ET Distribution Firewall

Outline Design

概要设计说明书

1 引言

1.1 编写目的

本系统是一个分布式入侵检测实验系统和一个分布式防火墙的组合体。现在的个人防火墙基本上是各自为政，缺乏分布式协作检测，很容易受到外界的恶意入侵。本软件提倡的正是一种协作机制，优化网络资源和效率。本系统还立足于现在没有一个实验系统可以同时兼容多种误用检测和异常检测的系统。为广大安全人员提供简易有效的实验平台。并且本系统可能是国内第一个 windows 系统下基于统计的异常检测系统。

本系统采用所有最新技术如 SOAP 架构，WEBSERVICE，ASP.NET,ADO.NET 等技术（详情请看详细设计），技术跨越从驱动层到高级应用层。完全按照现在网络安全急切需求设计，希望能带给大家一种全新的感受。

1.2 背景

说明：参考需求文档

系统的名称：ET Distribution Firewall

项目的开发者：张智卓、朱志强， 林茂

1.3 定义

列出本文件中用到的专门术语的定义和外文首字母组词的原词组。

1.4 参考资料

- **[1]Selection, Combination, and Evaluation of Effective Software Sensors for Detecting Abnormal Computer Usage *KDD’04*, August 22－25, 2004, Seattle, Washington, USA.**
- **[2] Simple, State-Based Approaches to Program-Based Anomaly Detection C. C. MICHAEL and ANUP GHOSH Cigital Labs**

- **[3]Documents of Snort**
- **[4]Documents of Bro**
- **[5] Analysis and Mathematical Justification of a Fitness Function used in an Intrusion Detection System, Pedro A. DiazGomez, Dean F. Hougen**
- **[6] http://www.ll.mit.edu/IST/ideval/index.html**
- **[7] Immunity-Based Intrusion Detection System Design, Vulnerability Analysis, and GENERTIA's Genetic Arms Race,** Haiyu Hou, Gerry Dozier
- **[8] Characterization of Network-Wide Anomalies in Traffic Flows,** Anukool Lakhina, Mark Crovella, Christophe Diot
- **[9] http://www.xfilt.com**
- **[10] http://www.xfocus.net**
- **[11] http://www.checkpoint.com**

2 总体设计

2.1 需求规定

详情请参考“需求分析”

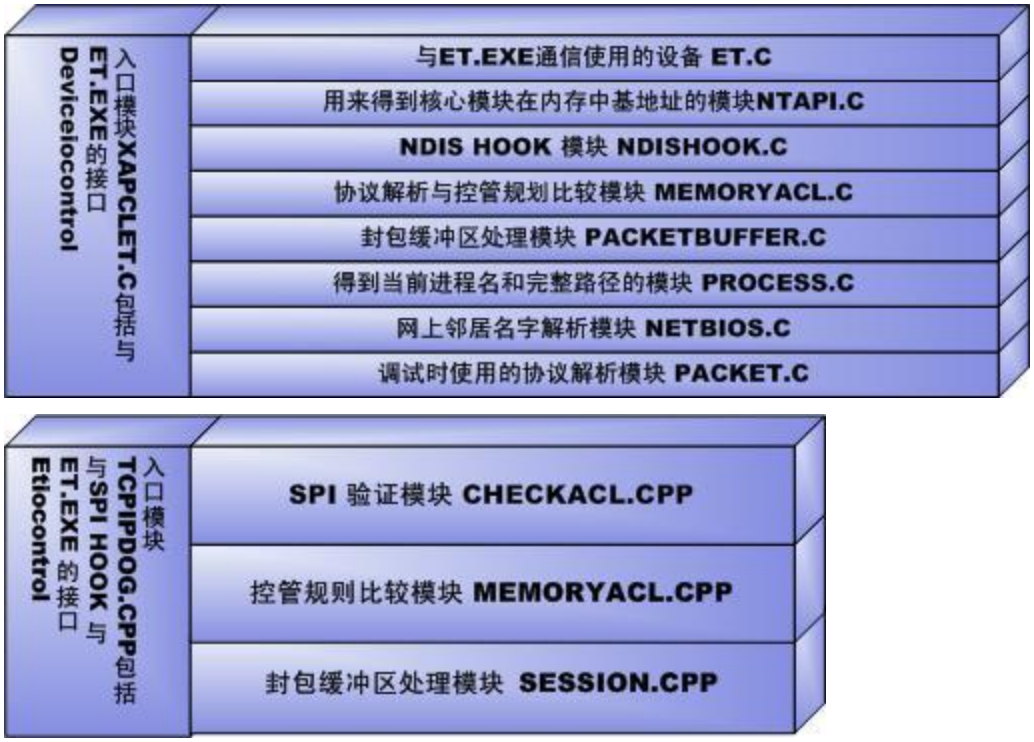
2.2 运行环境

- Processor ： Pentium III 1G or higher
- Operating System: Win2k , .NETFramework 1.0 or higher
- Memory : 128M minimum ,256M or more recommended.
- Hard disk: 10 GB minimum, 50 GB or more for statistic files recommended.
- User privileges: PowerUser of win2k
- (Server) SuperAdmin of SQLServer and IIS
- Network Interfaces: one or more Ethernet Card

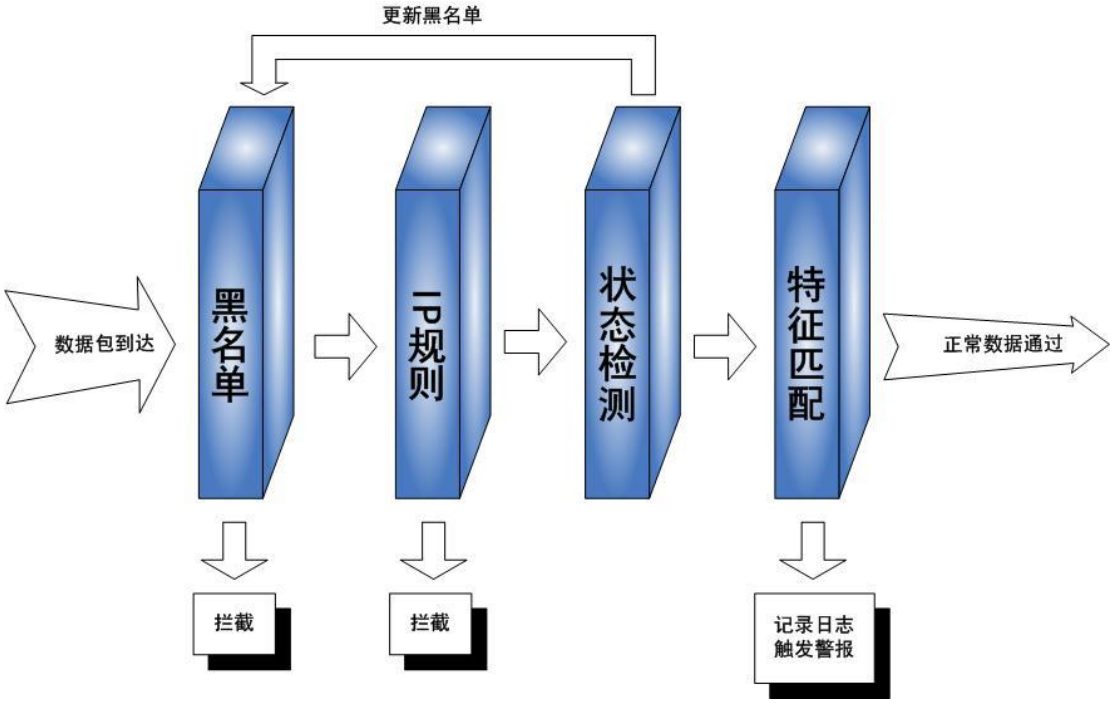
2.3 基本设计概念和处理流程

- **Driver Layer（驱动层）**

- **Application Layer（应用层）**



- **Network Layer（网络层）**
- **DataAccess Layer（数据层）**



2.4 功能结构

- **Firewall（防火墙基本功能）**

- **Communication（通信机制）**
- **Strategy And FeedBack（策略与反馈）**
- **Misuse Detection（误用检测）**
- **Anomaly Detection（异常检测）**

2.5 功能需求与程序的关系

	防火墙基本模块	分布式通信模块	策略与反馈模块	误用检测模块	异常检测模块
拦截非法网络访问	√				
状态检测（防御 DDOS 攻击）	√			√	√
设置 IP 规则	√				
采取黑名单机制	√	√	√		
共享黑名单		√			
灾难恢复		√			
访问控制、询问	√				
WebService Interface		√	√		
发送存活信息		√	√		
全局广播		√	√		
升级特征规则库		√		√	
邮件报警		√	√		
短信报警		√	√		
Web 远程管理		√	√		
服务端远程监控		√			
自定义策略脚本			√		
自定义专家动作脚本				√	
自定义特征脚本				√	
自定义算法脚本					√
动态编译脚本			√	√	√
6000~~10000 个系统统计接口					√
基于上下文关系的特征匹配				√	
检测类型选项				√	
日志功能	√		√	√	

2.6 人工处理过程

ASP.NET 和 SQLSEVER 的安装和设置需要用户人工处理。

2.7 尚未问决的问题

误用检测尚未能够独立处理 HTTP 流，现在先用 TCP 流代替处理。

3 接口设计

3.1 用户接口

提供一个分页的用户主界面，该界面是一个标准的Windows 有模式窗体，
提供一个快速用户向导，在用户主界面左方提供一个类似QQ 的方便的功能菜单，点击；
对应每一个功能菜单，切换一个分页供用户交互选择功能， 跟主流单机防火墙风格相似。

3.2 外部接口

WebService 的各个方法接口：
自定义脚本接口：

```
using System;
using System.IO;

namespace ET
{
    public class Plugins
    {
        public Plugins()
        {
        }

        public bool Counter(object parameter)
        {
            object[,] o=(object[,])parameter;
            double[,] d=new double[o.GetUpperBound(0)+1,o.GetUpperBound(1)+1];
```

```
                for(int i=0;i<o.GetUpperBound(0)+1;i++)
                {
                    for(int j=0;j<o.GetUpperBound(1)+1;j++)
                        d[i,j]=Double.Parse(o[i,j].ToString());
                }
///下面编写你自己的处理脚本
    return true;
        }
    }
}
```

动态编译接口：

特征规则定义：

4 系统数据结构设计

详情请查看“数据结构说明”

5 系统功能模块设计

5.1 防火墙基本功能

1、核心技术：NDIS-HOOK 和 DeviceIOControl

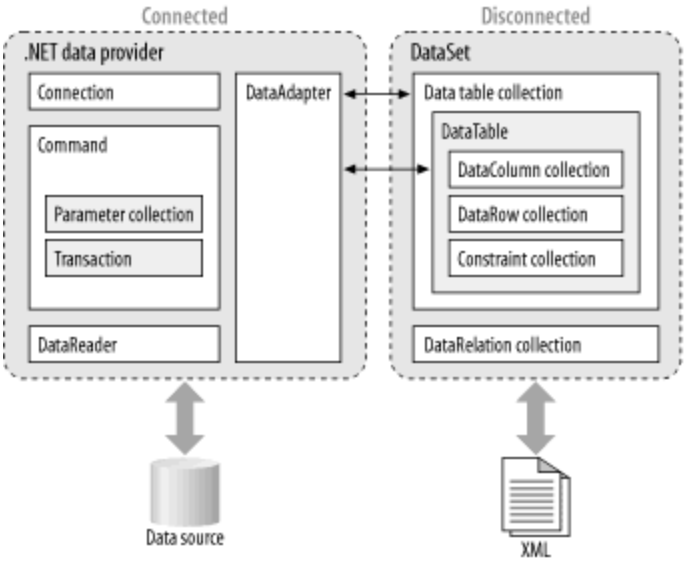
2、状态检测：

- **首先，对于一个会话我们使用什么来区分。从最简单的角度出发，我们可以使用源地址、目的地址和端口号来区分是否是一个会话。**
  当通过使用一个 SYN 包来建立一个会话时，防火墙先将这个数据包和规则库进行比较。如果通过了这个数据连接请求，它被添加到状态检测表里。这时需要设置一个时间溢出值，参考 CHECK-POINT FW-1 的时间值，将其值设定为 60 秒。然后防火墙期待一个返回的确认连接的数据包（ACK 包），当接收到如此的包的时候，防火墙将连接的时间溢出值设定为 3600 秒。对于返回的连接请求的数据包的类型需要做出判断，已确认其含有 SYN/ACK 标志。（注：对于时间溢出值，是参考国外著名的网络安全公司 Check-Point 的标准）

5.2 分布式通信模块

1、与 SQLServer 通讯

采用 ADO.NET 技术，更高效地处理分布式的数据访问。



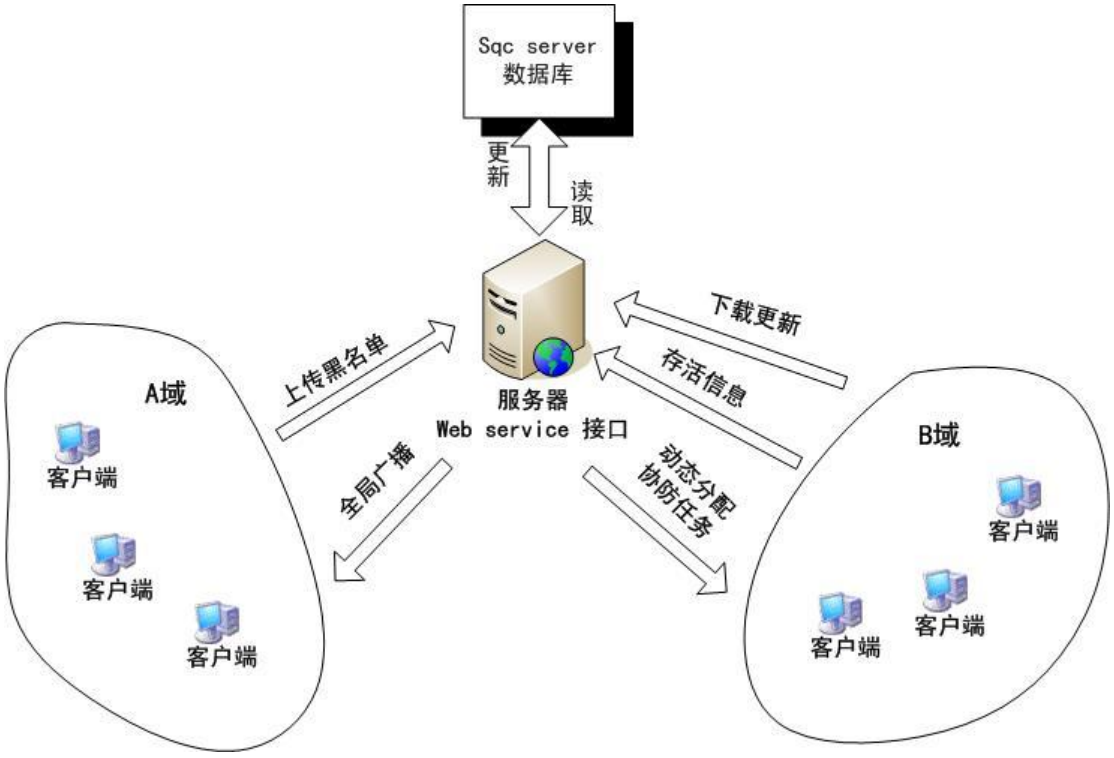
2、webservice 架构

基本上提供服务端所有方法接口，客户端可以穿透防火墙和路由访问服务端的 SOAP 接口，服务端需要 IIS 和 SQLServer 支持。

5.3 策略与反馈模块

1、允许自定义黑名单和防火墙规则，威胁级别高的黑名单会发送到服务端，由服务端分发给各个客户端共享。

2、全局广播，可以在服务端设置或编写自动策略脚本，但满足一定条件，服务端将自动按照策略脚本调节全局或特定 IP 段的客户端的最低警戒级别以及发送升级信息或广播信息。



5.4 误用检测模块（NIDS）

1、模块介绍：这个误用检测模型基于网络入侵检测专家系统，很大地吸收了国外著名的 NIDS 系统 Snort 和 Bro 优秀高效的特点，并且结合.Net 平台的开发优势。

2、强大的特征匹配功能

本系统采用基于上下文关系的高级规则匹配模式，用正则表达式来定义数据包内容。例如：

```
signature rpc-dcom_bind-req {   # 定义一个叫 rpc-dcom_bind-req 的规则，此规则匹配一个攻击会话
                        # 中必定先于实际攻击数据包出现的绑定 RPC DCOM 的数据包，以此规
                        # 则匹配出来的事件不单独触发告警，用于被以下的两个规则做关联。
  header ip[9:1] == 6          # 匹配 IP 包头的协议字段为 6，也就是 TCP 包。
  header tcp[2:2] == 135       # 匹配 TCP 包头的目标端口字段为 135，也就是 RPC DCOM 漏洞常用的攻击端口。
  tcp-state originator,established # 匹配 TCP 包状态为已建立且从连接的发起者发出。
  payload
/\x05\x00\x0b.{100,}\xa0\x01\x00\x00\x00\x00\x00\xc0\x00\x00\x00\x00\x00\x00\x46
.*\x04\x5d\x88\x8a\xeb\x1c\xc9\x11\x9f\xe8\x08\x00\x2b\x10\x48\x60/ # 用正则表达式匹配 RPC DCOM BIND 包的 payload 特
                                # 征(RPC 包头及两个特定的 UUID)。
  event "RPC DCOM BIND request"          # 定义可能出现在告警日志的事件信息。
  }

signature rpc-dcom_servername-overflow { # 定义名为 rpc-dcom_servername-overflow 的规则，此规则
                        # 匹配 RPC DCOM 长主机名的栈溢出攻击。
  header ip[9:1] == 6
  header tcp[2:2] == 135
  requires-signature rpc-dcom_bind-req   # 指定需要 rpc-dcom_bind-req 规则先要匹配到本规则才能
                        # 成立，也就是说此规则关联了同一个 TCP 会话中的另一个
                        # 规则，这是 Bro 规则提供的特有功能。具体到本规则，表
                        # 达的意思就是只有在看到有 RPC DCOM BIND 操作以后才会
                        # 再去匹配接下来的数据包中是否有攻击数据包。
  tcp-state originator,established
  payload /.*\x05\x00\x00.{100,}\x5c\x00\x5c\x00[^\\]{32,}/ # 匹配攻击数据包的 payload 特征，用正
                                # 表达式检查是否存在超过 32 字节的主机
                                # 名。
  event "RPC DCOM servername stack overflow attempt"
```

}

signature rpc-dcom_pathname-overflow { # 定义名为 rpc-dcom_pathname-overflow 的规则，此规则
匹配 RPC DCOM 长路径名的堆溢出攻击。
header ip[9:1] == 6
header tcp[2:2] == 135
requires-signature rpc-dcom_bind-req # 同上条规则一样的前提条件。
tcp-state originator,established
payload /.*\x05\x00\x00.{100,}\x5c\x00\x5c\x00[^\\]{5,32}\x5c\x00.{520,}\x00\x00/ # 匹配攻击数据
包的 payload 特
征，用正则表达
检测是否存在超
长的路径名。
event "RPC DCOM pathname heap overflow attempt"
}

3、对 Snort & Bro 的兼容、支持

- **signature rpc-dcom_servername-overflow**
- **{**

header ip[9:1] == 6
header tcp[2:2] == 135
tcp-state originator, established
requires-signature rpc-dcom_bind-req
payload /.*\x05\x00\x00.{100,}\x5c\x00\x5c\x00[^\\]{32,}/
event "RPC DCOM servername stack overflow attempt"
eval “RPC_DCOM.dll”
}

与 Bro 规则唯一不同的是 eval 项填入.Net 的动态链接库来处理对应事件
详细定义请参考（Bro-Ref-Manual.pdf）

4、自定义脚本插件

用户可以用上.NET 平台的任何一种语言（包括 C++,VB,JAVA,C#）来开发你自定义的事件动作脚本，比起 Bro 和 Snort 只能用 Perl 来编写脚本有更大的优越性，并且本系统同样支持动态编译(C#).

5.5 异常检测模块（Anomaly Detection）

1、模块介绍：本模块的架构设计主要参考 MIT（美国麻省理工）在 1998—2000 年的一个开放项目，

- **基于状态的异常检测方法（State-Based Approaches of Anomaly Detection.）**

提供 6000—10000 个系统统计接口（因不同用户的系统所安装的服务、版本而不同）覆盖系统每个方面：

- **与误用检测相结合**

同时提供 30 多个误用检测的统计接口，让用户可以更加准确地描述入侵行为。

- **算法模型（参考论文）**

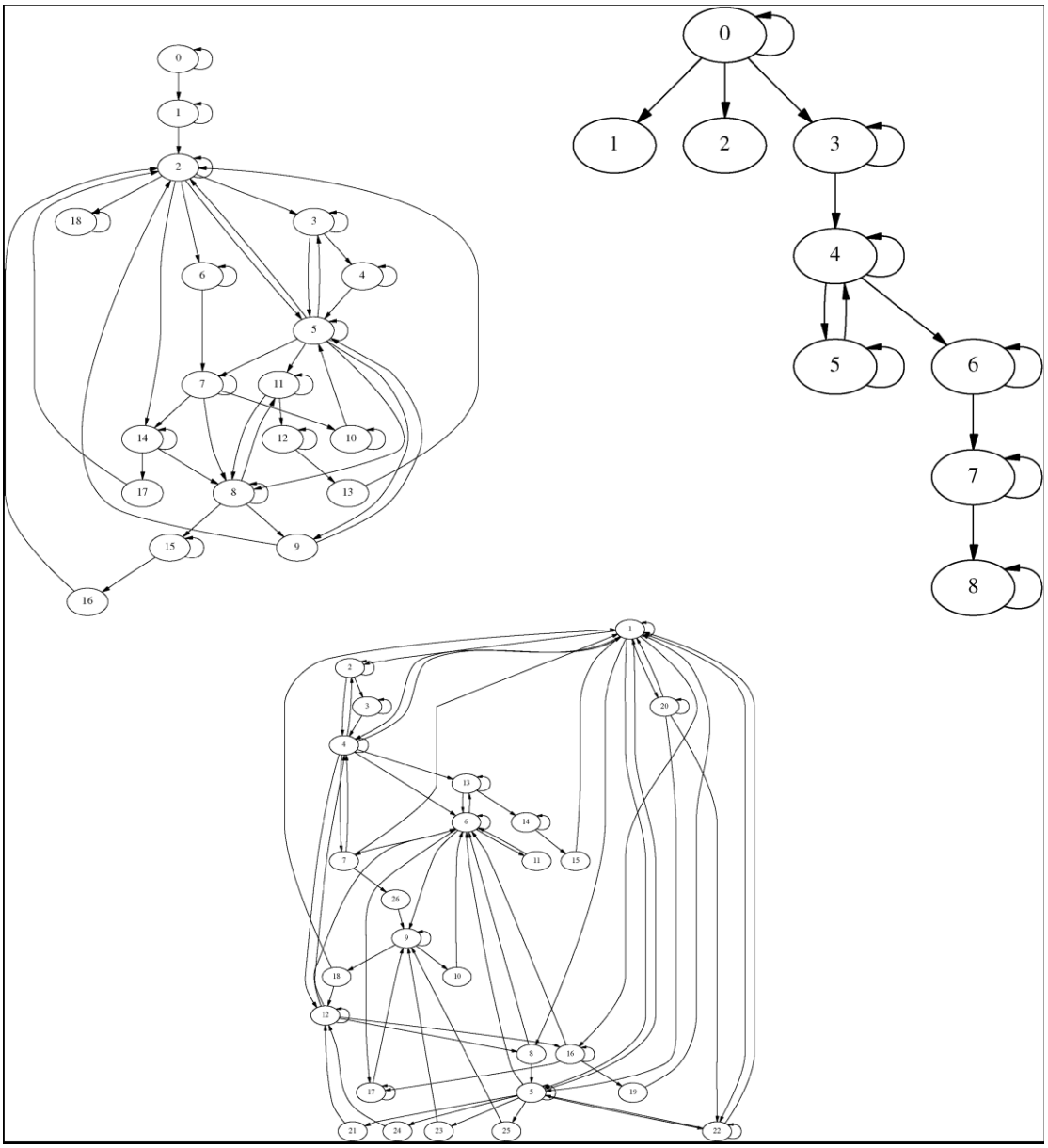
这里我们提供一个基于遗传算法和统计概率算法的模型

模型设计：

通过选取多个系统特征状态变量[**Detecting Abnormal Computer Usage**]，通过每格W秒采集一次数据，假设A个样本可以描述一个系统状态，每个样本包含M个特征，并且M个特征相互独立。我们可以采用高斯分布来描述每个特征的概率分布，最后通过乘法原理来得到该样本集出现的概率。把得到的概率基数与显著程度 r 比较，可以得出该样本集描述的状态是已有的状态还是新的状态。

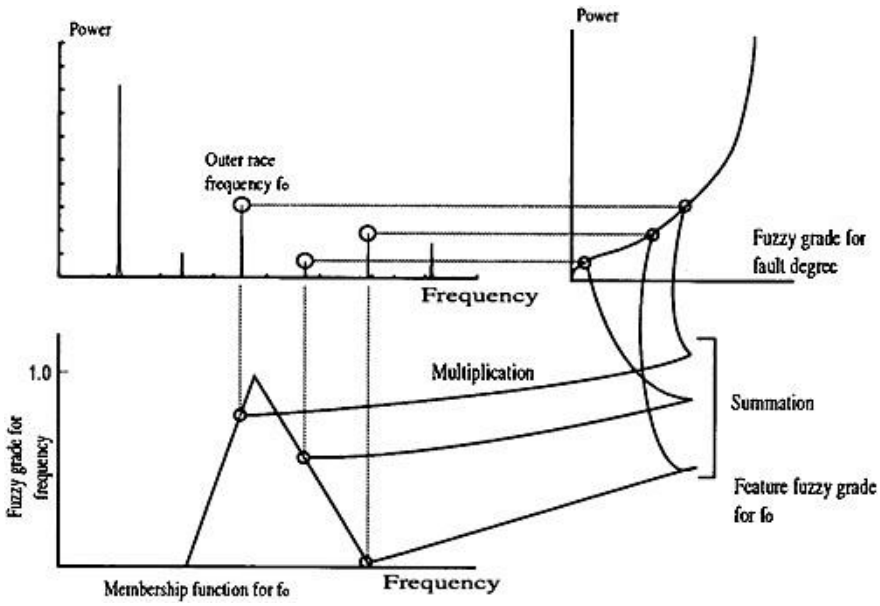
分三个过程：

1、创建状态过程

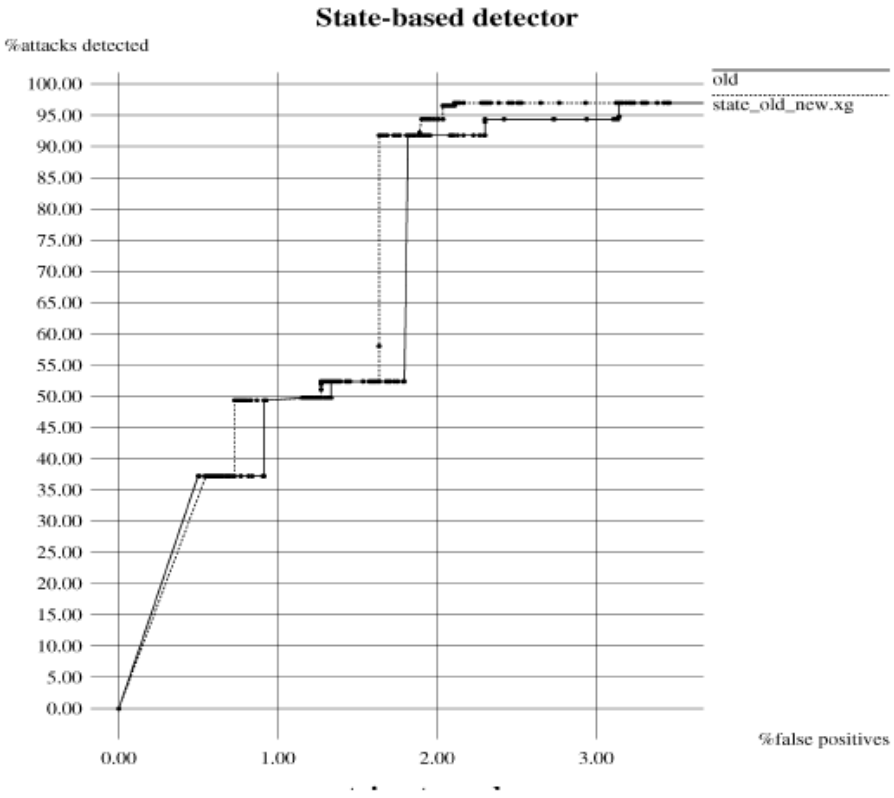


2、人工智能学习过程：

通过神经网络对各个接口的权值进行学习和调整。


Power
Power
Outer race
frequency fo
Fuzzy grade for
fault degree
Frequency
1.0
Multiplication
Summation
Fuzzy grade for
frequency
Feature fuzzy grade
for fo
Membership function for fo
Frequency

3、遗传算法选择过程


State-based detector
%attacks detected
100.00
95.00
90.00
85.00
80.00
75.00
70.00
65.00
60.00
55.00
50.00
45.00
40.00
35.00
30.00
25.00
20.00
15.00
10.00
5.00
0.00
0.00
1.00
2.00
3.00
%false positives
old
state_old_new.xg