Firewall Force: Final Project Report

Project Title: Security for Drone Networking Systems

Group Members: Connor Plonka, Nicholas Paceski, Andrew Destacamento, Lazaro Loureiro, Samson Silver, Richardson Jacques

Mentor: Nicholas Taylor

Objective and Scope

The objective of this project is to find a variety of methods to help protect a network of drones from being attacked and compromised. With the use of drones increasing rapidly in the current age, being used for such tasks as mapping out the landscape or being used in light shows, it is pertinent to have strong security measures protecting them and the data they use. Having multiple different methods that work on separate aspects of security help create a more robust defense and understanding of how to protect these networks and the drones that use them.

The scope of the project is limited to a simple setup between personal drones and a central command unit, in this case a truck that gets data and sends commands back to the drones. Since we are only a group of students, we had to make sure we don’t try to overcomplicate our goals and possible applications. With that in mind, we get the scope small, in connection with using personal drones. It also is limited to the software side of protection for the network, and hardware protections were not focused on.

Methodology and Process

We approached this project by aiming to design different methods to increase security in the drone network, focusing on secure communication over the network and authentication procedures between the drones and the truck on the network. These approaches were done using virtual environments, specifically using virtual machines on a limited network running on a Debian-based operating system.
The main focus areas for our approaches include:

- IP and MAC Address Rotation
- Rate Limiting and Flooding Prevention
- Secure P2P Networking
- WPA3 Authentication
- Firewall Configurations

We implemented these approaches through linux system administration by changing certain settings of the devices, creating scripts for authentication and secure connections, and then experimenting with them by running them in the virtual environments, as mentioned above.

Implementation Details

1. IP and MAC Address Rotation

IP and MAC address rotation provides protection by making it harder for attackers to pinpoint a specific address to track or identify, and thus target for attacks. Alternatively, an attacker may attempt to impersonate a valid drone by spoofing the drone's IP and MAC addresses. Rotating both of these would hinder an attacker if they can only spoof one address. Two different ways of implementing address rotation were designed:

- IP Address Rotation:
    - Used router settings and dnsmasq on OpenWRT to randomize IP addresses.
    - IP lease times randomized between 2 to 60 minutes.
- MAC Address Rotation:
    - Implemented using the GNU macchanger package via a systemd service.
    - Randomized MAC changes every 3 to 13 minutes to reduce tracking risks.

2. Rate Limiting and Authentication Flooding Prevention

To prevent attempts to flood available connections and make it so drones are unable or limited in their ability to connect to central command, authentication practices were put in place to allow connections from validated devices and disallow from invalid devices, as well as block the addresses of those attempting to continually connect. Connection is also limited through time intervals, so as to not flood the system continuously with connection requests.The implementation designs for this are as follows:

- Secure Token-Based Authentication:
    - Drones generated tokens using a secure key shared only with the truck.
    - Trucks validated drone authentication by matching tokens during communication.
- Simulation Environment:
    - Implemented using two Ubuntu VMs with dual network adapters (LAN and host-only).
    - Flask and Python scripts simulated drone and truck servers.
    - Flooding attempts with fake requests were tested and blocked successfully.
- Dynamic IP Handling:
    - Drones could dynamically update their IP addresses while maintaining static tokens.
    - Unauthenticated IPs or mismatched tokens were blocked.

3. Secure P2P Networking: VXLAN over IPSec

Secure networking helps ensure that any data sent over a network cannot be read by attackers from the outside, helping to protect from sniffing attacks and man-in-the-middle attacks. The data is encrypted through a tunnelling configuration between the valid drone and central command, with the design as follows:

- Configuration:
  - Used strongSwan for IPSec tunneling.
  - Employed IKEv2 protocol with Kyber768 + X25519 for key exchange.
  - Encryption used AES-GCM-256 cipher.
- Security Testing:
  - Simulated sniffing and man-in-the-middle attacks.
  - Ensured only authenticated and encrypted traffic passed through the tunnel.

4. WPA3 Authentication

Similar to the earlier authentication method, WPA3 authentication allows for authenticating drones and central command before a connection is formed between the two for data to traverse. In this case, authentication is done using certificates that are saved within the devices themselves, without needing to rely on address verification to ensure only valid drones can connect and communicate to central command, and vice versa. This method was designed as followed:

- Architecture:
  - Implemented WPA3-Enterprise simulation using TLS and X.509 certificates.
  - No centralized authentication server; enforced Zero Trust principles locally.
- Verification:
  - Drones authenticated using certificates verified by signature, expiration, and issuer.

5. Firewall Defense

Firewalls act as a fundamental level of security, limiting the allowed connections that can be made and thus protecting devices from connections by malevolent devices and attackers. The firewall for the truck/central command and drones were designed thusly:

- Firewall Rules:
  - Truck firewall allowed traffic only from verified drone IP addresses and ports.

        - Port 5000 was used, to work with other units.
    - Drone firewall limited incoming connections exclusively from the truck.

Challenges and Solutions

There were various challenges with the implementation of these methods. First, because we had to divide the labor between us all to ensure we were able to finish our parts, it made it so certain methods did not necessarily work with others fundamentally. For this reason, we were able to work with what we have to create at least two different combinations of the methods for protection, one which utilizes static addresses and one which uses dynamic or changing addresses. That allows us to show the merits of the methods even if they were not able to all work together.

Second, there are issues of errors with the programs themselves, in regards to how they worked together. This was solved through experimentation and editing of the code to ensure that all the scripts made were able to work together where they could and allow for multiple methods to be implemented at once. While it took a good amount of work, it did lead to some beneficial results, though there were still issues that we may have not resolved yet.

Lastly, a great challenge had less to do with the programming and design itself than the collaborative effort: at times we felt that we didn’t have enough time to work on the project together due to other projects from different courses, making it difficult for us to work as a team rather than individually. While more could have been done to solve the issue, we did what we could to work together collaboratively through Discord as well as meeting up in person to better edit and create a functional product of the different methods to then be able to share and present.

Results and Evaluation

In the end we have a finished result of five functional methods or techniques for drone network security, though they all may need future refinement. The techniques allow for two different scenarios utilizing different combinations of these techniques, which we tested through two demonstrations:

Demonstration 1: Authentication Flood Prevention and Secure Tunneling

In this demonstration we used the Firewall Configuration, P2P Networking, and Authentication Flood Prevention methods together at the same time, to show that they can work together to provide greater security for a network where the drones and central command use static or unchanging IP and MAC addresses.

- Demonstrated successful rejection of unauthorized servers.
- Demonstrated secure tunnel setup preventing data leakage.

Demonstration 2: WPA3 Authentication

This demonstration utilized the WPA3 authentication and MAC address rotation methods together, to show that they can work together in a way that provides security for a network with devices using dynamic or changing addresses instead of static addresses.

- Authenticated drones with valid certificates connected successfully.
- Drones with invalid certificates were denied access.

Testing Summary

While there were some hiccups with the tests, it overall was done in a way that showed that all the methods were able to work to some capacity, with the results as follows:

- Rate-limiting successfully thwarted simulated flooding attempts.
- Token-based dynamic IP management allowed secure, continuous authentication.
- Secure P2P networking ensured data confidentiality against network sniffing.
- WPA3 simulation verified robust local certificate-based authentication.

Reflection and Lessons Learned

After having done this project, we can reflect on the problems we faced throughout the semester working as a team. As stated before, we had difficulties combining our modules due to the fact we split the work up in the group. This also ties into the minimal knowledge we share on drones and networks. We can learn from this experience by doing research on the subject beforehand and having initial meetups ensuring everybody is on the same page. Even though things did turn out for the better in the end, this was a hurdle we spent a large chunk of our time evaluating.

This project didn't just entail the development of a program for a system of trucks and drones, it also taught us how to develop an entire simulation for any type of device and server setup. If implemented properly, the setup for this project could just as easily be used for a different group of devices, like an authentication token used to let users verify their entrance to their account. Not only can we use the practical skills developed here to create a simulation for any device we would want to work with, we would also have the knowledge to secure and protect it assuming the simulation was put to work in real-life application.