北京邮电大学

网络空间安全学院



**软件概要设计报告**

**项目： 基于DNS流量分析的僵尸网络检测工具**

**组员： 王硕、彭致远、李懿飞、王晨旭**

**2020年11月6日**

目录

[1 引言 3](#_Toc58492456)

[1.1 背景 3](#_Toc58492457)

[1.2 编写目的 3](#_Toc58492458)

[1.3 项目及范围 3](#_Toc58492459)

[1.4 读者对象和阅读建议 4](#_Toc58492460)

[1.5 定义、术语和缩略语 4](#_Toc58492461)

[2 系统设计概述 5](#_Toc58492462)

[2.1 限制和约束 5](#_Toc58492463)

[2.2 设计原则和设计要求 5](#_Toc58492464)

[2.3 系统模块划分 6](#_Toc58492465)

[2.3.1 系统模型 6](#_Toc58492466)

[2.3.2 系统模块划分 6](#_Toc58492467)

[3 功能模块设计 7](#_Toc58492468)

[3.1 DNS数据包捕获模块 7](#_Toc58492469)

[3.2 PCAP文件解析模块 8](#_Toc58492470)

[3.3 DNS指纹生成模块 8](#_Toc58492471)

[3.4 机器学习特征分析模块 9](#_Toc58492472)

[3.5 机器学习分类模块 9](#_Toc58492473)

[3.6 辅助输出模块 10](#_Toc58492474)

[4 界面设计 10](#_Toc58492475)

[5 软件环境设计 10](#_Toc58492476)

[5.1 开发环境 10](#_Toc58492477)

[5.2 运行环境 11](#_Toc58492478)

[6 安全设计 11](#_Toc58492479)

[6.1 系统备份设计 11](#_Toc58492480)

[6.2 系统容错设计 11](#_Toc58492481)

# 引言

### 1.1 背景

DNS隐蔽通道，是隐蔽信道的一种，通过将其他协议封装在DNS协议中传输建立通信。因为在我们的网络世界中DNS是一个必不可少的服务，所以大部分防火墙和入侵检测设备很少会过滤DNS流量，这就给DNS作为一种隐蔽信道提供了条件，从而可以利用它实现诸如远程控制，文件传输等操作，现在越来越多的研究证明DNS隐蔽通道也经常在僵尸网络和APT攻击中扮演着重要的角色。

企业网络经常面临网络攻击者窃取有价值和敏感数据的威胁。复杂的攻击者越来越多地利用DNS通道来泄露数据，以及维护恶意软件的隧道C&C（命令和控制）通信。这是因为DNS对于几乎所有应用程序来说都是如此重要的服务，从本地计算机到Internet的任何通信（不包括基于静态IP的通信）都依赖于DNS服务，限制DNS通信可能会导致合法远程服务的断开，因此，企业防火墙通常配置为允许UDP端口53（由DNS使用）上的所有数据包，即DNS流量通常允许通过企业防火墙而无需深度检查或状态维护。从攻击者的角度来看，这使得DNS协议成为数据泄露地隐蔽通信通道。

DNS这种穿透防火墙的能力为攻击者提供了一个隐蔽的通道，尽管是低速通道，通过将其他协议（例如，SSH，FTP）隧道传输到命令和控制中心，可以通过该通道泄露私有数据并保持与恶意软件的通信。现代恶意软件和网络攻击在很大程度上依赖于DNS服务，使其活动可靠且难以跟踪。

本项目为一种基于DNS流量分析的僵尸网络检测工具，提取DNS流量的特征，通过机器学习将流量分为恶意流量和正常流量。在DNS特征的提取上，利用机器学习的特征分析，筛选出重要的一些特征，过滤掉不重要的特征，通过模型的调参、完善，生成一个准确率很高的分类器。

本工具能够通过网络中的数据流量，检测出一个网络中的僵尸主机。工具应具有文件检测和实时检测的功能，并提供友好的用户交互与完善的输入输出系统。该工具应能够部署在网络边界上，有效地检测通过DNS隐蔽信道传递信息的主机，区分出正常主机与感染了病毒的僵尸主机，检测准确率较现有的检测工具高，误报、漏报率低，时长尽可能短，能够记录僵尸主机的活动、行为，留下检测记录。

### 1.2 编写目的

本概要设计说明书是基于需求说明书编写。目的在于在需求分析的基础上，结合文献调阅资料与实际情况，规定软件、硬件、技术选择，设计系统总体系统架构、总体功能模块以及各个子模块。为之后系统的详细设计以及程序编写、测试提供指导说明。

### 1.3 项目及范围

* + 项目名称：基于DNS流量分析的僵尸网络检测工具
  + 项目成员：北京邮电大学网络空间安全学院“网络安全分析实践”课程开发小组
    - 王硕（组长）：2018213641
    - 彭致远：2018213646
    - 李懿飞：2018213632
    - 王晨旭：2018213636
  + 系统范围：具有python2.7.9环境的Windows系统计算机
  + 用户：无限制
  + 实现项目的计算机网络：校园网

本设计文档适用于基于DNS流量分析的僵尸网络检测工具项目进行系统分析、详细设计以及编码实现

### 1.4 读者对象和阅读建议

本说明书的预期读者为系统设计人员、软件开发人员、软件测试人员和项目评审人员。其中系统设计人员、软件开发人员、软件测试人员为小组内部成员，软件评审人员为课程老师或助教。

### 1.5 定义、术语和缩略语

|  |  |  |
| --- | --- | --- |
| 序号 | 术语或缩写 | 解释 |
| 1 | DNS | 域名系统服务协议，是一种分布式网络目录服务，主要用于域名与 IP 地址的相互转换，以及控制因特网的电子邮件的发送 |
| 2 | DNS特征 | 用来衡量主机进行DNS通信过程的行为 |
| 3 | DNS指纹 | 根据特征工程，提取主机DNS通信特征，对主机行为进行了多维度标识 |
| 4 | Botnet | 僵尸网络，是指采用一种或多种传播手段，将大量主机感染bot程序（僵尸程序）病毒，从而在控制者和被感染主机之间所形成的一个可一对多控制的网络 |
| 5 | 僵尸主机 | 本文表示处于僵尸网络中的主机 |
| 6 | 域/域名 | 域名（英语：Domain Name），是由一串用点分隔的名字组成的Internet上某一台计算机或计算机组的名称，用于在数据传输时对计算机的定位标识 |
| 7 | MX记录 | 邮件交换记录 (MX record)是域名系统（DNS）中的一种资源记录类型，用于指定负责处理发往收件人域名的邮件服务器。 |
| 8 | PTR记录 | PTR记录，是电子邮件系统中的邮件交换记录的一种；另一种邮件交换记录是A记录（在IPv4协议中）或AAAA记录（在IPv6协议中）。PTR记录常被用于反向地址解析。 |
| 9 | TLD | top level domain，顶级域名 |
| 10 | SLD | second level domain，二级域名 |
| 11 | DGA域名 | dga是一种算法，作用生成随机数的。用dga算法生成的域名，这种域名通常硬编码在恶意软件中。 |

# 2 系统设计概述

### 2.1 限制和约束

受小组成员知识水平和调研结果，现列出在项目开发过程中需要准守的一些标准和规则如下：

1. 开发期限：2020年12月4日前完成
2. 硬件限制：小组内成员设备对本项目的功能和要求没有问题
3. 编程语言：Python、HTML
4. 界面语言：尽量使用英文
5. DNS解析工具：Python的dpkt包
6. 开发工具：PyCharm2020.2.3

### 2.2 设计原则和设计要求

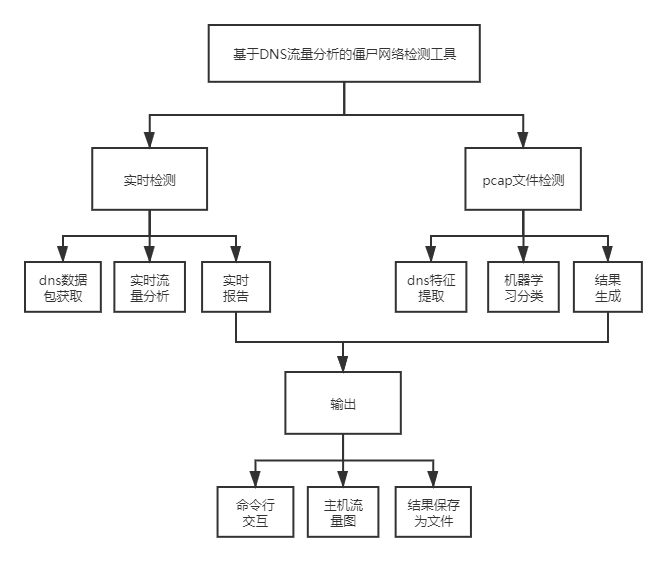
为提高小组合作效率，保证分工独立、软件的易用性、稳定性较高，本项目考虑如下设计原则和要求：

1. 命名规则：下划线分割小写字母的方式命名，如find\_abnormal\_hosts()，或者以“骆驼命名法”，为以大写字母开头的英文单词的组合，如dnsQueryCount
2. 模块独立性原则：每个模块有单独的输出，也可以被其他模块调用执行。
3. 系统容错率高：对用户输入、外界因素等做检查，防止程序出错。
4. 软件高效：优化程序的算法，降低时间复杂度，提高程序执行效率。
5. 用户友好性：提高交互性。

### 2.3 系统模块划分

#### 2.3.1 系统模型

本工具的系统功能图如下所示：



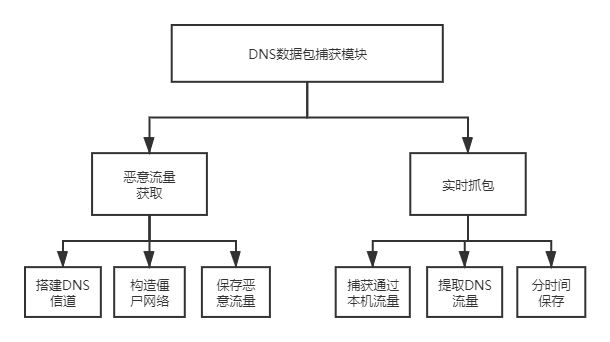
#### 2.3.2 系统模块划分

根据系统功能图，可以将本项目划分为六个模块，分别为：

1. DNS数据包捕获模块
2. PCAP文件解析模块
3. DNS指纹生成模块
4. 机器学习特征分析模块
5. 机器学习分类模块
6. 辅助输出模块

# 3 功能模块设计

### 3.1 DNS数据包捕获模块

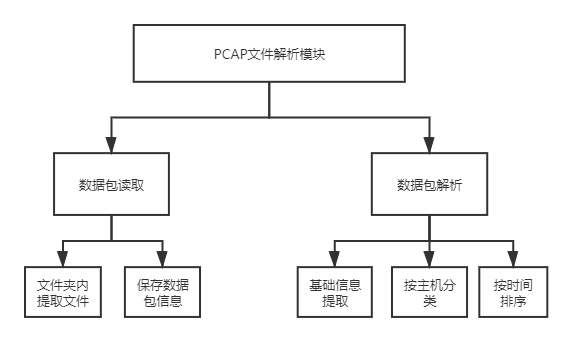


本模块的功能为获取DNS数据包。

一方面用于机器学习，包括获取正常的数据包和恶意的数据包。正常的数据包可以从公开流量数据集中获取，异常的数据包考虑构建一套DNS数据制造和收集的自动化框架，涵盖几种常见的DNS隧道工具（iodine/ozymandns/dns2tcp/dnscat2/Cobalt Strike）。

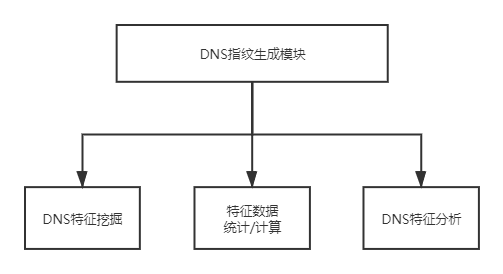
另一方面用于实时检测中的DNS数据包的抓包、过滤、切片。

### 3.2 PCAP文件解析模块



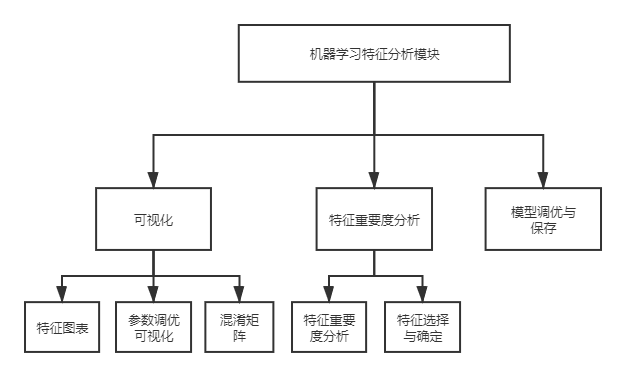
本模块主要功能为解析pcap文件，根据每一条请求/响应数据包，进行统计，为DNS特征挖掘做准备。

### 3.3 DNS指纹生成模块



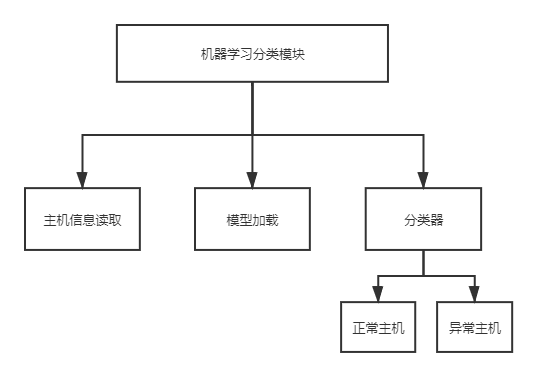
本模块的主要功能为进行DNS特征的提取，保存为CSV文件。

### 3.4 机器学习特征分析模块



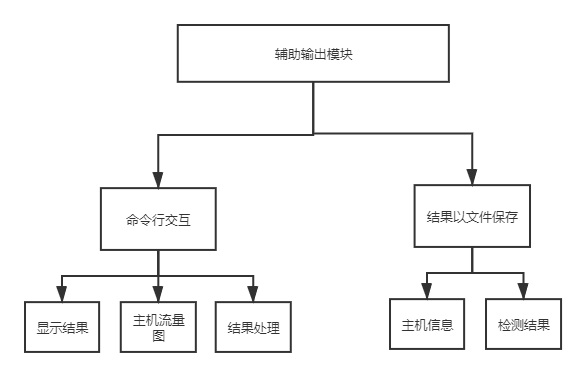
本模块的功能主要为通过机器学习，生成一个准确度最高的模型，用于分类网络中的DNS流量，包括对特征重要度的分析、特征的选择、模型参数的调整等

### 3.5 机器学习分类模块



本模块的主要功能为利用机器学习的结果，以生成好的特征值文件作为输入，输出分类好的僵尸主机或正常主机。

### 3.6 辅助输出模块



本模块主要是辅助模块，是与用户交互的界面，询问/帮助用户如何使用程序、处理检测的结果。

# 4 界面设计

界面最好是以图形化界面的形式展示，但是考虑到时间问题，可以采用命令行的方式进行交互，但是要注意以下几点：

1. 命令行界面应当友好，即提示语句排列清楚，输入人性化
2. 对于软件运行过程中会有实时输出，避免卡顿/运行不容易区分
3. 对于用户错误输入，应当能够检测，并有所提示，然后为用户提供再次输入的机会

# 5 软件环境设计

### 5.1 开发环境

程序开发阶段需要进行网络数据包捕获、测试机器学习、文件分析等，考虑到时间和小组成员条件等因素，故对开发环境有一定的要求如下：

1. 操作系统：Windows 10 / MAC
2. Python环境：python2.7.9
3. 开发工具：PyCharm、office-Excel、notepad++等
4. 联网要求：需要能连接互联网
5. 虚拟机要求：至少两台虚拟机，系统为：kali linux2020，Ubuntu18
6. 其他工具：wireshark、浏览器等

### 5.2 运行环境

程序运行时会利用事先机器学习训练好的模型，不需要有过高的系统要求或配置要求，故对程序运行环境（测试环境）有下列要求：

1. 操作系统：Windows 7 及以上 / MAC / Linux均可
2. Python环境：python2.7.9
3. 运行工具：PyCharm / 命令行均可
4. 联网要求：需要能连接互联网

# 6 安全设计

### 6.1 系统备份设计

为防止因硬件、软件原因导致的程序异常中断而造成数据丢失，因对数据具有备份以及及时保存措施，包括

1. 实时检测的网络通信数据包
2. 文件检测的数据包文件备份
3. 已提取的特征值输出到文件保存备份
4. 机器学习结果保存下来备份
5. 异常检测结果实时保存输出到文件进行备份

### 6.2 系统容错设计

为提高系统容错能力，防止由于网络数据包复杂、用户输入不合法、检测文件异常等问题，开发程序时应注意以下内容

1. 用户的错误命令输入，应当有所检查，并给予用户提示
2. 当用户没有将网络数据包文件放置到正常存储目录时，应该有所检查，并给予用户提示
3. 当用户通过抓包的方式获取网络数据包，但是网络出现问题时，应当有所检查，并给予用户提示
4. 当用户误删项目中的一些输出、输入文件夹、实时检测备份文件时，应当有所检查、并重新生成这些文件。