Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

网络未知协议逆向技术

硕士研究生 徐菊彬

2024年12月22日

问题回溯

- 相关内容
  - 灵通测

内容提要

- 预期收获
- 题目内涵解析
- 研究背景与意义
- 研究历史与现状
- 知识基础
- 算法原理
  - BinaryInferno
  - MDIplier
- 特点总结与工作展望
- 参考文献

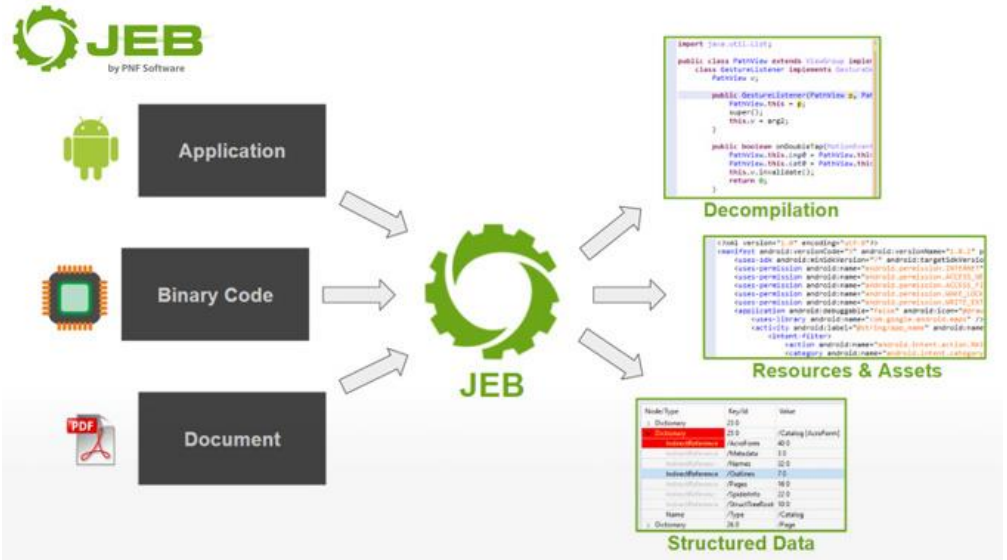
预期收获

- 预期收获
  - 1. 了解网络未知协议逆向的基本概念和研究方向
  - 2. 理解两种协议逆向方法的基本原理
  - 3. 了解现有方法的缺陷以及未来发展方向

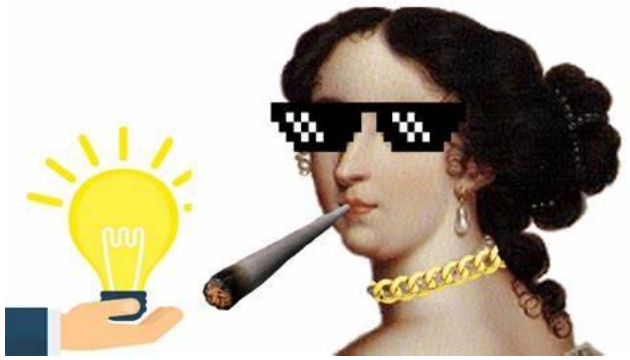
什么是逆向工程?

- 逆向工程
  - 识别目标设备、系统、软件或过程的组成部分
  - 了解组件之间的相互作用
  - 对目标进行推演

战争领域B-29→Tu-4



反编译

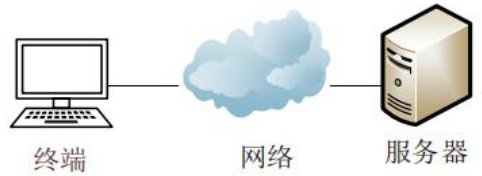


反汇编

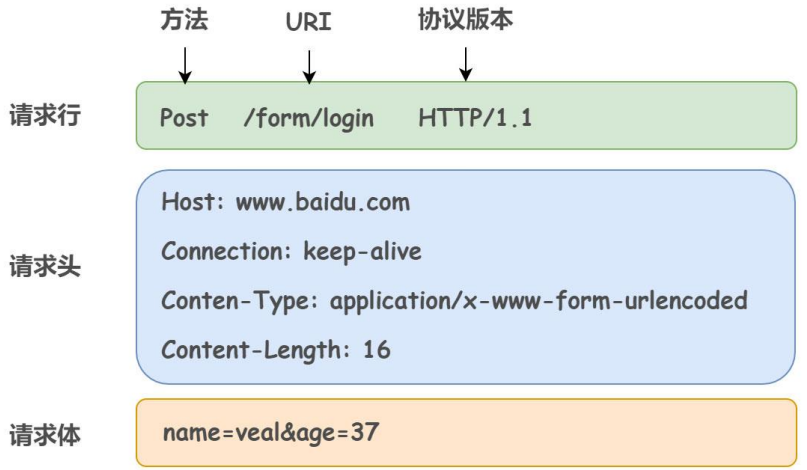
网络协议逆向

题目内涵解析 网络未知协议逆向

- 网络协议
  - 计算机网络中进行数据交换而建立的规则、标准或约定的集合
  - 保证节点之间的相互通信



- 网络协议分类
  - 是否公开协议规范
    - 已知协议：标准化、公开且广泛使用的协议
      - HTTP、TCP、FTP
    - 未知协议：不公开标识字段、报文格式或交互通信模式等协议规范
      - 用于工业控制系统、物联网设备间通信等私有协议



  - 可读性
    - 文本协议：主要由编码字符组成，如ASCII、Unicode等，人类可读
    - 二进制协议：使用任意字节值对任何类型的数据进行编码，机器可读

题目内涵解析 网络未知协议逆向

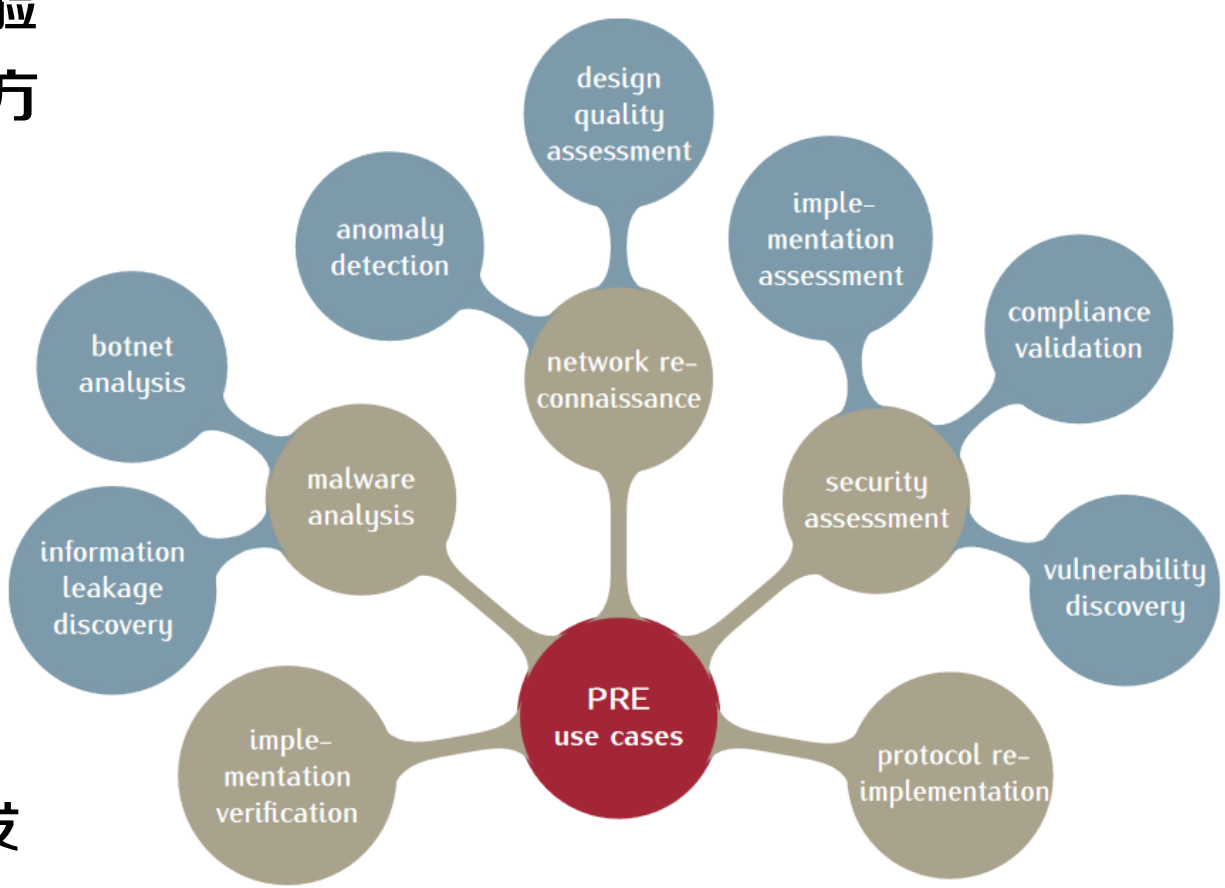
- 协议逆向工程（Protocol reverse engineering，PRE）
  - 旨在通过分析通信流量或通信实体来推断未知的协议规范，如标识字段、报文格式、交互通信模式等

分析方法	条件	适用场景	优点	缺点
动态实体分析	程序+环境均可运行	需要深入了解协议逻辑及动态生成数据	可观察运行时行为，获取动态数据	实现复杂度高
静态实体分析	程序+反汇编工具	目标程序复杂，需要分析实现细节	不受运行环境限制，可分析完整逻辑	较难处理加密代码
动态流量分析	实时捕获流量数据	快速掌握通信模式和协议结构	分析直观	较难分析加密内容
静态流量分析	流量数据pcap包	无法运行程序，仅有流量数据	分析简单，应用范围广	无法处理加密内容

- 案例思考
  - 评估携带植入式心律转复除颤器的患者的安全性影响
  - 专有设备的无线协议逆向，分析方法选择?

研究背景与意义

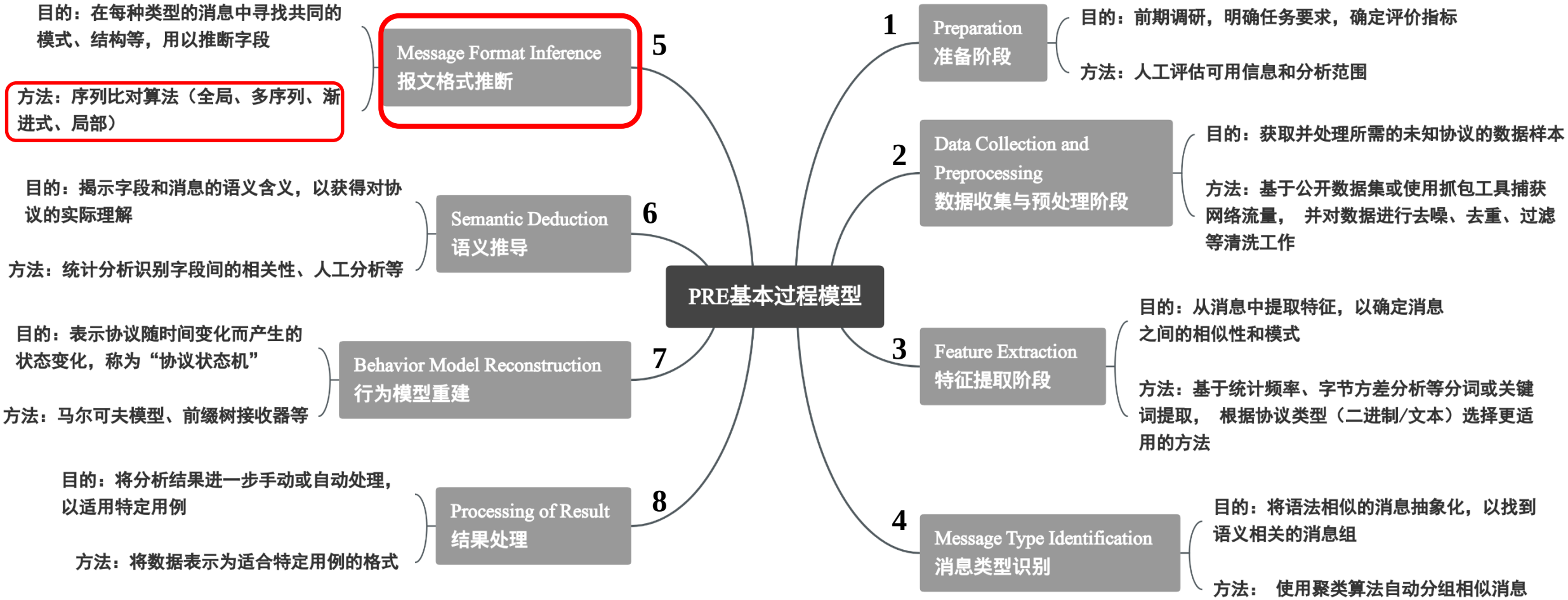
- 研究背景
  - 在未知协议重建、智能模糊测试、重建验证、网络侦察、恶意分析、安全评估等方面应用众多
  - 实现自动化协议逆向仍面临挑战
- 研究意义
  - 网络安全防御
    - 逆向未知协议，可帮助分析恶意流量、识别攻击行为，保障网络安全
  - 工业与信息系统安全
    - 理解工业控制协议等私有协议，帮助发现潜在漏洞，提升系统安全性



灵通测：助力智能化测试套件生成

协议逆向过程

- 网络协议逆向过程可总结为如下8个模块



研究历史与现状

报文格式推断

Kleber等人总结过往基于SAT的协议逆向方法，提出一个显示、结构化的协议逆向工程过程模型。并根据该过程模型，分阶段调研总结相关算法与逆向工具。

Lin等人提出的ReFSM模型是一种仅从网络数据包推断协议EFSM（扩展有限状态机）的新方法。与传统PRE只考虑控制流信息不同，构造EFSM时，考虑由状态转移上的数据保护和内存所代表的数据流。

Zhao等人提出一种基于深度学习的二进制协议格式提取工具ProsegDL，并设计一种生成训练数据集的方法。利用图像语义分割和孪生网络技术，专注于提取字段的特征并识别固定格式协议的字段边界。

Chandler等人提出了一个用于逆向工程二进制报文格式的全自动工具BinaryInferno 。该工具使用一个检测器集合来推断部分描述的集合，然后自动将部分描述集成到一个语义有意义的描述中，该描述可用于解析未来相同格式的数据包。

消息类型识别

Kleber等人利用二进制协议的内在结构特征，提出了一种准确区分消息类型的方法。结合Hirschberg算法和DBSCAN聚类算法，将离散的字节映射成特征向量进行相似性度量。

Ye等人提出一种新的基于概率网络的协议逆向工程技术Netplier。通过引入随机变量来表示代表消息类型的单个字段的可能性，从而对问题的固有不确定性进行建模。

Tang等人提出一种新的基于关系推理的网络协议逆向工程方法，数据包的n元词串具有上下文关系，通过考虑关键词间的上下文信息，有助于推断协议格式。

Liang等人提出MDIplier工具，利用协议消息的层次结构，在每个消息层进行定制化分析。MDIplier执行迭代推理过程，在每次迭代过程中，它识别用于层分离的消息分隔符，并分别为每层推断格式，优化了可用字段信息的使用。

知识基础

- 序列比对算法
  - 比较和对齐网络流量信息间的相似性
  - 应用于基因功能预测、进化关系分析等任务
- 算法分类
  - 全局比对（ Needleman–Wunsch algorithm ）
    - 对齐整个序列，适用于长度相近的序列
  - 局部比对（ Smith-Waterman algorithm ）
    - 寻找局部最优匹配区域，适用于长度差异较大的序列
  - 多序列比对（Multiple sequence alignment）
    - 同时比对多个序列，揭示共同特征
    - 渐进式比对（Clustal algorithm）

		G	E	T		/	i	n	d	e	x	.	h	t	m	l		H	T	T	P	/	1	.	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
E	0	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
T	0	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	0	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
/	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6
H	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	6	7	7	7	7	7	7	7	7
T	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	6	7	8	8	8	8	8	8	8
T	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	6	7	8	9	9	9	9	9	9
P	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	6	7	8	9	A	A	A	A	A
/	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	6	7	8	9	A	B	B	B	B
1	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	6	7	8	9	A	B	C	C	C
.	0	1	2	3	4	5	5	5	5	5	5	6	6	6	6	6	6	7	8	9	A	B	C	D	D
0	0	1	2	3	4	5	5	5	5	5	5	6	6	6	6	6	6	7	8	9	A	B	C	D	E

Table 1: A completed matrix processed by the Needleman Wunsch algorithm. Highlighted is the path that is traced back to the origin that is used to determine the presence of gaps in either sequence.

- 全局比对（Needleman–Wunsch algorithm）
  - 初始化得分矩阵
    - $(1,1)=0$，$(1,j)=j*d$，$(i,1)=i*d$
    - d空位罚分，$d=-1$
    - s匹配得分，$s=1\ or\ 0$
  - 填充得分矩阵
    - 左上方位置$(i-1,j-1)$的得分+s
    - 上方位置$(i-1,j)$的得分+ d，序列1插入空位
    - 左方位置$(i,j-1)$的得分+ d，序列2插入空位
    - 取三种得分的最大值作为当前位置得分
  - 回溯获取最优比对路径
    - 确定两序列的全局最优比对方式
    - 字符匹配、空位插入情况等

Needleman-Wunsch

match = 1　　mismatch = -1　　gap = -1

		G	C	A	T	G	C	G
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

知识基础

- 局部比对（Smith-Waterman algorithm）
  - 初始化得分矩阵
    - $(1,j)=0$，$(i,1)=0$
    - d空位罚分，$d=-1$，s匹配得分，$s=3\ or\ -3$
  - 填充得分矩阵
    - 左上方位置$(i-1,j-1)$的得分+s
    - 上方位置$(i-1,j)$的得分+ d，序列1插入空位
    - 左方位置$(i,j-1)$的得分+ d ，序列2插入空位
    - 取三种得分的最大值作为当前位置得分，若为负则记0
  - 回溯获取最优比对路径
    - 从最大值开始回溯，到0停止
    - 确定两个序列之间的局部最优比对方式

初始化得分矩阵

		T	G	T	T	A	C	G	G
	0	0	0	0	0	0	0	0	0
G	0								
G	0								
T	0								
T	0								
G	0								
A	0								
C	0								
T	0								
A	0								

置换矩阵：$S(a_i,b_j)=\begin{cases}+3, & a_i=b_j\\ -3, & a_i\neq b_j\end{cases}$

空位罚分：$W_k=kW_1$，$W_1=2$

BinaryInferno: A Semantic-Driven Approach to Field Inference for Binary Message Formats

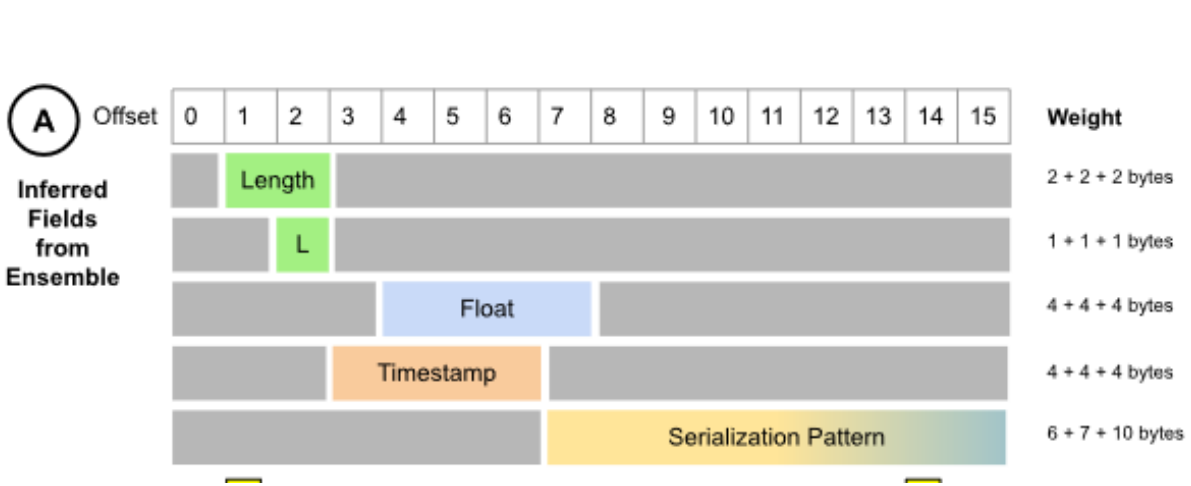
算法原理 TIPO

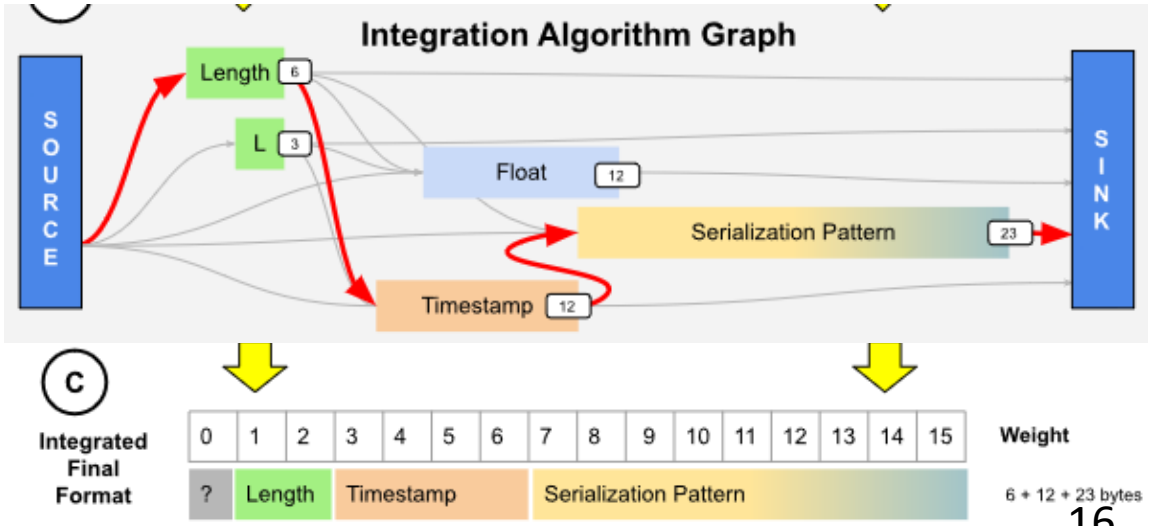
T	目标	二进制协议逆向工具，实现报文格式推断
I	输入	10种二进制协议数据包（bgp、dhcp、dnp3、mavlink、mirai、smb2、ntp48、smb、tutorial等）
P	处理	1.使用3种探测器对输入的消息样本进行分析 a.原子探测器识别基本的语义数据类型 b.字段边界探测器利用香农熵识别相邻字段边界 c.模式探测器推断可变长度字段 2. 结果集成
O	输出	一个语义上有意义的报文格式描述， 包括准确的字段边界和对应的字段的数据类型

P	问题	字节级操作限制、格式覆盖范围有限、样本数据敏感等
C	条件	假设：固定宽度字段和可变长度有效负载在消息中的起始位置是固定的
D	难点	二进制数据的二义性、探测器结果冲突、可变长度字段处理
L	水平	NDSS 2023 CCF A

算法原理 创新说明

- **BinaryInferno**
  - 使用一个由多种专门探测器组成的集成体来解决二进制报文格式的字段推断问题，而不依赖单一复杂的推理方法
  - 处理多种常见的二进制数据类型和模式，包括原子数据类型（如 IEEE 浮点数、时间戳、固定长度字段）、利用香农熵确定字段边界以及通过搜索常见序列化模式发现可变长度序列
  - 使用最大化样本数据量的集成算法来解决不同探测器对消息格式中特定字节的冲突

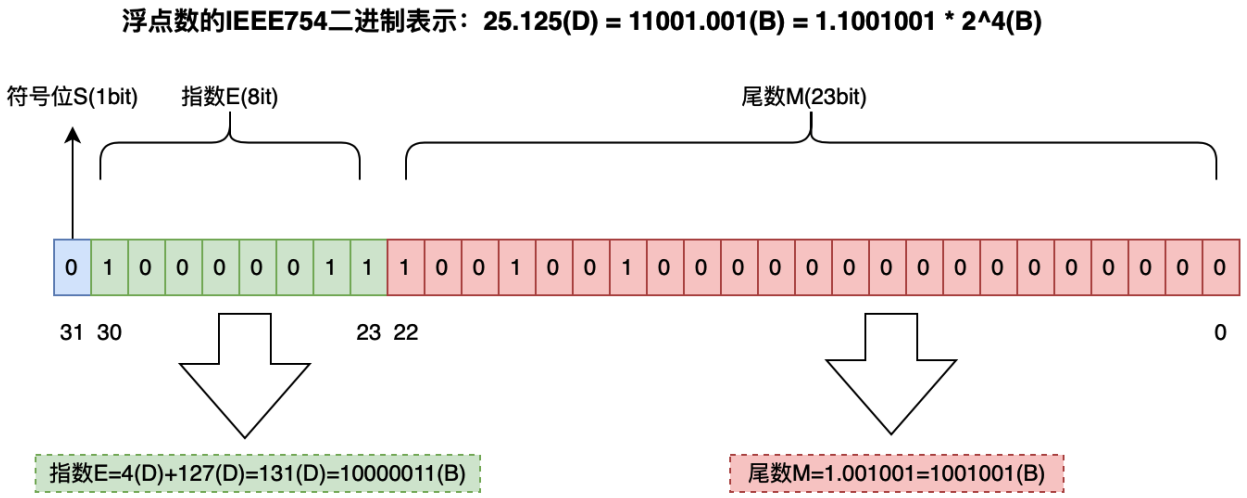


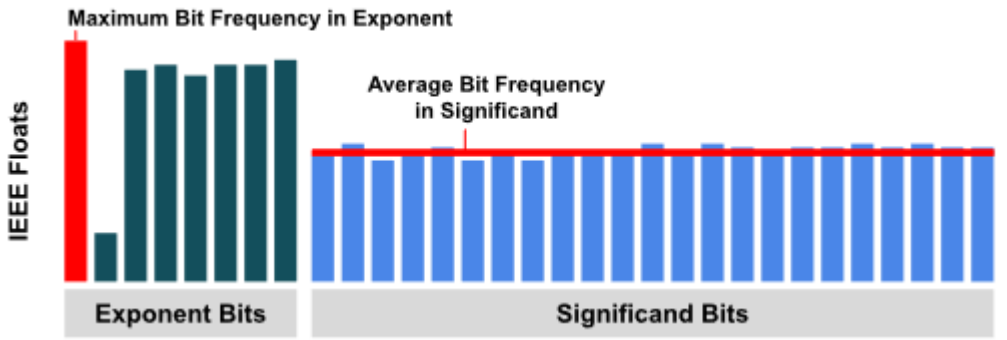


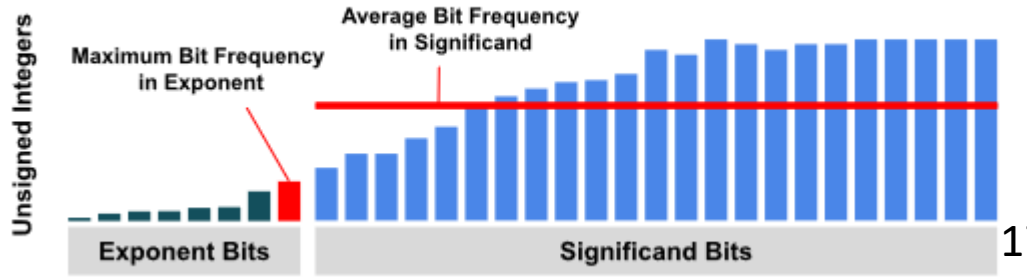
算法原理 BinaryInferno

- Atomic Detector（原子探测器）
  - 浮点数、时间戳、长度
- Float Detector（浮点数探测器）
  - 基于 IEEE 754 浮点表示的分布特征，识别潜在浮点数表示
  - 实际数据集中，浮点数往往集中在较小范围，相邻的实数值会使用相同的指数值，但有效数值不同
  - 指数位中 1 比特频率相对较高，有效数位中 1 比特频率相对较低
  - 量化表示：

$$L_{radio} = \frac{Average\ Bit\ Frequency\ in\ Significand}{Maximum\ Bit\ Frequency\ in\ Exponent}$$







算法原理 BinaryInferno

- **Timestamp Detector（时间戳探测器）**
  - 已知静态网络数据包的捕获时间

$$isTimestamp(X) = \bigvee_{x \in X} start \le f(x) \le end$$

- **Length Detector（长度探测器）**
  - 先编码数据的无符号整数长度，再编码数据本身
  - 切片是字节级、切片中的值与消息长度相对应

$$isStructLength(X, L) = \bigvee_{x_i \in X} ((x_i + k) = L_i)$$

    - k是非负常数，L_i是消息长度
    - L是实际后续数据长度
  - 案例：未知协议的二进制数据流（16进制编码）
    - 04 41 42 43 44 03 61 62 63 02 78 79

(2) 数据的实际长度集合

假设我们从每个字节开始，计算后续的实际数据长度（不包括当前字节本身）：

对于数据流 `04 41 42 43 44 03 61 62 63 02 78 79`:

- 从第 1 字节开始，后续数据长度为 **11**。
- 从第 2 字节开始，后续数据长度为 **10**。
- 从第 3 字节开始，后续数据长度为 **9**，以此类推。

我们可以构造一个**实际长度集合** L，表示从每个字节开始后续的数据长度。

(3) 遍历候选切片 X

我们假设每个字节可能是一个**长度字段**，即从数据流的每个位置开始，尝试用其解释后续子段的长度。

1. **第 1 字节：`04`**
   - 候选切片 $x_1 = 04$。
   - 其值为 4，表示后续数据段应该有 4 字节。
   - 实际后续数据为：`41 42 43 44`，长度确实为 4。
   - 差值 $k = L - x_1 = 4 - 4 = 0$。

   结论：`04` 是一个可能的长度字段。

(4) 判断是否为严格长度字段

通过以上计算，我们发现：

- 当候选切片值 $x_i + k = L$ 且 k 为固定值时，切片 x_i 可以解释为长度字段。
- 在本例中，`04`、`03` 和 `02` 都满足条件，且 $k = 0$。

因此，这些字段被识别为**严格长度字段**。

算法原理 BinaryInferno

- **Field-Boundary Detector（场边界探测器）**
  - 核心原理：信息论中的香农熵概念
  - 多字节字段在不同字节中包含的信息量往往不同
    - 整数字段：最低有效位变化更频繁，包含的信息量更多
    - IEEE754：有效数位的最低有效字节相比指数位的最高有效字节变化更多
  - 香农熵计算公式

$$H(M_k) = -\sum_{v \in M_k} P(v) \log P(v)$$

    - M_k是样本M中每个消息的第k个字节的集合
    - $P(v)$是值v在M_k中出现的概率

$$H(A) - H(B) \geq 1.0$$

  - 阈值取1.0，要求信息更丰富的切片(字节)拥有两倍的信息

算法原理 BinaryInferno

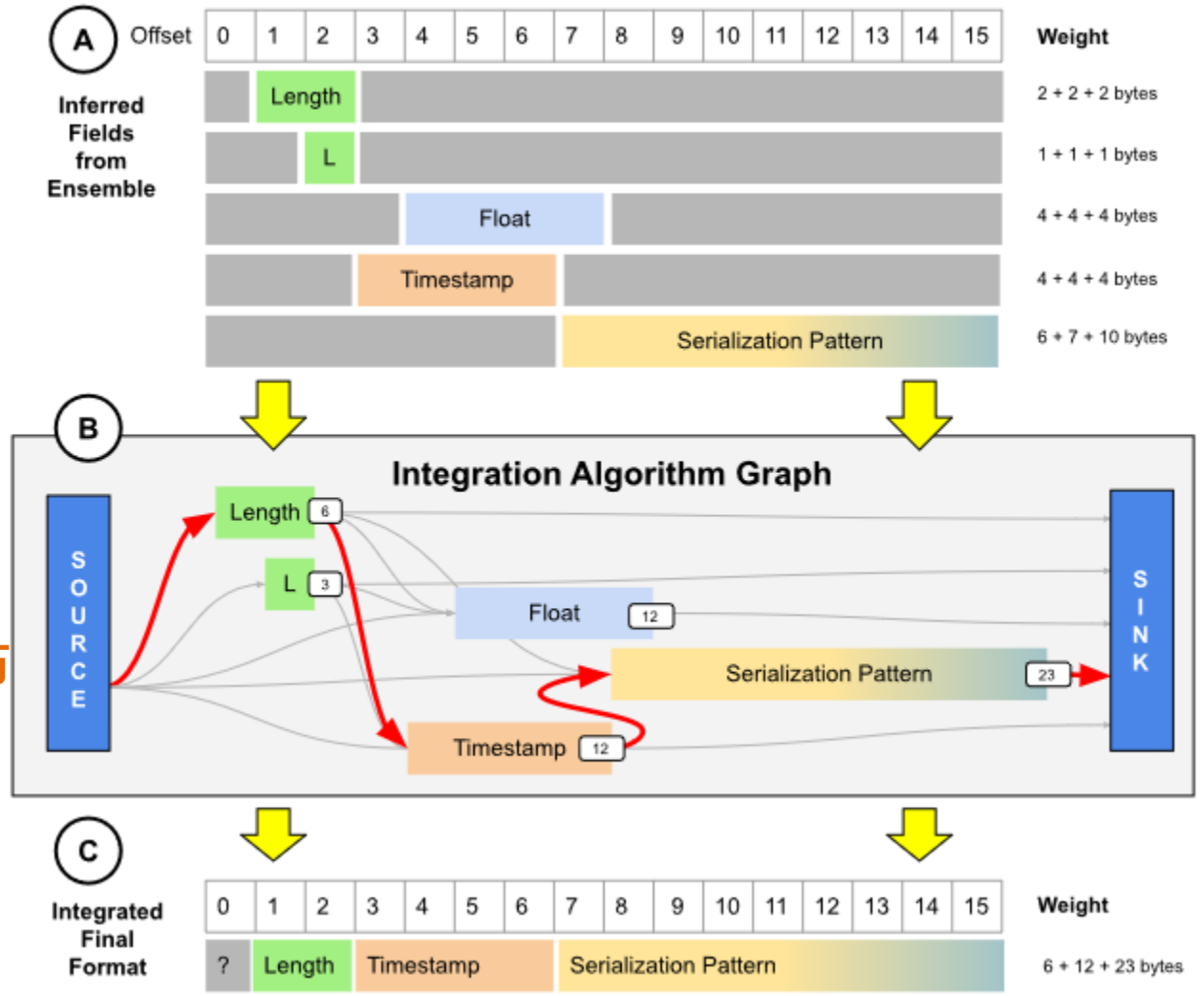
- **Pattern-based Detector（模式探测器）**
  - 识别常见的序列化模式来准确描述可变长度字段结构
  - 模式表示
    - 使用扩展巴科斯范式（EBNF）语法来表示消息格式
  - 模式推断
    - 深度优先搜索算法
      - 在语法定义的模式中寻找能够解释消息样本的模式
  - 优化策略
    - 记忆化：记录<VLFIELD>在各偏移处的模式，避免重复解释想相同消息与模式
    - 常量值切片处理：限制常量值对模式推断的误导
    - 并行化搜索

$$
\begin{aligned}
\langle PATTERN \rangle &::= L(V)^{|L|} \\
&\mid TL(V)^{|L|} \\
\langle T,L,Q \rangle &::= \mathrm{BYTE}^{N}\ (1 \le \mathrm{N} \le 4) \\
\langle V \rangle &::= \mathrm{BYTE} \\
\langle FIELD \rangle &::= \langle VLFIELD \rangle \\
&\mid \mathrm{BYTE}^{P}\ (\mathrm{P} \ge 1) \\
\langle VLFIELD \rangle &::= \langle PATTERN \rangle \\
&\mid Q(\langle PATTERN \rangle)^{|Q|} \\
&\mid Q(VV)^{|Q|} \mid \ldots \mid Q(VVVVVVVV)^{|Q|} \\
\langle FORMAT \rangle &::= \langle FIELD \rangle \\
&\mid \langle FIELD \rangle\ \langle FORMAT \rangle
\end{aligned}
$$

字符表示	含义
<PATTERN>	序列化模式
<T><L><Q><V>	类型、长度、数量、值
<FIELD>	字段是变长或定长
<VLFIELD>	模式or模式重复
<FORMAT>	连接字段

算法原理 BinaryInferno

- 集成算法
  - 将各检测器的描述集成为单一描述，解决推断冲突问题
- 构建有向无环图（DAG）
  - 节点：各探测器推断出的字段
  - 边：根据各字段的偏移量进行判别
- 权重分配
  - 边权重：各消息中样本字段覆盖字节数，反映字段信息量大小
  - 寻找最大权重路径来找到最佳描述




```
01000D60A67AED054150504C45
01000E60A67AF9064F52414E4745
01001160A67B0504504C554D0450454152
```

实验设计　数据资源

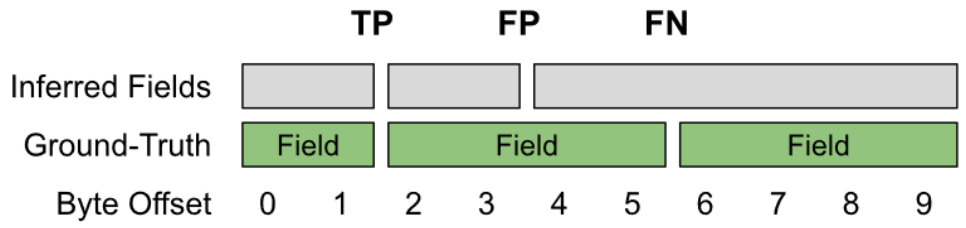
- 数据资源

协议类型	协议名称	数据来源	数据量
网络协议	bgp	网络安全竞赛	1000
	dhcp	公开数据集	1000
	ntp	公开数据集	1000
	smb	公开数据集	1000
	smb2	公开数据集	1000
工业控制协议	dnp3	公开数据集	1000
	modbus	网络安全竞赛	1000
专有协议	mavlink	模拟生成	1000
	mirai	模拟生成	1000
自建	tutorial	实际网络通信	1000

- 对比方法（字段推理）
  - 使用语义类型进行字段推理：Awre（2019）、FieldHunter（2015）
  - 比特一致性的字段边界推断：Nemesys（2018）
  - 多序列比对技术：Netplier（2021）
  - N-W序列对齐技术：Netzob（2014）
- 评价指标

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN} \qquad F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$



实验结果 对比实验

- 实验结果-协议样本测试

包含字节序信息、日期信息等先验知识

		Top-level Protocol Samples																				
		BI+			BI			AWRE			FIELDHUNTER			NEMESYSR			NETPLIER			NETZOB		
Sample	# Msgs	Pre.	Rec.	FPR	Pre.	Rec.	FPR	Pre.	Rec.	FPR	Pre.	Rec.	FPR	Pre.	Rec.	FPR	Pre.	Rec.	FPR	Pre.	Rec.	FPR
bgp	1000	**1.00**	**0.94**	**0.00**	**1.00**	**0.94**	**0.00**	0.51	0.06	0.005	**1.00**	**0.94**	**0.00**	0.06	0.50	0.36	0.22	0.53	0.09	0.00	0.00	0.02
dhcp	1000	**0.67**	0.75	0.02	0.66	**0.82**	0.03	0.07	0.01	**0.01**	*	*	*	0.49	0.36	0.02	0.18	0.31	0.08	-	-	-
dnp3	1000	0.61	0.31	**0.02**	0.61	0.31	**0.02**	0.00	0.00	0.07	*	*	*	0.34	0.54	0.13	0.23	**1.00**	0.42	**0.64**	0.40	0.03
mavlink	1000	**1.00**	0.67	**0.00**	0.72	0.84	0.03	0.50	0.17	0.03	**1.00**	0.33	**0.00**	0.39	0.78	0.11	0.17	**0.99**	0.43	**1.00**	0.50	**0.00**
mirai	1000	**1.00**	**1.00**	**0.00**	**1.00**	**1.00**	**0.00**	0.00	0.00	0.05	0.33	0.10	0.03	0.35	0.24	0.07	0.63	0.55	0.05	0.86	0.60	0.02
modbus	1000	**1.00**	0.40	**0.00**	0.60	0.60	0.05	0.00	0.00	0.09	0.76	0.60	0.02	0.54	0.59	0.06	0.23	**0.80**	0.34	0.60	0.60	0.05
ntp48	1000	**1.00**	0.70	**0.00**	0.26	0.50	0.16	0.00	0.00	0.04	*	*	*	0.39	0.61	0.11	0.22	**1.00**	0.42	0.38	0.31	0.06
smb	1000	**1.00**	0.59	**0.00**	0.73	0.73	0.02	0.01	0.002	0.04	*	*	*	0.17	0.25	0.07	0.39	**0.87**	0.08	0.41	0.45	0.04
smb2	1000	**1.00**	0.59	**0.00**	0.50	0.59	0.03	0.67	0.08	0.004	0.00	0.00	0.01	0.21	0.44	0.08	0.18	**0.99**	0.22	-	-	-
tutorial	1000	**1.00**	**1.00**	**0.00**	0.80	**1.00**	0.03	0.00	0.00	0.08	**1.00**	0.25	**0.00**	0.16	0.09	0.06	0.45	0.50	0.08	0.67	0.50	0.03
Average Performance		**0.93**	0.70	**0.005**	0.69	0.73	0.04	0.18	0.03	0.04	0.68	0.37	0.01	0.31	0.44	0.11	0.29	**0.75**	0.22	0.57	0.42	0.03
bgp	500	**1.00**	**0.94**	**0.00**	**1.00**	**0.94**	**0.00**	0.52	0.06	0.005	**1.00**	**0.94**	**0.00**	0.06	0.50	0.36	0.19	0.53	0.11	0.00	0.00	0.02
dhcp	500	**0.71**	0.67	0.02	0.70	**0.76**	0.02	0.07	0.005	**0.01**	*	*	*	0.49	0.37	0.02	0.18	0.24	0.07	-	-	-
dnp3	500	0.61	0.31	**0.02**	0.61	0.31	**0.02**	0.00	0.00	0.07	*	*	*	0.34	**0.54**	0.13	0.24	**1.00**	0.40	**0.64**	0.39	0.03
mavlink	500	**1.00**	0.67	**0.00**	0.72	0.84	0.03	0.50	0.17	0.03	**1.00**	0.33	**0.00**	0.39	0.78	0.11	0.16	**0.88**	0.41	**1.00**	0.50	**0.00**
mirai	500	**1.00**	**1.00**	**0.00**	**1.00**	**1.00**	**0.00**	0.00	0.00	0.05	0.33	0.10	0.03	0.35	0.24	0.07	0.51	0.60	0.09	0.86	0.60	0.02
modbus	500	**1.00**	0.40	**0.00**	0.60	0.60	0.05	0.00	0.00	0.09	0.76	0.60	0.02	0.55	0.59	0.07	0.26	**0.80**	0.30	0.60	0.60	0.05
ntp48	500	**1.00**	0.70	**0.00**	0.26	0.50	0.16	0.00	0.00	0.04	*	*	*	0.39	0.62	0.11	0.22	**1.00**	0.42	0.43	0.30	0.05
smb	500	**1.00**	0.59	**0.00**	0.70	0.66	0.02	0.01	0.002	0.04	*	*	*	0.20	0.28	0.07	0.31	**0.79**	0.10	0.38	0.46	0.05
smb2	500	**0.80**	0.59	0.01	0.47	0.59	0.03	0.68	0.08	**0.004**	0.00	0.00	0.01	0.22	0.45	0.08	0.18	**0.99**	0.23	-	-	-
tutorial	500	**1.00**	**1.00**	**0.00**	0.80	**1.00**	0.03	0.00	0.00	0.08	**1.00**	0.25	**0.00**	0.16	0.09	0.06	0.44	0.50	0.08	0.67	0.50	0.03
Average Performance		**0.91**	0.69	**0.005**	0.69	0.72	0.04	0.18	0.03	0.04	0.68	0.37	0.01	0.31	0.45	0.11	0.27	**0.73**	0.22	0.57	0.42	0.03
bgp	100	**1.00**	**0.89**	**0.00**	**1.00**	**0.89**	**0.00**	0.00	0.00	0.04	**1.00**	**0.89**	**0.00**	0.06	0.51	0.32	0.12	0.55	0.17	0.00	0.00	0.02
dhcp	100	0.05	0.14	0.15	0.05	0.14	0.15	0.07	0.01	**0.01**	0.00	0.00	**0.01**	**0.44**	0.34	0.02	0.37	**0.53**	0.05	-	-	-
dnp3	100	0.61	0.29	**0.02**	0.61	0.29	**0.02**	0.00	0.00	0.06	0.14	0.03	**0.02**	0.29	0.46	0.14	0.23	**1.00**	0.40	**0.64**	0.37	**0.02**
mavlink	100	**1.00**	0.67	**0.00**	0.51	0.51	0.04	0.00	0.00	0.05	*	*	*	0.40	0.80	0.11	0.15	**0.85**	0.42	**1.00**	0.50	**0.00**
mirai	100	**1.00**	**1.00**	**0.00**	**1.00**	**1.00**	**0.00**	0.00	0.00	0.05	0.00	0.00	0.01	0.34	0.24	0.07	0.44	**1.00**	0.19	0.86	0.60	0.01
modbus	100	**1.00**	0.40	**0.00**	0.60	0.60	0.05	0.00	0.00	0.09	0.77	0.60	0.02	0.54	0.58	0.07	0.25	**0.80**	0.32	0.60	0.60	0.05
ntp48	100	**0.75**	0.60	**0.02**	0.17	0.30	0.18	0.00	0.00	0.04	*	*	*	0.37	0.56	0.11	0.22	**1.00**	0.42	0.43	0.31	0.05
smb	100	**1.00**	0.59	**0.00**	0.78	0.73	0.01	0.004	0.002	0.05	*	*	*	0.19	0.30	0.08	0.22	**1.00**	0.21	0.33	0.42	0.05
smb2	100	**0.82**	0.71	0.01	0.45	0.71	0.05	0.69	0.08	**0.004**	*	*	*	0.24	0.48	0.08	0.18	**0.99**	0.24	0.20	0.76	0.15
tutorial	100	**1.00**	**1.00**	**0.00**	0.81	**1.00**	0.03	0.08	0.04	0.08	1.00	0.24	**0.00**	0.17	0.09	0.06	0.34	0.61	0.15	0.67	0.48	0.03
Average Performance		**0.82**	0.63	0.02	0.60	0.62	0.05	0.08	0.01	0.05	0.48	0.29	**0.01**	0.30	0.44	0.10	0.25	**0.83**	0.26	0.53	0.45	0.04

实验结果 对比实验

- 实验结果-有效载荷测试
  - 先验知识对BinaryInferno的提升较大

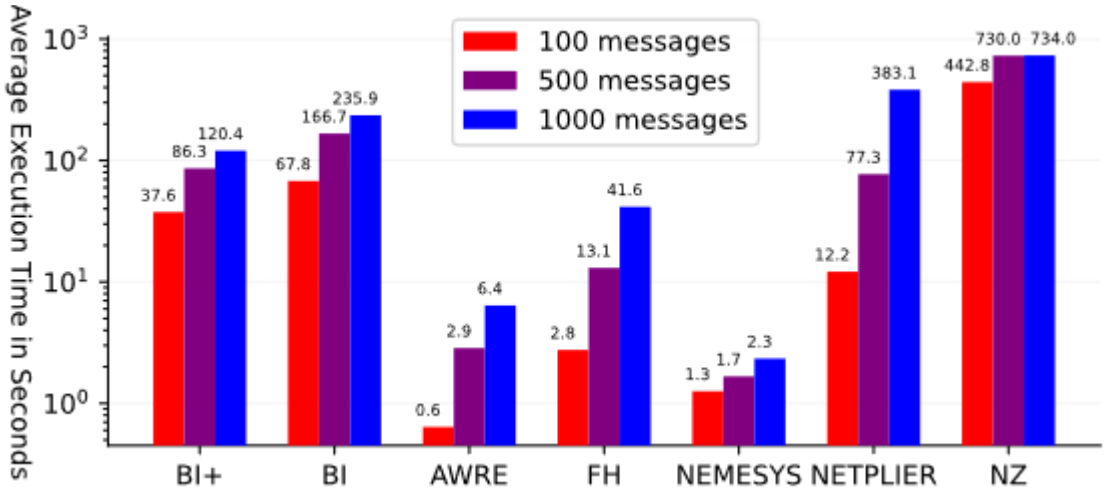
		Encapsulated Payload Samples																				
		BI+			BI			AWRE			FIELDHUNTER			NEMESYS			NETPLIER			NETZOB		
Sample	# Msgs	Pre.	Rec.	FPR	Pre.	Rec.	FPR	Pre.	Rec.	FPR	Pre.	Rec.	FPR	Pre.	Rec.	FPR	Pre.	Rec.	FPR	Pre.	Rec.	FPR
bgp01	22	**0.93**	0.77	**0.01**	**0.93**	0.77	**0.01**	0.00	0.00	0.02	0.51	0.23	0.04	0.29	0.51	0.24	0.48	**0.90**	0.18	0.46	0.53	0.12
bgp02	104	**1.00**	0.46	**0.00**	0.67	0.46	0.01	0.00	0.00	0.02	0.74	**0.67**	0.01	0.09	**0.67**	0.25	0.06	0.54	0.31	0.21	0.23	0.03
bgp04	1000	**1.00**	**1.00**	**0.00**	**1.00**	**1.00**	**0.00**	**1.00**	**1.00**	**0.00**	**1.00**	**1.00**	**0.00**	0.00	0.00	0.43	**1.00**	**1.00**	**0.00**	**1.00**	**1.00**	**0.00**
mavlink001	206	**1.00**	0.67	**0.00**	**1.00**	0.67	**0.00**	**1.00**	0.33	**0.00**	*	*	*	0.22	0.66	0.12	0.75	**1.00**	0.02	**1.00**	**1.00**	**0.00**
mavlink002	51	**1.00**	**1.00**	**0.00**	0.25	**1.00**	0.14	0.00	0.00	0.15	*	*	*	0.18	**1.00**	0.21	0.12	**1.00**	0.32	0.33	**1.00**	0.09
mavlink004	33	**1.00**	**0.50**	**0.00**	0.25	**0.50**	0.12	0.00	0.00	0.18	*	*	*	0.15	0.33	0.15	0.14	**0.50**	0.24	0.33	**0.50**	0.08
mavlink024	400	**1.00**	0.62	**0.00**	0.40	0.50	0.12	0.00	0.00	0.10	*	*	*	0.37	0.47	0.13	0.33	**0.88**	0.27	0.45	0.62	0.12
mavlink026	1000	*	*	*	0.00	0.00	0.05	0.00	0.00	0.09	*	*	*	-	-	-	0.00	0.00	0.07	**0.50**	**1.00**	**0.02**
mavlink030	1000	**1.00**	**1.00**	**0.00**	0.60	**1.00**	0.08	0.00	0.00	0.07	*	*	*	0.22	0.38	0.16	0.22	**1.00**	0.43	0.50	0.17	0.02
mavlink031	1000	**1.00**	**1.00**	**0.00**	0.78	**1.00**	0.04	0.00	0.00	0.06	*	*	*	0.23	0.38	0.15	0.23	**1.00**	0.43	0.50	0.14	0.02
mavlink032	1000	**1.00**	**1.00**	**0.00**	0.38	0.83	0.16	0.00	0.00	0.07	*	*	*	0.22	0.37	0.16	0.22	**1.00**	0.43	0.50	0.17	0.02
mavlink033	1000	**0.75**	0.75	**0.04**	0.46	0.75	0.15	0.25	0.12	0.10	*	*	*	0.45	0.63	0.13	0.31	**1.00**	0.38	0.51	0.63	0.10
mavlink042	371	**1.00**	**1.00**	**0.00**	0.00	0.00	0.33	0.00	0.00	0.55	**1.00**	**1.00**	**0.00**	-	-	-	0.00	0.00	0.33	0.00	0.00	0.33
mavlink046	5	**1.00**	**1.00**	**0.00**	0.00	0.00	0.33	0.00	0.00	0.56	**1.00**	**1.00**	**0.00**	-	-	-	0.00	0.00	0.33	0.00	0.00	0.33
mavlink076	93	*	*	*	*	*	*	0.00	0.00	0.06	*	*	*	0.06	**0.58**	0.26	**0.33**	0.50	**0.03**	**0.33**	0.50	**0.03**
mavlink083	385	**0.89**	0.89	**0.02**	0.58	0.78	0.08	0.00	0.00	0.08	*	*	*	0.26	0.35	0.14	0.25	**1.00**	0.42	0.60	0.33	0.03
mavlink085	377	**0.80**	0.73	**0.02**	0.37	0.64	0.13	0.00	0.00	0.13	*	*	*	0.30	0.43	0.12	0.25	**1.00**	0.37	0.25	0.18	0.07
mavlink087	247	**1.00**	**0.75**	**0.00**	0.50	**0.75**	0.03	0.00	0.00	0.04	*	*	*	0.37	0.66	0.05	0.15	0.50	0.11	0.29	0.50	0.05
mavlink111	410	**0.50**	**1.00**	**0.03**	0.33	**1.00**	0.07	0.00	0.00	0.11	*	*	*	0.10	**1.00**	0.30	0.17	**1.00**	0.17	0.33	**1.00**	0.07
mavlink140	383	**1.00**	0.57	**0.00**	0.50	0.57	0.05	0.00	0.00	0.05	*	*	*	0.27	0.39	0.10	0.26	**0.86**	0.23	0.33	0.43	0.08
mavlink141	388	**0.67**	0.80	**0.03**	0.33	0.60	0.10	0.00	0.00	0.06	*	*	*	0.28	0.61	0.13	0.24	**1.00**	0.28	0.33	0.40	0.07
mavlink147	207	**1.00**	0.40	**0.00**	**1.00**	0.40	**0.00**	0.00	0.00	0.05	*	*	*	0.18	0.36	0.12	0.62	**1.00**	0.05	**1.00**	0.60	**0.00**
mavlink230	39	**1.00**	**1.00**	**0.00**	0.89	**1.00**	0.01	0.00	0.00	0.05	*	*	*	0.29	0.42	0.11	0.26	**1.00**	0.31	0.45	0.62	0.08
mavlink241	30	**1.00**	**1.00**	**0.00**	0.80	**1.00**	0.02	0.14	0.25	0.16	*	*	*	0.29	0.54	0.09	0.19	0.75	0.22	0.50	0.50	0.03
mavlink242	14	**1.00**	0.44	**0.00**	**1.00**	0.44	**0.00**	*	*	*	*	*	*	0.53	0.77	0.06	0.31	**0.89**	0.19	0.55	0.67	0.05
mavlink245	191	*	*	*	*	*	*	**1.00**	**1.00**	**0.00**	*	*	*	-	-	-	**1.00**	**1.00**	**0.00**	**1.00**	**1.00**	**0.00**
Average Performance		**0.94**	**0.80**	**0.01**	0.54	0.65	0.08	0.14	0.11	0.11	0.85	0.78	**0.01**	0.24	0.52	0.17	0.30	0.78	0.24	0.47	0.53	0.07

实验结果 模块性能验证

- **Field-Boundary Detector**
  - 给定字节序（大端or小端模式），性能优良
  - 对复杂语义字段处理性能有限
  - 依赖数据的分布特征
- **Pattern-based Detector**
  - 能够自动地从消息样本中挖掘出序列化模式
  - 数据依赖：样本数过少无法正确推断
  - 对于未知协议逆向的贡献有待评估
- 测试时间分析
  - BinaryInferno模式搜索占时较长

Sample	Endianess	Precision	Recall	FPR	F1
Top-level	Correct	0.88	0.35	0.00	0.47
Payload	Correct	0.96	0.63	0.00	0.68
Top-level	Incorrect	0.38	0.14	0.04	0.18
Payload	Incorrect	0.26	0.15	0.11	0.06

Sample	Pattern	100	500	1000
dhcp	$(TL(V)^{[L]})^*$	N	Y	Y
mirai	$Q(V^5)^{[Q]} \cdot Q(TL(V)^{[L]})^{[Q]}$	Y	Y	Y
smb	$Q(VV)^{[Q]} \cdot LL(V)^{[LL]}$	Y	Y	Y
tutorial	$(L(V)^{[L]})^*$	P	Y	Y



算法总结

- 算法贡献
  - 构建一系列专用探测器，独立分析消息样本中的特定语义类型信息
    - 原子探测器识别浮点数、时间戳等基本语义数据类型
    - 模式探测器用于推断常见的序列化模式
  - 提出一种基于香农熵的场边界探测器
  - 设计集成算法，最大限度地整合个探测器推断信息
- 算法不足
  - 数据集质量敏感
  - 模式检测局限性，无法处理特殊的序列化模式
    - DNS中：长度或数量与所描述值存在特定距离关系
  - 算法处理字节级别，无法处理其他级别的数据格式

MDIplier: Protocol Format Recovery via Hierarchical Inference

算法原理 TIPO

T	目标	识别协议层次结构，推断报文格式
I	输入	8种协议样本数据包（dhcp、dns、ntp、smb、smb2、dnp3、modbus、S7comm等）
P	处理	1.数据预处理，去重，补充源、目标IP、端口号 2.分隔符识别，推断可能的划分消息头与消息体的分隔符 3.层次推断，对消息头与消息体进行分析，并进行结果整合
O	输出	协议报文格式信息，包括字段边界、字段类型等

P	问题	分层结构不能适用所有协议，样本数据敏感
C	条件	协议具有层次结构，包含消息分隔符
D	难点	设计适用多种协议的分层结构识别策略 获得协议层次结构后，如何减少字段信息丢失，提高字段推断性能
L	水平	ISSRE 2024 CCF B

算法原理 创新说明

- **MDIplier**
  - 提出一种层次化的协议格式推理方法：利用协议本身的层次结构，对每个消息层进行定制分析
  - 解决传统方法对协议消息进行全局聚类，未考虑消息本身的层次结构导致的字段信息丢失，进一步导致错误推断字段边界
  - 设计分隔符识别模块，实现消息的合理分层
  - 综合字段推断策略
    - 消息头：多序列比对方法
    - 消息体：基于关键字的聚类方法

F_1		F_2		F_3		F_4	F_5	F_6		F_7	
00	01	00	00	00	06	FF	05	00	00	FF	00
00	0B	00	00	00	06	FF	05	00	01	00	00
00	15	00	00	00	06	FF	05	00	02	FF	00
00	02	00	00	00	04	FF	01	01	01		
00	04	00	00	00	06	FF	01	00	00	00	63
00	0C	00	00	00	06	FF	01	00	01	00	01

(a) Ground truth of message fields

Cluster 1

F_1	F_2			F_3				F_4	F_5	F_6	
00	01	00	00	00	06	FF	05	00	00	FF	00
00	0B	00	00	00	06	FF	05	00	01	00	00
00	15	00	00	00	06	FF	05	00	02	FF	00

Cluster 2

F_1	F_2		F_3		F_4		F_5	F_6	F_7	F_8	F_9
00	02	00	00	00	04	FF	01	01	01	--	--
00	04	00	00	00	06	FF	01	00	00	00	63
00	0C	00	00	00	06	FF	01	00	01	00	01

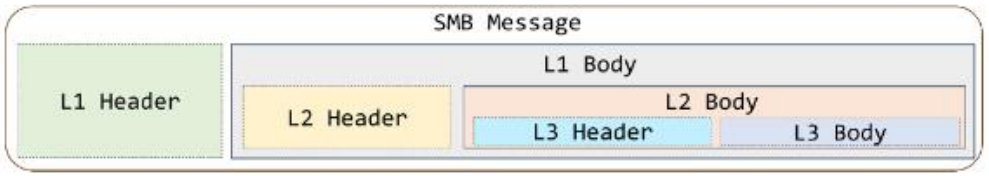
Message Header　　Message Body

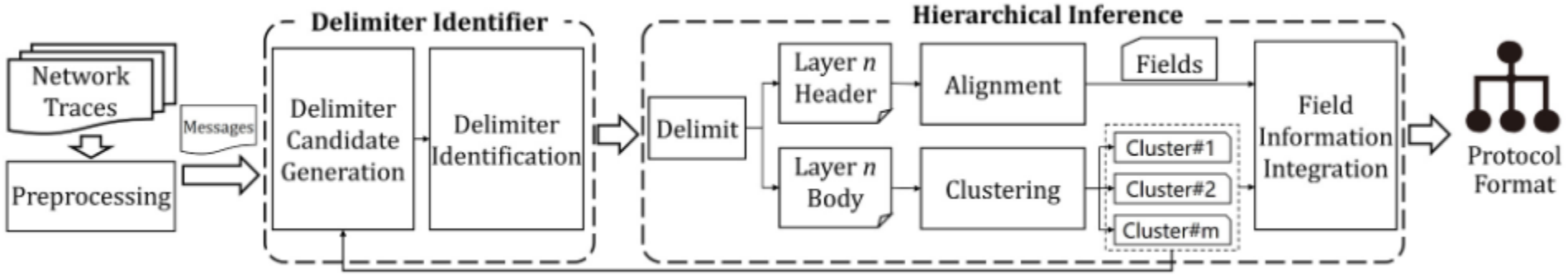
(b) Keyword-based clustering and field inference of each cluster

算法原理 MDIplier

- 算法结构
  - 预处理模块
  - 分隔符识别模块
    - 分析消息数据，确定用于将消息分层的分隔符
  - 层次推断模块
    - 消息头：结构固定，字段值在所在消息中的位置相对一致
    - 消息体：结构多样，字段值变化大

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0000h:	00	00	00	68	FF	53	4D	42	32	00	00	00	00	18	43	C8
0010h:	00	00	00	00	00	00	00	00	00	00	00	00	B4	61	17	E5
0020h:	80	22	12	00	0F	24	00	00	00	0A	00	FF	FF	00	00	00
0030h:	00	00	00	00	00	00	00	24	00	44	00	00	00	68	00	01
0040h:	00	01	00	27	00	00	44	20	16	00	56	05	06	00	04	01
0050h:	00	00	00	00	5C	00	74	00	6D	00	70	00	5F	00	73	00
0060h:	6D	00	62	00	31	00	5C	00	2A	00	00	00				





算法原理 MDIplier

- 分隔符识别模块
  - 准备阶段：初始化3个列表
    - L_c存储候选分隔符
    - L_d记录消息中相邻字符之间的差异度量
    - L_p记录潜在分隔符位置
  - 迭代阶段
    - 迭代次数：数据集中所有消息的最小字节数
    - 逐字节进行迭代，遍历数据集中每个消息M
    - 计算当前字节位置i在消息M与下一条消息M_next的字符差异
    - 判断当前字节i是否为潜在分隔符
  - 综合L_d， L_p，得到候选分隔符

Algorithm 1: Delimiter Candidate Generation

Input : Γ - messages data under analysis
Output: L_c - inferred delimiter candidate

```
1  L_c ← []; L_d ← []; L_p ← []
2  minlen ← maxnum
3  foreach message M in Γ do
4  |  minlen = ComputeMinLength(minlen, |M|)
5  end
6  for 0 < i < minlen do
7  |  diff_msg ← 0; pos_flag ← True
8  |  foreach message M in Γ do
9  |  |  diff_msg = CalcMsgDiff(M, M_next)
10 |  |  pos_flag = CompMsgDiff(M, M_next)
11 |  end
12 |  L_d[i].append(diff_msg)
13 |  L_p[i].append(pos_flag)
14 end
15 L_c = DelimCandRules(L_d, L_p)
16 return L_c
```

算法原理 MDIplier

- 层次推断模块
  - 消息头
    - 结构固定，字段值在所有消息中相对一致
    - 多序列比对算法划分字段→可变字段+常规字段
    - 合并连续常规字段得到消息头字段推断结果
  - 消息体
    - 结构多样、字段值变化大
    - 利用多序列比对算法划分字段，提取可变字段生成关键字候选字段列表
    - 遍历列表，计算每个字段作为关键字的后验概率
    - 选择概率最高的字段作为关键字进行聚类
    - 在每个聚类中再次进行多序列比对和字段推断
    - 得到消息体字段推断结果

实验设计　数据资源

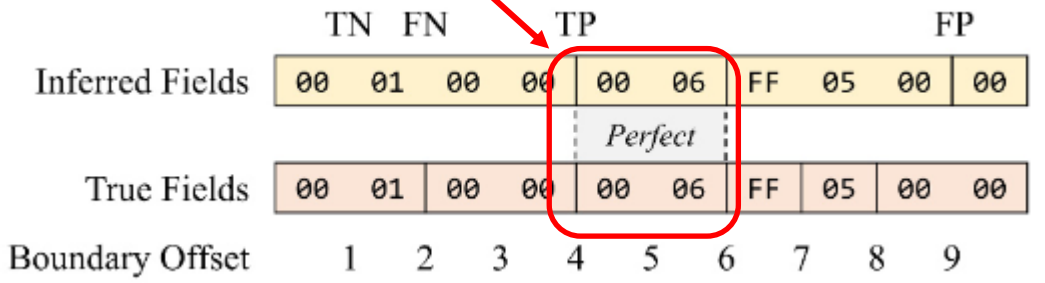
- 数据资源

协议类型	协议名称	数据来源	数据量
网络协议	dhcp	公开数据集	1000
	dns	公开数据集	1000
	ntp	公开数据集	1000
	smb	公开数据集	1000
	smb2	公开数据集	1000
工业控制协议	dnp3	公开数据集	1000
	modbus	公开数据集	1000
	S7comm	公开数据集	1000
专有协议	Yeelight Light	实际网络通信	1000
	TPLink Router	实际网络通信	1000
	Philips Bridge	实际网络通信	1000

- 对比方法（字段推理）
  - 使用语义类型进行字段推理：FieldHunter（2015）
  - 多序列比对技术：Netplier（2021）
  - N-W序列对齐技术：Netzob（2014）
  - 集成检测算法：BinaryInferno（2023）
- 评价指标
  - 完善性指标
    - *perfection*

$$Accuary = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$



实验结果　对比实验

- 实验结果

Protocol	#msg	MDIplier			Netplier			BinaryInferno			Netzob			FieldHunter		
		Acc.	F1	Perf.	Acc.	F1	Perf.	Acc.	F1	Perf.	Acc.	F1	Perf.	Acc.	F1	Perf.
DHCP	100	0.81	**0.34**	**0.34**	0.81	0.32	**0.34**	**0.90**	0.24	0.05	0.89	0.00	0.00	**0.90**	0.18	0.04
DNP3	100	**0.86**	**0.72**	**0.42**	0.81	0.41	0.10	0.78	0.67	0.41	0.75	0.59	0.29	0.68	0.37	0.00
DNS	100	**0.78**	0.35	**0.00**	0.76	0.15	**0.00**	0.66	0.00	**0.00**	0.70	**0.40**	**0.00**	0.73	0.00	**0.00**
Modbus	100	**0.81**	**0.68**	**0.47**	0.71	0.32	0.01	0.65	0.63	0.16	0.41	0.36	0.05	0.55	0.47	0.00
NTP	100	0.66	0.38	**0.36**	0.69	0.40	**0.36**	0.70	**0.53**	0.27	**0.80**	0.48	0.00	0.70	0.50	0.18
S7comm	100	**0.85**	0.40	0.07	0.83	0.09	0.00	0.64	**0.44**	**0.11**	0.66	0.09	0.00	0.61	0.23	0.00
SMB2	100	0.78	**0.38**	**0.03**	0.79	**0.38**	**0.03**	0.79	0.35	0.02	0.78	0.17	0.02	**0.80**	0.27	0.00
SMB	100	0.73	**0.38**	**0.21**	0.74	0.36	0.17	**0.81**	0.33	0.11	0.78	0.13	0.06	0.81	0.33	0.11
Average-100		**0.78**	**0.45**	**0.24**	0.77	0.30	0.13	0.74	0.40	0.14	0.72	0.28	0.05	0.72	0.29	0.04
DHCP	500	**0.94**	**0.43**	**0.27**	**0.94**	0.39	0.24	0.90	0.24	0.06	0.89	0.00	0.00	0.90	0.20	0.05
DNP3	500	0.76	0.62	**0.41**	0.76	0.56	0.31	**0.78**	**0.67**	**0.41**	0.75	0.59	0.30	0.68	0.37	0.00
DNS	500	0.68	0.24	0.00	0.68	0.23	0.00	0.66	0.00	0.00	0.70	**0.40**	0.00	**0.77**	0.37	**0.01**
Modbus	500	**0.79**	**0.66**	**0.40**	**0.79**	0.61	**0.40**	0.65	0.63	0.16	0.42	0.37	0.01	0.55	0.47	0.00
NTP	500	0.63	0.36	**0.36**	0.64	0.37	**0.36**	0.62	0.36	0.09	0.72	**0.38**	0.01	**0.79**	0.26	0.08
S7comm	500	**0.85**	0.40	0.07	0.83	0.09	0.00	0.64	**0.44**	**0.11**	0.66	0.09	0.00	0.61	0.23	0.00
SMB2	500	0.78	0.39	0.03	0.79	0.38	0.04	**0.81**	**0.40**	**0.05**	0.75	0.21	0.00	**0.81**	0.35	**0.05**
SMB	500	0.72	0.46	**0.26**	0.73	0.43	0.17	**0.80**	**0.49**	0.22	0.71	0.25	0.07	0.76	0.44	0.12
Average-500		**0.77**	**0.44**	**0.23**	**0.77**	0.38	0.19	0.73	0.40	0.14	0.70	0.29	0.05	0.73	0.34	0.04
DHCP	1000	**0.94**	**0.43**	**0.27**	**0.94**	0.39	0.24	0.90	0.24	0.05	0.89	0.00	0.00	0.90	0.20	0.05
DNP3	1000	0.76	0.62	**0.41**	0.76	0.56	0.31	**0.78**	**0.67**	**0.41**	0.75	0.59	0.29	0.68	0.37	0.00
DNS	1000	0.76	0.33	0.00	0.74	0.17	0.00	0.66	0.00	0.00	0.70	**0.40**	0.00	**0.83**	0.61	**0.01**
Modbus	1000	**0.80**	**0.66**	**0.40**	**0.80**	0.61	**0.40**	0.65	0.63	0.16	0.42	0.37	0.01	0.55	0.47	0.00
NTP	1000	0.63	0.36	**0.36**	0.66	0.37	0.32	0.55	0.28	0.00	0.74	**0.40**	0.01	**0.79**	0.26	0.08
S7comm	1000	**0.85**	0.40	0.07	0.83	0.09	0.00	0.64	**0.44**	**0.11**	0.66	0.09	0.00	0.61	0.23	0.00
SMB2	1000	**0.83**	**0.43**	**0.06**	0.81	0.39	0.03	0.80	0.34	0.00	0.75	0.26	0.01	0.82	0.36	0.05
SMB	1000	**0.79**	**0.50**	**0.24**	**0.79**	0.49	0.22	0.78	0.47	0.19	0.71	0.27	0.07	0.76	0.46	0.12
Average-1000		**0.79**	**0.47**	**0.23**	**0.79**	0.38	0.19	0.72	0.38	0.12	0.70	0.30	0.05	0.74	0.37	0.04

实验设计 未知协议评估

- 实验结果
  - 利用MDIplier推断未知协议格式，并基于推断格式生成新消息发送回设备，检查能成功触发原始行为的数量
  - MDIplier实际未知协议逆向是有效的， 9条行为全部触发

Device	Messages Behaviors	Message Format	# Triggered Behaviors
Yeelight Light	Set Bright	R(9) M(1) R(8) M(5) R(12) M(2) R(14) M(4) R(4)	2/2
	Turn On/Off	R(9) M(1) R(8) M(5) R(1) M(3) R(6) M(2) R(2) M(1) R(1) M(18) R(3)	
TPLink Router	Add Forbidden Domain	R(2) M(5) R(1) M(4) R(2) M(99) R(1)	5/5
	Delete Forbidden Domain	R(2) M(5) R(1) M(4) R(2) M(46) R(1)	
	Get Forbidden Domain	R(15) M(6) R(2) M(1) R(7) M(6) R(2) M(10) R(1) M(3) R(2)	
	Get Wireless Status	R(2) M(7) R(4) M(4) R(3) M(1) R(1) M(4) R(1) M(2) R(1) M(12) R(1) M(3) R(2)	
	Set Signal Strength	R(2) M(2) R(1) M(5) R(2) M(14) R(2) M(15) R(1) M(3) R(1) M(38) R(1)	
Philips Bridge	Set Name	R(1) M(11) R(1) M(11) R(1) M(5) R(1) M(8) R(1) M(7) R(1) M(3) R(1) M(7) R(1) M(12) R(1) M(13) R(6) M(7) R(1) M(3) R(2) M(2) R(3) M(2) R(11) M(4) R(1) M(5) R(5) M(1) R(19)	2/2
	Create group	R(1) M(12) R(1) M(11) R(1) M(5) R(1) M(8) R(1) M(7) R(1) M(3) R(1) M(7) R(1) M(14) R(1) M(13) R(6) M(7) R(1) M(3) R(2) M(2) R(3) M(2) R(11) M(4) R(1) M(4) R(6) M(5) R(3) M(13) R(1) M(1) R(2) M(6) R(2) M(1)	

算法总结

- 算法贡献
  - 提出一种基于协议消息分层结构的分析方法
    - 消息头+消息体
    - 针对不同部分采用定制的划分策略
  - 设计了一种迭代式分层推理过程
    - 在每次迭代中精确识别消息层和推断各层格式，优化利用了输入消息中的字段信息，逐步深入解析协议结构，显著提升了对复杂协议的分析能力
- 算法不足
  - 数据集质量敏感
  - 加密数据处理困难
  - 多序列比对算法瓶颈
    - 计算复杂度高、资源消耗大
    - 随着数据量增加，仍可能面临效率和准确性方面的挑战

特点总结与未来展望

特点总结与未来展望

- 特点总结
  - BinaryInferno
    - 设置多种类型的字段类型探测器，独立工作，获取所有可能的数据信息
    - 利用香农熵构建场边界探测器
    - 设计集成算法，最大限度地整合个探测器推断信息
  - MDIplier
    - 关注协议的层次结构，抽象为消息头+消息体，分层制定字段划分策略
    - 迭代式分层推理，深入解析协议结构
- 未来展望
  - 深度学习技术应用、大模型生成等
  - 对加密未知协议的处理
  - 内存开销优化

参考文献

[1] Chandler J, Wick A, Fisher K. BinaryInferno: A Semantic-Driven Approach to Field Inference for Binary Message Formats. NDSS[C]. San Diego, CA, USA :NDSS, 2023: 1-18.

[2] Liang K, Luo Z, Zhao Y, et al. MDIplier: Protocol Format Recovery via Hierarchical Inference. 2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE) [C]. Piscataway, NJ: IEEE, 2024: 547-557.

[3] Kleber S, Maile L, Kargl F. Survey of protocol reverse engineering algorithms: Decomposition of tools for static traffic analysis[J]. IEEE Communications Surveys & Tutorials, 2018, 21(1): 526-561.

知人者智，自知者明。胜人者有力，自胜者强。知足者富。强行者有志。不失其所者久。死而不亡者，寿。

谢谢！