**网络流水印注入工具软件v1.0**

**用**

**户**

**手**

**册**

目 录

[1 引言 4](#_Toc505974121)

[1.1编写目的 4](#_Toc505974122)

[1.2软件概述 4](#_Toc505974123)

[2 系统需求 5](#_Toc505974124)

[2.1硬件需求 5](#_Toc505974125)

[2.2 软件需求 6](#_Toc505974126)

[3 方法简介 7](#_Toc505974127)

[4 操作说明 9](#_Toc505974128)

[4.1启动 9](#_Toc505974129)

[4.2 通过交互式界面指定参数 10](#_Toc505974130)

[4.2.1 选择网络流水印分类 10](#_Toc505974131)

[4.2.2 选择网络流水印方法 11](#_Toc505974132)

[4.2.3 指定目的IP 11](#_Toc505974133)

[4.2.4 指定目的端口 11](#_Toc505974134)

[4.3 命令行参数配置方法 12](#_Toc505974135)

[4.3.1 命令行参数-t <type index> 12](#_Toc505974136)

[4.3.2 命令行参数-m <method index> 12](#_Toc505974137)

[4.3.3 命令行参数-h <target IP> 13](#_Toc505974138)

[4.3.4 命令行参数-p <target port> 13](#_Toc505974139)

[4.3.5 命令行参数-f <file name> 14](#_Toc505974140)

[4.3.6 命令行参数-d [file name] 14](#_Toc505974141)

[4.3.7 命令行参数-s 15](#_Toc505974142)

[4.3.8 命令行参数—help 15](#_Toc505974143)

[4.4 注入网络流水印 15](#_Toc505974144)

[4.5各种网络流水印方法的使用 17](#_Toc505974145)

[4.5.1 IBW方法 17](#_Toc505974146)

[4.5.2 ICBW方法 17](#_Toc505974147)

[4.5.3 IPD方法 18](#_Toc505974148)

[4.5.4 DSSS方法 18](#_Toc505974149)

[4.5.5 Long-DSSS方法 19](#_Toc505974150)

[4.5.6 MMARSSW方法 19](#_Toc505974151)

[4.5.7 ICBSSW方法 20](#_Toc505974152)

[5应用场景 21](#_Toc505974153)

[5.1. 应用场景 21](#_Toc505974154)

[5.2 效果分析 21](#_Toc505974155)

[6功能结构 24](#_Toc505974156)

[附录：命令行参数使用的样例 25](#_Toc505974157)

[参考文献 26](#_Toc505974158)

# 1 引言

## 1.1编写目的

网络流水印作为一种主动的流量分析手段，能有效追踪恶意匿名通信的用户与跳板机链后的真实攻击者，具有准确率高、误报率低和测试时间短等优点。因而在攻击源追踪、网络监管和攻击取证等领域受到广泛应用。网络流水印注入工具软件v1.0，方便用户向目标网络流注入网络流水印（共支持七种典型的网络流水印方法[1-7]），以达到流量跟踪等目的。本手册介绍了软件的功能和性能，详细说明使用步骤，并给出了实际操作的例子。

## 1.2软件概述

随着互联网的推广，人们的通信和信息获取也越来越依赖网络，各种网络安全事件层出不穷。网络攻击者经常使用一系列攻击手段，如ARP欺骗嗅探攻击、DDoS轰炸攻击、XSS蠕虫脚本攻击等破坏个人电脑、服务器和互联网上的信息，给个人和企业造成无法弥补的损失。为了隐藏真实的IP，防止被网络监管者追踪，网络攻击者还经常采用Tor和匿名代理（Shadowsocks）等工具实现匿名通信，对网络安全提出了严峻的挑战。因此，如何有效跟踪网络数据流进而确定攻击源，是网络安全和网络监管的重要课题。

传统的流量跟踪方法通过分析数据包的基本信息来追踪攻击源，例如IP包头部信息、数据包时间戳和大小等。这类方法往往依赖于网络基础设施的建设，需要分析大量网络数据，且追踪效率较低，难以有效地定位攻击源。随着攻击者对网络流干扰的手段越来越多，例如加密、数据包重组和添加垃圾数据包等，通过IP追踪真正的攻击者变得困难。因此需要设计一种主动有效的流量追踪方法。

网络流水印通过调制时间特征和其它特征将水印信息注入到网络流，是一种新兴的主动的流量分析手段[8]。它通过添加特殊标记来跟踪网络流，模型如图1。水印注入者将水印注入到目标网络流中，携带水印信号的目标网络流会经过复杂的网络路径到达水印检测者。因此检测者获取的网络流同时携带水印信号和噪声，需从中检测水印，以达到流量追踪的目的。和传统的方法相比，网络流水印技术能够缩短追踪时间，减少分析的数据量，从而降低计算和通讯成本。

**水印注入者**

水印

目标网络流

携带水印

的网络流

**复杂网络路径**

携带水印和

噪声的网络流

**水印检测者**

目标网络流

水印检测

**图1 网络流水印技术模型**

具体实施可分为两个步骤：（1）当目标网络流在路由器中转时，水印注入者可以改变或调制它们的载荷、时间间隔和流速率信息来注入水印信息。由此每一条数据流都在发送者和接收者之间建立了唯一的联系；（2）水印检测者则负责检测对象中是否存在网络流水印，并从中提取有效信息。网络流水印注入工具软件V1.0为水印注入者（即用户）提供了很大的便利，可以便捷的向目标网络流注入七种不同的网络流水印：Interval-based Flow Watermarking (IBW水印)[1]、Interval Centroid Based Watermarking (ICBW水印)[2]、Inter-packet Delay (IPD水印)[3]、DSSS-based Flow Marking (DSSS水印)[4]、Long PN Code Based DSSS Watermarking (Long-DSSS水印)[5]、Multi-flow Attacks Resistant Spread Spectrum Watermarks (MMAR-SSW水印) [6]和Interval Centroid Based Spread Spectrum Watermarking (ICBSSW水印)[7]。

# 2 系统需求

## 2.1硬件需求

本软件操作便捷，并且对硬件要求较低。计算机具体要求如下：

* 具有Pentium IV或更高版本的处理器
* 最低1GB内存
* 最小20 GB硬盘
* 以太网适配器或无线网络适配器（网卡）
* 鼠标和键盘

## 2.2 软件需求

本软件可以运行于两种常见的操作系统。若使用Linux操作系统，则要求系统为Centos 7.0或更高版本；若使用Windows操作系统，则要求系统为Windows 7或更高版本。需要预装的工具软件包括：

Python 3及以上版本，具体详见<https://www.python.org/>；

Tcpdump（只支持Linux操作系统），具体详见<http://www.tcpdump.org/>。

# 3 方法简介

本软件支持的七种典型网络流水印方法，可分为三大类：

一、时间间隔基础类（Interval-based），包括以下三种方法。

* IBW方法[1]：首先以某时间偏移量offset为起点，按照固定的时间间隔*T*切分数据流。注入水印时，随机选取*r*个连续的时间间隔对*<Ti, Ti+1>*，(*i=1,2…r*)，并将间隔*Ti*内的数据包通过延迟的方式填充到*Ti+1*中。因此，*Ti*间隔内的数据包被清空，*Ti+1*间隔内的数据包以翻倍的方式注入水印信号，每个信号冗余到*r*对时间间隔内。
* ICBW方法[2]：随机选取一个时间偏移量offset，并以offset为起点，从数据流中切割出*2\*n*个时间间隔为*T*的数据片段。然后，将*2\*n*个数据片段随机分成A和B两组，每组各包含*n*个间隔*T*的数据片段。接着，再次将*n*个间隔*T*的数据片段，随机分割成*l*组，每组包含*r=n/l*个数据片段。这样水印信号可以通过一对分别在A和B组中的*r*个数据片段来注入。计算*r*个时间间隔*T*内所有数据包的平均质心，然后通过移动数据包的时间下标来改变A和B组对应的平均间隔质心。规定A和B间隔质心的差值大于零或者小于零，用来分别注入水印信号1和0。例如，*Ai*和*Bi* (*i=1,2…l*)分别代表了A组和B组内第*i*个小分组对应的平均质心，*Yi=Ai-Bi*的正负性可以通过调整*Ai*和*Bi*来改变，并由此完成水印信号1（即*Yi>0*）和水印信号0（即*Yi<0*）的注入。而要改变*Ai*或者*Bi*则需改变*r*个时间间隔*T*内所有数据包的时间下标。例如，延迟*r*个时间间隔内的数据包，增大他们的时间下标，可以提高*Ai*的平均质心。
* IPD方法[3]：选取一段包含*n*个数据包的数据流，将*n*个数据包等分成*m*个组，每个组内包含*r*个数据包（*r=4s, s=1, 2……*），组内数据包对应的时间下标为*ti*（*i=1,2…r*）。将每个组内的数据包切分成*2s*个数据包对*<Pi, Pi+2s>* ，其时间下标为<*ti, ti+2s*>，则计算*IPDi=ti+2s-ti*。由此可以得到*2s*个IPD数值并随机均分成两组，每组平均值记为*IPDavg1*和*IPDavg2*。在注入水印时，只需根据水印信息位是0还是1，来调整两个IPD的平均值差值，*IPDdiff=IPDavg1-IPDavg2*。例如，当注入“1”时要求*IPDdiff>0*，因此需要增大*IPDavg1*。主要做法是延迟*IPDavg1*对应的*s*个包对的数据包，即使得数据包的时间*ti*增大。

二、直序扩频基础类（DSSS-based）

DSSS-based类方法的基本流程是：首先将原始水印信号通过PN码进行扩频，然后通过调整数据流（如恒定速率数据流，CBR流）的流速来注入网络流水印。注入强信号“1”时需提高流速率，可通过在短时间内填充大量数据包的方式来提高速率。当注入弱信号“0”时需采取相反的操作，例如在短时间内仅填充少量的数据包。

根据对原始信号扩频方式的不同，这类方法包括DSSS[4]、Long-DSSS[5]和MMAR-SSW[6]。DSSS方法中原始信号位“0”和“1”对应的PN码是固定的，所以很容易被均方差自相关检测（Mean-square autocorrelation，MSAC）攻击。Long-DSSS方法为了克服这个问题，使用了Long PN Code，使得每一个原始信号对应的PN码都不相同。这就有效避免了DSSS方法中常出现的网络流速率呈现周期性波峰波谷的现象，减少被攻击的概率。MMAR-SSW采用了正交PN码（Orthogonal PN Code），使得扩频后的水印信号注入位置不再是连续的而是随机选取的，更好的消除了网络流水印的周期性，可以进一步提高网络流水印的鲁棒性以抵抗MSAC和多流攻击。

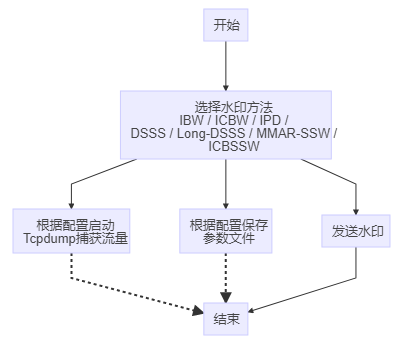
三、混合类（Hybrid）

ICBSSW方法[7]结合了DSSS和ICBW方法，主要利用DSSS-based类的PN码扩频的方式来扩展ICBW的信号。它先将原始信号使用PN码扩展后再使用ICBW方法注入扩展后的信号，即在原先ICBW方法的前端加上一个PN码编码器。因此，该方法继承了DSSS-based网络流水印方法使用PN码带来信息冗余的特性。

# 4操作说明

本软件的操作简单，其工作流程如图2所示。用户首先启动本软件（参见4.1），选择一个网络流水印方法。然后设置相应的方法参数，可采用两种方式：（1）通过交互式界面（Interactive Shell），根据提示依次输入方法参数（参见4.2）；（2）通过命令行直接指定部分或全部的参数（参见4.3）。确定方法和相关参数后，用户通过交互式界面输入待注入的水印信息，并完成注入。

为了验证网络流水印的注入是否成功（即水印是否存在），本软件会将选中方法在解密检测时所需参数以json格式存储到文件中，例如随机种子和DSSS方法生成的PN码[[1]](#endnote-1)等。



**图2 网络流水印注入工作流程**

## 4.1启动

假设本软件的所有文件放置在某个合适的目录下，以下称为*<ROOT>*。启动本软件的方法如下：

1. 启动一个终端；
2. 执行shell命令：*cd <ROOT>*
3. 执行shell命令：*python3 wms.py*

（备注：用户可以通过命令行方式在步骤（3）启动软件的同时设置相关参数，具体方法请参见4.3节，操作样例请参见附录。）

## 4.2 通过交互式界面指定参数

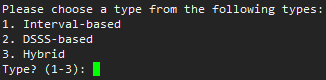
启动后，若用户没有通过命令行给定所需的参数，则进入交互界面，提示用户输入所需参数。若输入的参数不合法，本软件会提示用户重新输入。输入参数的顺序为：

1. 水印分类编号；
2. 选定分类中的水印方法编号；
3. 目标网络流（水印注入对象）的目的IP地址；
4. 目标网络流（水印注入对象）的目的端口。

当所需参数设置完毕后，交互式界面将提示用户输入待注入的水印信息。用户也可以通过重定向的方法从指定的文件中读取待注入的水印信息。（备注：若用户已经通过命令行完成某个参数的设置，则交互式界面将不会再出现该参数的设置提示。）

### 4.2.1 选择网络流水印分类

本软件支持的七种典型网络流水印方法可以分为三类，具体信息见表1。用户可以根据需要选择使用哪一类网络流水印，如图3的界面提示用户输入网络流水印分类的编号。用户要选择哪一个分类，请直接输入对应的分类编号。



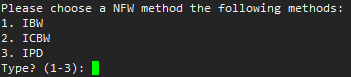
**图3 网络流水印分类选择的界面**

|  |  |  |  |
| --- | --- | --- | --- |
| 类别编号 | 类别名称 | 方法编号 | 方法名称 |
| 1 | 时间间隔基础类（Interval-based） | 1 | IBW[1] |
| 2 | ICBW[2] |
| 3 | IPD[3] |
| 2 | 直序扩频基础类（DSSS-based） | 1 | Long-DSSS[5] |
| 2 | MMARSSW[6] |
| 3 | DSSS[4] |
| 3 | 混合类（Hybrid） | 1 | ICBSSW[7] |

**表1 网络流水印的分类列表**

### 4.2.2 选择网络流水印方法

选定方法类别后，本软件会列出该类别下的网络流水印方法（参见表1），并提示用户输入网络流水印方法的编号。用户只需要输入被选中方法的编号即可。例如图4，第一步选择了分类1（时间间隔基础类），软件提示用户选择此分类下的三种方法IBW、ICBW和IPD。



**图4 网络流水印方法选择的界面**

### 4.2.3 指定目的IP

本软件提示用户输入一个目的IP地址，如图5。用户只需要输入目标网络流（水印注入对象）的目的IP地址，并回车即可。



**图5 指定目的IP**

### 4.2.4 指定目的端口

本软件提示用户输入一个目的端口，如图6。用户只需输入目标网络流（水印注入对象）的目的端口即可（备注：合法的端口数是0~65535）。



**图6 指定目的端口**

## 4.3 命令行参数配置方法

本软件提供了通过命令行设置参数的快捷操作，所支持的命令行参数选项有：（1）*-t <type index>*；（2）*-m <method index>*；（3）*-h <target IP>*；（4）*-p <target port>*；（5）*-f <file name>*；（6）*-d [file name]*；（7）*-s*；（8）*--help*。

使用语法为：python3 wms.py [options...]。本节将逐一说明这些命令行参数的作用和使用方法。

### 4.3.1 命令行参数-t <type index>

-t指定并选择对应编号的网络流水印类别。<type index>是选中的网络流水印分类编号（参见表1）。若参数<type index>无效，则进入交互式界面提示用户重新输入合法的分类编号。

示例：

1. 选择时间间隔基础的水印类别：

*python3 wms.py -t 1*

1. 选择直序扩频基础的水印类别：

*python3 wms.py -t 2*

### 4.3.2 命令行参数-m <method index>

-m指定并选择某一类别中网络流水印方法的编号。<method index>是选中的网络流水印方法编号（参见表1）。此命令行参数通常会配合*-t <type index>*使用（参见4.3.1）。通过这两个命令行参数可以唯一地指定一个网络流水印方法。若参数<type index>无效，则进入交互式界面提示用户重新输入合法的方法编号。

示例：

1. 选择时间间隔基础的水印类别中的IBW方法：

*python3 wms.py -t 1 -m 1*

1. 选择直序扩频基础的水印类别中的DSSS方法：

*python3 wms.py -t 2 -m 3*

### 4.3.3 命令行参数-h <target IP>

-h设置目标网络流（水印注入对象）的目的IP地址。若<target IP>不合法，则进入交互式界面提示用户重新输入一个合法的IP地址。

示例：指定水印注入的目的IP为114.114.114.114：

*python3 wms.py -h 114.114.114.114*

### 4.3.4 命令行参数-p <target port>

-p设置目标网络流（水印注入对象）的目的端口（取值范围：0~65535）。此命令行参数需配合*-h <target IP>*（参见4.3.3）使用。通过这两个命令行参数可以唯一地指定一个通信接口。若<target port>无效，则进入交互式界面提示用户重新输入一个合法的端口号。

示例：指定目标网络流的目的IP地址为114.114.114.114，端口port为53：

*python3 wms.py -h 114.114.114.114 -p 53*

### 4.3.5 命令行参数-f <file name>

注入网络流水印后，本软件将解密此水印所需信息保存到一个json文件中。通过此命令行参数-f能够指定保存的json文件的文件名（不含扩展名“.json”），支持相对路径和绝对路径。若未指定文件名或指定文件名无效，本软件将会使用默认的文件名，并将此文件保存于当前工作目录。默认的文件名格式为：*year-month-day\_rand.json*，其中“year-month-day”为当前系统时间的年、月、日以“-”连接的形式，“rand”为8位随机的16进制数。

示例：

1. 保存解密水印所需信息到当前工作目录下的文件“info.json”中：

*python3 wms.py -f info*

1. 保存解密水印所需信息到“/home/pi” 目录下的文件“info.json”中：

*python3 wms.py -f /home/pi/info*

### 4.3.6 命令行参数-d [file name]

本软件可以在水印注入过程中启用Tcpdump捕获流量，并保存到指定的文件中，[file name]为可选参数。若只设置了-d参数，本软件会使用默认的文件名，并将文件保存到当前工作目录下。默认的文件名格式为：*year-month-day\_rand.pcap*，其中“year-month-day”为当前系统时间的年、月、日以“-”连接的形式，“rand”为8位随机的16进制数。

本功能在Linux/Unix操作系统下有效，同时要求预装Tcpdump，并授予本软件超级用户的权限。在Windows系统中运行时，本软件将忽略此命令行参数。若想要在Windows下捕获流量，需手动运行Wireshark。

示例：

1. 在注入水印的同时启动Tcpdump捕获流量，并保存到当前工作目录下的capture.pcap文件中：

*python3 wms.py -d capture.pcap*

1. 在注入水印的同时启动Tcpdump捕获流量，并保存到目录“/home/pi”下的capture.pcap文件中：

*python3 wms.py -d /home/pi/capture.pcap*

### 4.3.7 命令行参数-s

本软件会默认地向标准输出流（终端或屏幕上）输出一些必要的提示信息。设置了-s安静模式后，本软件将不会向标准输出流输出任何提示信息。

示例：

1. 开启安静模式：

*python3 wms.py -s*

1. 不开启安静模式（正常启动）：

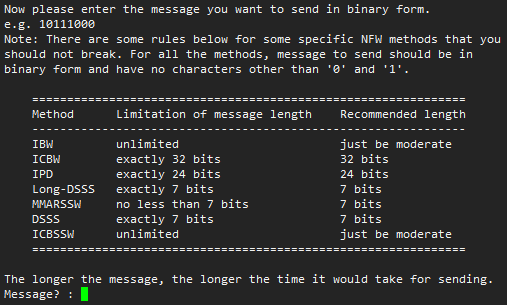
*python3 wms.py*

### 4.3.8 命令行参数—help

显示本软件的帮助信息。

## 4.4 注入网络流水印

完成注入水印所需参数的设定后，本软件将提示用户输入待注入的水印信息，如图7。需要注意的是，输入的水印信息必须是由“0”和“1”组成的二进制形式的字符串，且中间不能有其它无关的字符（如空格）。



**图7 输入待注入的网络流水印信息**

每种网络流水印方法对信息长度的限制不同，详情参见表2。水印信息越长，注入水印所耗费的时间就越长，请用户耐心等待。表3结合文献[1-7]和实验，列出各网络流水印方法的最优参数组合，以提高水印注入的性能。

|  |  |  |
| --- | --- | --- |
| **方法名称** | **信息长度限制** | **推荐长度** |
| IBW | 无 | 适中即可 |
| ICBW | 32位 | 32位 |
| IPD | 24位 | 24位 |
| Long-DSSS | 7位 | 7位 |
| MMARSSW | 不少于7位 | 7位 |
| DSSS | 7位 | 7位 |
| ICBSSW | 无限制 | 适中即可 |

**表2 网络流水印方法信息的长度限制**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **网络流水印方法** | **方法参数** | | | | | | |
| **网络流水印长度(bit)** | **时间间隔**  **(s)** | **偏移量** | **信号位**  **(bit)** | **PN码长度** | **码时间**  **(s)** | **符号时间**  **(s)** |
| ICBW | 32 | 600 | 2000 | / | / | / | / |
| IBW | 32 | 600 | 2000 | / | / | / | / |
| IPD | 32 | / | 10 | / | / | / | / |
| DSSS | / | / | / | 7 | 7 | 1 | 50 |
| Long-DSSS | / | / | / | 7 | 7 | 1 | 50 |
| MMAR-SSW | / | / | / | 7 | 7 | 1 | 50 |
| ICBSSW | / | / | / | 7 | 7 | 1 | 50 |

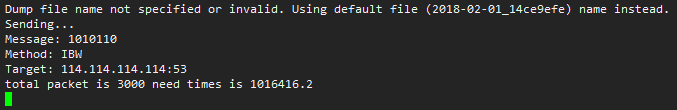
**表3 网络流水印方法的最优参数组合表**

输入水印信息后，本软件根据指定方法和相应参数向目标网络流注入网络流水印。水印注入完成后，本软件会将解密水印所需的参数保存至文件（详见4.3.5）。

## 4.5各种网络流水印方法的使用

### 4.5.1 IBW方法

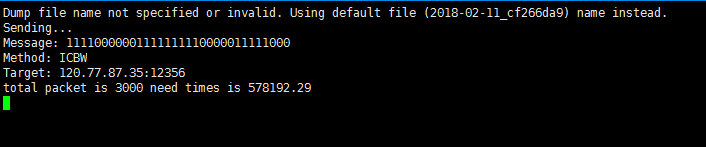
IBW方法是1号类别的1号方法。使用交互式界面选择水印方法时，需在选择类别时输入1，在选择方法时输入1；若使用命令行参数指定水印方法，需输入-t 1和-m 1命令行参数。该方法对水印长度没有限制，但水印信息越长，注入水印所需的时间越长。使用IBW方法注入水印的示例如图8所示。该例子的返回信息包括json文件名称（2018-02-01\_14ce9efe.json）、水印注入涉及的数据包数目（3000个）和发送完所有数据包耗时（1016416.2秒）。



**图8 IBW方法注入水印的界面**

### 4.5.2 ICBW方法

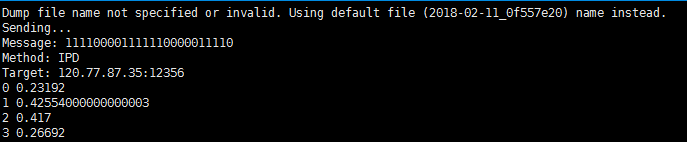
ICBW方法是1号类别的2号方法。使用交互式界面选择水印方法时，需在选择类别时输入1，在选择方法时输入2；若使用命令行参数指定水印方法，需输入-t 1和-m 2命令行参数。ICBW要求待注入的水印长度恰好为32位。使用ICBW方法注入水印的示例如图9所示。该例子的返回信息包括json文件名称（2018-02-11\_cf266da9.json）、水印信号（Message的内容）、目的地址和端口（Target的内容）、水印注入涉及的数据包数目（3000个）和耗时（578098.05秒）。



**图9 ICBW方法注入水印的界面**

### 4.5.3 IPD方法

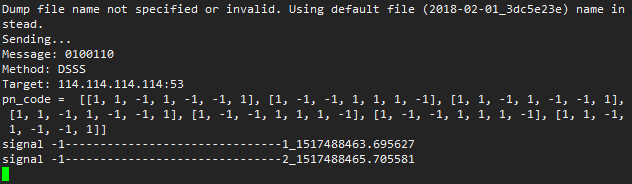
IPD方法是1号类别的3号方法。使用交互式界面选择水印方法时，需在选择类别时输入1，在选择方法时输入3；若使用命令行参数指定水印方法，需输入-t 1和-m 3命令行参数。IPD方法要求待注入的水印长度恰好为24位。使用IPD方法注入水印的示例如图10。该例子返回信息包括json文件名称（2018-02-10\_61a26b17.json）、水印信号（Message的内容）、目的地址（Target的内容），以及当前发送的数据包的标号和耗时。例如，3号数据包发送耗时0.26692秒。



**图10 IPD方法注入水印的界面**

### 4.5.4 DSSS方法

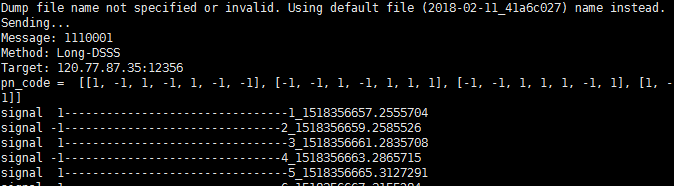
DSSS方法是2号类别的3号方法。使用交互式界面选择水印方法时，需在选择类别时输入2，在选择方法时输入3；若使用命令行参数指定水印方法，需输入-t 2和-m 3命令行参数。DSSS方法要求待注入的水印长度恰好为7位。使用DSSS方法注入水印的示例如图13。该例子返回信息包括json文件名称（2018-02-01\_3dc5e23e.json）、水印信号（Message的内容）、目的地址（Target的内容）、水印注入所使用的PN码（pn\_code矩阵），以及发送的编码信号位（signal）和发送的时间戳（信号位的发送间隙为1秒）。例如，第一个编码信号位是-1，它的发送时间戳是1517488463.695627秒。



**图13 DSSS方法注入水印的界面**

### 4.5.5 Long-DSSS方法

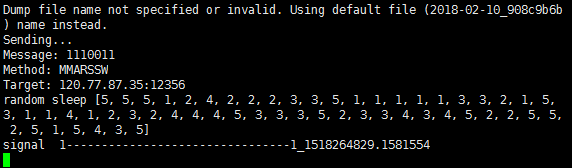
Long-DSSS方法是2号类别的1号方法。使用交互式界面选择水印方法时，需在选择类别时输入2，在选择方法时输入1；若使用命令行参数指定水印方法，需输入*-t 2*和*-m 1*命令行参数。Long-DSSS方法要求待注入的水印长度恰好为7位。使用Long-DSSS方法注入水印的示例如图11。该例子返回信息包括json文件名称（2018-02-01\_41a6c027.json）、水印信号（Message的内容）、目的地址（Target的内容）、水印注入所使用的PN码（pn\_code矩阵），以及发送的编码信号位（signal）和发送的时间戳（信号位的发送间隙为1秒）。例如，第一个编码信号位是1，它的发送时间戳是1518356657.2555704秒。



**图11 Long-DSSS方法注入水印**

### 4.5.6 MMARSSW方法

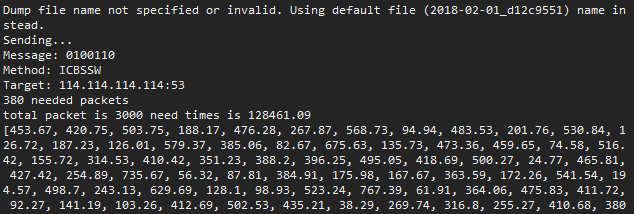
MMARSSW方法是2号类别的2号方法。使用交互式界面选择水印方法时，需在选择类别时输入2，在选择方法时输入2；若使用命令行参数指定水印方法，需输入*-t 2*和*-m 2*命令行参数。MMARSSW方法要求待注入的水印长度不少于7位；但水印信息越长，注入水印所需的时间越长。使用MMARSSW方法注入水印的示例如图12。该例子返回信息包括json文件名称（2018-02-10\_908c9b6b.json）、水印信号（Message的内容）、目的地址（Target的内容）、水印注入所使用的PN码（pn\_code矩阵），随机睡眠时间（random sleep矩阵，依次记录了水印信号位发送的间隙，即每个水印信号的嵌入位置是随机的）以及发送的编码信号位（signal）和发送的时间戳（信号位的发送间隙为1秒）。例如，第一个编码信号位是1，它的发送时间戳是1518264829.1581554秒。



**图12 MMARSSW方法注入水印**

### 4.5.7 ICBSSW方法

ICBSSW方法是3号类别的1号方法。使用交互式界面选择水印方法时，需在选择类别时输入3，在选择方法时输入1；若使用命令行参数指定水印方法，需输入*-t 3*和*-m 1*命令行参数。ICBSSW方法对水印长度没有限制；但水印越长，注入水印所需的时间越长。使用ICBSSW方法注入水印的示例如图13。该例子的返回信息包括json文件名称（2018-02-01\_d12c9551.json）、水印信号（Message的内容）、目的地址（Target的内容）、水印注入涉及的数据包数目（3000个）和总耗时（128461.09秒），以及水印信号对应的每个数据包的延迟时间（行向量[453.67, 420.75, …]，单位是毫秒）。



**图14 ICBSSW方法注入水印的界面**

# 5应用场景

## 5.1.应用场景

经过实验验证，本软件注入的网络流水印均是有效的。以图15的应用场景为例：一台主机为目标网络流的发送者，一台云端服务器为目标网络流的接收者；两者之间通过匿名服务器形成一条持续的有效的背景流（可视作噪音），背景流分为bulk-transfer和ssh流。用户（水印注入者）通过本工具软件向目标网络流注入网络流水印，水印检测者则在匿名服务器和接收者之间检测水印。



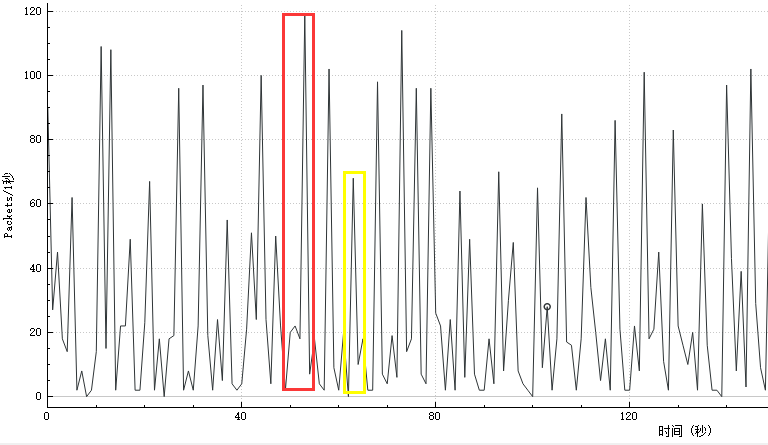
**图15 应用场景图**

为了保证实验的公平性，所有的测试都采用相同的背景流：（1）使用iperf3工具[9]，发送bulk-transfer流量，平均速率为2Mbps，每个包的大小为150字节；（2）使用从WIDE backbone[10]收集的ssh流，每组具有5000个以上的数据包，平均速率为2.89个数据包/秒，其中80％的数据包到达时间小于500ms。

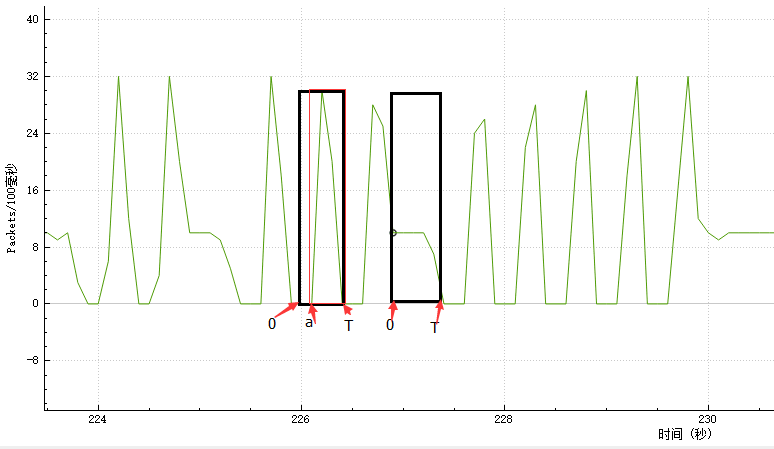
## 5.2 效果分析

本节以DSSS和ICBW两种网络流水印方法为例，验证本软件注入水印的有效性。为了观察网络速率随时间变化的关系，本软件启动的同时也开启Tcpdump捕获数据包，并根据所捕获的数据包作I/O图。I/O图显示网络速率随时间变化的关系，横轴为时间，纵轴为单位时间内捕获的数据包个数。

一般地，水印信号会隐藏于网络噪声中，不易观察。为便于展示，我们增强了水印信号的强度，所捕获的DSSS水印数据包的I/O图如图16所示。红色框出的部分是一位强信号“+1”通过在固定的时间段内注入大量的数据包，明显提升网络速率；黄色框出的部分是一位弱信号“-1”注入少量的数据包，对网络流速率产生的影响较小。



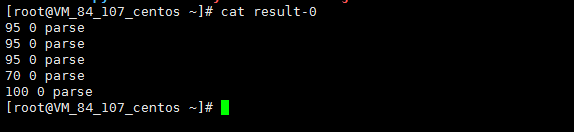
**图16 DSSS网络流水印信号（增强版）**



**图17 ICBW网络流水印信号**

图17为捕获的ICBW水印数据包的I/O图。图中的黑色矩形表示的范围为一个固定时间间隔*T*，其中部分数据包被挤压到*（a, T）*范围内，导致该时间间隔内数据包的发送时间整体偏移（质心后移）。红色矩形则表示数据包被挤压到*（a, T）*范围内的所有数据包。没有水印信号注入的间隔*T*内数据包发送时间没有被挤压，例如第二个黑色矩形中的数据包在整个时间间隔*T*内均匀分布。时间间隔基础类（interval-based）方法的I/O图与此类似。

本软件可以通过实现七种网络流水印方法，将水印信号注入到目标网络流中，并对网络流速产生预期的影响。因此，水印检测者可以对捕获的数据包进行分析，在网络背景流等因素干扰下，检测出水印信号。



**图18 解密正确信号的百分比**

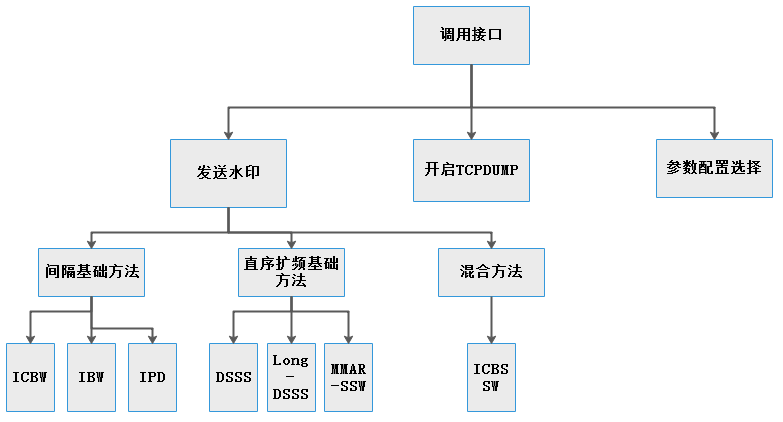
图18展示的是水印检测路由器通过捕获数据包文件，正确检测出的网络流水印信号的百分比。水印检测的正确率大都在90%以上，偶有70%的结果是因为当时网络存在严重干扰。为了进一步验证网络流水印检测效果，设计一系列实验对每种方法都给予不同程度的网络干扰（主要包括数据包延迟、丢包、数据流拆分、数据包重组等），取每种干扰的100次实验结果的检测正确率平均值作记录，如表4所示。以上实验表明，本网络流水印注入工具软件，可以采用不同方法有效完成水印注入的工作，可应用于数据流量的追踪。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **方法** | ICBW | IBW | IPD | DSSS | Long-DSSS | MMARSSW | ICBSSW |
| **检测率** | 94% | 96% | 91% | 93% | 92% | 96% | 98% |

**表 4 网络流水印方法的平均检测率**

# 6功能结构

本网络流水印注入工具软件包含的功能结构，如图19所示。所支持的七种网络流水印方法使用统一的入口函数（简称调用接口）进行调用，其输入参数是一个包含了水印方法名、水印信息和目的地址的字典。在选择网络流水印方法和相关参数配置之后，可以待注入的水印发送至目标网络流。



**图19软件的功能结构图**

# 附录：命令行参数使用的样例

1. 不指定任何命令行参数，直接启动本软件：

*python3 wms.py*

1. 指定并选择直序扩频基础的（DSSS-based）方法类别中的DSSS方法：

*python3 wms.py -t 2 -t 3*

1. 指定并选择直序扩频基础的（DSSS-based）方法类别中的DSSS方法，并指定目的IP地址和目的端口分别为210.34.0.12和443：

*python3 wms.py -t 2 -t 3 -h 210.34.0.12 -p 443*

1. 指定并选择直序扩频基础的（DSSS-based）方法类别中的DSSS方法，指定目的IP地址和目的端口分别为210.34.0.12和443，指定写入的解密发送的水印所需的必要信息的json文件名为info.json：

*python3 wms.py -t 2 -t 3 -h 210.34.0.12 -p 443 -f info*

1. 指定并选择直序扩频基础的（DSSS-based）方法类别中的DSSS方法，指定目的IP地址和目的端口分别为210.34.0.12和443，并在发送水印时开启Tcpdump捕获流量保存到文件capture.pcap：

*python3 wms.py -t 2 -t 3 -h 210.34.0.12 -p 443 -d capture.pcap*

1. 指定并选择直序扩频基础的（DSSS-based）方法类别中的DSSS方法，指定目的IP地址和目的端口分别为210.34.0.12和443，开启安静模式，指定写入的解密发送的水印所需的必要信息的json文件名为info.json，从文件message.txt中读取欲注入的水印信息，并在发送水印时开启Tcpdump捕获流量保存到文件capture.pcap：

*python3 wms.py -t 2 -t 3 -h 210.34.0.12 -p 443 -s -f info.json -d capture.pcap < message.txt*

# 参考文献

[1] Pyun Y J, Park Y, Reeves D S, et al. Interval-based flow watermarking for tracing interactive traffic[J]. Computer Networks. 2012, 56(5): 1646-1665.

[2] Wang X, Chen S, Jajodia S. Network flow watermarking attack on low-latency anonymous communication systems[C]. Proceedings of IEEE Symposium on Security and Privacy, 2007: 116-130.

[3] Pan Z, Peng H, Long X, et al. A watermarking-based host correlation detection scheme[C]. Proceedings of Management of E-Commerce and E-Government. 2009: 493-497.

[4] Yu W, Fu X, Graham S, et al. DSSS-based flow marking technique for invisible traceback[C]. Proceedings of IEEE Symposium on Security and Privacy. 2007: 18-32.

[5] Huang J, Pan X, Fu X, et al. Long PN code based DSSS watermarking[C]. Proceedings of IEEE Conference on Computer Communications. 2011: 2426-2434.

[6] Zhang L, Wang Z, Wang Q, et al. MSAC and Multi-flow Attacks Resistant Spread Spectrum Watermarks for network flows[C]. Proceedings of IEEE International Conference on Information and Financial Engineering. 2010:438-441.

[7] Luo J, Wang X, Yang M. An interval centroid based spread spectrum watermarking scheme for multi-flow traceback[J]. Journal of Network and Computer Applications, 2012, 35(1): 60-71.

[8] Biswas A K, Ghosal D, Nagaraja S. A Survey of Timing Channels and Countermeasures [J]. ACM Computing Surveys (CSUR), 2017, 50(6): 1-39.

[9] iPerf - The ultimate speed test tool for TCP, UDP and SCTP. <https://iperf.fr>

[10] WIDE Internet of [WIDE Projec](http://www.wide.ad.jp/)t, http://two.wide.ad.jp/.

1. PN码是一种具有与白噪声类似的自相关性质的0和1所构成的编码序列。 [↑](#endnote-ref-1)