The impact of DoS attacks on the AR.Drone 2.0

Gabriel Vasconcelos, Gabriel Carrijo, Rodrigo Miani, Jefferson Souza
Faculty of Computing (FACOM)
Federal University of Uberlândia, Uberlândia, Brazil
E-mail: {gabrielvasconcelos, gabriellander, miani, jrsouza}@ufu.br

Vitor Guizilini
School of Information Technologies (SIT)
University of Sydney, Sydney, Australia
E-mail: v.guizilini@acfr.usyd.edu.au

***Abstract*—A key challenge for the use of unmanned aerial vehicles (UAVs) is the security of their information during navigation to accomplish its task. Information security is a known issue, but it seems to be overlooked from a research perspective, that tends to focus on more classical and well-formulated problems. This paper addresses an experimental evaluation of three Denial of Service (DoS) attack tools to analyze the UAV's behavior. These tools are executed in real-time on the robot while it navigates an indoor environment (inside the University building). We present experiments to demonstrate the impact of such attacks on a particular UAV model (AR.Drone 2.0) and also show a description of existing vulnerabilities. Our results indicate that DoS attacks might cause network availability issues influencing critical UAVs applications, such as the video streaming functionality.**

***Keywords*-Unmanned Aerial Vehicles; DoS; AR.Drone 2.0;**

I. Introduction

Aerial robotics has seen significant progress recently with several technology companies, such as Google, Amazon and Facebook. Today, UAVs are successfully employed in many outdoor applications, such as product delivery, agriculture [1], area monitoring [2], maritime patrol [3], mapping [4] and search & rescue [5]. In these situations, an aerial robot is expected to operate autonomously or remotely piloted for long periods of time while navigating potentially unexplored and highly dynamic areas.

According to Wilson [6], a drone is a UAV without a pilot that can be maneuvered by remote control or by onboard computers providing autonomous behaviours. While most people are capable of remotely piloting a drone, they usually do not care about the information contained in it (i.e. data from the camera, sonar, laser, radar, inertial measurement unit (IMU), Global Positioning System (GPS), among others sensors).

Information security [7] is a critical aspect of any research area, but it is specially important when UAVs are involved. Assuming that a robot does not have the minimum security for autonomous navigation, hackers can bring it down immediately. Furthermore, private information contained in the aerial robot can be stolen, such as camera images and GPS position. Finally, hackers can get full control of the drone and perform several unexpected activities, such as collide with objects near it.

Our goal is to perform an experimental evaluation of several Denial of Service (DoS) attack tools on the AR.Drone 2.0, while keeping the vehicle as safe as possible. We apply three different DoS attack tools, namely Low Orbit Ion Cannon (LOIC) [8], Netwox [9] and Hping3 [10]. During the experiment, we explore the features of each one of these tools. The proposed methodology involves: delimiting all the variables in the experiments section (pilot, drone and the attacker); explaining the configuration and the specification of the AR.Drone 2.0; and describing the step by step of the experiments with the DoS attack tools.

The main contribution of this work is to provide an empirical evaluation of the effects caused by DoS attacks on UAVs such as the AR.Drone 2.0.

The remainder of this paper is organized as follows. Section II shows the theoretical background of computer security, reconnaissance attacks and DoS attacks behind our experimental. Section III reviews the state of the art in applications of drone security. Section IV presents the proposed methodology, detailing the scenario used to validate the experiments, and also provides a brief overview of the configuration and specification of the AR.Drone 2.0. Section V details and discusses the experimental setup, results and analysis. Finally, section VI concludes the paper suggests directions for future work.

II. THEORETICAL BACKGROUND

In this section, we provide a brief overview of the theoretical background behind our experimental evaluation. More specifically, we first discuss some computer security concepts, then the notion of reconnaissance attacks and lastly the fundamentals of DoS attacks.

A. Computer Security Principles

Computer networks and information technology systems have made a huge impact in our society, from powerful smartphones to e-commerce and cloud computing solutions. Within this scenario, there is a need to secure the systems that hold data about citizens, corporations, and government agencies [11].

Computer security [12], as a field, is the study of how to make computer systems resistant to misuse. One example of abuse is a cyber attack. Any action taken to undermine the functions of a computer network or device can be viewed as a cyber attack [13]. The Ponemon Institute, in a recent survey [14], showed that the mean annualized cost for protecting and dealing with cyber attacks, for an organization, is around $15 million per year.

Three different aspects decompose information security: confidentiality, integrity, and availability. Jonsson and Pirzadeh [15] provide the definition for each one:

- **Confidentiality** - is the ability of the computing system to prevent disclosure of information to unauthorized parties;
- **Integrity** - is the ability of the computer system to guard against improper information modification or destruction;
- **Availability** - is the ability of the system to in fact deliver its service.

Suppose that Alice, the pilot, wants to send a message to the drone without anybody else learning its contents. If no one else apart from Alice and the drone can hear the message, then we say Alice and the drone have confidentiality.

Integrity is the correctness of the data, for example, ensuring that the message the drone receives the same one that Alice intended to send.

Availability is being able to always have a communication channel between Alice and the drone. A typical availability attack occurs when an attacker cuts the communication channel between Alice and the drone. The focus of this paper is the study of availability attacks towards the AR.Drone 2.0, and in particular an attack known in the literature as Denial of Service.

B. Reconnaissance

The first step for any cyber attack consists of information gathering about a targeted network or device. This phase is called reconnaissance. Port scan is one of the most used reconnaissance attacks. A port scan is used to check for open or closed network ports and for used or unused services. The services may or may not have a vulnerability that the attacker could exploit [16]. An Internet Control Message Protocol (ICMP) port scan, for example, is used to check the availability of a target device and the fingerprint of the target operating system. The Network Mapper or Nmap [17] is an automated tool that discovers hosts and services on a computer network through port scans.

C. Denial of Service Attacks

A Denial of Service attack is characterized by an attempt of an attacker to prevent legitimate users of a service from using the desired resources [18]. Moore et al. [19] groups DoS attacks in two different classes: logic attacks and resource attacks.

Logic attacks exploit existing software flaws to cause remote devices to crash or substantially degrade in performance. A well-known example is the Ping of Death, that causes the operating system to crash by sending an ICMP ping packet larger than 65,535 bytes. Upgrading faulty software or filtering particular packet sequences prevents many of these attacks, but they remain a serious and ongoing threat [19].

Resource attacks overwhelm the victim's computer resources (CPU and memory, for instance) or network resources by sending incessant streams of spurious packets. Because there is typically no simple way to differentiate the valid packets from the malicious packets, it can be hard to defend against this type of attack. A well-known resource attack is the Synchronize (SYN) Flood, which exploits a weakness in the Transmission Control Protocol (TCP) connection sequence (three-way handshake). The attacker sends multiple SYN requests to the server but does not respond to the server's SYN-ACK response. The server continues to wait for an acknowledgment (ACK) for each one of these requests, binding resources until no new connections can be made, and ultimately resulting in the denial of service.

In this paper, we are interested in analyzing the impact produced by DoS resource attacks on the AR.Drone 2.0. Logic attacks are out of the scope of this work.

III. RELATED WORK

UAVs have recently experienced a massive increase in their areas of application, due to a combination of advancements in hardware (higher payloads, more precise sensors, component miniaturization) and software (more efficient algorithms, scaling in data acquisition and storage, embedded processing). These areas of application include aerial photography, surface mapping, surveillance, scene reconstruction and 3D modeling, target tracking and product delivery. The quick spread of UAVs – allied to a large variety of shapes, sizes and prices – have also led to their adoption by a broad range of consumers, from hobbyists to billion-dollar companies, each with its own requirements and goals.

Currently the biggest challenge in the spread of UAVs to an even wider audience is not technological, but legal. Such ubiquitous tool, capable of addressing in a similar manner problems that once needed specific equipment and specialization, needs above all to be robust and reliable, to minimize the chances of costly accidents. Careful engineering can precisely quantify the odds of mechanical and software failure, but there are also security risks, that come with malicious intent [20]. The analysis of such threats is crucial to promote commercially viable products; that satisfy all legal requirements of safety and security.

In [21], the potential risks of unencrypted connections (both for communication and video streaming) are explored, including the possibility of hijacking the vehicle. Pleban, Band, and Creutzburg [22] addresses the possibility of spying on unencrypted video streams, which could reveal confidential information about the user and its surroundings.

Deligne [23] briefly describes DoS attacks that could lead to the corruption of communication between user and machine, along with means to reproduce these attacks under different circumstances. In this paper, the author executes a very simple attack using the Hping3 tool. Deligne does not discuss several issues: analyzing network delay issues caused by the attack, investigating other DoS attack types, such as TCP SYN (that can be performed using LOIC and Netwox) and evaluating the impact of the attack on the drone functionalities.

The security studies conducted in [21], [23] and [22] focused on the same platform explored on this paper (the Parrot AR.Drone 2.0, a low-cost consumer drone). However, the same methodology can be readily applied to any other wireless vehicle.

IV. METHODOLOGY

Our primary goal is to understand and analyze the consequences resulting from launching the DoS attacks on UAVs such as the AR.Drone 2.0. The proposed methodology describes a usual scenario where a Pilot is sending a series of commands to the UAV and an attacker launches reconnaissance and DoS attacks.

The following steps summarize the methodology:

1) Establish a connection between Pilot and AR.Drone;
2) Pilot sends a set of commands to AR.Drone (taking off, short flights and landing) to understand its behavior in normal conditions (no attackers);
3) Establish a connection between Attacker and AR.Drone;
4) Attacker performs reconnaissance attacks on AR.Drone using the port scan tool;
5) While Pilot is sending a series of commands to AR.Drone, Attacker uses information obtained in step 4 to launch a DoS attack towards AR.Drone.

Figure 1 shows the components of the experiment:

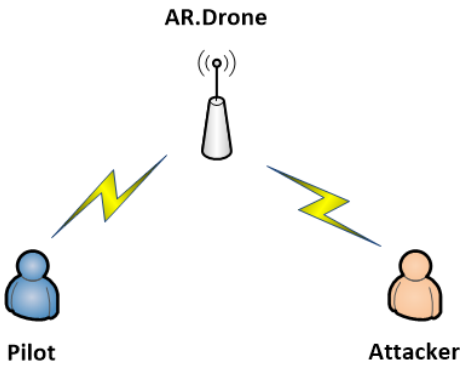

Figure 1. Main components of the proposed experiment.

- **Pilot** - It is a person that connects to the wireless network of the AR.Drone 2.0 using a standard laptop. Also, this person uses a PS3 (Playstation 3) joystick to maneuver and control the AR.Drone 2.0. We use ROS[1] to interface the laptop with the AR.Drone 2.0;
- **AR.Drone** - It is a quad-rotor helicopter that can be piloted by a mobile device on the iOS or Android systems. Furthermore, the AR.Drone 2.0 can be piloted through of a standard laptop (as in our case). This drone has the following features: Wi-Fi b/g, MEMS 3-axis accelerometer, 2-axis gyroscope, high-efficiency propellers, structure of carbon fiber, four brushless motors, lithium-polymer battery, front and vertical cameras, and ultrasonic altimeter sensor;
- **Attacker** - It is a person that connects to the wireless network of the AR.Drone 2.0 using an another standard laptop. This person uses tools to scan ports of the AR.Drone 2.0 and applies a series of DoS attack tools, as LOIC, Netwox, and Hping3 (section IV-B).

[1]Robot Operating System: http://wiki.ros.org/

A. Reconnaissance

The reconnaissance attacks on the AR.Drone will be done using a security tool called Nmap ("Network Mapper"). Nmap is an open source tool for network exploration and security auditing [17]. Nmap was designed to determine which hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, and several of other features.

The output from Nmap is a list of scanned targets, with supplemental information on each depending on the options used. In our case, we are interested in the lists of scanned ports. This list shows the port number and protocol, service name and state. The state is either open, filtered, closed or unfiltered. According to the Nmap manual [17], open means that an application on the target machine is listening for connections/packets on that port. Filtered means that a firewall or another network obstacle is blocking the port, so that Nmap cannot tell whether it is open or closed. Closed ports have no application listening on them, though they could open up at any time.

B. DoS Attacks Tools

DoS attacks will be launched using three automated tools: Low Orbit Ion Cannon (LOIC), Netwox and Hping3. The DoS attacks tools are implemented in the following programming languages: C# (LOIC), C (Netwox and Hping3). Next, we provide some details for each tool.

Praetox Technologies originally developed LOIC, allegedly as a network stress-testing tool [8]. The source code for LOIC is still available on the now-unmaintained Praetox website, but it has since been modified in the public domain through various updates and has been widely used by Anonymous as a DDoS tool [24]. LOIC is very simple to use. The Windows version just needs the user to enter a target address and click the "IMMA CHARGIN MAH LAZER" button, although there are some optional settings. Users can select a variety of options, such as the type of packets sent (TCP, UDP or HTTP), port numbers and so on. In this paper, we will use the option "TCP" and set a particular port to launch TCP SYN resource attacks.

Netwox is a powerful open source toolbox that can be used to perform multiple network tests and also some attacks. In our experiment, we will use the Netwox tool number 76 to launch the SYN flood attack.

Hping3 is an open source tool designed as a packet generator and analyzer for the TCP/IP protocol (Internet Protocol - IP). The new version, hping3, is scriptable using the Tcl language and implements an engine for string based, human readable description of TCP/IP packets, so that the programmer can write scripts related to low-level TCP/IP packet manipulation and analysis in a short time.

In this work we will use Hping3 to launch a resource DoS attack by sending multiple spurious packets to the AR.Drone as fast as possible (–fast –flood option).

V. EXPERIMENTAL RESULTS

To evaluate the capabilities and performance of the methodology, we applied three DoS attack tools and tested their performance on an aerial robot, the AR.Drone 2.0 (Figure 2). We assessed the AR.Drone 2.0 inside the university building and our evaluation was performed in real-time on a standard laptop with Ubuntu 14.04.

Figure 2. Parrot AR.Drone 2.0 robotic platform used in the experiments.

The first step is to establish a connection between Pilot and AR.Drone. This procedure can be easily done, since the AR.Drone works as an Access Point, creating a wireless network under the name "ardrone2-044078". This network has no security capabilities (WPA-1 or WPA-2, for instance) or passwords, in other words, any device equipped with a wireless network card and within range may be able to establish a connection with the AR.Drone. A wireless network with no security capabilities represents a critical issue for ensuring confidentiality and integrity between Pilot and AR.Drone. One way to overcome this problem is updating the drone to support WPA-1 or WPA-2. Araos [25] developed a non-official update for that.

The next step consists of sending a series of commands to AR.Drone to understand its behavior in normal conditions, or with no attackers. We measured the network latency[2] between Pilot and AR.Drone for 5 (five) minutes using the ICMP ping. The average network latency for this period was 20.92 ms.

Now, the Attacker should establish a connection with the AR.Drone and perform reconnaissance attacks. As previously stated, any device equipped with a wireless network card will be able to connect to the AR.Drone. Using a standard laptop, the Attacker readily joined the AR.Drone wireless network. By verifying the new IP address as "192.168.1.3", the Attacker can assume that the AR.Drone's IP address is "192.168.1.1". Using this information the Attacker could launch port scan attacks using the Nmap tool.

Figure 3 shows the results produced by the Nmap. We can see the IP address of the AR.Drone ("192.168.1.1") and the port number and protocol, service name and state.

[2]We use the Round Time Trip - which measures the time required for a packet to travel from a specific source to a specific destination and back again - to estimate the network latency between two devices.

Three TCP ports, representing three different available services: 21 (FTP - File Transfer Protocol), 23 (Telnet) and 5555 (Freeciv - AR.Drone 2.0 video camera streaming application). Both ports 21 and 23 provide direct access to the AR.Drone 2.0 through the following shell commands: "ftp 192.168.1.1" and "telnet 192.168.1.1". None of these services are password protected. The Attacker might use telnet to get a root shell and be able to execute malicious remote commands, for example, a shutdown.

```
Starting Nmap 6.40 ( http://nmap.org ) at 2016-05-12 15:14 BRT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
 Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.1.1
Host is up (0.054s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
23/tcp   open  telnet
5555/tcp open  freeciv

Nmap done: 1 IP address (1 host up) scanned in 0.77 seconds
```

Figure 3. Nmap scanning on ports.

The Attacker is now able to launch DoS attacks on the AR.Drone. However, since some of the DoS attacks will be targeted in specific TCP ports, we also need to measure the network latency in regular conditions for every discovered TCP port (21, 23 and 5555). The average network latency for each case is depicted in Table I. The following DoS commands were executed from the attacker computer:

- Netwox - "*netwox 76 –dst-ip 192.168.1.1 –dst-port 21*", "*netwox 76 –dst-ip 192.168.1.1 –dst-port 23*" and "*netwox 76 –dst-ip 192.168.1.1 –dst-port 5555*"
- Hping - "*hping3 –fast –flood 192.168.1.1*" ([23])
- LOIC - loic was performed via Graphical User Interface (GUI) with the following parameters: *192.168.1.1*, Method *TCP* and ports *21* and *23* (LOIC does not support sending packets to port 5555).

The command line performed for the DoS attack tools are shown in Figures 4, 5 and 6. Table I shows the values obtained by the tools in each one of the attack rounds.

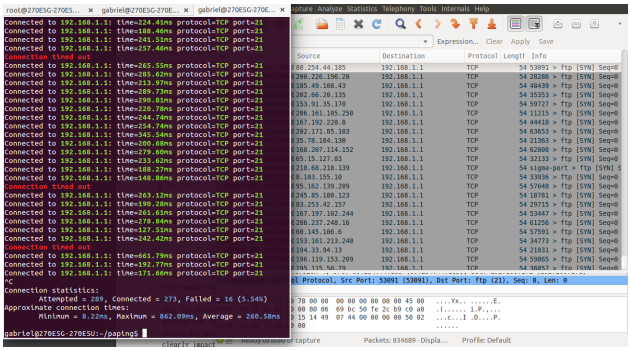

Figure 4. The Netwox command line to port 21 can be seen on the left side. The Wireshark tool is executed and shown on the right side.

The results presented in Table I show a substantial increase in the network latency during the DoS attacks for all the tools. Higher values of network latency are an indicator that something is wrong with the connection between two devices. This means that sending incessant spurious network packets to the AR.Drone caused

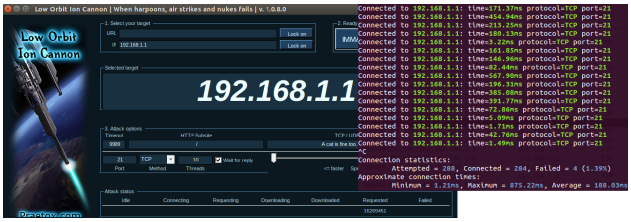

Figure 5. The GUI for the LOIC on port 21 can be seen on the left side. The average latency results by LOIC are shown on the right side.

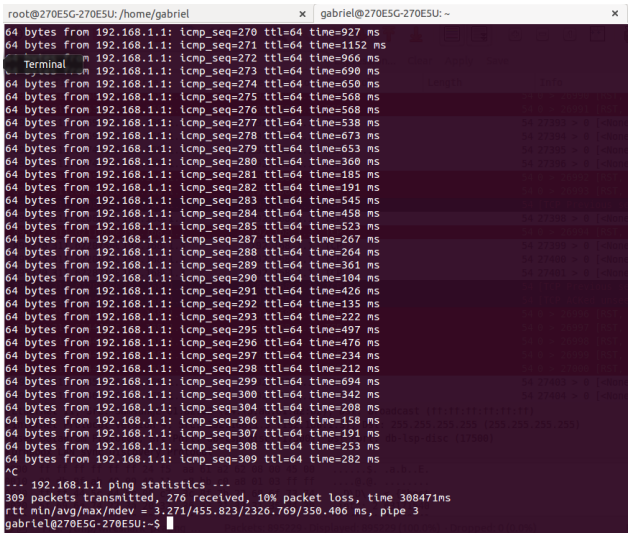

Figure 6. Hping3 is run to show the results of average latency.

Table I
THE AVERAGE LATENCY FOR THE DOS ATTACK TOOLS.

Regular conditions	Hping3	LOIC	Netwox
20.92ms (ICMP)	**455.82ms**	-	-
24.41ms (TCP Port 21)	-	**188.03ms**	**260.58ms**
57.60ms (TCP Port 23)	-	90.54ms	212.90ms
81.97ms (TCP Port 5555)	-	-	110.82ms

a direct impact on its network resources, validating our experiment. The less powerful processor embedded in the Drone could be one of the reasons behind the success of the DoS resource attack. However, the absence of basic security configurations (open wireless network and WPA, for instance) is also a factor that contributes to the attack.

We can observe in Table I that the highest value of the average latency of the network obtained by the three DoS attack tools was the Hping3 tool with 455.82 milliseconds. Table II shows the latency increase rate produced by three DoS attack tools. For example, 21.788 is the result of the division between 455.82 (Drone under attack) by 20.92 (regular network conditions) as shown in Table I.

Table II
THE LATENCY INCREASE RATE OF THE DOS ATTACK TOOLS.

	Hping3	LOIC	Netwox
ICMP	**21.788**	-	-
TCP Port 21	-	7.702	10.675
TCP Port 23	-	1.571	3.696
TCP Port 5555	-	-	1.351

We can see in Table II that the Hping3 DoS attack tool produced the highest value of latency increase rate compared to the other two DoS attack tools (LOIC and Netwox). Deligne [23] also launches a DoS attack using Hping3. In his paper, the behavior of the drone is haphazard and gets out of control, either by hitting an obstacle or shutting down the system board in less than a second. We were not able to reproduce this behavior, even with a five-minute attack. We believe that the company (Parrot) might have upgraded the firmware to deal with a high number of network packets sent to the drone. However, Hping3 can still be considered a serious threat to the AR.Drone due to the network resources issues caused by sending incessant streams of spurious packets. It is possible to note that the drone is also vulnerable to other DoS attack types, in our case, the TCP SYN Flood.

The impact of DoS on the AR.Drone 2.0 can be better explained by analyzing the behavior of the video streaming application (port 5555) during an attack. There is a significant number of video streaming applications using drones. Today for the world of filming, drones are almost indispensable with their cameras, because they can achieve extraordinary angles images. Also, drones may be able to transmit risk situations of inaccessible places (e.g. mines and landslide) and survey and act in dangerous situations. Another relevance is the successful identification of criminals through aerial images. We will show that DoS attacks can compromise this critical application.

Using “rostopic hz” in ROS we obtain the video frame rate exchanged between Pilot and Drone. Higher values indicate a good video quality for the user. Figure 7 shows the average frame rate exchanged between Pilot and AR.Drone 2.0 camera per second for the three tools.

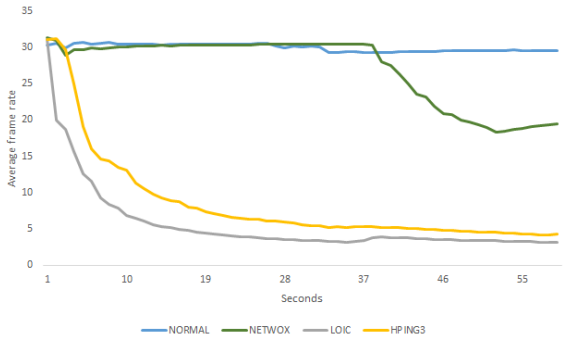

Figure 7. AR.Drone 2.0 camera average frame rate per seconds.

We observe that the Frame Rate (FR) is constant when there is no DoS attack. During an attack, the FR dramatically decreases, which may have a direct impact on video quality for the user. We can see that the tools that most influenced the average FR of the drone camera were the LOIC and Hping3 tools in a 1-minute analysis (Figure 7).

We have a video on the tools that were most influenced in the average latency in Table I (LOIC on port 21, Netwox on port 21 and Hping3). The impact of the DoS attack can be seen in the **Video Streaming Application**[3].

[3]Video Streaming Application: https://youtu.be/6QlGMn3_9XQ

We have made another video to show the vulnerability of the wireless network of the AR.Drone 2.0, where an attacker accesses the network via telnet on port 23. Afterwards, the attacker runs the poweroff command and the AR.Drone 2.0 falls from the air (**Poweroff Application**[4].).

VI. CONCLUSIONS

An evaluation of the information security of the AR.Drone 2.0 has been proposed, using tools of DoS attack from a UAV in indoor environments. These tools (LOIC, Netwox, and Hping3) were processed on a laptop through an attacker to bring down a drone. After a series of comparisons, the Hping3 tool showed the high impact on the UAV and resulted in the lowest average frame rate of the AR.Drone 2.0 camera. The DoS attacks performed in this experiment compromises the video streaming application along with any other computer vision application.

We believe that the contributions made in this paper are an important step towards information security for UAV. The tools of DoS attacks were fundamental to show the problematics of video streaming for the AR.Drone 2.0 when it is being attacked. The Hping3 tool showed the highest value of average latency of the network, resulting in the increase of network latency rate and the lowest average frame rate from the front camera of the AR.Drone.

As future work, we will consider other DoS attack tools. We will also try to adjust the network performance of the drone by tunning some TCP/IP parameters, such as increase the TCP/FIN timeout. Furthermore, we will improve our experiments to perform Distributed Denial of Service (DDoS) attacks. We believe that a DDoS attack may cause an even greater impact on the AR.Drone 2.0 resources. Finally, we will test another aerial robotics platform, known as SOLO, manufacted by the 3D Robotics company. These can help to build a complete experiment and a better understanding of the scenario.

VII. ACKNOWLEDGMENT

The authors would like to thank the financial support of the Federal University of Uberlândia, FAPEMIG and Faculty of Engineering & Information Technologies, The University of Sydney, under the Faculty Research Cluster Program.

REFERENCES

[1] S. Efron, "The Use of Unmanned Aerial Systems for Agriculture in Africa," pp. xxxii–334, 2015.

[2] S. D'Oleire-Oltmanns, I. Marzolff, K. D. Peter, and J. B. Ries, "Unmanned aerial vehicle (UAV) for monitoring soil erosion in Morocco," *Remote Sensing*, vol. 4, no. 11, pp. 3390–3416, 2012.

[3] R. Hopcroft, E. Burchat, and J. Vince, "Unmanned aerial vehicles for maritime patrol: human factors issues," *Science And Technology*, pp. 1–43, 2006.

[4] H. Eisenbeiss, "The Potential of Unmanned Aerial Vehicles for Mapping," *Photogrammetrische Woche 2011*, pp. 135–145, 2011.

[4]Poweroff Application: https://youtu.be/XK-aVEpAruM

[5] V. B. Hammerseth, "Autonomous Unmanned Aerial Vehicle In Search And Rescue," no. June, 2013.

[6] R. L. Wilson, "Ethical issues with use of Drone aircraft," *2014 IEEE International Symposium on Ethics in Science, Technology and Engineering, ETHICS 2014*, 2014.

[7] M. T. Scholar, "Security Incident Management in Ground Transportation System Using UAVs," no. i, 2015.

[8] P. Farina, E. Cambiaso, G. Papaleo, and M. Aiello, "Understanding DDoS Attacks from Mobile Devices," *Int. Conf. on Future Internet of Things and Cloud*, pp. 614–619, 2015.

[9] "Netwox," 2016. [Online]. Available: http://ntwox.sourceforge.net/

[10] "Hping3," 2016. [Online]. Available: http://www.hping.org/hping3.html

[11] M. Bishop, "What is computer security?" *Security & Privacy, IEEE*, vol. 1, no. 1, pp. 67–69, 2003.

[12] J. Bau and J. C. Mitchell, "Security modeling and analysis," *IEEE Security and Privacy*, vol. 9, no. 3, pp. 18–25, 2011.

[13] O. A. Hathaway, R. Crootof, P. Levitz, H. Nix, A. Nowlan, W. Perdue, and J. Spiegel, "The law of cyber-attack," *California Law Review*, vol. 100, no. 4, pp. 817–885, 2012.

[14] Ponemon Institute, "2015 Cost of Cyber Crime Study: Global," Tech. Rep., 2015.

[15] E. Jonsson and L. Pirzadeh, "A Framework for Sec. Metrics Based on Operational System Attributes," in *Int. Work. on Security Measurements and Metrics*, 2011, pp. 58–65.

[16] S. Panjwani, S. Tan, K. Jarrin, and M. Cukier, "An Exp. Evaluation to Determine if Port Scans are Precursors to an Attack," in *Int. Conf. on Dependable Sys. and Net.*, 2005.

[17] "Network Mapper," 2016. [Online]. Available: https://nmap.org/

[18] F. Lau, S. Rubin, M. Smith, and L. Trajkovic, "Distributed denial of service attacks," *International Conf. on Systems, Man and Cybernetics*, vol. 3, pp. 2275–2280, 2000.

[19] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," vol. 24, no. 2, pp. 115–139, 2006.

[20] S. S. Janghel, "Drone Security Analyzer," Ph.D. dissertation, 2015.

[21] F. Samland, J. Fruth, M. Hildebrandt, T. Hoppe, and J. Dittmann, "AR.Drone: security threat analysis and exemplary attack to track persons," *The International Society for Optical Engineering*, vol. 8301, 2012.

[22] J.-S. Pleban, R. Band, and R. Creutzburg, "Hacking and securing the AR.Drone 2.0 quadcopter: Investigations for improving the security of a toy," *The International Society for Optical Engineering*, vol. 9030, 2014.

[23] E. Deligne, "ARDrone corruption," *Journal in Computer Virology*, vol. 8, no. 1-2, pp. 15–27, 2012.

[24] S. Mansfield-Devine, "Anonymous: Serious threat or mere annoyance?" *Network Security*, no. 1, pp. 4–10, 2011.

[25] D. Araos, "Ardrone WPA2." [Online]. Available: https://github.com/daraosn/ardrone-wpa2