Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心

特定安全攻防场景中的对抗样本生成方法

硕士研究生　张荣倩

2021年07月25日

内容提要

- 背景简介
- 基本概念
- 算法原理
- 应用总结

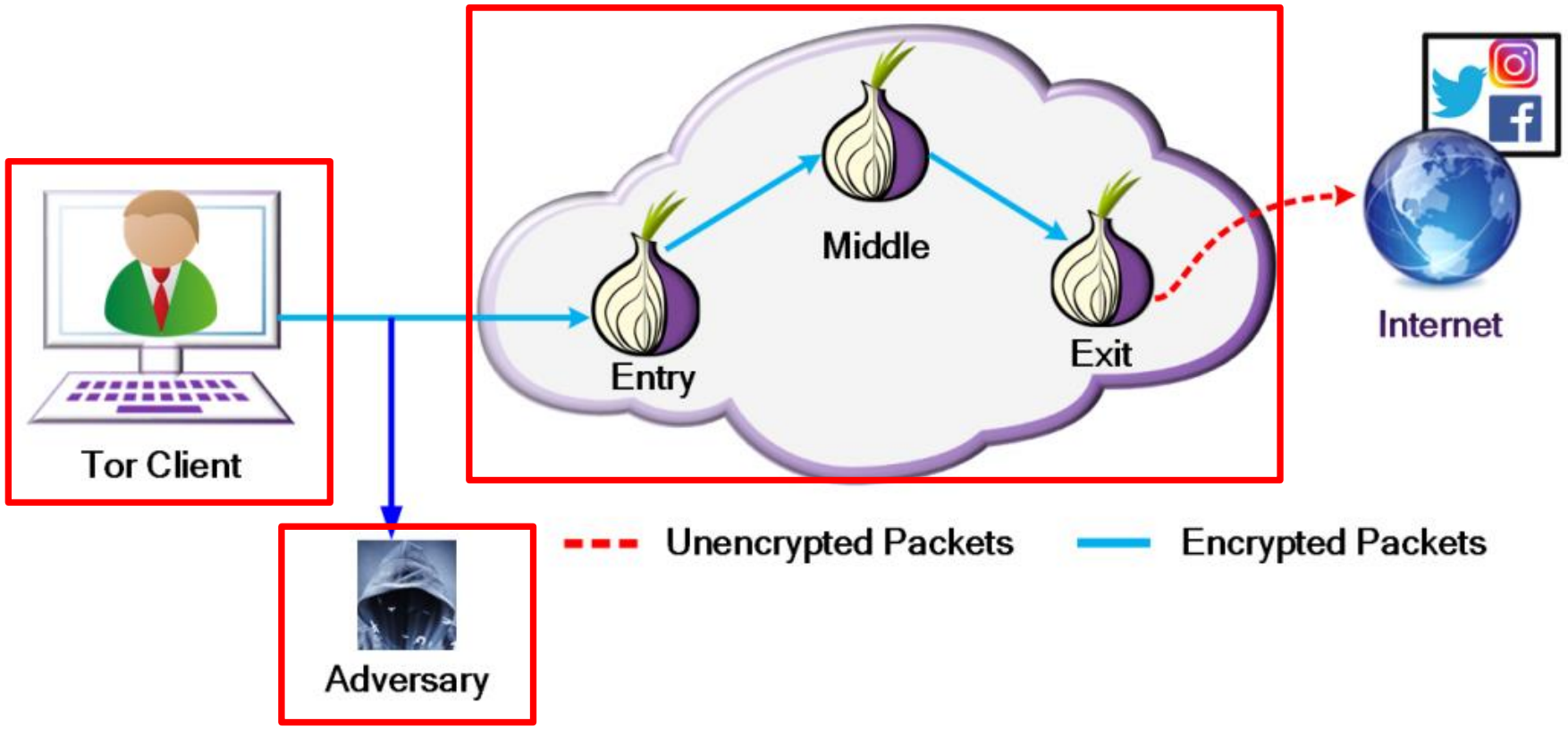
预期收获

- 预期收获
  - 1. 了解网站指纹的基本概念
  - 2. 了解对抗样本在网站指纹防御的应用
  - 3. 了解DGA的基本概念
  - 4. 了解对抗样本在DGA领域的应用

网站指纹防御

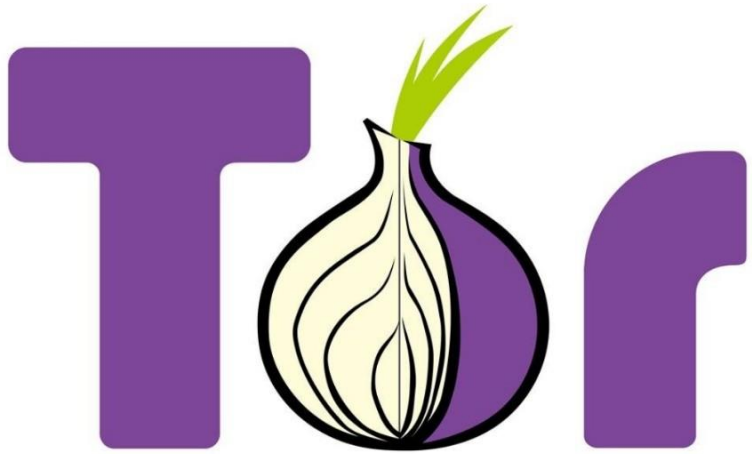
背景介绍

- 网站指纹 Website fingerprinting（WF）
  - 对用户浏览活动匿名性最严重的威胁之一
  - 可以从网站中提取流量模式，分析用户访问的网站来破坏隐私增强技术的保护
  - 组成：
    - 用户
    - 对手
    - 隐私增强技术



背景介绍

- 目标用户（受害者）
  - 攻击者监视和窃取隐私的对象
  - 通过隐私增强技术来保持其通信的机密性和匿名性



- 隐私增强技术（PET）
  - 增强和保护数据隐私，提供匿名性和防止窃听
  - 举例：Tor

- 窃听者（对手）
  - 位于本地
  - 被动性：只进行观察和记录，不可丢弃、延迟或修改流量流中的真实数据包

基本概念

- WF攻击者（对手）
  - 目的：推测用户访问哪些网站
  - 操作：
    - 捕获用户和服务器往来的加密流量包，提取独特的特征获得trace（数据包的方向序列）
    - 多次访问各种网站并从中收集trace集，在此基础上训练有监督分类器(关键)
    - 预测目标用户访问的网站
  - 分类器：
    - 输入：网站流量trace
    - 输出：预测置信度和预测标签（网站域名）
  - 前提：
    - 对手无法解开Tor网络提供的加密（限制）
    - 大多数PET公开可用，可以进行对抗训练（攻击者优势）

基本概念

- WF防御者
  - 目标：混淆对手的分类器
  - 操作：生成对抗性trace（能够掩盖网站流量特征的对抗样本）
  - 手段：
    - 填充虚拟数据包（带宽开销增加）
    - 延迟真实数据包（延迟开销增加）
  - 前提：
    - 平衡开销和防御的有效性（难点）
    - 不仅要抵御当前目标攻击，而且还要抵御可以预见的攻击（可预见性）
    - 利用对抗样本的可转移性抵御对手的对抗训练（防御者优势）

基本概念

- 对抗样本
  - 用作攻击行为：生成精心制作的对抗样本，导致分类器预测错误（图像）
  - 用作防御行为：作为网站指纹域中对手分类器的防御机制（网站指纹）
  - 属性
    - 高错误分类率
    - 小扰动，低开销（轻量级）
    - 可转移性：针对某一分类器生成的对抗性样本可以使其他多个分类模型分类错误（防御者的优势）
- 对抗训练
  - 将对抗样本作为训练集输入到分类器训练
  - 是对抗对抗样本最有效的对策之一（攻击者的优势）

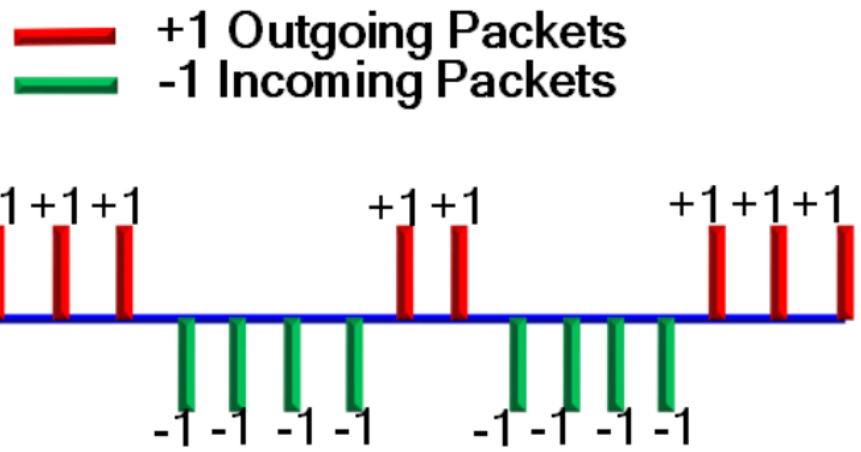
基本概念

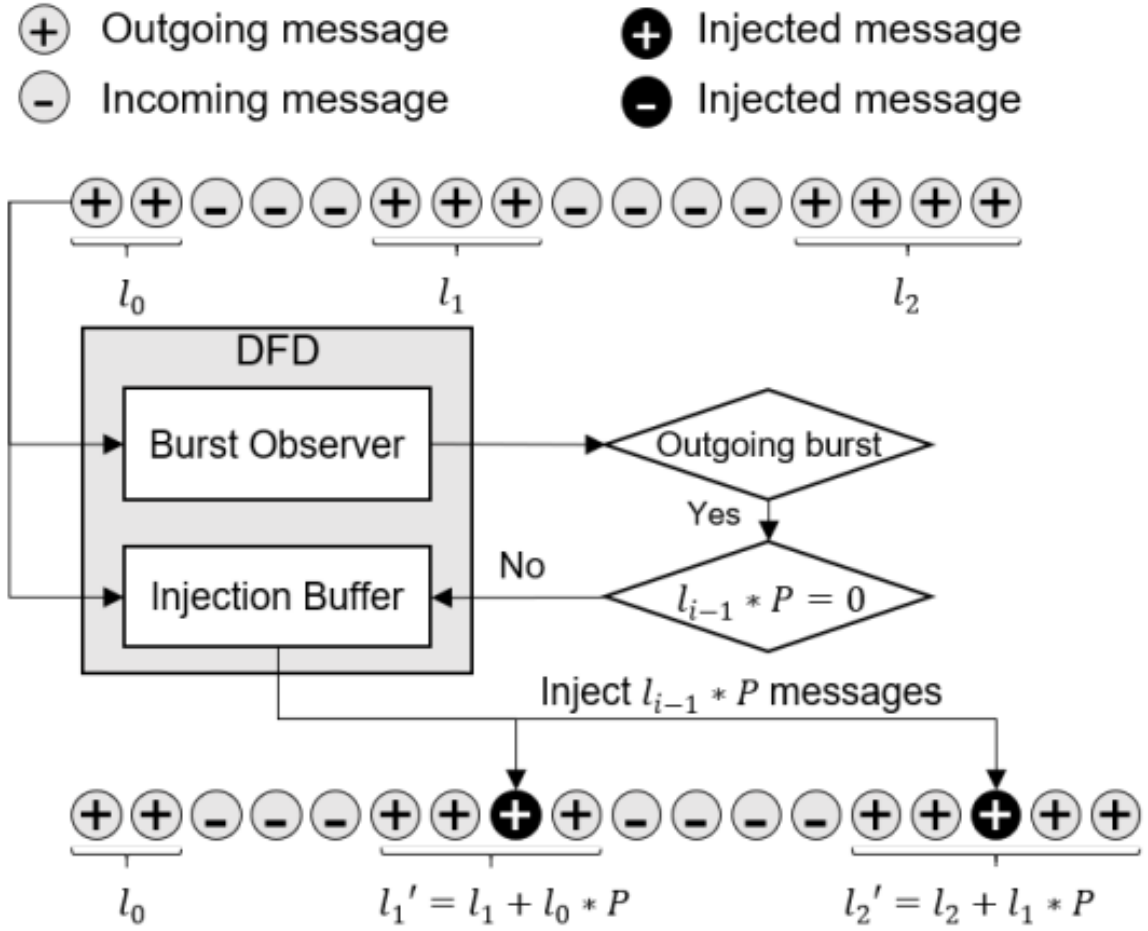
- 传统对抗样本生成方法
  - 快速梯度符号方法（FGSM）
  - 迭代快速梯度符号方法（IGSM）
  - 雅可比显着图攻击 (JSMA)
  - 基于优化的方法
  - Carlini & Wagner (C&W)

- 利用传统方法生成的样本能否成功混淆WF攻击者的分类器？-不可以
  - 在无对抗训练的场景下，运用传统对抗样本生成方法能达到很好的效果
  - 现实情况下，对手可以对分类器进行对抗训练，所以传统方法无效

基本概念

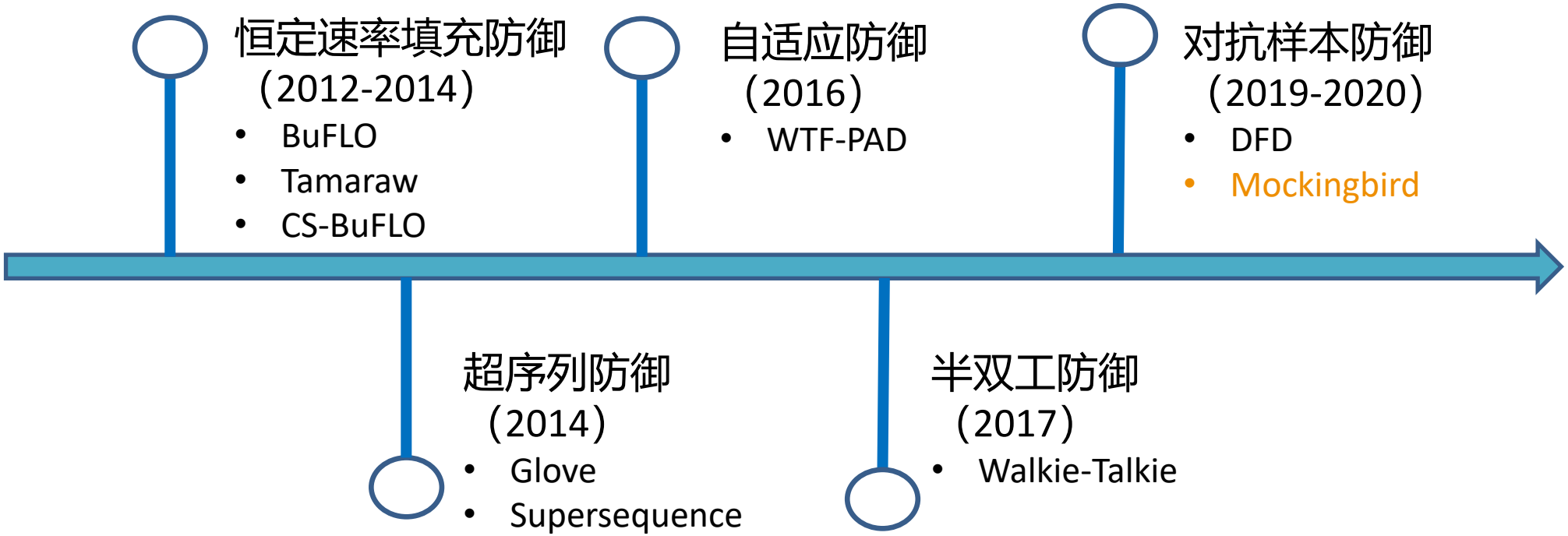
- 数据表示
  - 建模：流量trace - 传入(服务器到客户端)和传出(客户端到服务器)的突变序列
  - 突变序列：相同方向（传入或传出）的连续数据包序列
  - 字符表示：
    - $I_s = [l_0, l_1, \dots, l_n]$
    - $\Delta = [\delta_0, \delta_1, \dots, \delta_n]$
  - 例子：DFD





基本概念

- WF防御发展趋势



- 缺陷：带宽和延迟开销昂贵、对深度学习分类器脆弱等

算法原理

算法原理 Mockingbird

T	混淆对手的流量trace分类器
I	网站流量trace

P	从目标池中选出最接近源样本的样本
	For {
	1. 计算扰动向量
	2. 扰动向量附加于源trace样本
	3. 计算分类器对源trace样本的置信度和标签
	}

O	对抗性流量trace

P	无针对性的网站流量trace对抗样本生成
C	对手分类器掌握一定的对抗学习能力
D	生成更稳健的对抗性trace
L	IEEE Transactions on Information Forensics and Security 2020 1区

算法原理 Mockingbird

- 根本目的：保护源trace（敏感站点S）

- 思路：
  - 源trace I_s → I'_s，使 $f(I'_s) = t$，$t \neq s$

- 特点：
  - 不关注损失函数（如FGSM）
  - 使用优化方法会导致在搜索空间中遵循一组更可预测的路径

  - 仅旨在减少trace与目标trace的距离
  - 遵循直线到达不可预测的目标样本

算法原理 Mockingbird

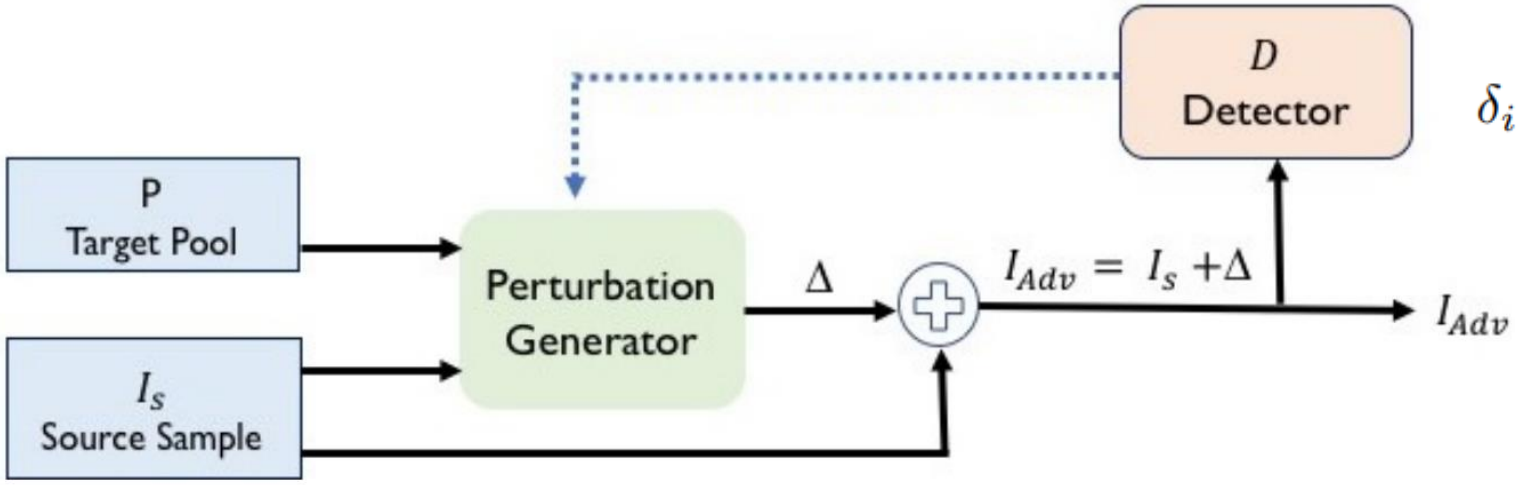
- 步骤
  - 源trace S，源样本 I_s，目标池 P_s
  - 从 P_s中选出最接近 S 的目标样本 I_t
  - 计算扰动向量 Δ
  - Δ 附加于 I_s 以接近 I_t
  - 当检测器输出错误判别结果，则停止

$$I_t = \underset{I \in P_s}{\operatorname{argmin}} D(I_s, I)$$

$$D(x, y) = l_2(x - y)$$

$$\Delta = [\delta_0, \delta_1, \cdots, \delta_n] \quad (\delta_i >= 0)$$

$$\nabla(-D(I, I_T)) = -\frac{\partial D(I, I_T)}{\partial I} = \left[-\frac{\partial D(I, I_T)}{\partial b_i}\right]_{i \in [0, \cdots, n]} \quad (5$$



$$\delta_i = \begin{cases} -\alpha \times \frac{\partial D(I, I_T)}{\partial b_i} & -\frac{\partial D(I, I_T)}{\partial b_i} > 0 \\ 0 & -\frac{\partial D(I, I_T)}{\partial b_i} \leqslant 0 \end{cases}$$

$$I_s^{new} = I_s + \Delta$$

算法原理 实验结果

- 实验结果对比
  - 白盒/黑盒条件下的带宽开销和准确率（越低越好）
    - 带宽开销：（case1/2，FD/HD）比其他方法低
    - 白盒：普遍效果很好
    - （top-k）准确率
      - 分类器推测某trace对应排名第k的站点
    - 传统ML攻击：效果更好

Cases	Dataset	BWO	DF [9]
Case I	FD	0.56	0.35
	HD	0.63	0.35
Case II	FD	0.56	0.55
	HD	0.73	0.29

Case	Dataset	BWO	DF [9]	Var-CNN [8]	CUMUL [22]	*k*-FP [23]	*k*-NN [21]	DF Top-2	Var-CNN Top-2
	Undefended (FD)	-	0.97	0.98	0.93	0.85	0.86	-	-
	Undefended (HD)	-	0.98	0.99	0.92	0.92	0.90	-	-
	WTF-PAD [14]	0.64	0.86	0.90	0.55	0.44	0.17	0.92	0.95
	W-T [15]	0.72	0.40	0.44	0.36	0.30	0.35	0.97	0.94
Case I	*Mockingbird* (FD)	0.58	0.42	0.35	0.19	0.21	0.08	0.56	0.50
	Mockingbird (HD)	0.62	0.41	0.33	0.22	0.28	0.10	0.57	0.47
Case II	*Mockingbird* (FD)	0.58	0.58	0.62	0.32	0.32	0.14	0.70	0.72
	Mockingbird (HD)	0.70	0.38	0.30	0.20	0.26	0.12	0.54	0.43

算法原理 优劣分析

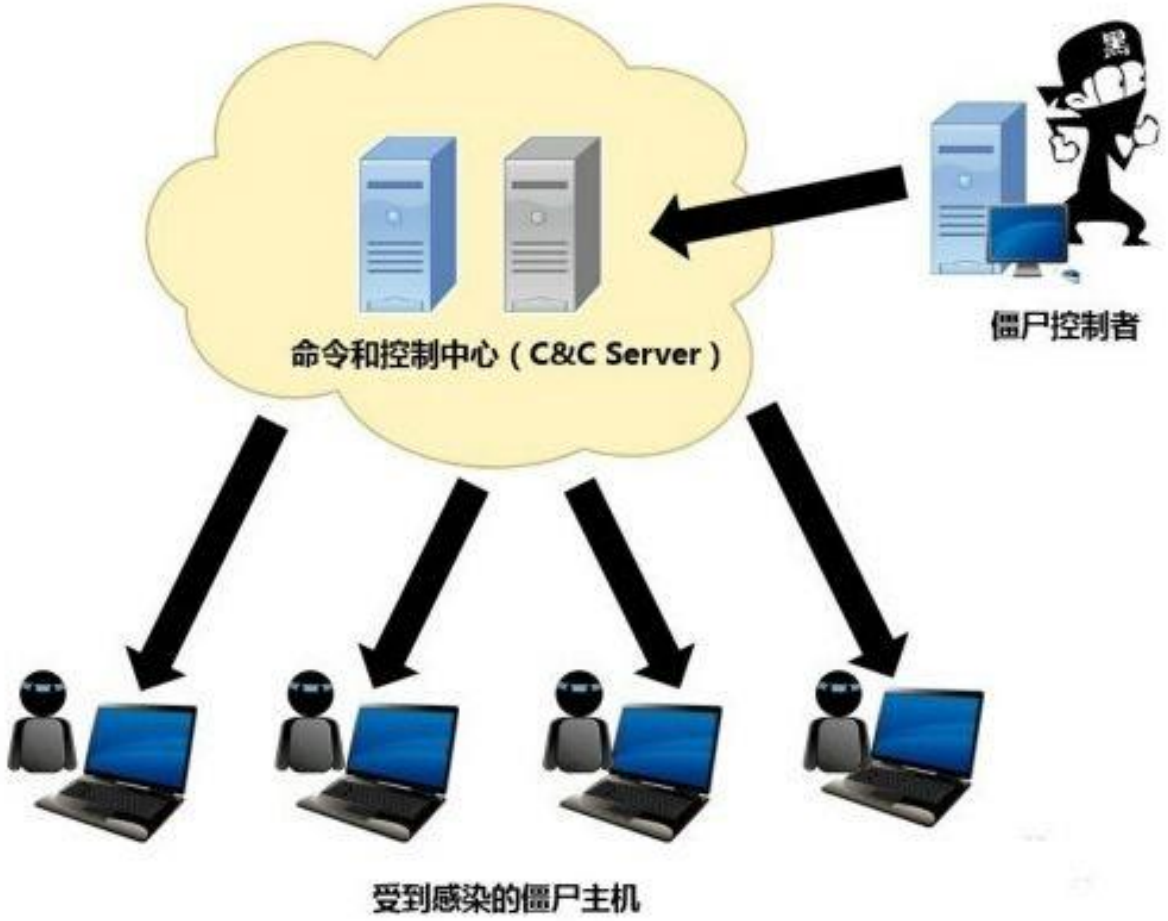
- 优势
  - 出色的 Top-k 混淆能力
    - W-T：两个站点混淆
    - WTF-PAD：相似站点混淆
    - Mockingbird：找到新站点
  - 普遍开销较低

- 缺陷
  - 可拓展性
  - 无法实时逐包生成对抗性trace
  - 计算量需求大

域生成算法

背景介绍

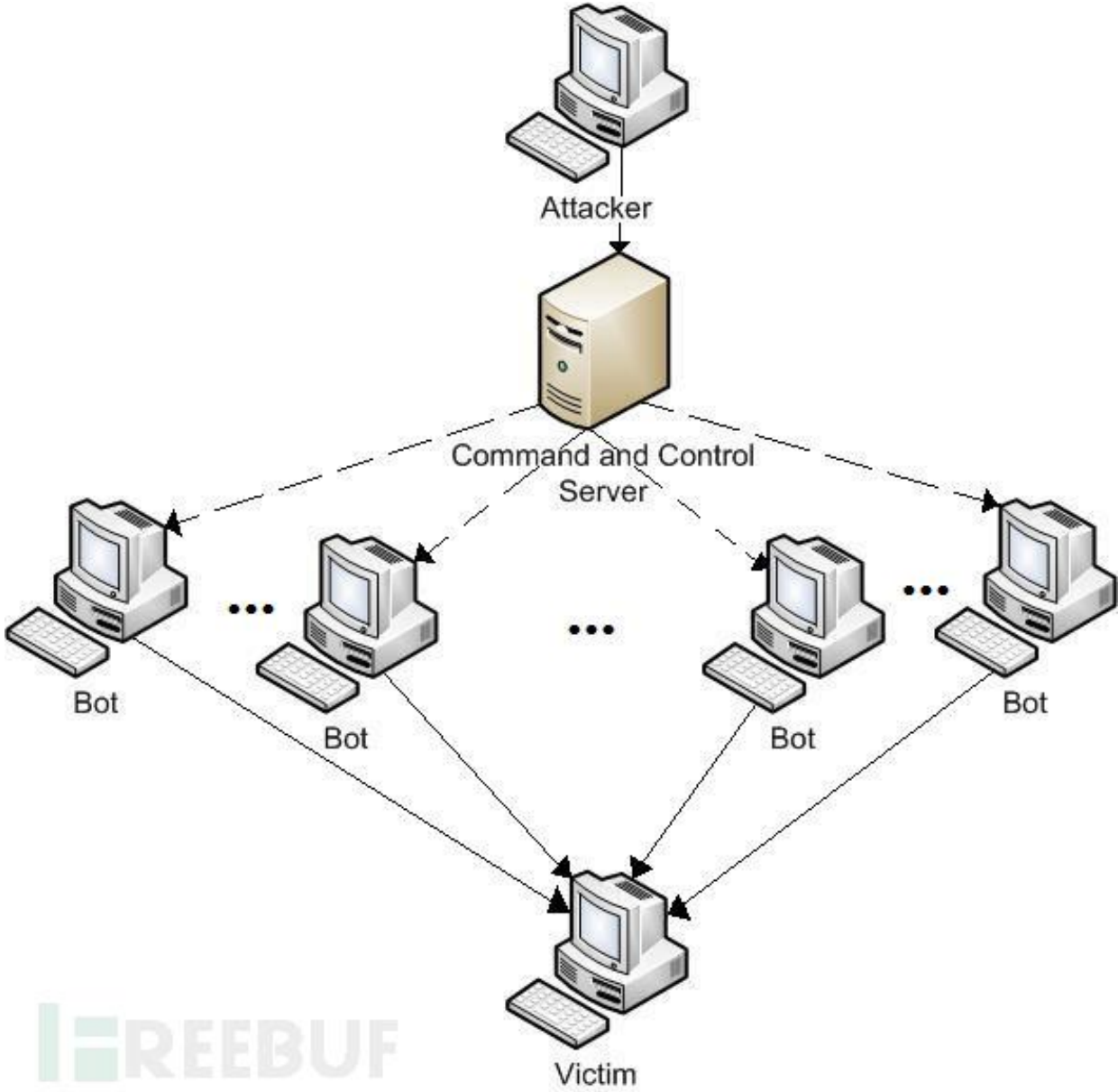
- 僵尸网络
  - 控制中枢：第三方（botmaster）可以通过命令和控制（C&C）服务器控制受感染的机器
  - 目标：隐藏C&C服务器地址
  - 方法：域通量（domain fluxing）



- 域通量
  - 不断更改僵尸网络C&C服务器的域名
  - 关键：域生成算法（DGA）

背景介绍

- DGA
  - 攻击者
    - 目的：不断改变C&C服务器的域名
    - Botmaster
      - 生成带有种子的域名
      - 注册尚未注册的域名
      - 映射到C&C服务器的IP地址
    - Bot
      - 恶意软件生成相同的域名
      - 逐个查询，若其中一个域名被成功解析，恶意软件就能够连接到 C&C 服务器
  - 防御者
    - 预测 C&C 服务器的域名
    - 沉没域名



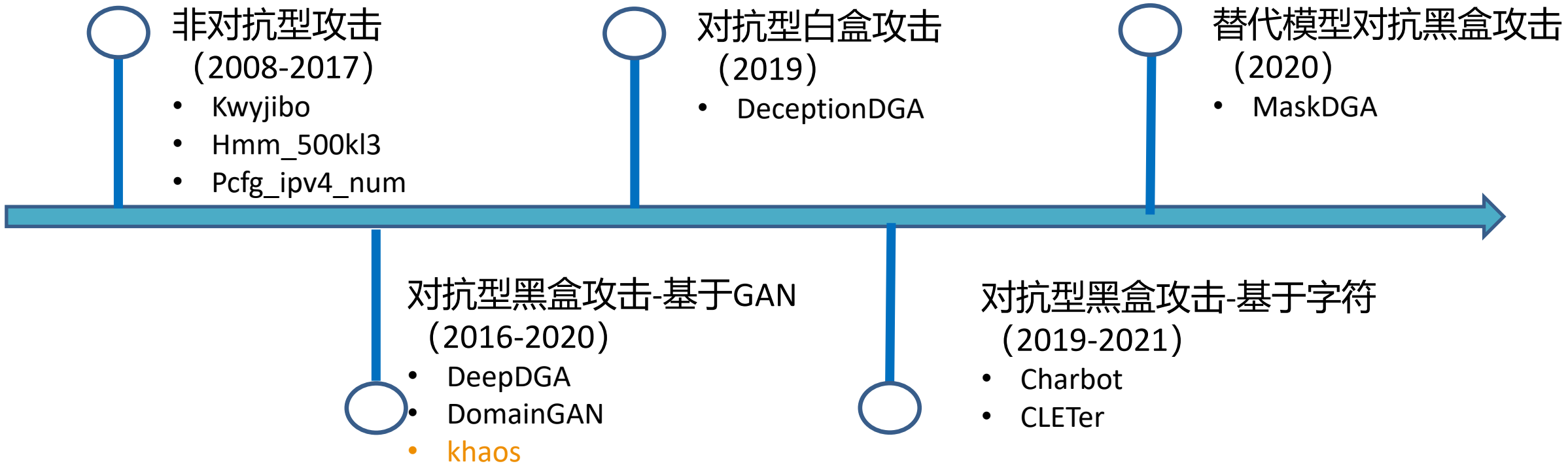
基本概念

- 域名
  - 顶级域（TLD）、顶级域标签
  - 二级域（2LD）、二级域标签
  - 三级域（3LD）、三级域标签
  - 例子：book.example.com
- 域名生成重点：二级域名标签

- N-gram
  - 基本思想：将文本内容按照字节进行大小为N的滑动窗口操作，形成字节片段
  - 向量特征空间：统计所有字节片段出现频度，形成关键字节片段列表

基本概念

- DGA攻击发展趋势



- 缺陷：只能生成固定长度、原始GAN训练困难

算法原理

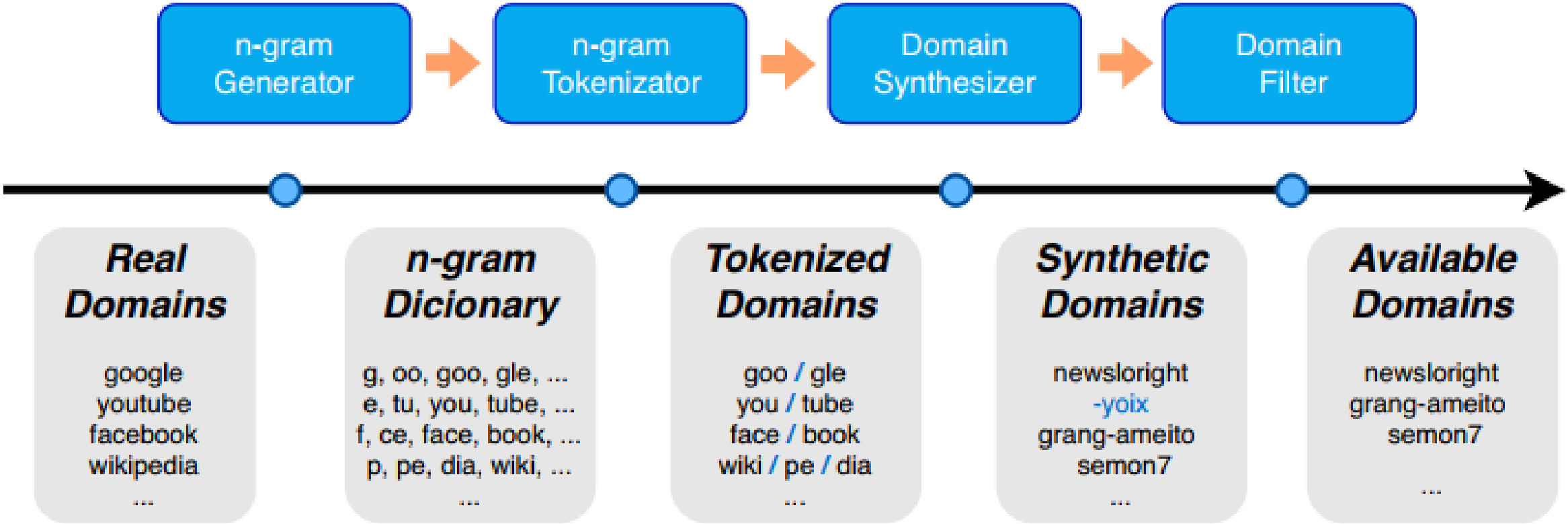
算法原理 khaos

T	混淆DGA检测器
I	真实域名
P	1. 提取n-gram构建字典 2. 根据字典拆分真实域名 3. 排列n-gram生成新的域名 4. 过滤不符合规定或冲突域名
O	与真实域名相似的混淆域名

P	利用真实域名生成混淆域名绕过DGA检测器
C	攻击者不了解DGA检测器的知识（黑盒）
D	不可能收集所有音节和首字母缩略词
L	IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY 1区

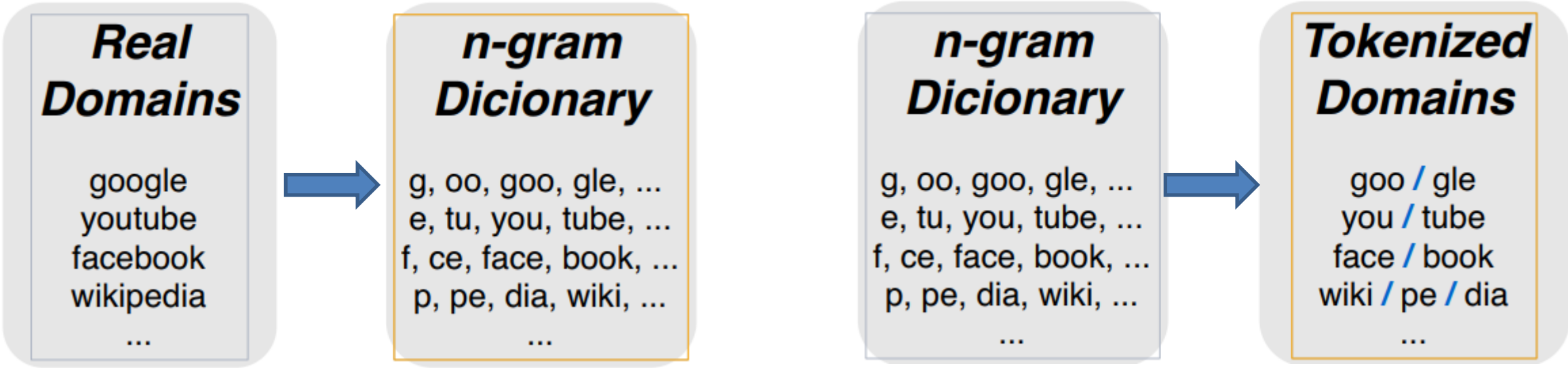
算法原理 khaos

- 基于神经语言模型和wasserstein生成对抗网络（WGAN）
- 域名由音节或首字母缩略词组成（facebook、wikipedia、bbc）
- 关键：将真实域名中音节和首字母缩略词替换为n-gram



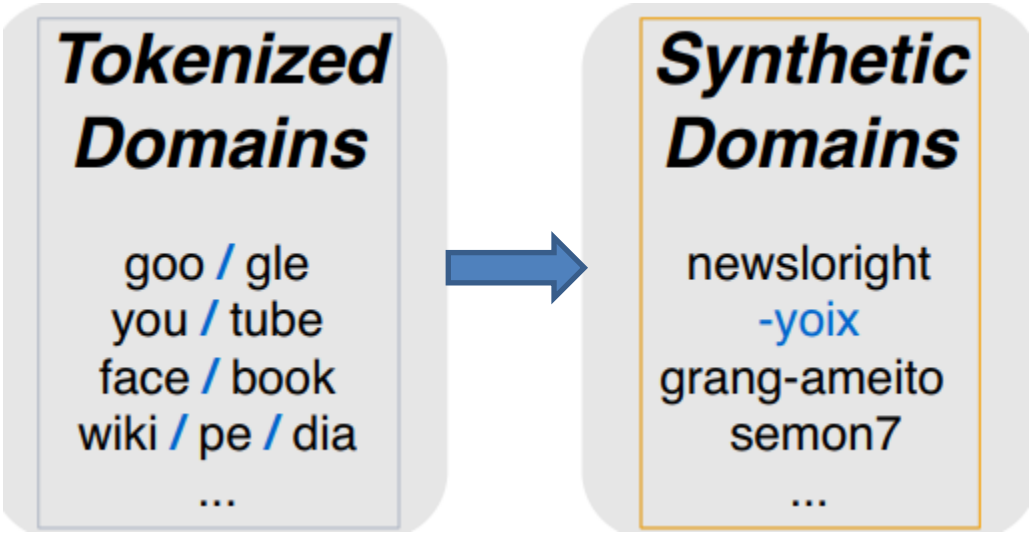
算法原理 khaos

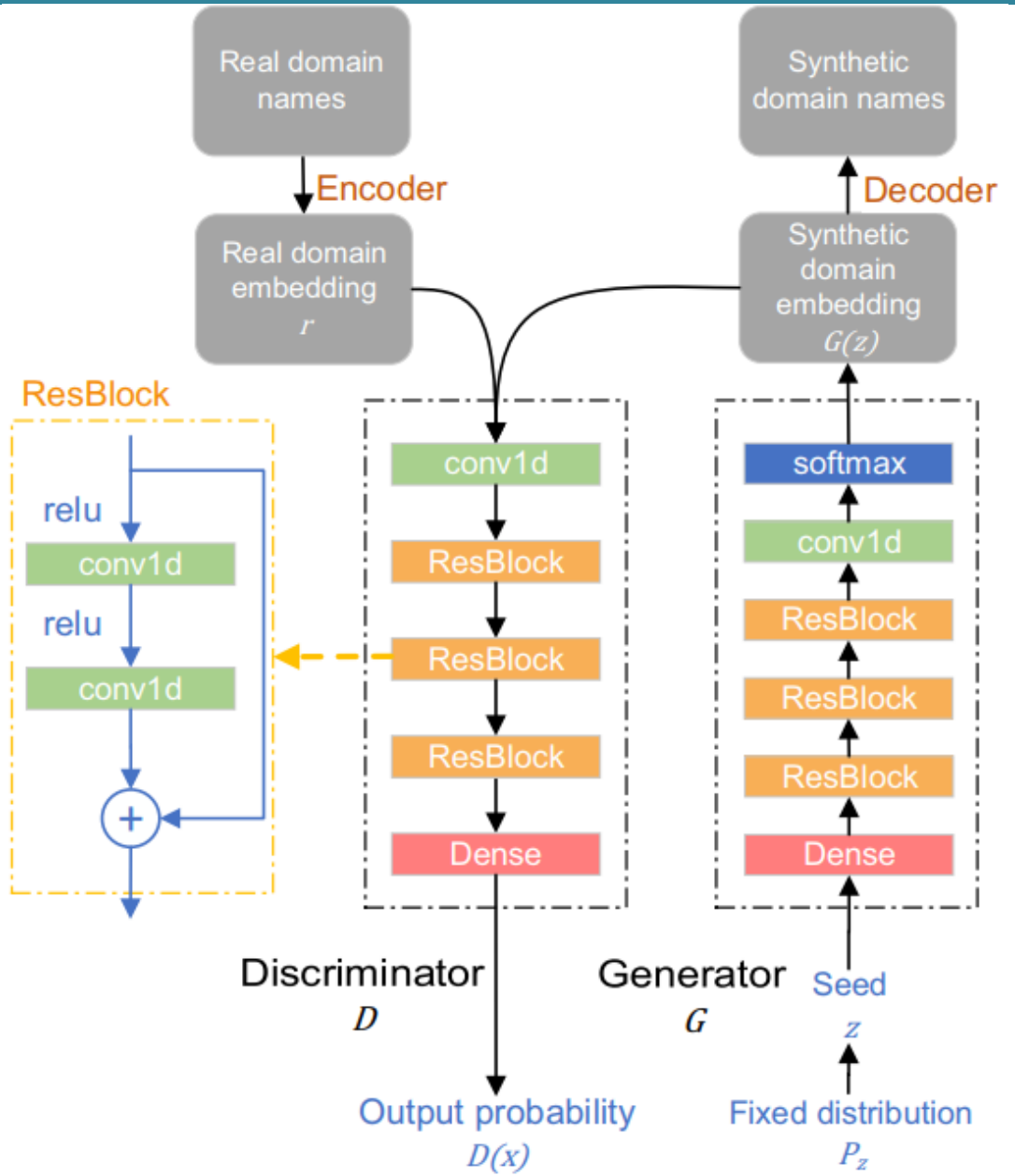
- n-gram Generator
  - 从真实域名中提取n-gram（n=1,2,3,4）
  - 按照出现频率对n-gram进行排序
  - 将前5000个n-gram构建为词典
- n-gram Tokenizer
  - 根据字典将真实域名拆分为 n-gram
  - 优化规则
    - goo/gle和g/oo/gle
    - wiki/pe/dia和wiki/p/edia



算法原理 khaos

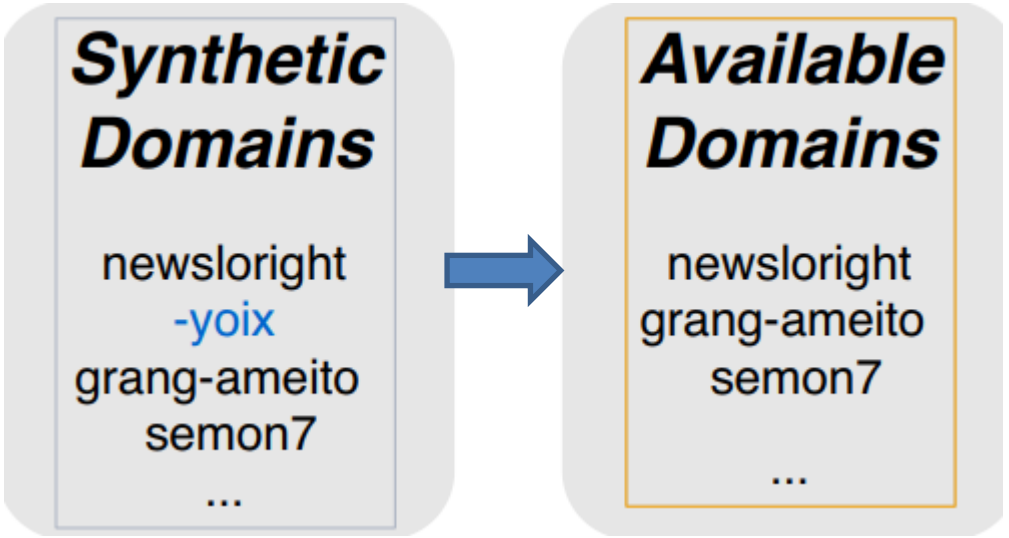
- Domain Synthesizer（核心）
  - 根据学习的规则排列 n-gram 生成新的域名
  - 组成
    - WGAN（G D）
    - 编码器：域名转换为域嵌入的向量
    - 解码器：将域嵌入转换为域名





算法原理 khaos

- Domain Filter
  - 过滤不符合命名约定的域名
    - RFC 1034 和 RFC 1035 规范
  - 过滤可能与现有域名冲突的域名
    - 丢弃长度小于 3 的域名
    - 已经注册



- 生成的不同长度域名的AUC

Length	3	4	5	6	7	8	9	10	All
AUC	0.53	0.63	0.70	0.71	0.76	0.73	0.76	0.75	0.76

算法原理 实验结果

- 基于统计和基于 LSTM 的检测方法下 DGA 的 AUC
  - 没有一个 DGA 可以完全混淆检测
  - Khaos效果相对较好
  - 对于khaos，基于统计的DGA检测器性能好

DGA	Statistics-based	LSTM-based	Difference
Khaos	0.76	0.57	-0.19
hmm_500KL3	0.80	0.81	0.01
pcfg_ipv4_num	0.82	0.89	0.07
DeepDGA	0.93	0.98	0.05
Kraken_v1	0.97	0.99	0.02
Nymaim	0.97	0.99	0.02
Pykspa_precursor	0.98	1.00	0.02
Matsnu	0.92	0.96	0.04
Suppobox	0.90	0.98	0.08
Gozi	0.95	0.99	0.04

- 在基于图的检测方法下，DGA 的精度、召回率和 FPR
  - FPR：FP/(FP+TP) 越高越好
  - 检测器对抗Suppobox实现高性能
  - Khaos效果略好

DGA	Precision	Recall	FPR
Khaos	0.68	0.98	0.47
Matsnu	0.69	1.00	0.46
Gozi	0.69	1.00	0.46
Suppobox	0.99	0.99	0.01

算法原理 优劣分析

- 优势：
  - 可以生成与真实域名高度相似的域名
  - 相比于其他DGA方法效果逃逸检测效果更好
  - 克服原始GAN难以收敛的问题，容易训练

- 局限性：
  - 可能会生成不包含任何单词的长域名，这和现实世界中长域名大都是由单词组成的规则相违背，因此看起来不像是真正的域名。

应用总结

应用总结

- 对抗样本生成在网络安全领域的应用
  - 恶意软件攻击
  - 恶意流量攻击

  - 网站指纹防御
  - 恶意URL生成
  - 网络钓鱼攻击
  - 垃圾邮件

参考文献

[1] Yun X, Huang J, Wang Y, et al. Khaos: An Adversarial Neural Network DGA With High Anti-Detection Ability[J]. IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. 2020, 15: 2225-2240.

[2] Gould N, Nishiyama T, Kamiya K. Domain Generation Algorithm Detection Utilizing Model Hardening Through GAN-Generated Adversarial Examples[C]// International Workshop on Deployable Machine Learning for Security Defense. Springer, 2020: 84-101.

[3] Rahman M S, Imani M, Mathews N, et al. Mockingbird: Defending against deep-learning-based website fingerprinting attacks with adversarial traces[J]. IEEE Transactions on Information Forensics and Security. 2020, 16: 1594-1609.

道德经

大成若缺，其用不弊。
大盈若冲，其用不穷。
大直若屈。大巧若拙。
大辩若讷。静胜躁，寒胜热。清静为天下正。

谢谢！