用程序（包括操作系统、档案和框架）暴露情况。

补密可以来自开放或封闭的来源。各种数据库共享一个常见收集之简历标识符 描述 都 这 目前 信息 关于 这脆弱性包括供应商产品时间脆弱形式这类之脆弱性和描述（鲁塞尔和杜伦，2018年）。这使得识别变得容易 开发 方向 为 应用 上 约特 或约 平台 跟 协议。

* + 与物合物集成时的漏洞：

由于物联网平台集成了各种物联网技术和协议，以单个单元运行，因此，物联网被视为一个不成熟的平台，容易受到多次安全攻击。可以采取一些具体的方法来整理、分析，最终确定物联网平台上的威胁模式。

## 攻击类型

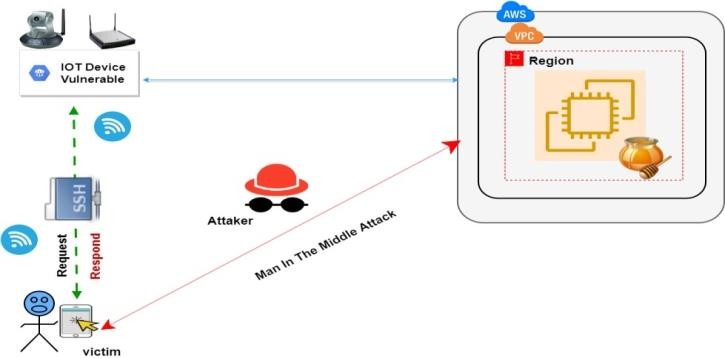
不同形式的入侵可以由恶意用户进行，如端口扫描、MITM 攻击（中间人）和拒绝服务 （DoS）。对这些攻击作了简要概述。

## 端口扫描攻击

这是被动攻击之一，这意味着它不会损害系统或服务器。但是，它可以存储与受害者的机器或服务器相关的所有详细信息。端口扫描旨在通过向客户发送请求（Cruz-Cunha，2014 年）在服务器的端口地址范围内查找任何活动端口。不同的端口扫描包括连接同步制版 （SYN） 扫描：UDP 扫描、XMAS 扫描、确认 （ACK） 扫描和 FINISH （FIN） 扫描通常被攻击者应用的扫描的流行所使用。

* + 1. 中间人 （麻省理工学院）

在这种攻击中，入侵者试图拦截两个伙伴之间的通信。攻击者仿效可靠的连接，将消息从一方传输到另一方，以建立这是私人对话的信念。图1显示了攻击是如何进行的;攻击者不是在受害者不知情的情况下传输信息，而是在受害者不知情的情况下进行这一部分;作为交换，攻击者可以访问所有传输的数据。



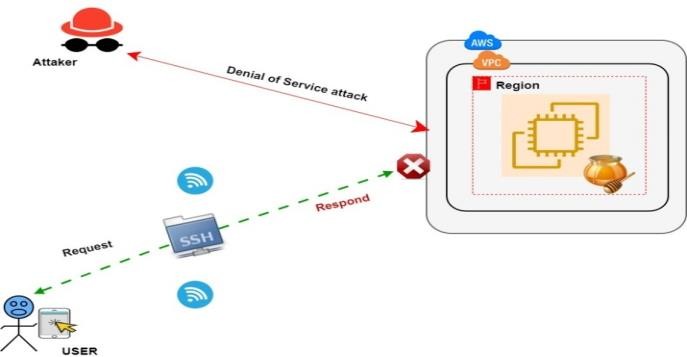
图（1）： 中间人

## 拒绝服务攻击

此攻击旨在关闭机器、网络或软件一段时间，以拒绝使用该服务。在此次攻击中，系统/网络被系统/网络无法处理的大量垃圾邮件请求 ping，因此会崩溃。政府和银行通常是这次袭击的主要目标，因为它导致它们失去时间和金钱来重新建立系统/网络。

此类型的攻击显示在图 2 中。

图（2）： 拒绝服务攻击



## 分布式拒绝服务攻击

这与拒绝服务攻击相同，但攻击者使用许多系统来取下系统/网络有所不同。典型的DDoS攻击有一个主人和一个僵尸。

主指通过探索系统中的所有漏洞并开始攻击的入侵者，通过恶意软件或绕过系统对系统施加感染，发现弱系统并控制系统。

* + 1. **嗅探攻击**

最常见的攻击类型之一是嗅探附加，通过连接和无线链接进行。它使攻击者能够访问和收集，并通过访问用户系统来更改来自用户系统的数据（Salman &体系，2017 年）。攻击者在嗅探中使用两种最重要的方法，包括 ARP 中毒和 TCP 会话被盗的方法。ARP 中毒作为嗅探方法，用于使用数据包欺骗入侵和基于路由器的弱点攻击网络。

另一方面，在以混杂模式从源头和目的地捕获数据包的IP 地址时，应用了 TCP 会话盗窃方法。攻击被说明。

## 蜜罐技术

在显示预先解释的协议确定正常行为的标准后，收集、分析和识别不常活动的过程在动态协议分析中完成，其中包括有效载荷（内容）处理、上下文和整体 pattern检查。就物联网而言，协议不仅定义物联网工具交换数据的方式，还定义其包含在物联网平台中的方式。

蜜罐技术被认为是处理数据和 承认 这 模式（萨尔马n & 耆那教的 2017) 主要地 蜜罐 科技 是 a 机制 那 是 使用 自 检测 恶意 活动通过 模拟 之 一 当前 系统 在 a 安全 环境。 因此 这 侵略者 是 引诱 到 这 系统 思维 它 是一个实际的：然而，它不会影响系统，因为它受到保护，和所有 活动 这 入侵者正在 记录。

关于蜜罐的主要想法是部署之前提到的软件上的通信协议，黑客攻击，同时记录他们的活动。

## 蜜罐的优势

蜜罐的一个优点是市场上存在多种安全解决方案，任何人都可以在互联网上探索选择，以找到最合适的解决方案，以满足他们的需求。根据 Mokube 和 Adams（2007 年）的说法，蜜罐捕获入侵ns 并提供有关攻击类型的信息，必要时，由于日志，人们可以看到攻击的附加信息。可以注意到新的威胁，并且可以通过查看它们来形成新的安全选项。进一步分析可以通过研究这种不当行为来完成。它有助于理解可能发生的更多攻击。在捕获数据方面，蜜罐并不重。他们主要处理恶意流量的流入。因此，捕获的数据并不是整个流量。只关注恶意流量简化了调查。因此，蜜罐变得非常有益。对于少数恶意攻击，无需使用大数据存储和新技术。

任何类型的计算机都可以配置为蜜罐系统。因此，蜂巢系统不会增加安全系统的成本。它们易于理解、配置和部署。它们没有复杂的算法，不需要更新和修改。由于蜜罐可以检测到任何恶意攻击者，因此它提出了更多想法，并确实加深了安全解决方案。

## 蜜罐的缺点

由于蜜罐的应用有几个重要优势，它们也有缺点。正如 Mokube & Adams （2007） 所陈述的，他们只能在凶残的入侵者实时攻击系统时选择数据，否则，捕获信息将变得困难。在不同的 IoT 系统上，蜜罐无法检测到发生在其他系统上的任何攻击。

因此，对外部蜜罐系统的攻击可能会损坏其他 IoT 系统并导致巨大的问题。蜜罐的第二个缺点是经验丰富的攻击者对蜜罐进行指纹识别。攻击者很容易研究他是否攻击蜜罐系统或实际系统，此识别可以确定他或她的活动。指纹识别能够区分两者，但不是 实验的理想结果。蜜 罐 可以用作僵尸，以达到外部 系统 和 削弱其 安全 系统，这是危险的。

相关工作，正如我们所做的研究一样，在解释蜜罐部署的完整方法的情况下，还没有进行过任何研究。此外，在 EKK 堆栈的帮助下模拟设备和收集攻击数据尚未用于查找物料网蜜罐的未知攻击。

这项研究的重点是安装物美网蜜罐。他们提出了一个程序，在物料网系统上安装必要的蜜罐，考虑到流行选择的工具的等级和要求，避免检测陷阱节点作为蜜罐中的一些最我们ed物科扫描仪。他们发现，这些物联网工具的设计没有适当的安全标准，他们在我们的生活中，特别是在我们的隐私和安全（Acien，尼托，费尔南德斯，和洛佩兹，2018年）中扮演重要和明智的角色。

为了获得这项研究的奥比克，下面的步骤如下

对基于物科网的共享协议（包括 MQTT、XMPP、AMQP 和UPnP）进行研究，以访问和理解其设计。在本次调查中，我们提供了一个平台，专注于精确的协议和可能的研究子主题。

在回答蜜罐技术是否适合识别物美网络平台上的安全问题的问题时，应用了事物点的概念证明 （POC）。通过主要执行 Thingpot POC 在四个节点上运行 46 天（节点的即用于 REST API，而一个节点用于 XMPP），它捕获了大量有价值的信息。通过进行详细的数据分析，他们发现并总结了除 IoT 系统之外黑客的行为。因此，ThingPot 展示了识别物联网平台上安全问题的能力，并证明蜜罐设备适合识别物联网系统上的安全隐患（Wang，2017 年）。

智能交互是一种自动智能构建 IoT 蜜罐的方法。这利用多种启发式以及交互式机器学习机制来自定义过程扫描，并增强响应逻辑以扩展以捕获漏洞代码。

他们提出了一个通用框架，为IoT工具构建一个智能交互蜜罐。他们解释了为什么以及我们如何需要智能交互式蜜罐（Wan g，2017）。

交互式智能的目的是从物超所学工具上的零知识中学习与客户互动的正确行为。对客户的适当反应应适合与潜在黑客扩大会话，诱骗他们通过检查，同时发送 Maarof等人描述的漏洞利用需求（2018 年）。 要实现这一目标，需要系统通过与黑客的互动自动收集作为候选人的有效响应：学习程序帮助蜜罐增强每个请求的正确行为（Ramirez 等人，2017 年）。源代码的任何版本都必须导致 2 个重要结果：

它不仅为创建一个活跃的僵尸网络提供了一个功能过程，而且为定制僵尸网络的创建提供了一个功能过程，而且最明显和最明显的是，它提供了网络罪犯和黑客。

第二个结果是，在 IoT工具和相关操作系统对简单恶意软件感染和奴役试验的弱点之后，对IoT 安全的需求形成。这些设备不仅弱，而且有些设备也受到感染。

从源代码的发布，即对物联网安全的日益需求，以及攻击者和机器人牧民对物联网设备的奴役和争夺。然而，约有60亿台物联网设备连接到互联网，预计到2020年将达到200亿台。 黑客们展开了激烈的竞争，促使恶意软件的启动从操纵设备中清除其他恶意软件，这简化了间谍活动、数据窃取或启动大量 DDoS 攻击，并四处寻找其他弱 devices 并将它们添加到僵尸网络（Antonakakis 等人，2017 年）。

IoTPOT 提出了一个独特的蜜罐，用于模仿许多 IoT 工具的 Telnet 服务，以详细监控和记录对 IoT 设备的连续主动攻击。物联网对与名为 IoTBOX的后端高交互计算机生成环境的合作具有显著的低行动响应。IoTBOX控制许多虚拟环境，这些虚拟环境通常通过固定系统应用于各种中央处理单元 （CPU） 设计 （Minn 等人，2015 年）。在39天的行动中，观察到76，605次恶意软件的尝试下载，比访问互联网协议（IP）的16，934次有所增加。（拉米雷斯等人，2017年）确认，目前的蜜罐不能捕获任何这些二元ies 。Telnet协议，作为电话网密码蜜罐和蜂蜜，因为他们无法管理通过黑客发送的不同流入命令（Shodan，2019年）。

分析捕获的恶意软件二进制文件;他们建议将 IoTBOX 作为 IoT设备恶意软件分析的第一个环境。物联网运行8个CPU架构，即水疗臂、PPC和MIPS。

通过物科工作台对17个样本进行沙盒勘探表明，这些样品 用于执行10种不同类型的DDoS攻击以及23号端口扫描。 最终，合并了物合物异位素观测结果和 沙盒 物合器分析结果，我们 确认：

至少有4个独特的恶意软件家庭正在通过Telnet传播

他们典型的活动是在电话网和 DDoS 上执行额外的广播

有些家庭发展迅速，更新迅速，甚至在39天的有限观察期内（Minn等人，2015年）内，也为CPU设计的品种运送二进制文件。开发了一个自动框架，用于检测和识别受破坏的僵尸网络开发和恶意软件分析。他们提出了一些基本的研究问题，如"僵尸网络受损机器的特征是什么？

Dionaea 一个低交互性、服务器端蜜罐，该蜜罐模拟易受攻击的系统或设备安装在亚马逊弹性计算云 （Amazon EC2） 上，以记录攻击数据和恶意软件二进制文件。病毒总计用于恶意软件分组，而 Shodan 则用于丹加维尔等人描述的设备识别（2014 年）。对恶意软件的顶级变化、设备类型、产品、国家、组织和开放端口进行了检查，以发现受攻击的设备，并且仍然存在关键路由器的弱点（Chinn， 2015）。

因此，我们提出了一个基于 IoT 蜜罐的欺骗工具。欺骗思想是网络安全保护的新兴类别。欺骗设备可以分析和检测弱点，并保护系统和设备免受零日攻击和激进攻击。它们能够自动、提供和准确感知内部网络中的恶意移动，这些恶意运动可能隐藏在其他形式的网络保护人员中。欺骗概念使安全系统具有更主动的地位，以寻求欺骗入侵者，发现他们，并征服他们，允许创新屈服于正常流程。

# 实验设置

本节使用基于：EKK 堆栈（亚马逊弹性搜索服务、亚马逊 Kinesis 和 Kibana）的中央日志和观察服务器描述实验设置，该堆栈有助于显示数据、创建可视化和用于任何大小的数据的仪表板。这项研究基于EKK堆栈（亚马逊基尼斯，亚马逊弹性搜索服务和基巴纳）。弹性搜索是指具有分析的搜索引擎。Logstash 是指用于处理数据的服务器端管道，该管道同时从 numero我们源中获取数据，在将其发送到弹性搜索等"藏匿处"之前对数据进行转换。Kibana 允许用户使用弹性搜索中的分析图表和图形可视化数据。

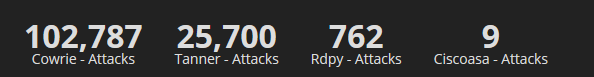
我们使用已知日志组合解决方案的替代方案，ELK 堆栈（弹性搜索、日志和基巴纳）：EKK 堆栈（亚马逊弹性搜索服务、亚马逊 Kinesis 和基巴纳）。EKK 解决方案可消除对日志聚合分辨率的部署、管理和缩放的简单繁重。通过 EKK 堆栈，我们强调对日志的分析和应用程序的调试，而不是对日志进行分类和向上扩展系统。

## 使用基巴纳的数据分析

在基巴纳，我们使用弹性搜索执行了几个关键字搜索。主要目标是在我们的蜜罐里发现攻击事件。我们已经从 log 数据分析中确定了几个事件。Kibana 被用作可视化和探索数据的开源工具。此外，它还适用于日志和时间系列分析、应用程序检查和操作智能使用案例。Kibana 被认为是一个非常有用的工具，有助于丰富功能，如条形图、趋势线、直方图、线图和其他视觉图形。Kibana 允许与亚马逊弹性搜索紧密集成，使其成为我们的可视化默认选择。

EKK 的实施消除了部署、管理和扩展日志聚合分辨率的同质重型升降机。重点是 EKK堆栈，重点是分析日志和更正应用程序，而不是管理和向上扩展系统来聚合这些日志。

## 最常见的攻击



图（3）： 最常见的攻击

考里蜜罐经历了超过10万次攻击。毫不奇怪， 大多数Telnet 和 Ssh 蜜罐的攻击最多。[Cowrie](https://github.com/cowrie/cowrie) 是一个标准的高交互蜜罐，计划引诱和记录物理力事件和攻击者的任何外壳交易。其主要目的是与攻击者互动，同时监控他们认为自己违反了系统时的行为。

毫不奇怪，在102，787起袭击中，32，607起来自中国，美利坚合众国以14，115次袭击位次。

## 他们登录后所做的事情

了解攻击者一旦进入即该怎么做是建立防御的关键。并不是每个攻击都会来自未知 IP 的未知用户。对于大多数 SIEM，如果用户使用列出的任何命令，则可以触发警报。这样，如果网络上的用户丢失了凭据，如果使用这些命令中的任何一个，我们可以将其标记为可疑行为，并设置规则来阻止帐户，直到确认它不是违规行为。

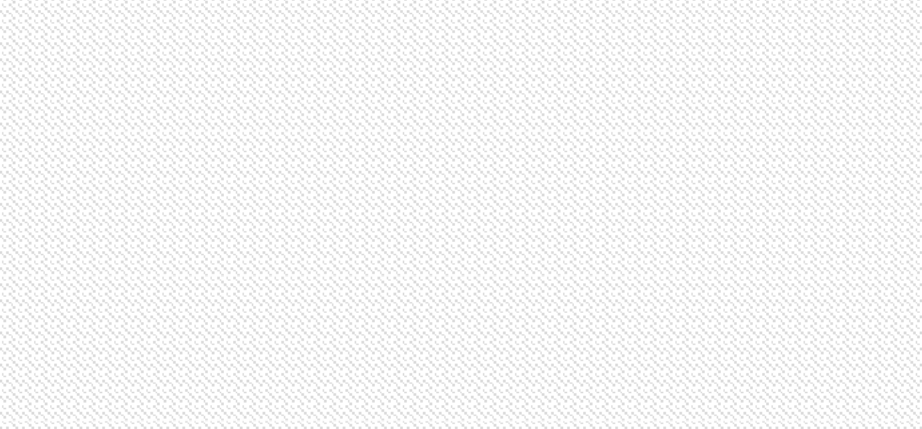
我们注意到，网络攻击不仅仅是四大攻击（俄罗斯、中国、美国和伊朗）。 对于安全团队来说，理解这一点对于构建您的战略至关重要。定期 查看服务攻击发生的位置有助于确定是否有任何内部 破坏 发生 。

然而，我们确实注意到，一些蜜罐码头工人继续撞倒后，一个小蛮力，我们单独设置了一个弹性集群，以监控服务器，看看它是否被攻击。我们建议使用弹性的最新 versi来使用其 SIEM 功能来监控一些测试环境。我们还建议未来的研究，清理蜜罐，并单独部署他们更好的分析，除了利用机器学习弹性。

## 统计学

对结果的统计分析为读者提供了更好的洞察力，了解我们在EKK实验中观察到的内容。本分析的第一部分包括对蜜罐的攻击的整个结构。它承载着更广泛的攻击载体以及连接不同蜜罐系统的普通IP的概念。它还携带在所有程序中攻击次数最多的IP。

从 REST 日志中分析此信息表明，入侵者注意到了物联网环境和所有设备。一般来说，所有攻击者都在寻找飞利浦休、贝尔金韦莫和 TPlink等设备。特别是，他们有兴趣获得有关智能设备的信息，更重要的是接管控制权。攻击者似乎更喜欢的方法首先是进行一般扫描以查找漏洞，然后是通过蛮力或模糊进行更有针对性和更具体的攻击。我们建立了T-蜂蜜罐（陷阱），模仿各种设备，并连接到互联网，看看他们发生了什么事'在野外'。通过整个24小时的时间跨度，连接攻击的尝试是在数以万计的from各种独特的IP地址。



200

180

160

190

188

170173

164

150152

160

148

140

120

100

80

60

40

20

0

123 123

114115

122

120

103

93

101

100

83

84

72

54

58

56

54

44

36

44

一天12小时的时间戳

**计数**

图(4): 来自独特的IP地址对蜜罐的独特点击数2月1日至2020年2月

2月29日至20日

2月28日至20日

2月27-20日

2月26日至20日

2月25日至20日

2月24-20日

2月23日至20日

2月22日至20日

2月21日至20日

20-2月-20日

2月19日至20日

2月18日至20日

2月17日至20日

2月16日至20日

2月15日至20日

2月14日至20日

2月13日至20日

2月12日至20日

2月11日至20日

2月10日至20日

09-2月-20日

08-2月-20日

07-2月-20日

06-2月-20日

05-2月-20日

04-2月-20日

03-2月-20日

02-2月-20日

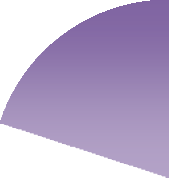
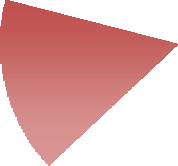
01-2月-20日

在许多情况下，尝试执行连接到电话网协议;其他人应用了SSH。

表（3）：按连接类型对未遂攻击进行贡献归类

|  |  |
| --- | --- |
| 连接类型 | 数 |
| 电话网 | 6700 |
| 嘘 | 1182 |
| 总 | 7882 |

我们分析了攻击源自的设备类型。超过 60% 的攻击可以以 DVR 服务和基于 IP 的摄像机的形式看到，但大约 17% 是各种形式的网络工具以及来自几家主要制造商的路由器。大约 1% 是 Wi-Fi 中继器，而其他网络硬件、电视调谐器、VOIP 设备、Tor 退出节点、打印机和构成"智能家居"的所有设备。其他设备无法识别，只是注册为无法识别的设备，形成 20%。



未识别 的设备

20%

电视/IP 电话

1%

网络 设备

16%

德维尔或IP 摄像机

63%

**德维尔或 IP** **摄像机**

**网络 设备**

**电视/IP 电话**

**未识别 的设备**

图 （5）：按设备类型分布攻击源 2020 年 2 月 10 日至 2 月 29 日多个连接试图进入蜜罐系统的一些 IP 地址

我们构建，他们响应了HTTP的要求。当然，许多设备通过 NAT 技术应用程序使用每个 IP 地址。人们通常会发现，响应 HTTP 服务器请求的设备并不是通常预期的那样对蜜罐造成攻击的设备。

有网页和设备控制面板，以响应和维护请求，并通过摄像机进行监控。返回的页面可以识别设备。下表显示了入侵设备检测到的网页最常用的标题：

表（4）：对设备的攻击数量，按标题分类

|  |  |
| --- | --- |
| 设备 | IP 地址数 |
| DVR或 IP 摄像机 | 4966 |
| 网络 设备 | 1261 |
| 电视/IP 电话 | 79 |
| 未识别 的设备 | 1576 |
| 总 | 7882 |

表（5）：对设备的攻击数量，按http标题分类

|  |  |
| --- | --- |
| 赫特普 标题 | 设备 |
| 网络 服务 | 2569 |
| 网 保网络 | 1432 |
| 德夫尔夫兹 | 621 |
| DVR 组件 下载 | 777 |
| 网德夫3 | 960 |
| 伊夫斯韦布 | 1523 |
| 总 | 7882 |

几个IP 地址是一个潜在的漏洞。以下显示了易受攻击的设备。

表（6）：脆弱设备的IP地址的数量，网络摄像机和录像机

|  |  |
| --- | --- |
| 赫特普 标题 | 设备 |
| 埃尔特克斯 Ntp | 653 |
| 路由 器路由器 | 823 |
| 庞家庭网关 | 1878 |
| TL-WR841N | 1969 |
| ZXHN H208N | 1038 |
| TD-W8968 | 642 |
| 伊盖特 GW040 庞特 | 879 |
| 总 | 7882 |

我们设置的他不仅记录了来自网络硬件的攻击，这些攻击被归类为家庭设备，而且还记录了企业级的硬件。

在所有发布攻击的IP地址中，其中一些地址使用主动和安全链接（如：

* 商店、餐馆和加油站的销售点设备
* 数字 电视 广播 系统
* 物理 安全 和 访问 控制 系统
* 环境 监测 设备
* 曼谷地震统计离子监测
* 工业级 可编程 微控制器
* 电源 管理 系统

要确认这些设备是否感染了病毒并不容易。但是，对蜜罐的一些攻击来自这些设备使用的 IP 地址，这意味着至少有一个或多个设备在它们所在的网络上受到感染。

## 受感染设备的地理位置

如前所述，大多数受感染的工具是基于 IP 的摄像机，而且 直流器。其中很多在中国很流行，越南。其他人被发现 在俄罗斯和巴西等许多国家。地理分布 装置s与攻击蜜罐的IP地址在馅饼中表示 图表 下面：



**4.485**%**。7：34%：6%**

**4.13%;4%** **5.97% ;6%**

**6.21%;6%**

**3.95%;4%**

**12.26%;12%**

**6.92%;7%**

**1..6895**%**：;2%**

**3.24%;3%**

**3.90%;4%**

**11..5357%：;21%**

**1.65%;2%**

**17.44%;17%13.95%：14%**

**01..9095**%**：;11%**

中国

越南 俄罗斯 巴西 土耳其 塔万 伊朗

韩国美国印度阿根廷巴基斯坦

意大利 澳大利亚 泰国 罗马尼亚

哥伦比亚联合金 多姆

图（6）：2020年2月22日至2月29日，按国家分列攻击设备IP地址

## 服务器 IP 地址的地理分布

我们记录了超过 100000次黑客攻击尝试和 8000 多个唯一的 IP 地址，从这些 IP 地址下载了物联网设备的恶意软件。IP 地址的细目如下所示。

表7将恶意软件wa下载到设备的服务器IP地址的地理分布

|  |  |
| --- | --- |
| 国家 | 独特的 IP |
| 越南 | 2136 |
| 中国台湾省 | 1356 |
| 巴西 | 1124 |
| 土耳其 | 696 |
| 韩国， 共和国 | 620 |
| 印度 | 504 |
| 美国 | 429 |
| 俄罗斯联邦 | 373 |
| 中国 | 361 |
| 罗马尼亚 | 283 |
| 总 | 7882 |

可以推断，这种差异是由于在这些国家中一些国家存在防弹服务器，这意味着它比感染物代设备更快、更容易传播恶意软件。

# 结论

蜜罐技术是网络安全战略中有帮助和最重要的部分。 尽管蜜罐技术在不断改进，但攻击者不断在物合物智能设备和蜜罐中搜索漏洞以识别它们。我们使用从AWS 收集的蜜罐数据分析了物科设备的外阴不平度。使用EKK堆栈分析数据，用于日志数据分析和可视化。值得注意的是，EKK使用弹性搜索，帮助识别了各种类型的漏洞。 很明显，蜜罐经常成为攻击者的目标。

在整个论文中，我们概述了如何使用 EKK技术建立蜜罐，并强调物联网设备的漏洞及其用户社区面临的威胁。互联网日常用户一般没有意识到这些设备的危险性，因为这些设备的设计没有足够的安全措施。此外，他们越来越多地扮演重要和明智的角色，并影响隐私和安全。实验结果为蜜罐的改良提供了参考，促进了利用EKK技术开发蜜罐技术。

与旧式安全发现方法不同，蜜罐结构，特别是在物联网研究领域，旨在通过探索网络包或日志文件来攻击和习惯性地监控可能的攻击。使用 EKK 技术，我们能够从这些数据中推断出IoT设备的漏洞和威胁、有效的角色策略、技术和程序。这提供了一个强大的工具，可用于生成更多的防御策略。

## 建议

在以后的工作中，我们的目标是在 AWS EKK POT 中使用机器学习扩展对物科网络设备漏洞攻击的分析。进一步研究的建议包括但不限于下列：

改进蜜罐机制，防止攻击者识别，并防止蜂巢捕获他们的行为。这需要智能技术来自动检查远程服务器是否运行蜜罐服务。

此外，研究应该扩展蜜罐的能力，以刺激更多的架构和环境，这是常见的物超网设备。

因此，我们建议进一步研究基于机器学习的检测框架的蜜罐，以防御基于物机的攻击。

随着未来的工作，我们可以在更大的公共网络中部署蜜罐，并 评估 其平均 会话 时间和 数据捕获性能。

# 参考文献

Acien, A., Nieto, A., Fernandez, G., & Lopez, J. (2018). A comprehensive methodology for deploying IoT honeypots. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*,*11033 LNCS*,229–243. https://doi.org/10.1007/978-3-319-98385-1\_16

Al Hasib, A., & Mottalib, M. A. (2010). Vulnerability analysis and protection schemes of Universal Plug and Play protocol. *Proceedings - 2010 13th IEEE International Conference on Computational Science and Engineering, CSE 2010*, 222–228. https://doi.org/10.1109/CSE.2010.37

Anagnostakis, K. G., et al. "Detecting targeted attacks using shadow honeypots." Proceedings of the 14th conference on USENIX Security Symposium. ACM, 2005. 129-144.

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Arbor, A., Bursztein, E., … Zhou, Y. (2017). Understanding the Mirai Botnet. *USENIX Security*, 1093–1110. https://doi.org/10.1016/j.religion.2008.12.001

Al-Shaer, E., Wei, J., & Hamlen, K. W. (2019). *Autonomous Cyber Deception: Reasoning, Adaptive Planning, and Evaluation of HoneyThings.* NY: Springer.

Brewster, T. (2013). How Attackers Can And Will Exploit UPnP Weaknesses[20]. Retrieved from https:/[/www.silicon.co.uk/workspa](http://www.silicon.co.uk/workspace/how-attackers-will-exploit-upnp-105868)c[e/how-attackers-will-exploit-upnp-105868](http://www.silicon.co.uk/workspace/how-attackers-will-exploit-upnp-105868) Chinn, R. (2015). Botnet Detection: Honeypots and the Internet of Things, *I*. Retrieved

from https://msmis.eller.arizona.edu/sites/msmis/files/documents/sfs\_papers/ryan\_chin n\_sfs\_masters\_paper\_0.pdf

Christian Seifert, Ian Welch, and Peter Komisarczuk. Taxonomy of honeypots, 2006.

Cruz-Cunha, M. M. (2014). *Handbook of Research on Digital Crime, Cyberspace Security, and Information Assurance.* NY: IGI Global.

CVE Details. (2019). CVE security vulnerability database. Retrieved from https:/[/www.cvedetails.com/](http://www.cvedetails.com/)

ElasticSearch-Contributors. (2013). ElasticSearch Documentation. Retrieved from <http://www.elasticsearch.org/guide/>

Fahrnberger, G., Gopinathan, S., & Parida, L. (2019). *Distributed Computing and Internet Technology: 15th International Conference, ICDCIT 2019, Bhubaneswar, India, January 10–13, 2019, Proceedings.* Los Angeles: Springer.

Görmüş, S., Aydın, H., & Ulutaş, G. (2018). Security for the internet of things: A survey of existing mechanisms, protocols, and open research issues. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *33*(4), 1247–1272. https://doi.org/10.17341/gazimmfd.416406

Guide, U. (2012). Amazon Elastic Compute Cloud User Guide for Linux Instances. *Copyright © 2016 Amazon Web Services*. Retrieved from <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-ug.pdf>

Mokube, I. and Adams, M. (2007). “Honeypots: concepts, approaches, and challenges,” in Proceedings of the 45th Annual Southeast Regional Conference, pp. 321–326, ACM, Winston-Salem, NC, USA, March 2007. View at Publisher

Kamesh, & Sakthi Priya, N. (2016). A survey of cybercrimes Yanping. *Security and Communication Networks*, *5*(February), 422–437. https://doi.org/10.1002/sec

Kemmerer, R.A. and G. Vigna. "Intrusion detection: a brief history and overview." Computer 2002: 27-30.

Lehto, M., & Neittaanmäki, P. (2018). *Cyber Security: Power and Technology.* NY: Springer.

L. Spitzner, (2003). Honeypots: Tracking Hackers. Addison-Wesley, [Online]. Available:

<http://www.tracking-hackers.com/book/>

L. Spitzner, (2003) “The honeynet project: trapping the hackers,” IEEE Security & Privacy, vol. 1, no. 2, pp. 15–23, 2003.

Maarof, A., Senhadji, M., Labbi, Z., & Belkasmi, M. (2018). Security on the internet of things. *Security and Privacy in Smart Sensor Networks*, 105–121. https://doi.org/10.4018/978-1-5225-5736-4.ch006

Matheu-García, S. N., Hernández-Ramos, J. L., Skarmeta, A. F., & Baldini, G. (2019). Risk-based automated assessment and testing for the cybersecurity certification and labeling of IoT devices. *Computer Standards & Interfaces, 62,* 64–83.

Mokube, I. & Adams M.,2007.Honeypots:Concepts,Approaches,and Challenges. ACMSE 2007, March 23-24, 2007, Winston-Salem, North Carolina, USA ,pp.321-325.

Melo, A. C. M. (2013). Simple Authentication and Security Layer (SASL). *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699. https://doi.org/10.1017/CBO9781107415324.004

Michael Mimoso. (2015). Filet-o-Firewall UPnP Security Vulnerability in Home Routers

\_Threatpost[19]. Retrieved from https://threatpost.com/upnp-trouble-puts-devices-behind-firewall-at-risk/114493/

Minn, Y., Pa, P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., & Rossow, C. (2015). IoTPOT: analyzing the rise of IoT compromises. *Emu*, *9*, 1.

Mohammed, M., & Rehman, H.-u. (2015). *Honeypots and Routers: Collecting Internet Attack.* Washington: CRC Press.

Nastase, L. (2017a). Security in the Internet of Things: A Survey on Application Layer Protocols. *Proceedings - 2017 21st International Conference on Control Systems and Computer, CSCS 2017*, (July 2016), 659–666. https://doi.org/10.1109/CSCS.2017.101

Nastase, L. (2017b). Security in the Internet of Things: A Survey on Application Layer

Protocols. *Proceedings - 2017 21st International Conference on Control Systems and Computer, CSCS 2017*, (July 2016), 659–666. https://doi.org/10.1109/CSCS.2017.101

Nicholson, D. (2017). *Advances in Human Factors in Cybersecurity: Proceedings of the AHFE 2017 International Conference on Human Factors in Cybersecurity, July 17−21, 2017, The Westin Bonaventure Hotel.* Los Angeles, California, USA: Springer.

Ramirez, D., Uribe, J. I., Francaviglia, L., Romero-Gomez, P., Fontcuberta I Morral, A., & Jaramillo, F. (2017). IoTCandyJar: Towards an Intelligent-Interaction Honeypot for IoT Devices. *Journal of Materials Chemistry C*, *6*(23), 6216–6221. https://doi.org/10.1039/C8TC01582A

Russell, B., & Duren, D. (2018). *Practical Internet of Things Security: Design a security framework for an Internet-connected ecosystem, 2nd Edition.* Chicago: Packt Publishing Ltd.

Salman, T., & Jain, R. (2017). Networking protocols and standards for the internet of things. *Internet of Things and Data Analytics Handbook*, 215–238. https://doi.org/10.1002/9781119173601.ch13

Shodan. (2019). Shodan Exploits. *Shodan*. Retrieved from https://exploits.shodan.io/welcome

Singh, M., Rajan, M. A., Shivraj, V. L., & Balamuralidhar, P. (2015). *Simple Authentication and Security Layer (SASL)*. *Proceedings - 2015 5th International Conference on Communication Systems and Network Technologies, CSNT 2015*. https://doi.org/10.1109/CSNT.2015.16

Matheu-Garc´ıa, S. N., Hern´andez-Ramos, J. L., Skarmeta, A. F., and Baldini, G. (2019). “Risk-based automated assessment and testing for the cyber-security certiﬁcation and labeling of IoT devices,” Computer Standards& Interfaces, vol. 62, pp.

64–83.

Roland Bodenheim, Jonathan Butts, Stephen Dunlap, and Barry Mullins. (2007). Evaluation of the ability of the Shodan search engine to identify Internet-facing industrial control devices. International Journal of Critical Infrastructure Protection, 7(2):114–123, 2014.

Talabis, Ryan. "Honeypots 101: A Honeypot By Any Other Name.".

Thangavel, D., Ma, X., Valera, A., Tan, H. X., & Tan, C. K. Y. (2014). Performance evaluation of MQTT and CoAP via a common middleware. *IEEE ISSNIP 2014 - 2014 IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Conference Proceedings*. https://doi.org/10.1109/ISSNIP.2014.6827678

Wang, M. (2017). *Understanding the security flaws of IoT protocols through honeypot technologiess Meng*. *Journal of the Optical Society of America*.