每周总结

数据科学与计算机学院 孔德宇

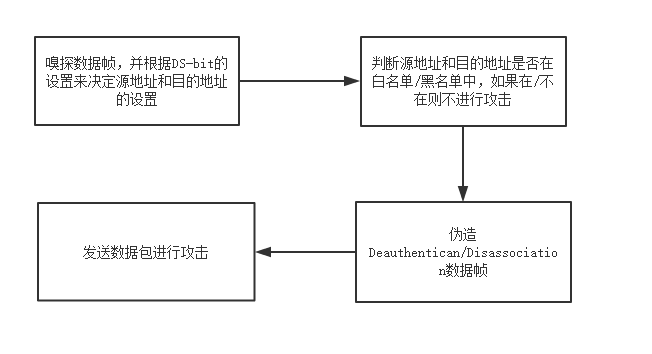
一、任务概述

● 阅读mdk3源代码，分析洪水攻击的实现方法。

● 编写代码，实现对AP上某一终端的洪水攻击。

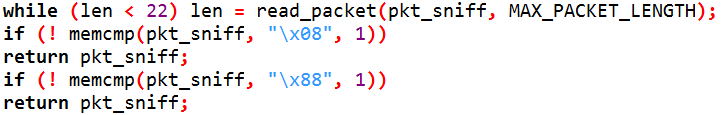
二、mdk3洪水攻击原理

1、基本流程图



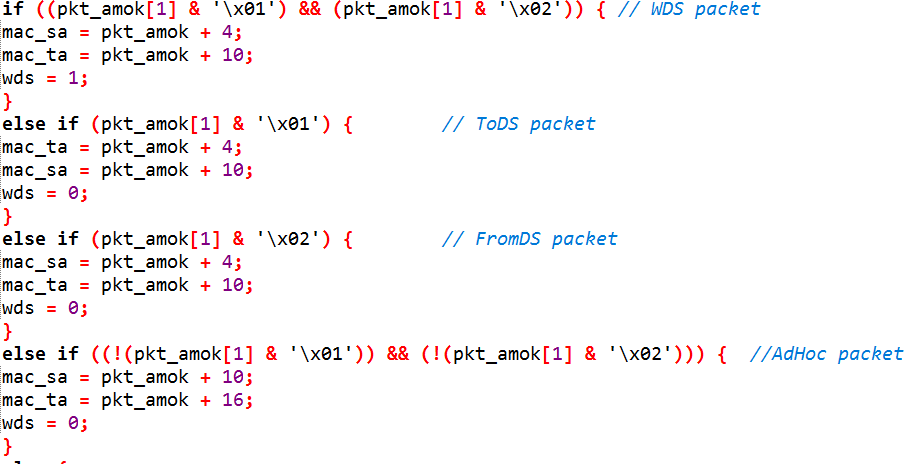
2、主要步骤分析

**①** 嗅探数据帧

mdk3实现的洪水攻击是对一片区域或指定AP的所有连接终端的全面攻击。因此如果想要获得周围区域所有的AP和终端信息，就需要对数据包进行嗅探。

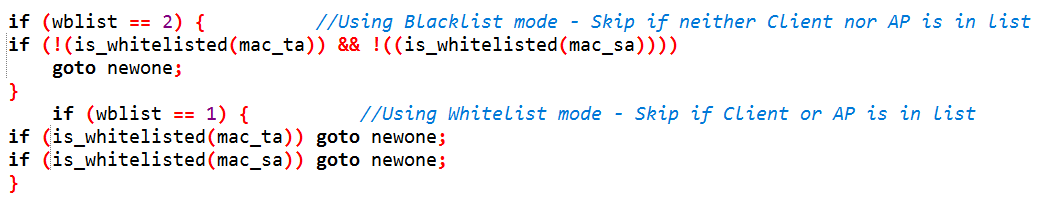
这里调用read\_packet（实际上是底层的recv函数）来进行数据包的嗅探，并取出其中的数据帧。

**②** 设置目的地址和源地址

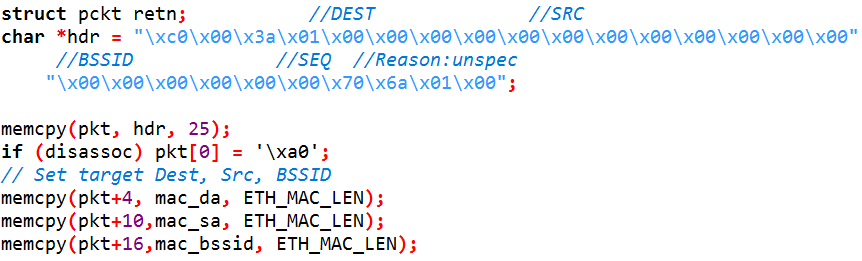
 数据帧里面有两个重要的bit ，TO DS和FROM DS。它们决定了数据的流向以及目的地是否为传输系统，通过它们我们可以对源地址和目的地址进行设置。

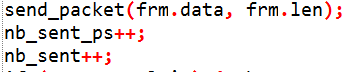
根据数据帧DS位的取值，从而决定源地址和目的地址在数据帧上的位置。并对其进行设置。因为

③ 对白名单/黑名单的处理

 如果目的地址和源地址都不在黑名单中，则不进行攻击。如果目的地址或源地址在白名单中就不进行攻击，否则进行攻击。

**④** 伪造Deauthentican/Disassociation帧

 根据Deauthentican/Disassociation帧的格式，进行帧的伪造。

**⑤** 发送数据包，进行洪水攻击

这里调用send\_packet（）函数（实际上为底层的send函数）来发送数据包从而达到洪水攻击的目的。

三、对AP上某一终端的洪水攻击

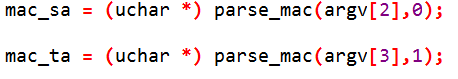
可以看出，mdk3工具虽然可以进行洪水攻击，但是攻击范围过大。于是在mdk3的基础上做了修改，使其能达到只断开某一终端连接的洪水攻击。

1、改进的方面

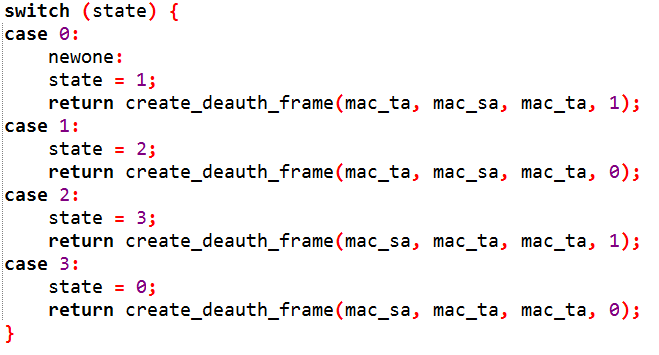
● 通过指定目的地址和源地址进行洪水攻击。

● 通过混合Deauthentican/Disassociation帧并且双向发送来提高攻击效率。（不确定、理论上可行）

2、改进分析

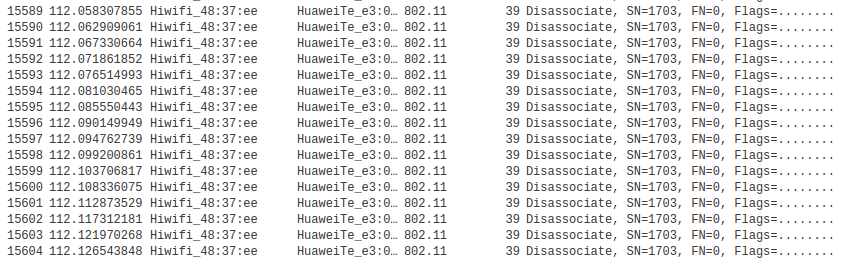
**①** 设置目的地址和源地址

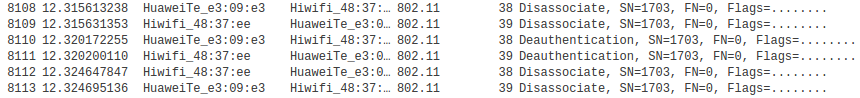
根据输入来设置目的地址和源地址。

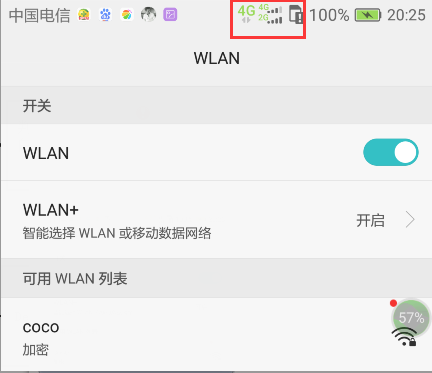
② 提高攻击效率

通过不断发送两种、双向的数据包，从而提高攻击的效率。

3、测试结果

单向攻击的wireshark抓包结果：

双向攻击的wireshark抓包结果：

攻击后终端状态：

可以看出，终端成功的被断开。

攻击后其它终端的状态：



可以看出其它终端仍然可以正常连接，说明攻击达到了目的。