Unleashing the Shrew: an intelligent attack on TCP traffic in WLANs

Zhang Yaofeng

Supervised by Brahim BENSAOU

1. Introduction

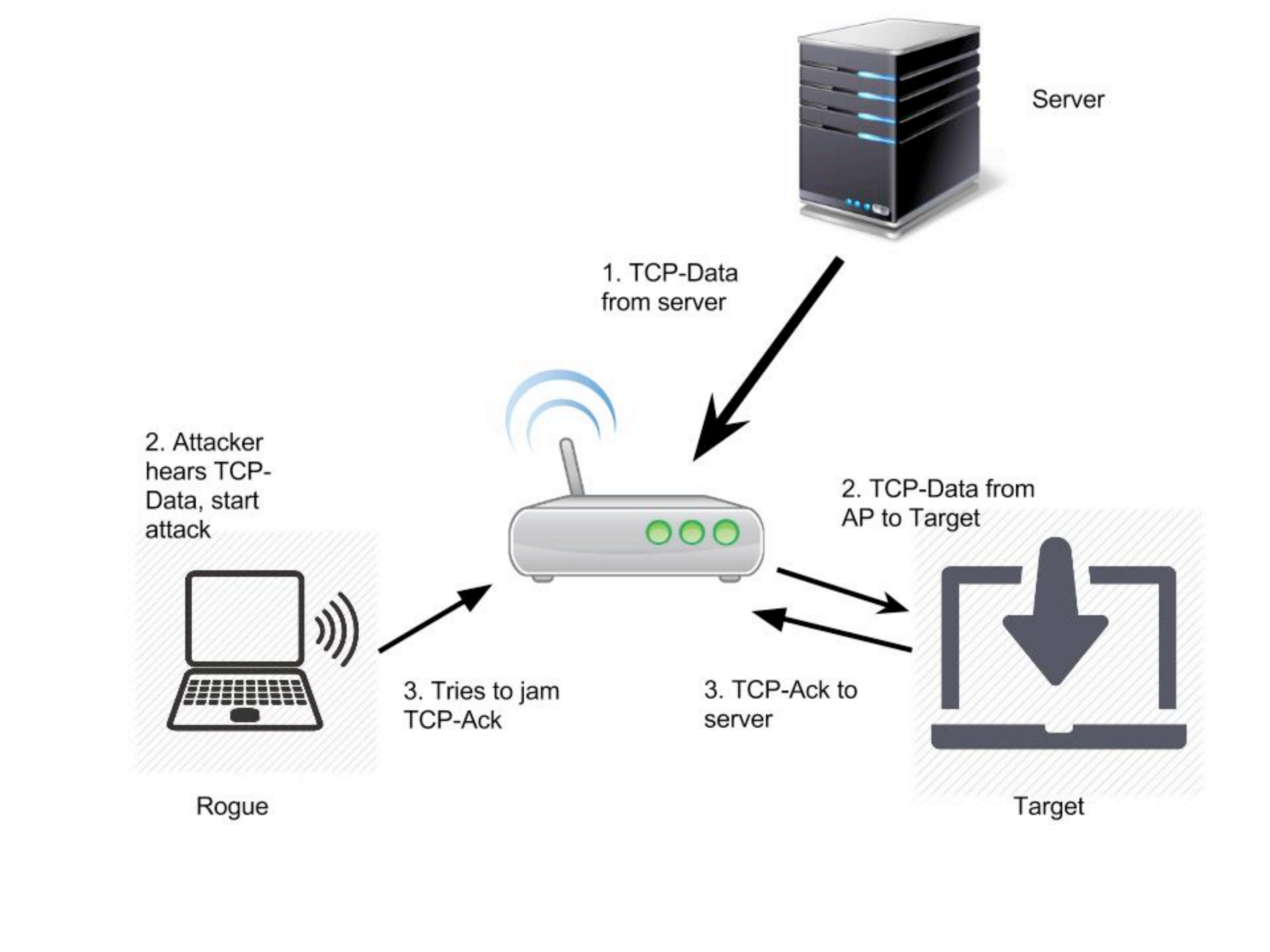
WiFi is a popular technology we use every day. However, the share nature of wireless medium makes WiFi susceptible to different kinds of misbehavior. Attacker may gain additional bandwidth by attacking traffic of other members in the same WLAN.

An attack model has been proposed by L. Gu and B. Bensaou in which the attacker targets downlink TCP traffic of a hidden terminal. By sending jamming signal according to a probablistic model, the attacker aims to jam TCP-Ack from target to its TCP source, so that a decrease of target's TCP transmission rate may increase the attacker's share of bandwidth in the network. Theoretical analysis and simulation have been performed. This project aims to implement and deploy the attack in real wireless network node.

2. Technical Description

Attack sequence is as follows.

1. Remote server established TCP connection to target
2. Server starts sending traffic to target
3. TCP-Data packets arrive AP, AP transmits to target, rogue node overhears the activity
4. Rogue node starts counter to mimic backoff counter behavior in target to predict the time target sends back TCP-Ack
5. Rogue node sends UDP packet to jam the transmission of TCP-Ack from target



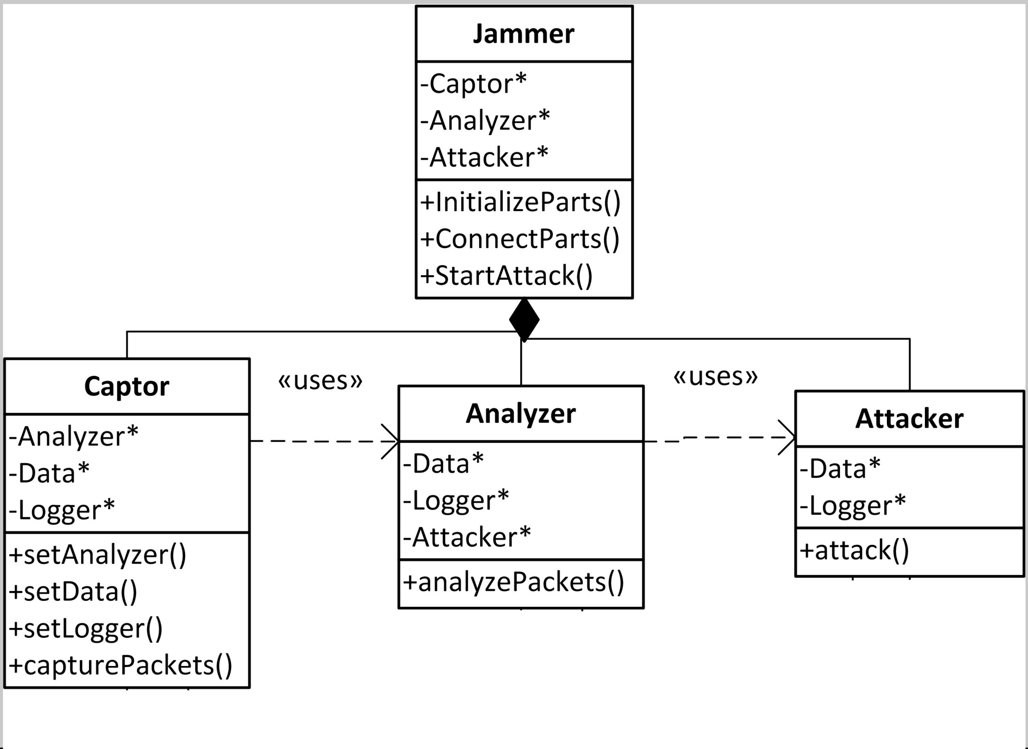
3. Design

3 main components are instantiated:

I. Captor: captures packets from wireless networks

II. Analyzer: extracts information from packets to decide whether to attack

III. Attacker: sends jamming signal when asked by Analyzer



Jammer
-Captor*
-Analyzer*
-Attacker*
+InitializeParts()
+ConnectParts()
+StartAttack()

Captor
-Analyzer*
-Data*
-Logger*
+setAnalyzer()
+setData()
+setLogger()
+capturePackets()

«uses»

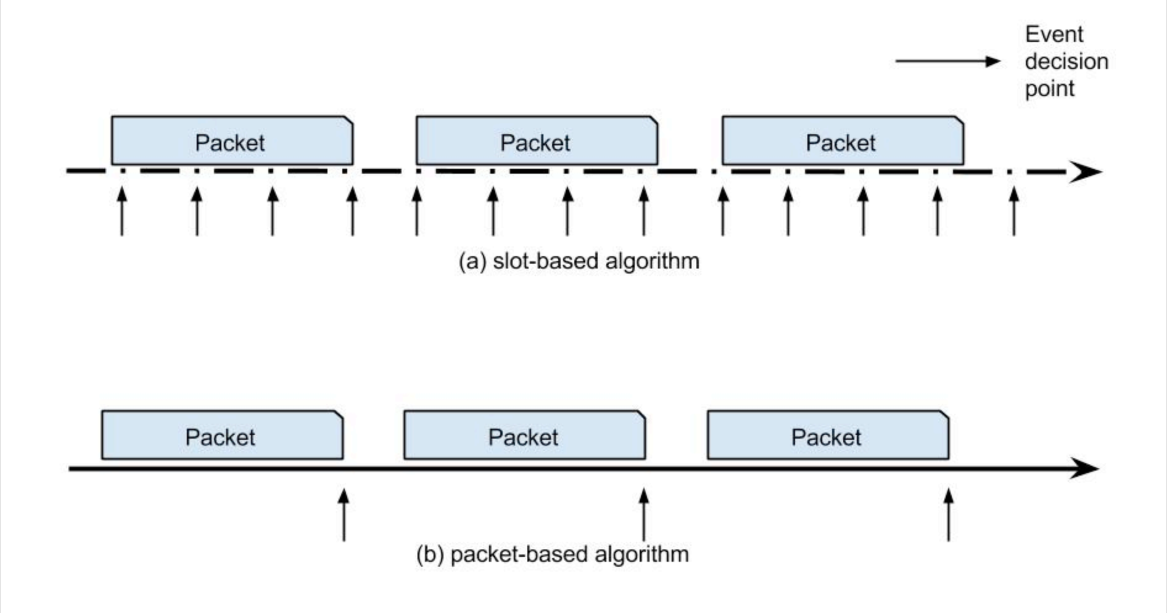
Analyzer
-Data*
-Logger*
-Attacker*
+analyzePackets()

«uses»

Attacker
-Data*
-Logger*
+attack()

4. Implementation

We use libpcap to capture packets. As it is a user space program, it is more natural to implement the program in packet-based instead of the original slot-based algorithm. Translation is required to perform correct updates of corresponding counters.



(a) slot-based algorithm

(b) packet-based algorithm

5. Evaluation

The program can correctly monitor the traffic and extract information from packets. It is robust enough to sustain long time running without crashing.

The program shows deficiencies in terms of effectiveness because of potential problems such as kernel-user space latency and priority of user space program.

6. Conclusion

A user space C++ program that realizes the jamming attack model proposed by L.Gu and B. Bensaou is preliminarily implemented by making use of libpcap and high resolution timers. Improvement can be achieved by moving the program in kernel space in future development.