计算机网络和信息集成
教育部重点实验室

面向全流量的网络APT智能检测方法

程 光

网络空间安全学院
计算机科学与工程学院、软件学院
计算机网络和信息集成教育部重点实验室

东南大学
2017年4月17日

报告提纲

1 研究背景

2 检测架构

3 检测方法

研究背景

- APT，即高级持续性威胁
- 高级性智能技术
  - 攻击行为特征不确定：“零日攻击”
  - 攻击渠道的多元化：水坑式攻击，鱼叉式攻击，社会工程学
  - 攻击空间的不确定性：多路径入侵
- 持续性隐藏技术
  - 攻击周期长：长时间潜伏，多阶段攻击
  - 潜伏手段隐蔽：Rootkit技术，清除日志，隐藏活动
  - 通信流量：低流量，低频率
- 威胁
  - 窃密活动：关键词，搜寻敏感数据
  - 数据回传：隐蔽通道，加密数据

研究背景

- **APT的大数据特征**
  - 数据的价值密度变得更小、更分散，很难聚焦高价值的信息；
  - 用于分析的数据类型和数据格式多种多样，日志信息的行为、内容、结构各异；
  - 数据体量巨大、增长速度快，时间跨度长；
  - 多点攻击事件协同分析、关联，分布式的检测体系
- **传统检测体系结构无法适应APT大数据的特点**

研究背景

- **APT的全流量采集和存储**
  - APT攻击的潜伏和持续性特点，数据的分析和存储需要适应巨大的时间跨度特性；
  - 单点单攻击检测无法形成对APT攻击的判定，需通过对历史数据回溯，进行关联分析；
  - 首次检测无法时完全发掘APT攻击的特征，对数据进行二次筛选、分析，进一步挖掘特征；
- **全流量数据并非全网、全数据量，而是对所需保护对象的全流量采集和长期数据存储**

研究背景

- APT的智能检测
  - 从海量的网络流量中进行数据挖掘
  - 恶意事件的关联分析和规则挖掘
  - 根据已发现的特征或知识对未知的APT攻击进行判定，对APT攻击进行预测和泛化
  - 对APT检测的动态性、大规模、复杂性进行自动管理和优化
- 人工智能技术
  - 机器学习、仿生智能计算、模糊神经网络等

报告提纲

1 背景介绍

2 检测架构

3 检测方法

东南大学

APT智能检测架构

检测应用

C&C通信特征检测

C&C通信机制发掘

加密流量分类

周期性检测

历史数据回溯

…

特征提取

Heavy Hitter

Entropy

HHH

SS

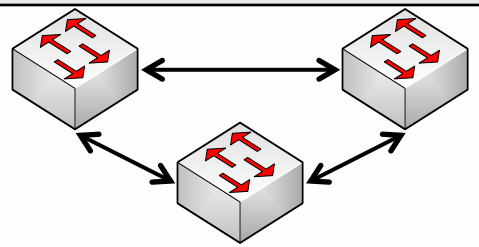
Traffic Change

…

资源分配和配置

数据预处理

配置



数据

APT智能检测架构

- **数据预处理**
  - 从网络中直接获取的实时网络流，进行会话还原，消除因网络条件造成的乱序、重传、延迟等对后续分析的干扰；
  - 进行应用协议识别；
  - 从非结构化的数据流中抽取结构化的**元数据信息**，以便后续的各类统计和关联分析。
- **元数据信息**
  - 对原始报文的描述数据
  - 包含原始报文的具体内容，如：开始时间、结束时间、源地址、目的地址、源端口、目的端口、协议类型、应用类型、上行流量、下行流量、数据包分布状况等

APT智能检测架构

- **特征提取**
  - 抽取元数据，根据检测需求组合成相应的检测特征
  - 利用元数据构建大数据存储及检索索引，为后续安全分析提供底层技术支撑
  - 特征提取分析中，只需操作元数据表，而在需要分析原始报文时，通过查询元数据表索引，即可快速定位到相关报文

检测方法

- 周期性检测
  - 心跳信号是检测未知APT 攻击的重要连接特征
  - 由于被控端一般会周期性地链接控制端，以更新状态、获取新的指令，连接的周期和频率较为固定
- C&C通信特征检测
  - 同一APT攻击恶意软件及其变体与C&C服务器通信流量的模式往往是一致的
  - C&C通信连接的URL通常采用特定的编码格式
  - 相当数量的APT攻击利用已知的恶意软件进行C&C通信

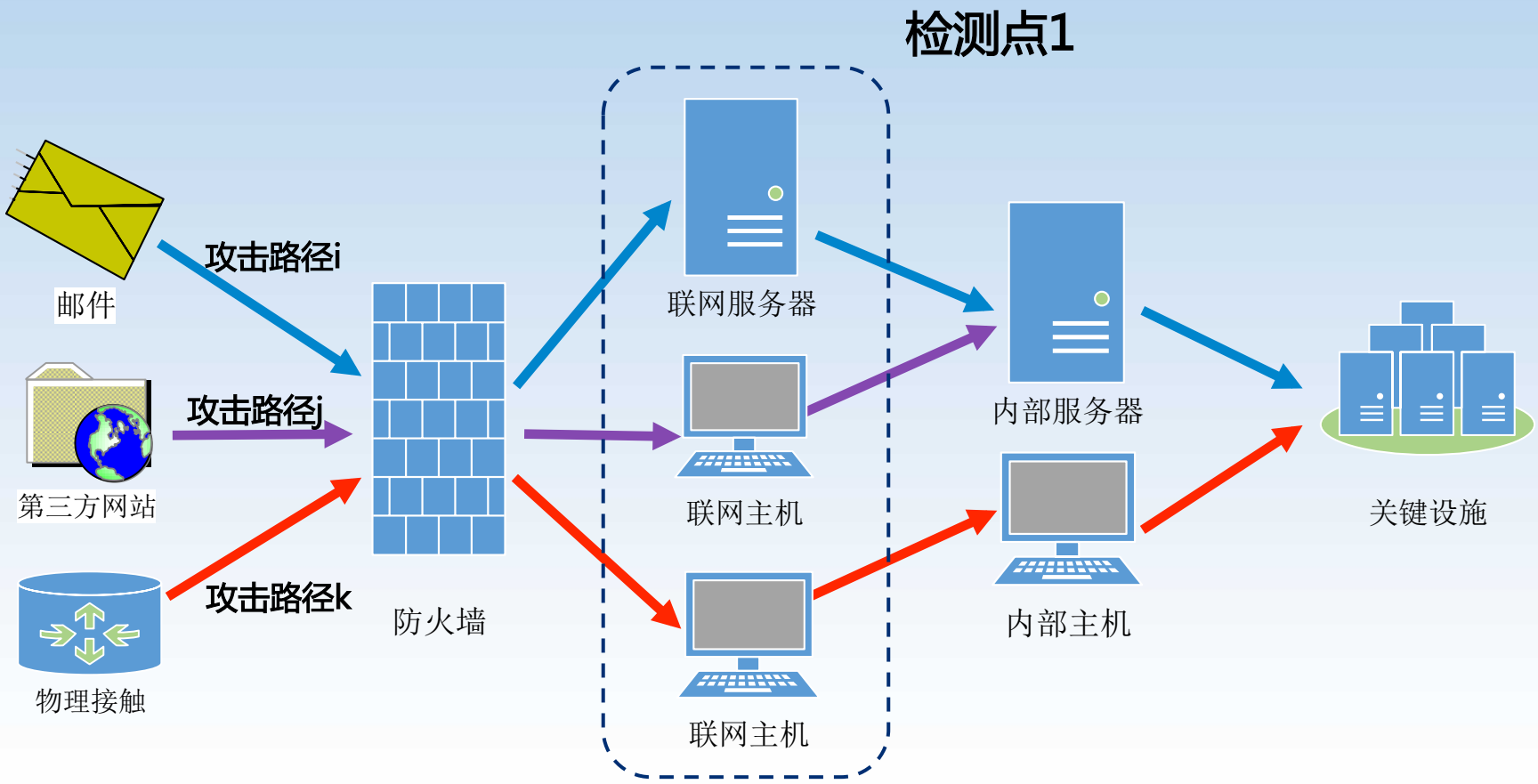
检测方法

- **命令控制机制**
  - 恶意软件将Web搜索服务用于命令控制服务器节点查找过程以提高其隐蔽性；
  - 以HTML代码为载体通过信息隐藏传送命令控制信息，以保持该过程过程隐蔽性与时空复杂度的折中；
  - 构建面向本地网络环境下同类恶意软件的命令控制信息分发策略，降低本地恶意软件被流量行为追踪的风险；
- **加密流量识别**
  - 恶意软件为规避检测，对传输命令和数据进行加密
  - SSL/TLS 协议滥用，SSL/TLS 隧道北广泛应用于躲避网络监管
- **数据回溯**
  - 对未知的APT攻击事件进行关联分析
  - 实现快速的历史数据定位

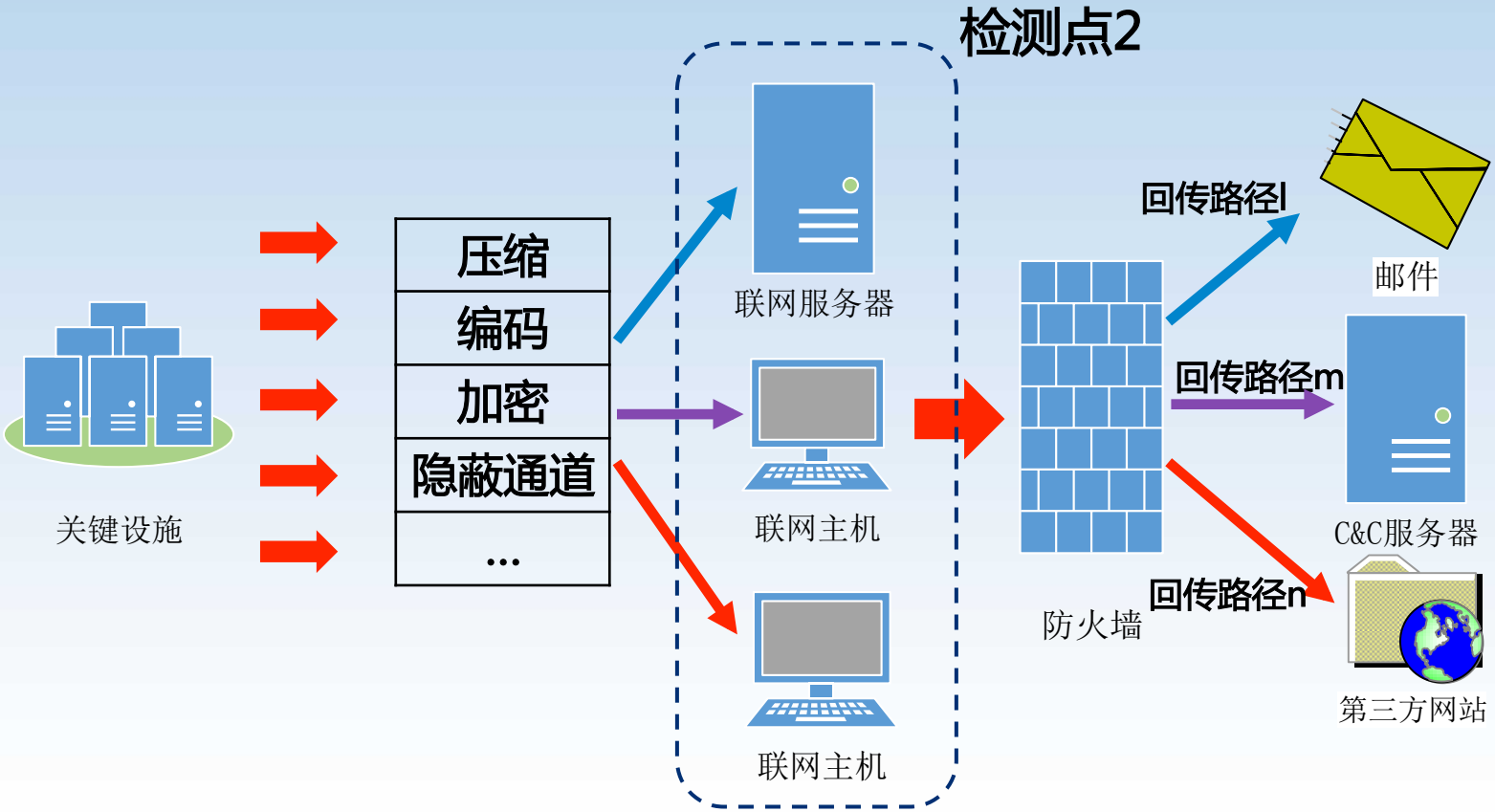
APT攻击检测方法

1. 周期性检测
2. C&C通信特征检测
3. 命令控制机制
4. 加密流量识别
5. 数据回溯

定向入侵模型

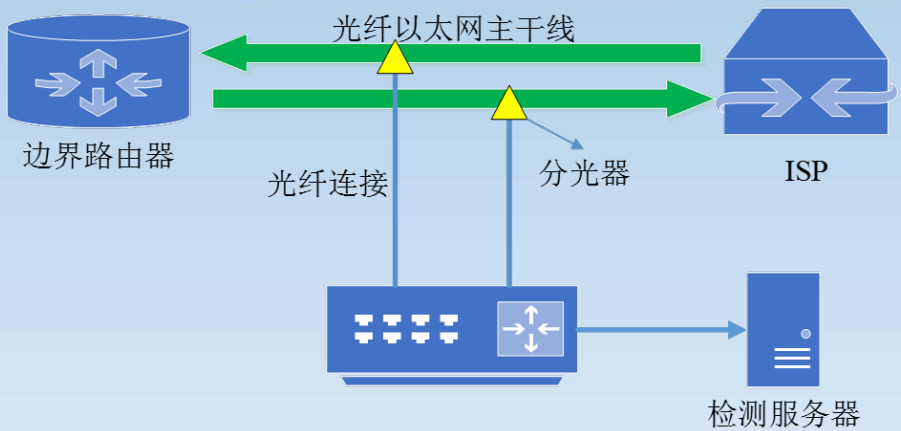

检测点1
邮件
攻击路径i
第三方网站
攻击路径j
物理接触
攻击路径k
防火墙
联网服务器
联网主机
联网主机
内部服务器
内部主机
关键设施

数据回传模型


检测点2
关键设施
压缩
编码
加密
隐蔽通道
...
联网服务器
联网主机
联网主机
防火墙
回传路径l
回传路径m
回传路径n
邮件
C&C服务器
第三方网站

明确被保护对象

- 采集点的部署
  - 采集教育网边界流量
  - 根据保护对象进行过滤
  - 全流量存储

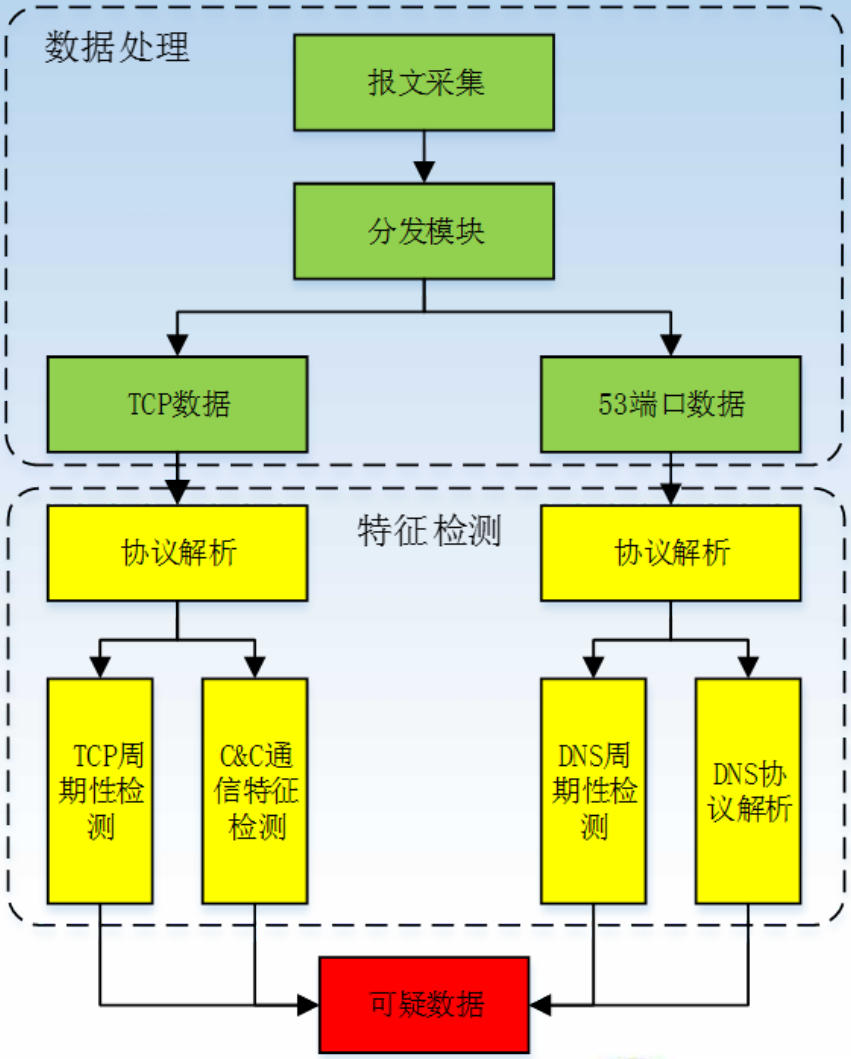


- 明确被保护的目标
  - APT是“指哪打哪”，具有高度的目标导向
  - 根据重要程度和价值，选取7个标的进行监控

单位	服务器IP地址	职能
某大学1	***.***.***.111	技术转移中心
	..***.93	教务处、科技处、研究生院
某大学2	***.***.***.16	技术转移中心
	..***.143	科技处
	..***.5	教务处
某大学3	***.***.***.60	科技处
	..***.83	教务处

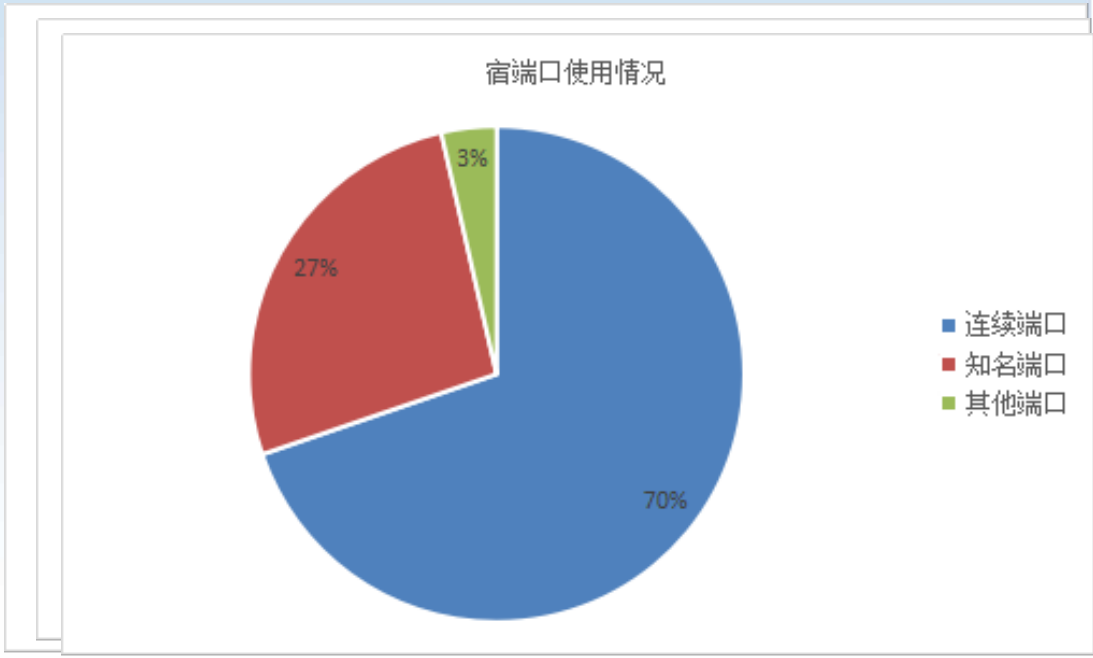
周期性检测流程

- 数据处理
  - 数据采集
  - 数据分发
- 特征检测
  - TCP周期性检测
  - DNS周期性检测
  - C&C通信特征检测
  - DNS协议解析



TCP数据的周期性特征

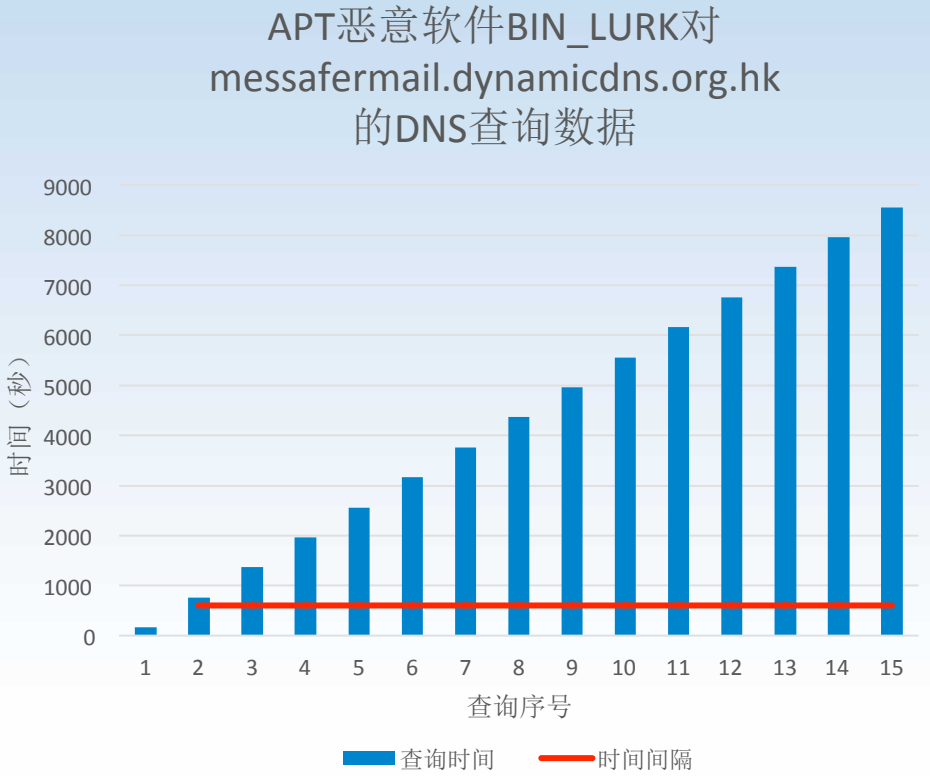
- TCP通信周期性特征
  - 通过周期性的建立TCP SYN连接继续通信
  - 传输具有周期性的数据序列，标志位为PSH ACK
- 周期性具体表现
  - 数据发送的时间间隔相同
  - 每次发送的数据在1~n个（n一般为较小的整数）
  - 使用的端口是连续的端口或形成 一个等差数列（避免流量特征过于集中）



DNS查询数据的周期性

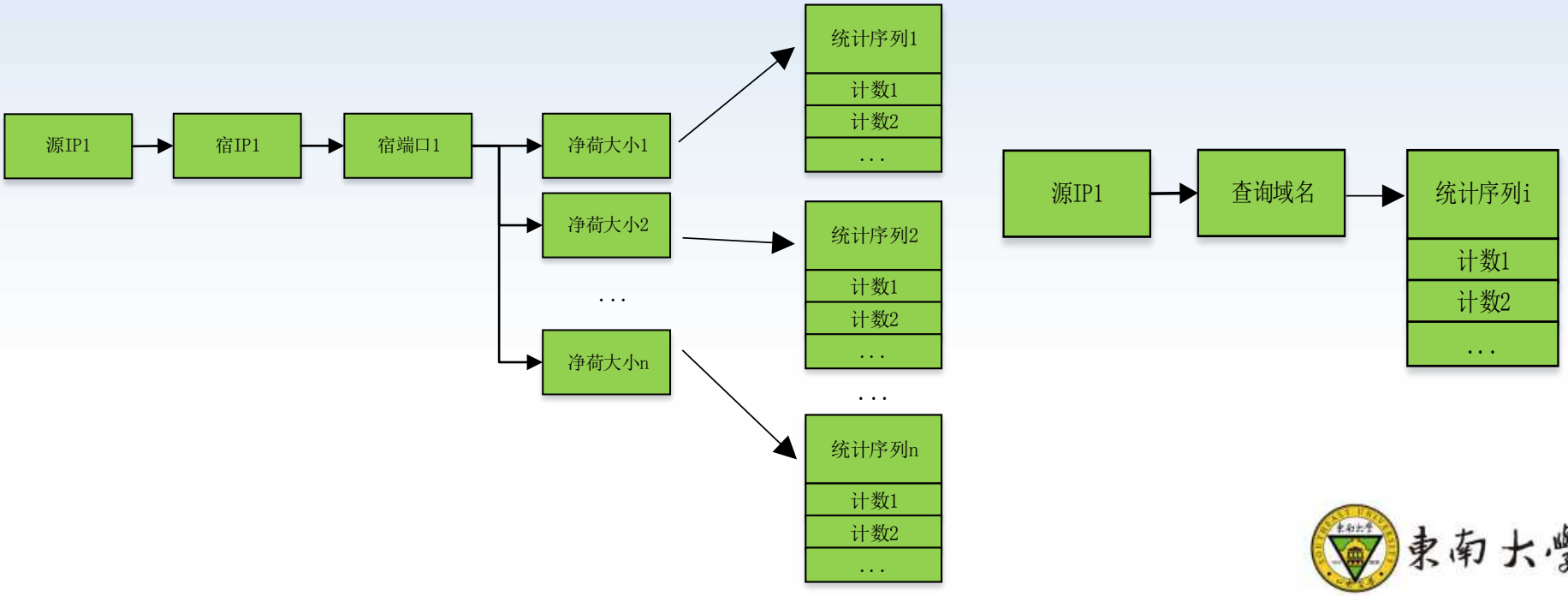
- DNS查询数据的周期性
  - 周期性的查询C&C服务器的域名
- 周期性具体表现
  - 对同一个域名的查询数据发送的时间间隔相同
  - 每次发送的DNS查询数据在1~n个

    （n为较小的整数）



周期性检测方法

- 数据聚类方式
  - DNS查询数据按照源IP、域名聚类
  - TCP数据按照源宿IP、宿端口、净荷大小进行聚类
- 检测的统计周期为T秒
  - 将T分为N个大小为t的时间窗口进行数据统计



周期性检测方法

- 使用自循环相关方法进行计算

- 将结果转换为0~1的数值，与阈值a进行比较判定

APT恶意软件BIN_LURK在检测周期内的DNS查询序列

t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
0	0	0	1	0	0	1	0	0	1

此序列被判定为具有周期性。

实验方案

- **实验数据**
  - 离线数据+APT样本流量
- **实验设计**
  - 将APT样本流量与离线数据混合检测
  - 计算检出率和误报率
- **处理速率**
  - 每秒处理20万+数据包，速率在130MB/s

数据	数据量	文件数	采集时长
离线数据	723GB	19030	5个月
APT样本	12.7MB	38	-

实验结果

- 实验结果
  - 检测结果表示
  - 结果覆盖20个APT样本
- 检出率

$$检出率 = \frac{检出的APT结果数量}{含有此特征的样本数量} \times 100\%$$

  - 检出率为76.92%
- 误报率

$$误报率 = \frac{非APT攻击的结果数量}{总的检测结果数量} \times 100\%$$

  - 误报率为20%

名称	描述
源IP	32位无符号整型
宿IP	32位无符号整型
事件类型	变长字串，记录事件的可疑类型
时间戳	变长字串，记录事件的第一个数据包时间戳

数据	SYN检测	PSH&ACK检测	DNS查询数据检测
离线数据	1	1	3
APT样本	19	1	2

APT攻击检测方法

1. 周期性检测
2. C&C通信特征检测
3. 命令控制机制
4. 加密流量识别
5. 数据回溯

C&C通信特征

- 特征来源
  - 安全防护公司的分析报告（赛门铁克、卡巴斯基、趋势等）
  - 恶意样本沙盒运行，流量采集
- Taidoor的URL请求编码格式：/[5 characters].php?id=[6 random numbers]{12 characters}
- IXESHE的URL请求格式：/[ACD][EW]S[SomeNumbers].jsp?[Encryped Base64 Blob]

```
GET /wwsyr.php?id=01576619113845C1EE HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: www.gov.toh.info
Connection: Keep-Alive
Cache-Control: no-cache
```

```
GET /AWS26329.jsp?UrFwUIOKTRyfxR9KNRqhg8lcPr/CGjUwP8yJUs=7RjH7OinJ/85cgrqJP8jKGjpqgb/
wTrO7OIjhxoHcGaFaURqK/aHophHLd23K=NHk=a9oQhvDQaLKy8qo/RnJz42A HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Host: dot.faawan.com:443
Connection: Keep-Alive
```

C&C通信的架构

- C&C架构
  - 安全公司通过关联分析恶意样本中出现的域名、IP，绘制出了攻击者的C&C通信架构

DOMAINS	REGISTRATIONS
mailru-vip.com yandex-vip.com google-officeonline.com office-helppane.com foxit-pro.com ymail-vip.com ymail-pro.com yandex-pro.com google-office.com mailru-pro.com	xiaohu wang bruce_tuner@yahoo.com +86.01089464156 fax: +86.01089464156 bei jing shi beijing beijing 102600 CN
hoticq.com redhag.com zadhc.com lasmail.com hotoicq.com	jason bush icqmaster@163.com +86.01062311307 fax: +86.01062311307 No.20 Xueyuan Road,Haidian District,Beijing beijing beijing 100083 CN

Lurid的C&C域名架构以及注册者

Table 6

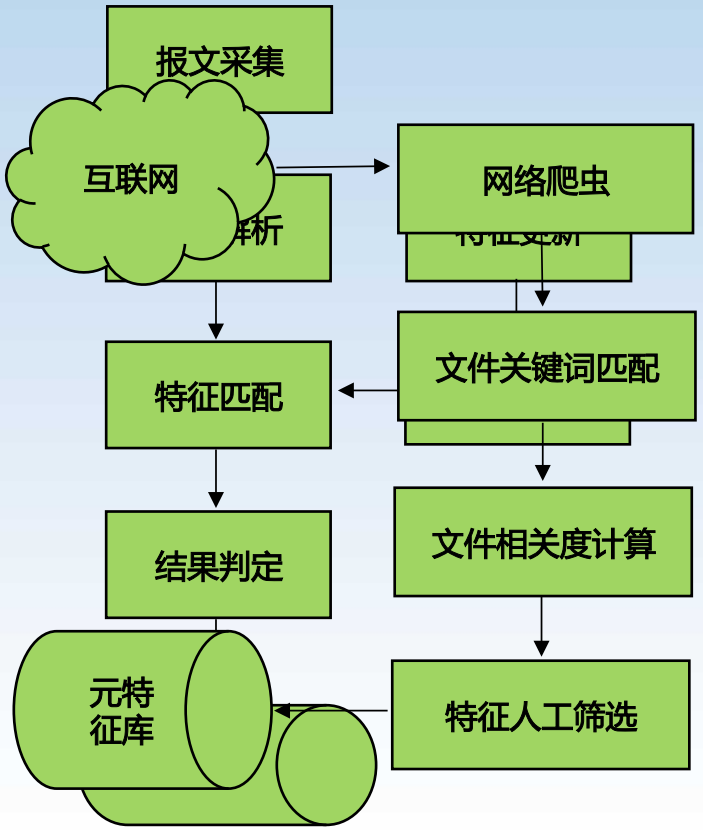
C&C servers

IP	Country	ASN	Registrar
110.142.12.95	Australia	1221	apnic
203.45.204.239	Australia	1221	apnic
220.245.107.203	Australia	7545	apnic
193.170.111.210	Austria	1853	ripencc
88.117.175.114	Austria	8447	ripencc
81.21.80.40	Azerbaijan	39280	ripencc
203.188.255.117	Bangladesh	9832	apnic
24.79.164.206	Canada	6327	arin
213.41.162.198	France	13193	ripencc
62.38.148.117	Greece	3329	ripencc
212.205.207.42	Greece	6799	ripencc
202.82.162.61	Hong Kong	4515	apnic
218.103.88.197	Hong Kong	4515	apnic

Taidoor的C&C 服务器地址

C&C通信检测

- 基于如下特征
  - C&C通信的URL地址满足特定的模式
  - 安全公司归纳的C&C域名和IP
- 系统架构
  - 根据以上特征就可以从网络层面完成对APT攻击活动的检测
  - 核心：特征更新、特征库、特征匹配



特征更新

各大安全防护公司海量的APT攻击分析报告

半自动的特征更新方法：机器和人工结合的方式

特征更新算法

- **文件相关度计算**
  - 采用**空间向量计算模型**，将关键词数量n作为空间向量的维度，设置关键词的权重$W↓i$作为一维分量的大小，该向量方法表示为：

$$\rho=(\alpha↓1,\alpha↓2,...,\alpha↓n),\ i=1,2,...,n$$

  - 计算关键词出现的**频次**，并求出对应的**频率**之比$X↓i$，文件对应向量的每一维分量为$X↓i\ W↓i$，文件的空间向量表示为：

$$\rho=(X↓1\ W↓1,X↓2\ W↓2,...,X↓n\ W↓n),\ i=1,2,...,n$$

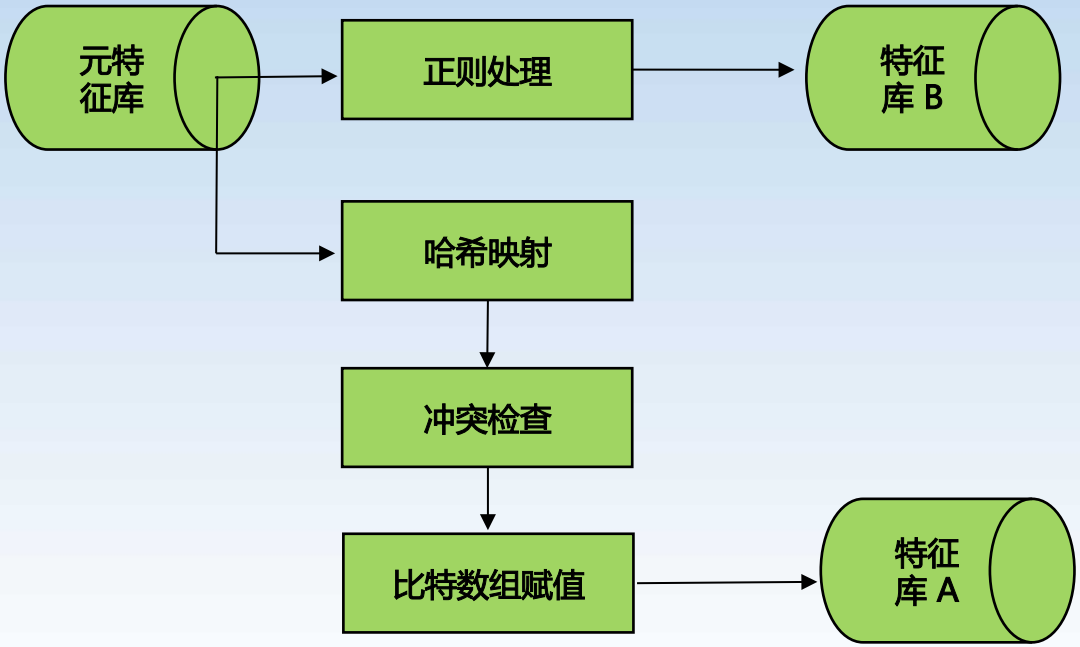
  - 用两个向量的余弦表示**文件的相关度**为：

$$\cos<\rho,q>=(\rho,q)/(|\rho||q|)=(X↓1\ W↓1↑2+X↓2\ W↓2↑2+...+X↓n\ W↓n↑2)/\sqrt{(W↓1↑2+W↓2↑2+...+W↓n↑2)(X↓1\ W↓1↑2+X↓2\ W↓2↑2+...+X↓n\ W↓n↑2)}$$

  - 设定阈值m，若$\cos<\rho,q>$大于m时，判定当前文为相关的，否则判定

特征库的构建

- 特征库
  - 分别构建供快速匹配的特征库
  - 精确匹配的特征库



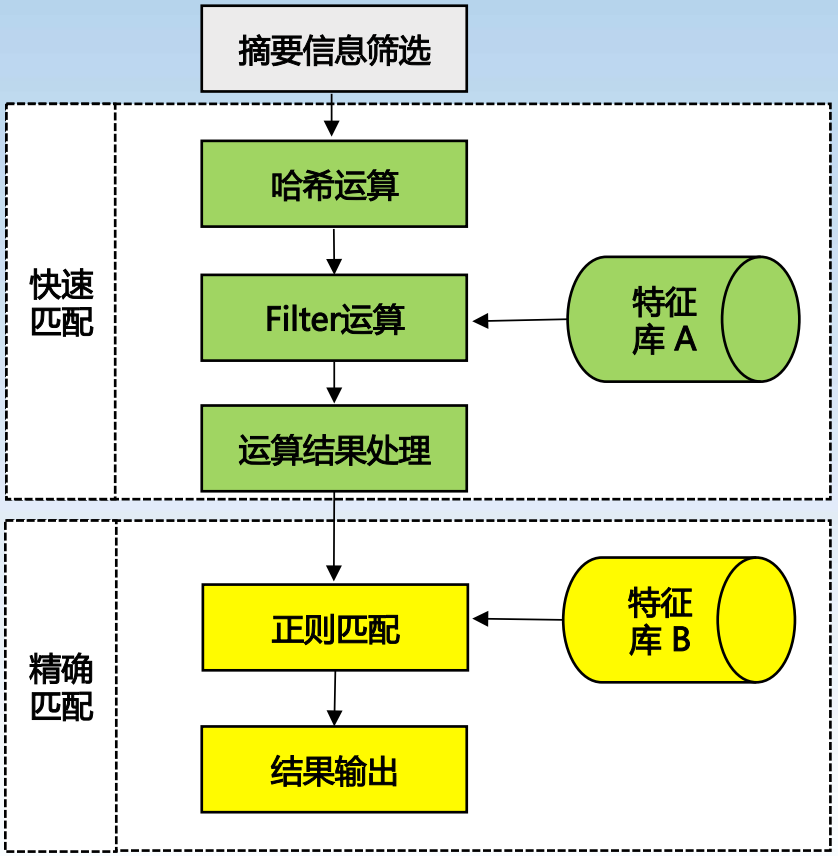
特征库建立流程图

特征库的构建

- 快速匹配特征库的构建算法
  - 采用BloomFilter算法
  - 数据集合S的大小为n， hash函数的个数为k，位数组A的大小为N，那么 Bloom Filter的误判率$P↓fp$（false positive, fp）估算公式如下：$P↓fp ≈(1−e↑−kn/N)↑k$。
  - 在实际的场景中，常常是已知集合大小n，预设误判率$P↓fp$，需要计算位数组大小N、hash函数的个数k。通过数学推导，可得到如下公式：$N=nlnP↓fp /(ln2)↑2$ ，$k=N/n\ ln2$。
  - 快速匹配的特征库就是Bloom Filter需要构建的位数组，集合就是元特征库中的特征字符串。

特征匹配

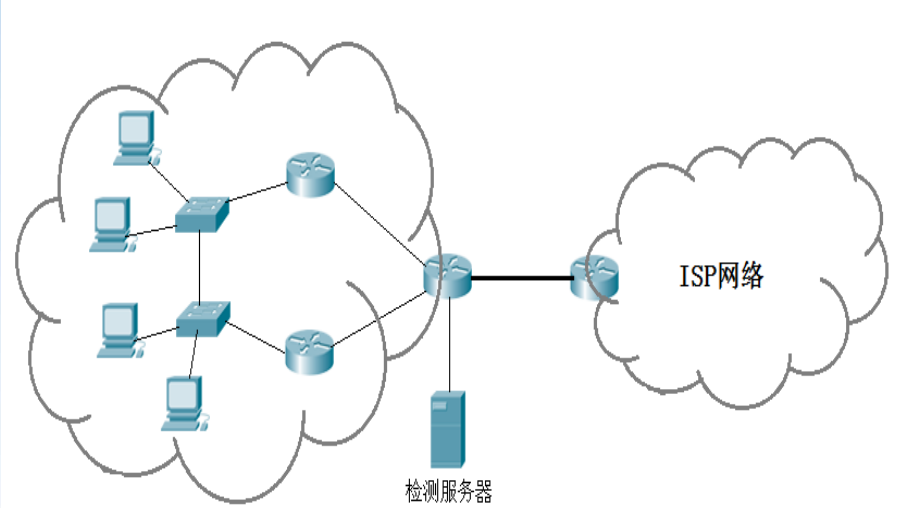
- 特征匹配
  - 快速匹配（Bloom Filter）
  - 精确匹配（正则）
  - 两次匹配策略充分考虑了检测的效率和准确率
- 匹配算法流程
  - 如右图所示



匹配算法流程图

C&C通信检测实验环境

- 实验拓扑
  - 利用若干交换机、PC机和服务器搭建的一个小型局域网环境
  - APT恶意软件安装在局域网内的PC机上
  - 检测系统运行在检测服务器上
- APT恶意样本来源
  - 安全公司APT攻击报告中样本文件的MD5或哈希值
  - 从VirusShare存储库中按文件的MD5下载恶意样本
  - 目前一共获取了包括IXESHE、Lurid、Mirage、Taidoor在内近10种APT活动累计数十个APT恶意软件样本



实验拓扑图

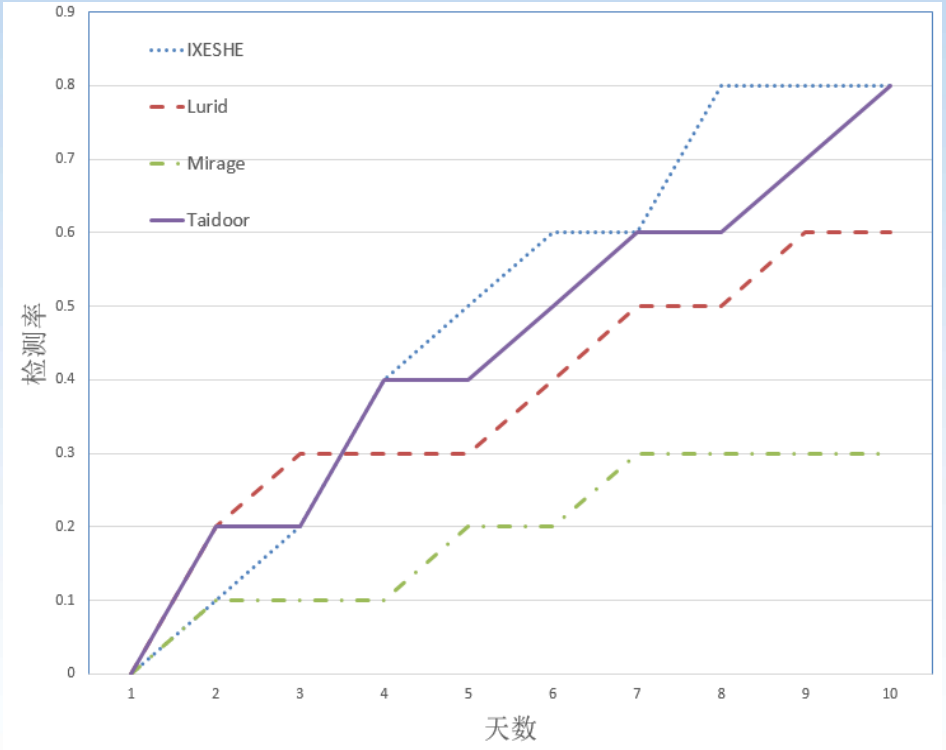
检测结果

- 检测结果
  - 下表中部分APT样本被成功检测，并发掘出了与之通信的C&C服务器
  - 部分样本未被检测出，原因如下：
    - 恶意样本未成功运行
    - 恶意样本和C&C服务器位成功连接
    - 恶意样本的通信特征在目前的APT活动分析中没有出现

名称	检测结果(Y/N)	C&C 服务器IP
IXESHE	Y	68.16.99.165
Lurid	Y	94.245.121.251
Mirage	Y	123.120.101.129
Snake	N	N/A
Sykipot	N	N/A
Taidoor	Y	61.7.158.11
Winnti	N	N/A
Careto	N	N/A

检测结果

- 检测结果
  - 如右图，随着时间的推移，检测率显著升高
  - 原因可能在于越来越多的恶意样本和C&C服务器建立了连接并逐渐产生通信流量



检测率与天数的关系图

APT攻击检测方法

1. 周期性检测
2. C&C通信特征检测
3. 命令控制机制
4. 加密流量识别
5. 数据回溯

基于Web搜索服务查找命令控制节点地址的机制

Web搜索结果页面示例

构造搜索关键字

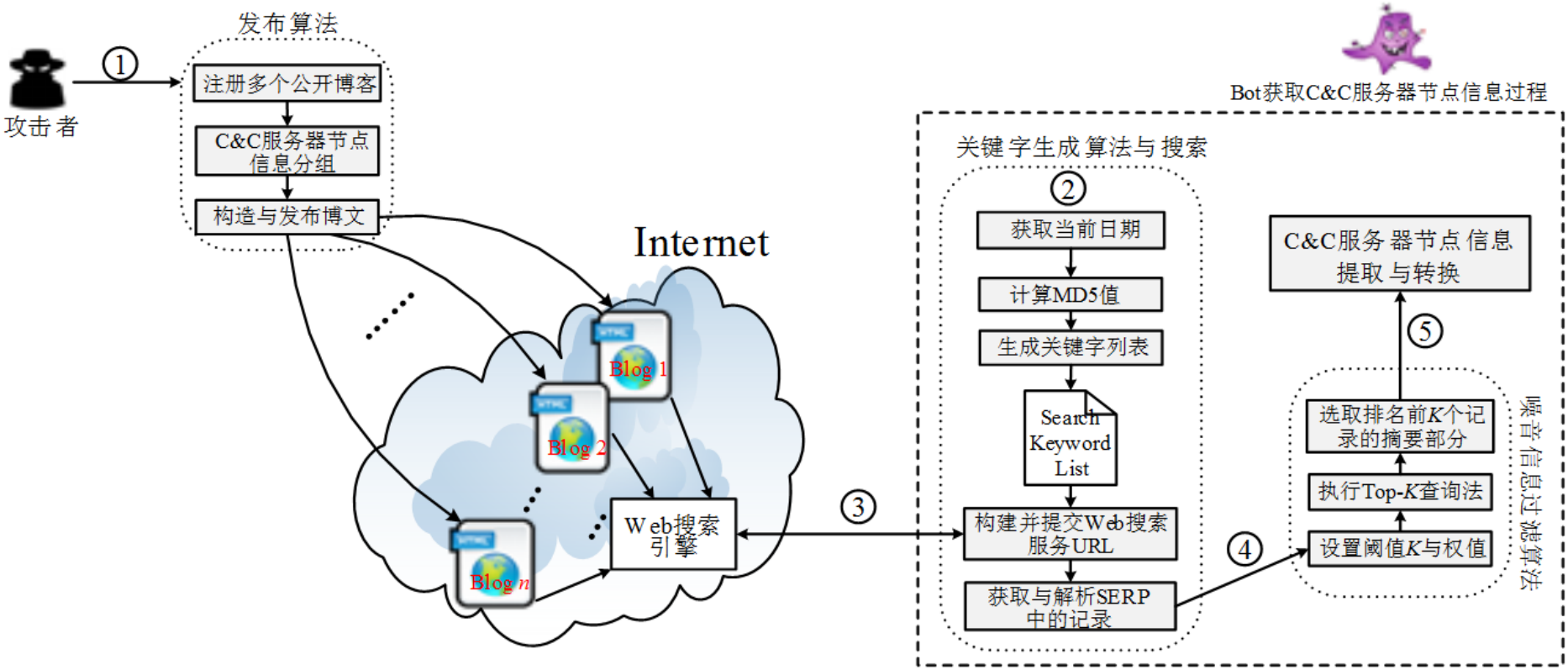
获取Web搜索结果页面

过滤页面记录

提取摘要内容的C&C服务器节点信息

基本思路

基于Web搜索服务查找命令控制节点地址的机制



基于Web搜索服务的C&C服务器节点信息查找机制 CAWSS

基于Web搜索服务查找命令控制节点地址的机制

关键字生成算法与搜索

关键字生成：以日期的MD5值作为搜索关键字，且每天产生关键字的数量设定为12个，如算法1所示。

搜索过程：针对不同搜索引擎，结合关键字构造检索URL（Uniform Resource Locator）字符串并提交给相应的搜索引擎。

算法 1　关键字生成过程

```
1   Function KeywordProduction( )
2   {
3       StringKlist[13];   //关键字列表
4       StringYear, Day, DataStr;
5       Year←get theyear of current date of victim system;
6       Day←get theyear of current date of victim system;
7       for ( inti=1 ; i<13; i++)
8           DateStr←Year+"-"+itoa(i)+"-"+Day; //格式为 YYYY-M-D
9           Klist[i]  ←MD5(DataStr);
10      end for
11  }
```

如百度搜索引擎的URL如下：

http://www.baidu.com/s?· word="+Keyword+"&rn=20　//Keyword为日期的MD5值

基于Web搜索服务查找命令控制节点地址的机制

噪音记录过滤算法

- 利用Top-*K*算法对搜索结果按分值降序排序，得到*K*条个得分最高的记录
- 利用模式匹配算法在此*K*条记录的abstract部分查找与提取IP地址信息字符串

噪音记录过滤算法结果示例

东南大學

基于信息隐藏的命令控制信息传送机制

- C&C信息序列化
  - 将传输的C&C命令转化为对应的二进制位序列
- C&C信息隐藏与发送
  - 将C&C命令对应的二进制位序列隐藏于载体Web页面的HTML代码中，并传送给恶意软件
- C&C信息接收与提取
  - 恶意软件接收隐藏C&C命令的载体Web页面，并从其HTML代码恢复出C&C命令

基于信息隐藏的命令控制信息传送机制

C&C信息序列化

- 设M表示一条C&C命令，$M=v_1v_2\ldots\ldots v_n$，v_i为右表中value
- BS为命令M所对应的二进制序列，且$BS=u_1u_2\ldots\ldots u_n$
- M所对应二进制序列由如下公式得到

$$key_i = K(v_i)$$

$$u_i = BIN(key_i)$$

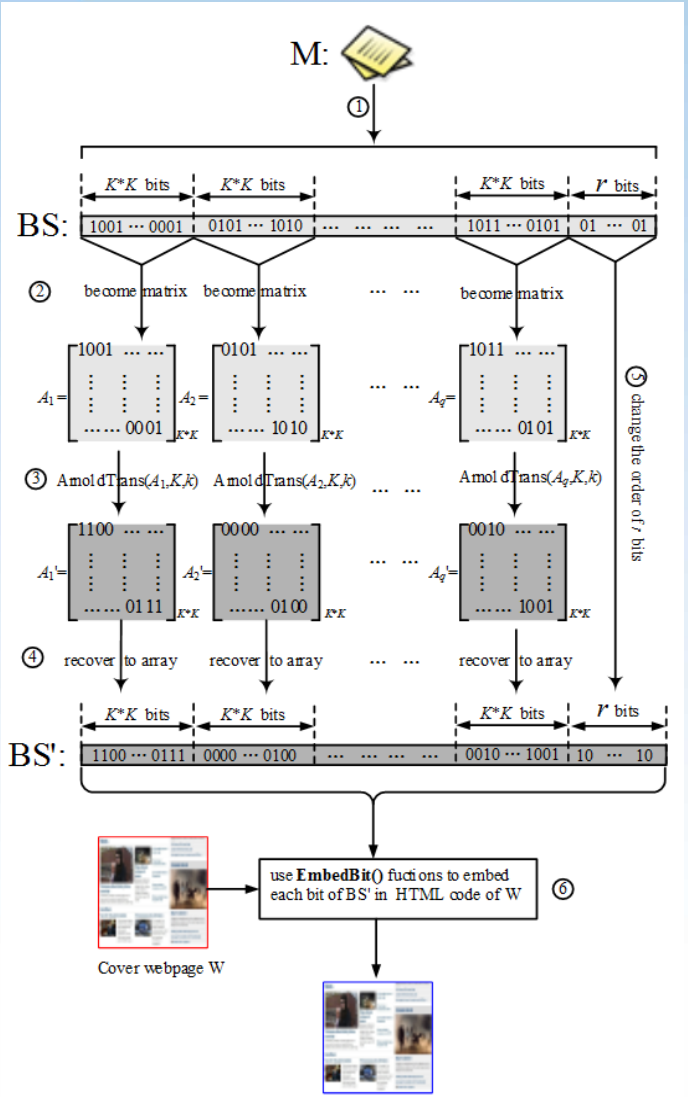
C&C字符编码表

key	value	key	value	key	value	key	value
1.	‘a’	16.	‘n’	31.	‘2’	46.	_
2.	‘b’	17.	‘o’	32.	‘3’	47.	|
3.	‘c’	18.	‘p’	33.	‘4’	48.	/
4.	‘d’	19.	‘q’	34.	‘5’	49.	*
5.	‘e’	20.	‘r’	35.	‘6’	50.	\
6.	‘f’	21.	‘s’	36.	‘7’	51.	!
7.	‘g’	22.	‘t’	37.	‘8’	52.	%
8.	‘h’	23.	‘u’	38.	‘9’	53.	~
9.	‘i’	24.	‘v’	39.	空格	54.	?
10.	‘j’	25.	‘w’	40.	[	55.	@
11.	‘k’	26.	‘x’	41.	]	56.	$
12.	‘l’	27.	‘y’	42.	(	57.	^
13.	‘m’	28.	‘z’	43.	)	58.	"
14.	‘n’	29.	‘0’	44.	:	59.	'
15.	‘o’	30.	‘1’	45.	-	60.	.

基于信息隐藏的命令控制信息传送机制

C&C信息隐藏与发送

- 使用EDA隐藏算法将C&C命令*M*的二进制序列BS隐藏在WEB页面的HTML代码中
- 将页面放置在攻击者自己建设的Web服务器上
- 恶意软件通过正常访问Web服务，获取这些页面，以实现C&C命令*M*传送到恶意软件



EDA隐藏算法主要过程

基于信息隐藏的命令控制信息传送机制

C&C信息接收与提取

恶意软件接收到含有C&C命令信息的载体Web页面W后，需要从W的HTML代码中提取出所隐藏的C&C命令信息*M*，步骤如下：

① 将 BS'分成具有大小 K*K bits 的 q 个数组和一个包含 r（r <K * K）bits 的数组；然后将 q 个数组直接变换为 K*K 方阵。

② 利用密钥 T, k 对所有 q 个方阵应用函数 ArnoldTrans()，然后将所得到的 q 个方阵直接转换为 q 个数组。

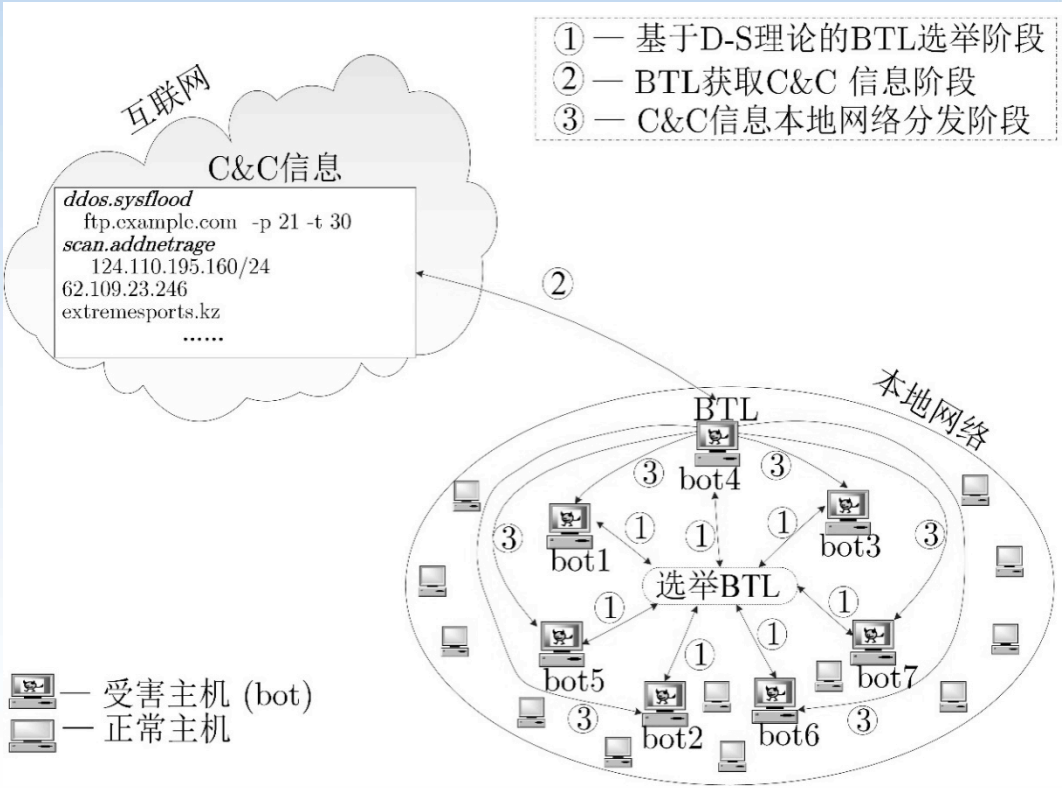
③ 使用函数 RestoreRestBitsOrder()恢复最后一个数组中被 EDA 算法的 ChangeRestBitsOrder()函数修改过顺序的 r bits。

面向本地网络的命令控制信息分发机制

该机制可分为三个部分：

- 基于D-S证据理论的BTL选举
- BTL获取C&C信息
- C&C信息本地网络分发

BLT为本地网络同类Bot的临时代表。



面向本地网络的命令控制信息分发机制

面向本地网络的命令控制信息分发机制

基于D-S证据理论的BTL选举

评价性能指标：**开机时间比** 和 **CPU利用率**

开机时间比：在BTL选举中，开机时间越长，表示该受害主机能正常持续运行的时间越长，处于活跃状态，越有利于该主机成为BTL。此处用开机时间比$g(j)$来衡量第j个恶意软件的开机时间，定义为：

$$g(j)=\frac{t(j)}{\Omega}$$

$t(j)$表示恶意软件j的累积开机时间，Ω为观察的时间窗口长度。

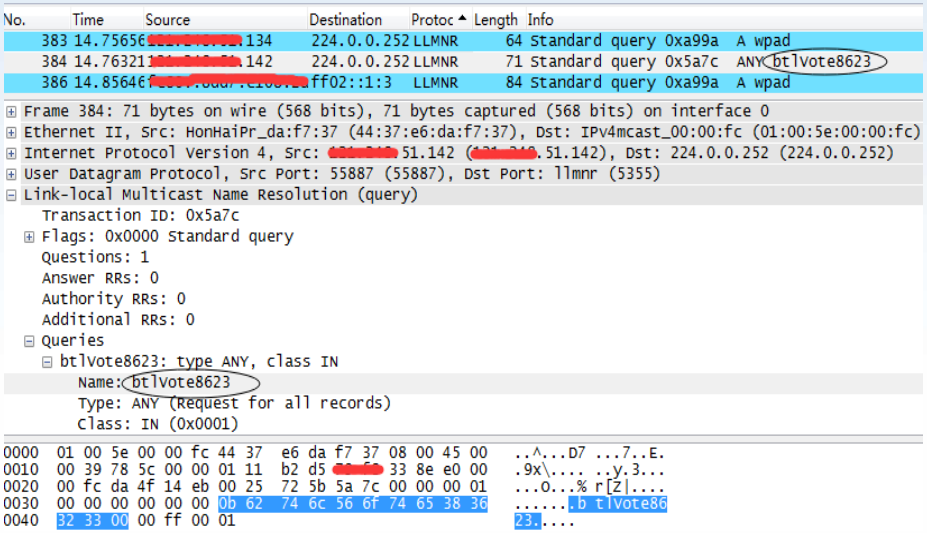
CPU利用率：所在受害主机CPU利用率小，表示能被该恶意软件所利用的主要硬件资源越多，代码执行时给受害主机所造成负担较轻，不易引起受害主机用户的觉察。

CPU利用率定义为在观察时间长度为Ω内的CPU利用率平均值，记为$h(j)$。

面向本地网络的命令控制信息分发机制

BTL选举过程步骤

- 恶意软件利用LLMNR协议的Query包在本地网内通告自己的开机时间比$g(j)$与CPU使用率$h(j)$
- 其它恶意软件根据收到的两个指标值，计算出BTL集合θ_1
- 采用集合θ_1中IP地址最大者对应的恶意软件作为BTL



含有两个评价指标的LLMNR Query包示例

面向本地网络的命令控制信息分发机制

- **BTL获取C&C信息**

BTL采用CAWSS机制来查找命令控制节点地址信息，然后再使用基于HTML代码信息隐藏方法来获取命令控制信息。

- **基于LLMNR协议的C&C信息分发**

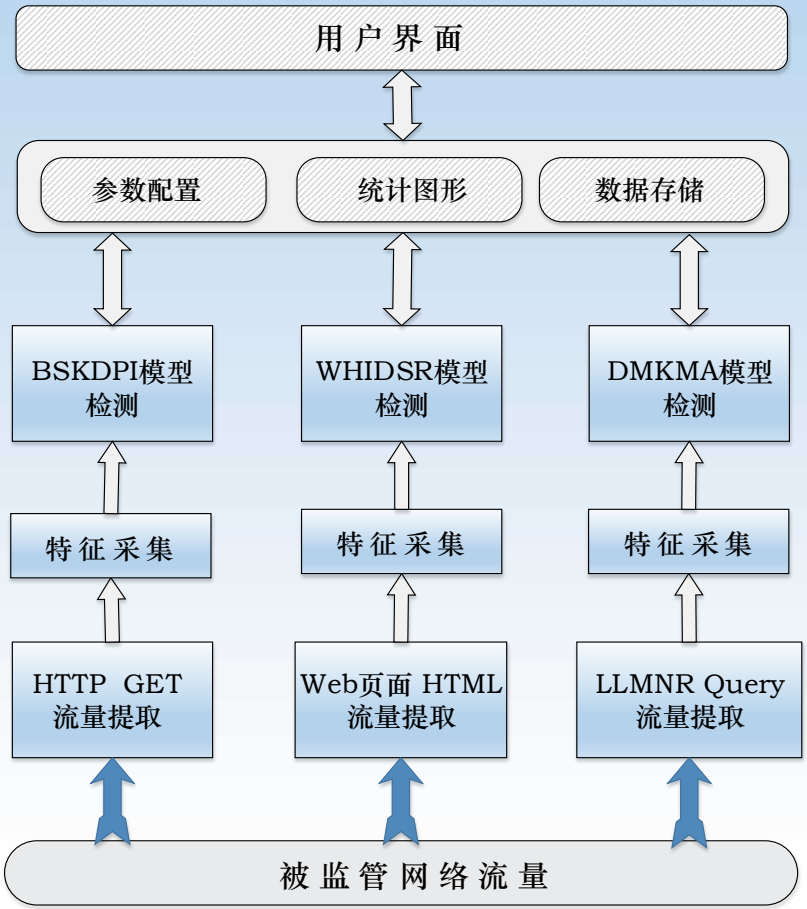
基本思想： 将C&C命令分成多个子部分，然后每一部分当作LLMNR Query报文的Name字段值进行发送。

采用格式：ccinfoXXSSF +命令控制信息内容

位名称	含义
XX	控制命令信息编号
SS	控制命令信息的分段编号
F	控制命令信息分段结束标志，F=0 表示未结束，F=1 表示结束。

检测原型系统设计与实现

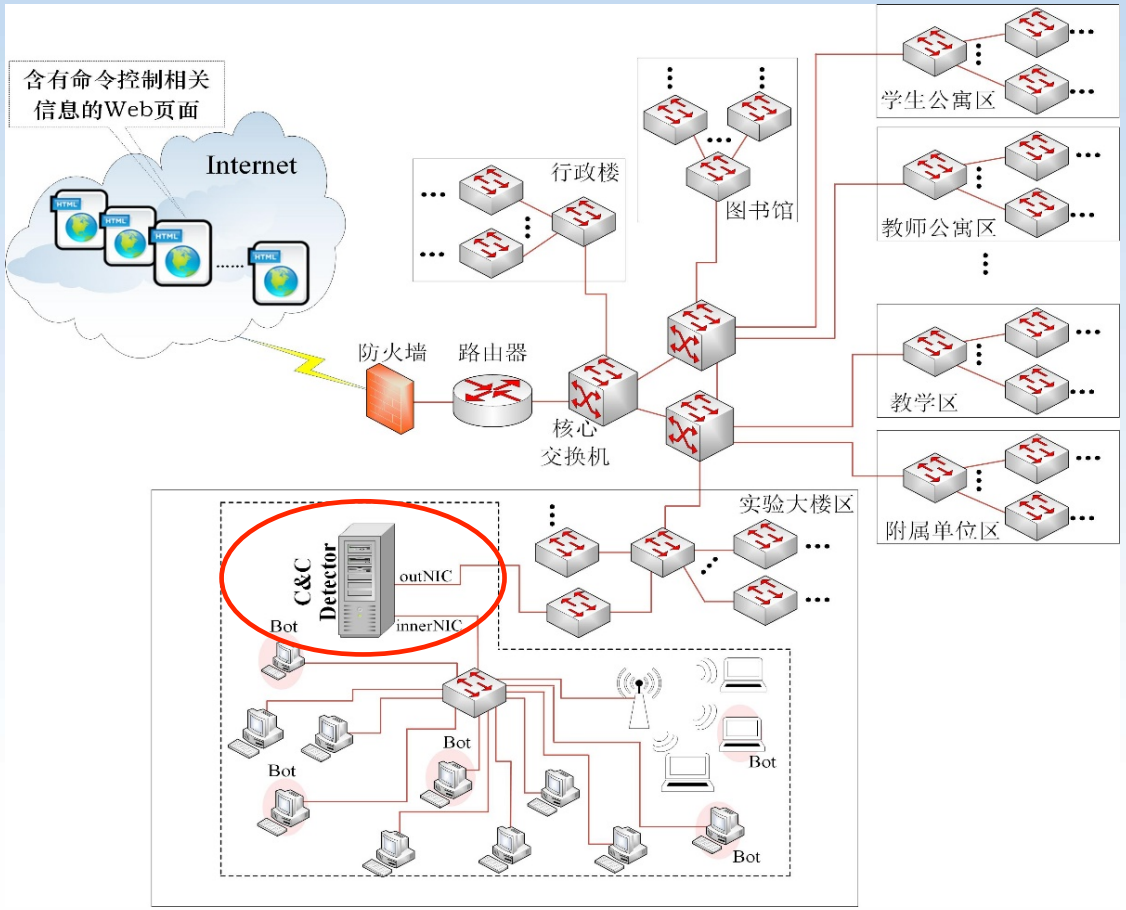
针对所提出的新型命令控制机制，设计并实现了检测原型，该系统采用集中式单点结构，系统总体框架如图所示。



检测原型系统框架

检测原型系统设计与实现

- 系统部署与测试



检测原型系统部署环境示意图

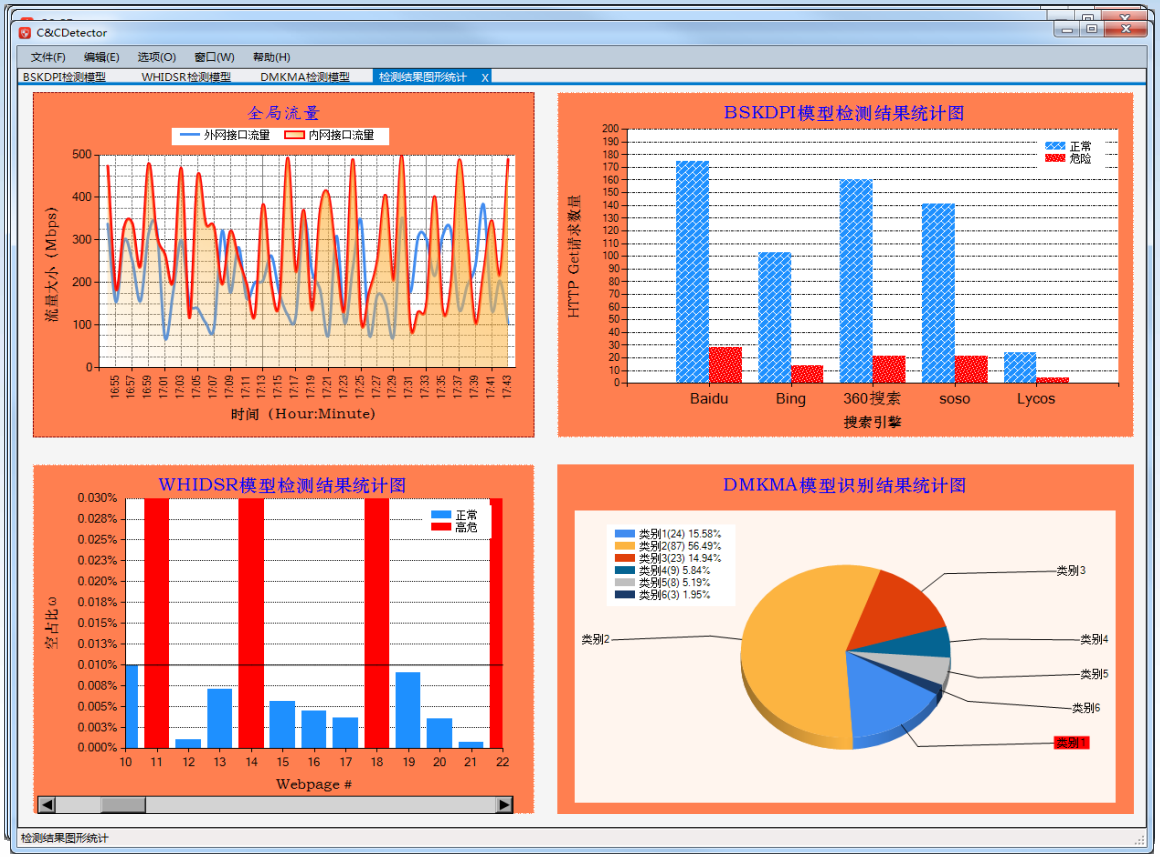
检测原型系统设计与实现

面向基于Web搜索服务查找命令控制节点地址过程的检测功能测试界面

面向基于信息隐藏的命令控制信息传送过程的检测功能测试界面

面向本地网络命令控制信息分发过程的检测功能测试界面

检测结果图形统计测试界面



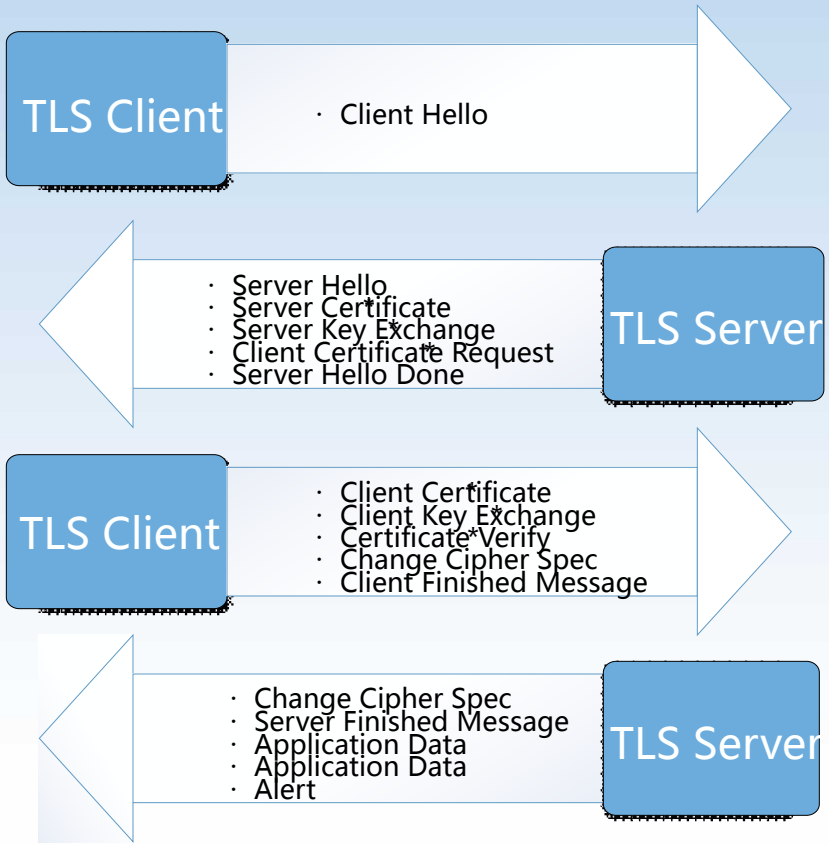
APT攻击检测方法

1. 周期性检测
2. C&C通信特征检测
3. 命令控制机制
4. **加密流量识别**
5. 数据回溯

加密流量识别存在的问题

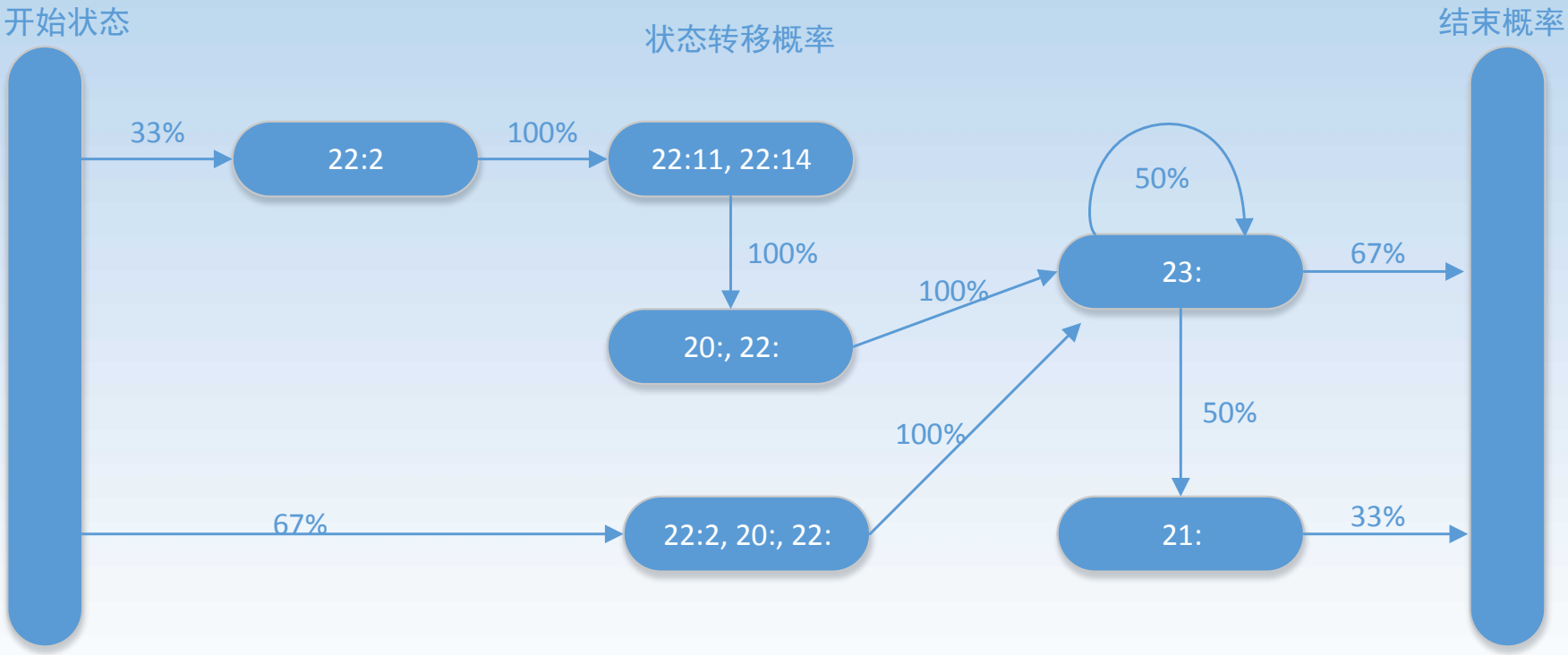
- 存在的问题
  - 基于一阶Markov链的识别方法，只考虑两个状态的转移
    - 状态种类少，SSL/TLS应用间区别性不足
  - 基于二阶Markov链，引入Certificate包长优化状态Markov状态转移概率的识别方法
    - 仍有少量应用的Certificate包长会聚类到相同的类中，造成误判
- 解决方法
  - 引入握手过程的包长与消息类型构成二位特征增加Markov状态的数量
  - 引入二阶Markov模型，状态转移考虑前两个状态，增加Markov状态转移多样性
  - 引入数据传输过程的包大小建立HMM模型，根据相邻包大小相关性改进HMM发射概率，增加应用间区分性

基于握手过程的Markov 链模型



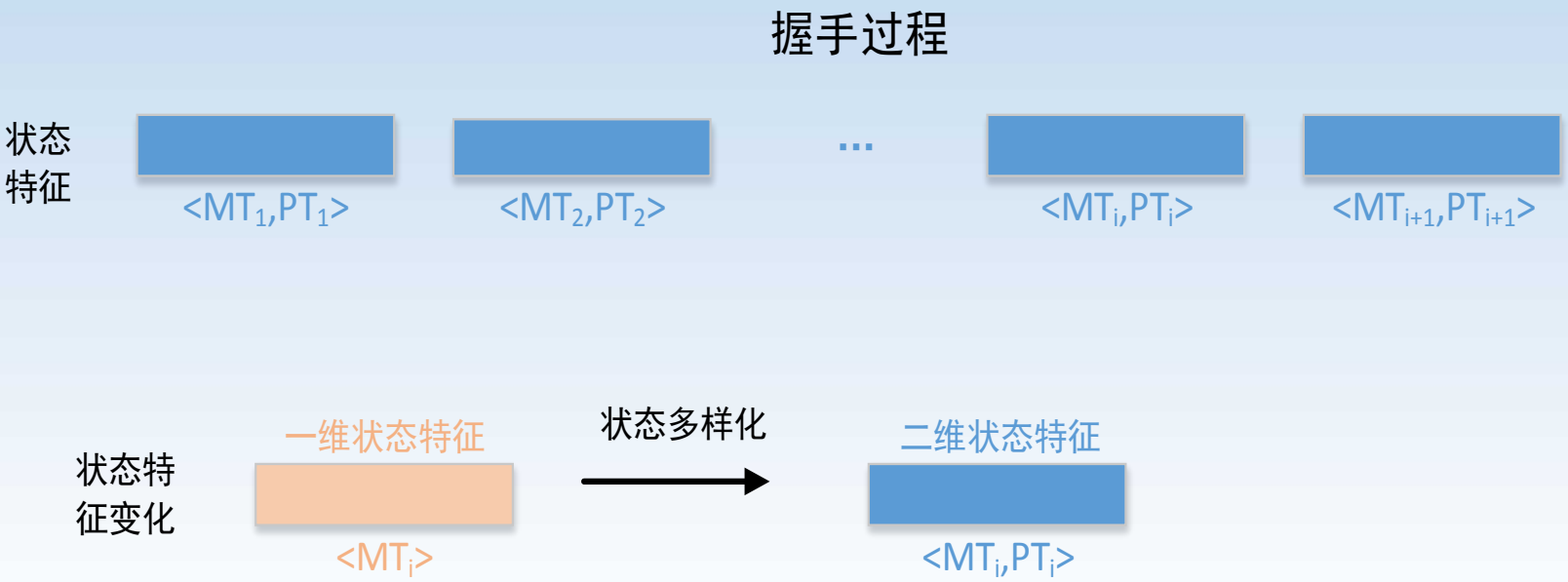
明文	SSL/TLS 消息类型
20	Change Cipher Spec
21	Alert
22:02	Server Hello
22:11	Certificate
22:12	Server Key Exchange
22:13	Certificate Request
22:14	Server Hello Done
22:17	Encrypted Handshake Message
22:18	New Session Ticket
22:20	Finished
23	Application Data

Markov链模型实现举例

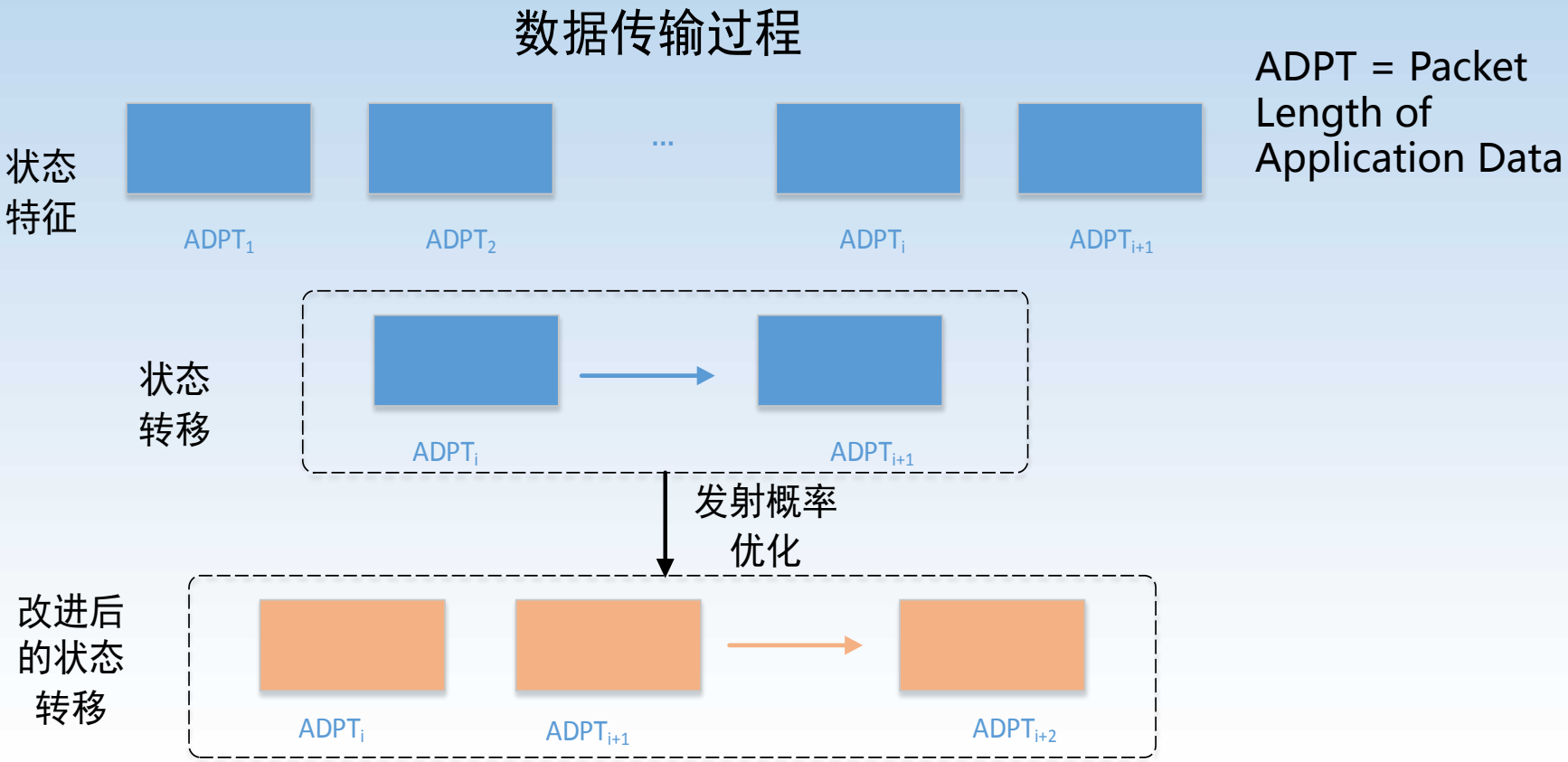


流量分类模型

Markov链状态特征多样化



基于数据传输过程的HMM模型改进



ADPT = Packet Length of Application Data

HMM 发射概率优化具体描述

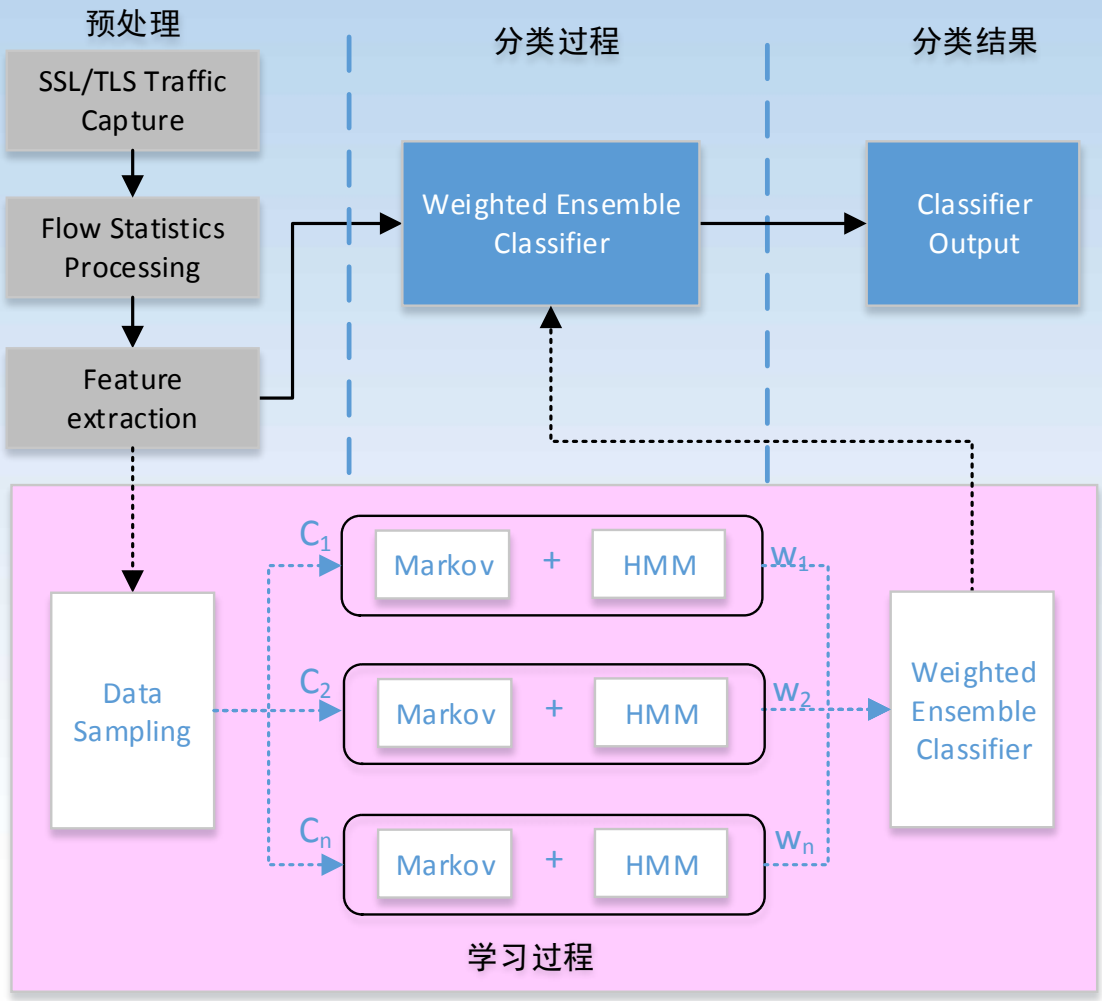
发射概率计算 $B=\{b\downarrow im\}$

改进后的发射概率 $B=\{b\downarrow im(o\downarrow t-1)\}$

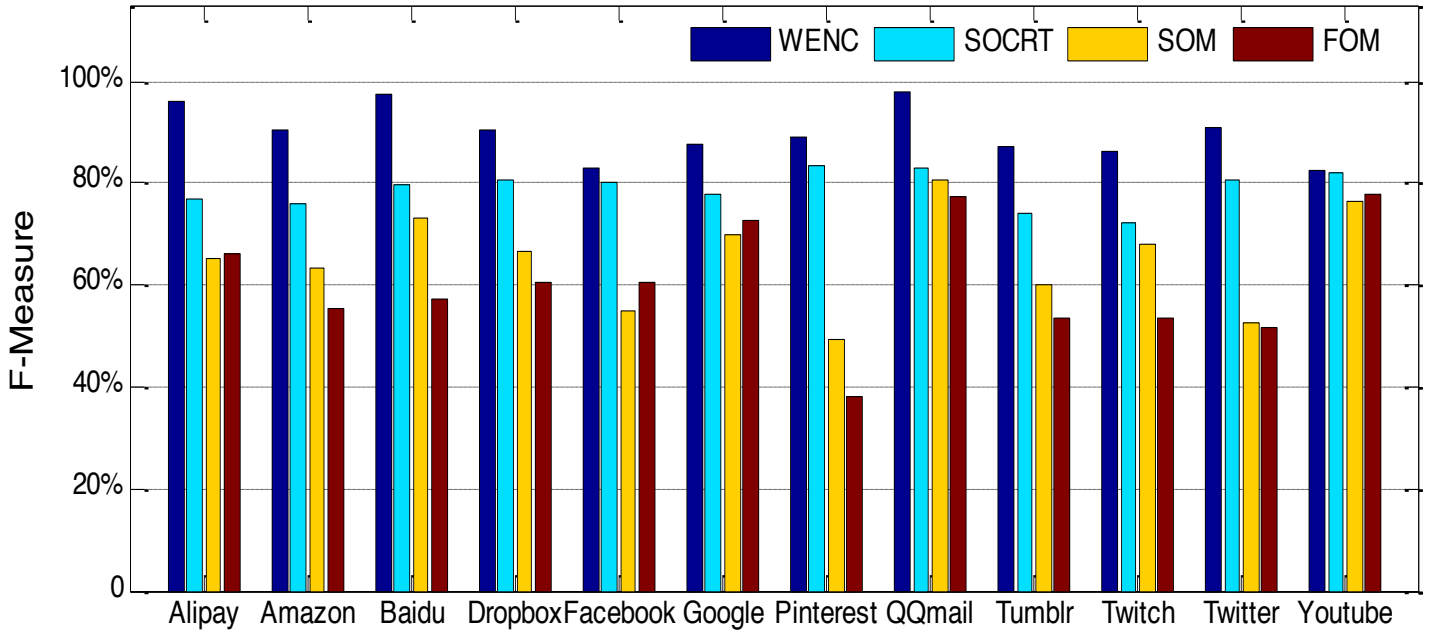
具体来说 $b\downarrow im(o\downarrow t-1)=P(v\downarrow m/q\downarrow i,o\downarrow t-1)$

改进后的发射概率：在t 时刻时，给定的上一个观察特征状态为o_{t-1} 和隐藏状态为q_i 情况下，观察特征取值为v_m 的概率。

加权集成分类器总体架构



性能分析


WENC
SOCRT
SOM
FOM
F-Measure
100%
80%
60%
40%
20%
0
Alipay
Amazon
Baidu
Dropbox
Facebook
Google
Pinterest
QQmail
Tumblr
Twitch
Twitter
Youtube

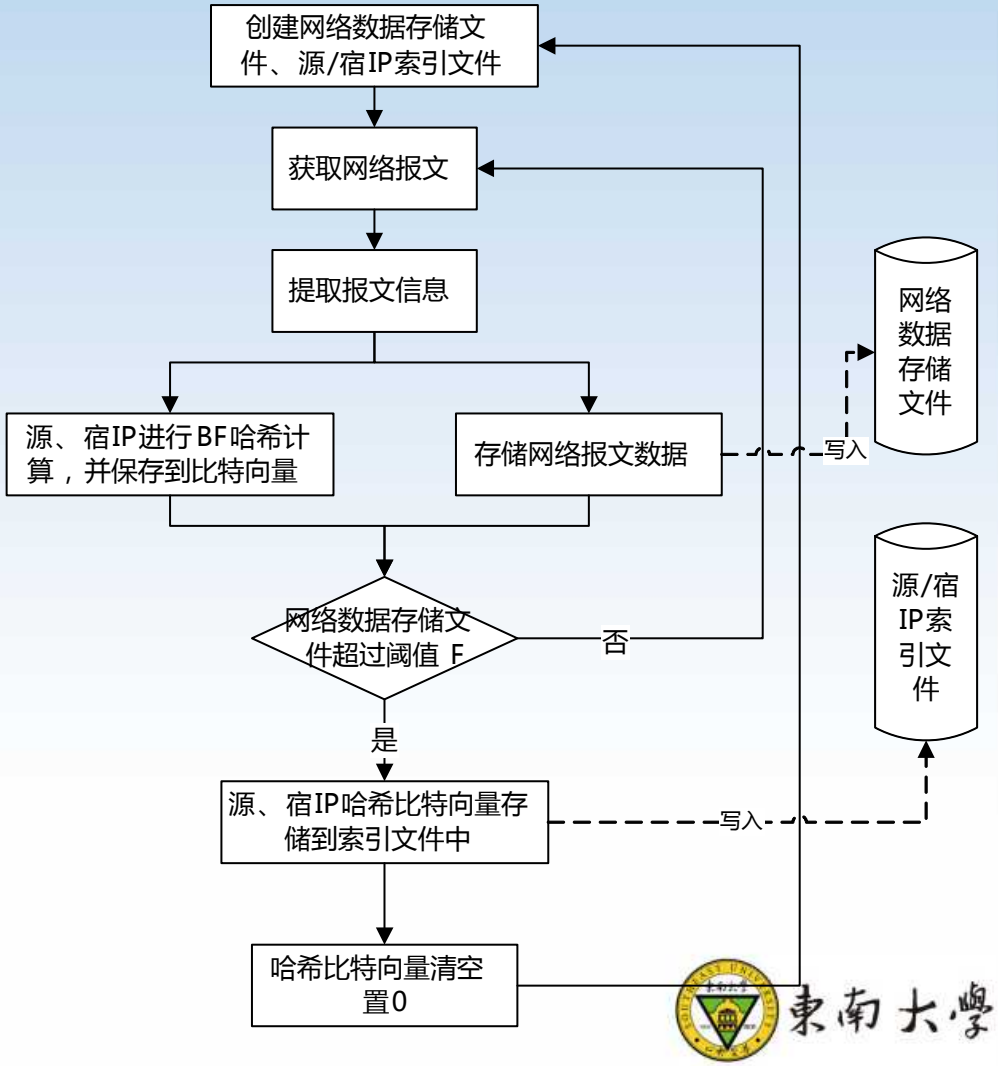
APT攻击检测方法

1. 周期性检测
2. C&C通信特征检测
3. 命令控制机制
4. 加密流量识别
5. 数据回溯

数据回溯算法

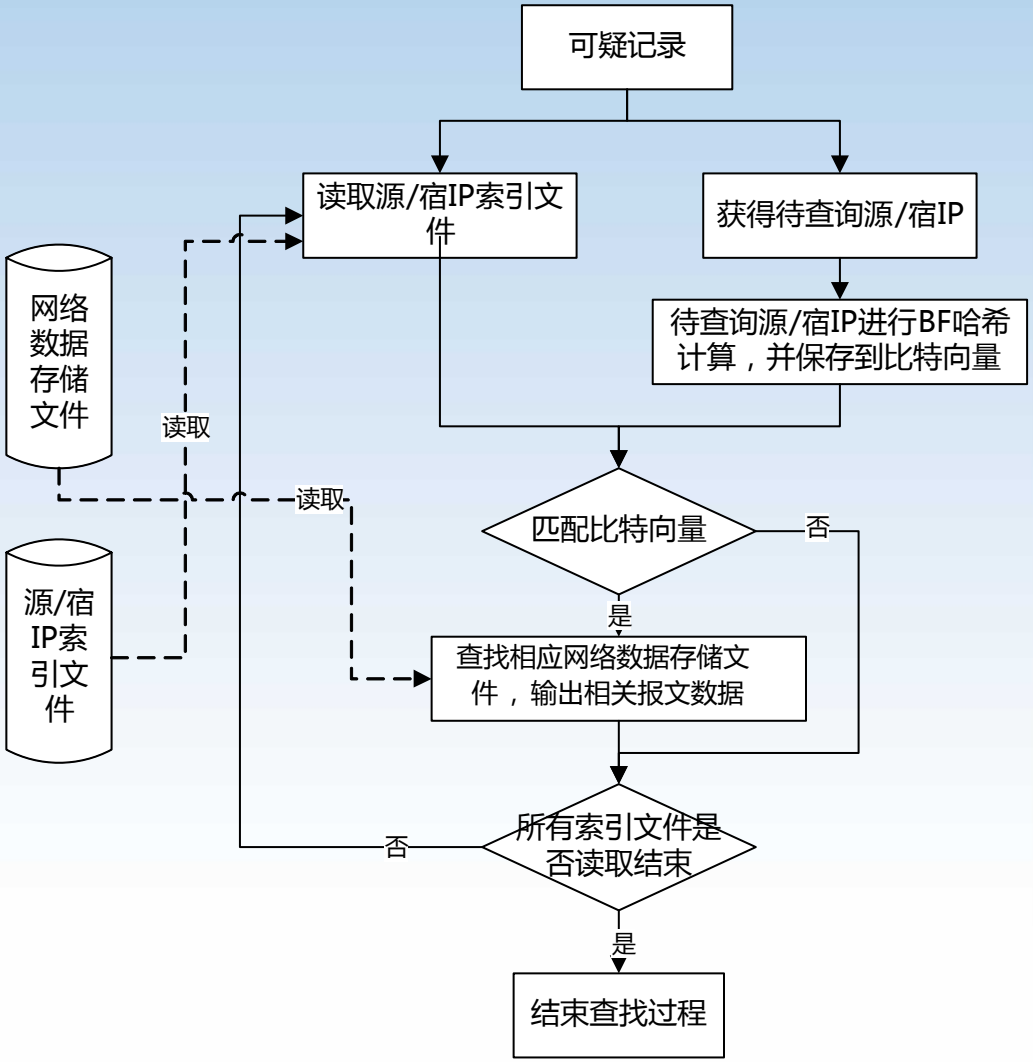
对数据流结构的海量网络数据进行存储和回溯查找

- **实时存储过程**
- 获取网络报文数据，截取特定长度的报文数据进行实时存储
- 提取报文数据中的源宿IP信息，使用Bloom Filter算法对源宿IP分别进行哈希计算，并在比特向量中置相应位为1
- 每个网络数据存储文件中的所有网络报文对应于一个索引结构，存储为索引文件



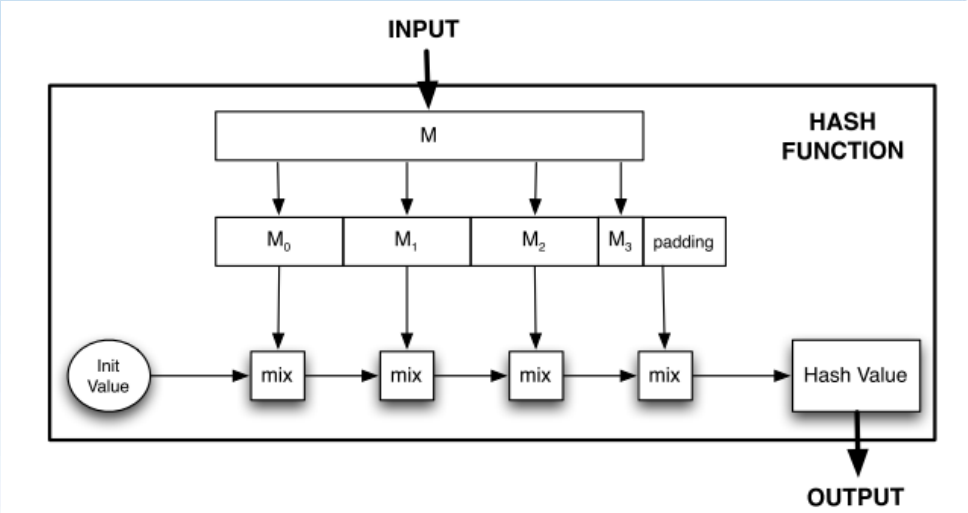
数据回溯过程

- **实时查找过程**
- 当APT攻击检测到可疑数据时，提取其中的源宿IP，对源宿IP分别进行哈希计算
- 查找相应的索引文件，如果索引文件中相应位上都为1，则查找成功
- 获取详细的报文数据信息，用于可疑信息的详细分析



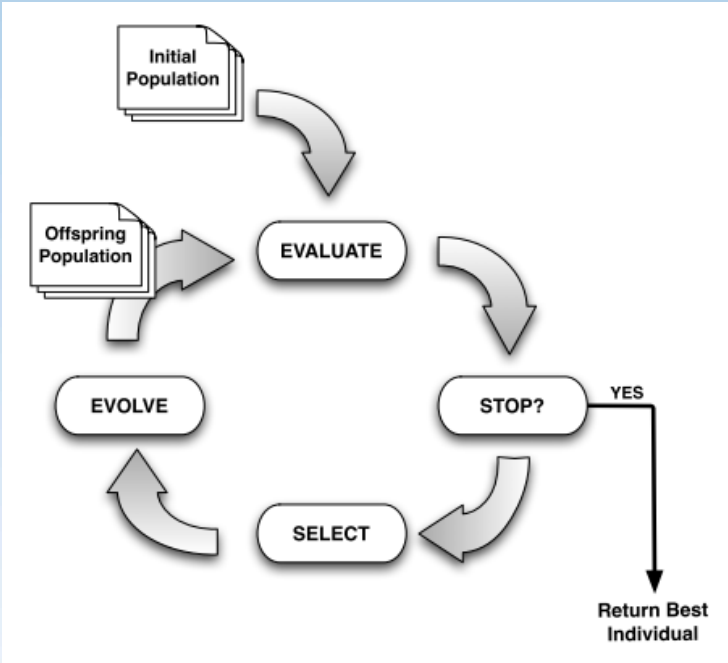
哈希算法的设计

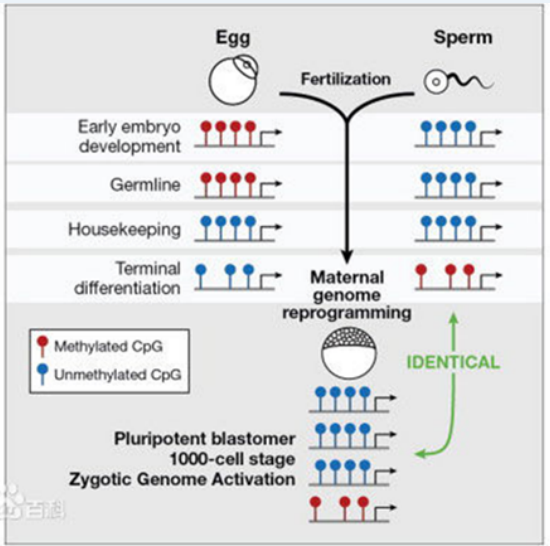
- 简化的Merkle-Damgard结构可用来构造一个表现较优的哈希算法，大多现有的字符串哈希算法均基于此结构进行设计
- 其将输入分割为多个等长的block，并对当前block和中间状态进行混合操作(mixing)，从而得到最终的哈希输出



基于遗传编程的哈希算法

- 采用遗传编程思想，在训练集搜索生成针对IP流量特征的哈希算法
- 充分学习流量特征，使生成的哈希算法随机性和运行速度基本达到最优
- **遗传编程**
  - 首先选取初始种群，对种群中的个体进行评估
  - 选取较优的个体，对其进行进化操作，从而形成下一代种群
  - 进化过程重复至找到最优的个体或达到最大的进化轮数





GP-Hash参数设置

GP-Hash	
Max Generations	35
Pop. Size	150
Max Tree Height	17
Function Set	{right rotation, xor}
Terminal Set	{a0: 当前需处理的block；hval: 中间状态}
Fitness	哈希熵值
Crossover(杂交)	Rate=0.7; Selection=Tournament; Tournament Size=4
Point Mutation(变异)	Rate=0.2; Selection=Tournament; Tournament Size=4
Reproduction(复制)	Selection=Tournament; Tournament Size=4
Initialization	Half and half, init depth2-4

使用异或、循环位移操作作为哈希算法的基本运算单元。

可选取随机性测度、冲突率、雪崩测度等作为种群中哈希个体的评估函数，从而得到相应测度上表现较优的哈希算法。

研究背景

- APT的智能检测
  - 从海量的网络流量中进行数据挖掘
  - 恶意事件的关联分析和规则挖掘
  - 根据已发现的特征或知识对未知的APT攻击进行判定，对APT攻击进行预测和泛化
  - 对APT检测的动态性、大规模、复杂性进行自动管理和优化
- 人工智能技术
  - 机器学习、仿生智能计算、模糊神经网络等

结论

- APT高级持续性威胁
  - 高级性智能技术、持续性隐藏技术、APT的大数据特征
  - 传统检测体系结构无法适应APT大数据的特点
  - 对所需保护对象的全流量采集和长期数据存储
  - 采用机器学习、仿生智能计算、模糊神经网络等智能技术
- APT智能检测架构
  - 数据预处理、特征提取、检测应用
- ATP检测方法
  - 周期性检测、C&C通信特征检测、命令控制机制、加密流量识别、数据回溯

感谢！ 敬请批评指正！