A look at Sundown exploit kit.

So recently a range of things have drawn my personal interest, one of them is understanding EK (exploit kits) more anyone in IT knows the best defense against attack is to understand the attack fully.

Understand the attack and thus where or how to break it and render it useless.

Sundown recently drew my interest as I was used to pouring over RIG EK, RIG EK … and look more….RIG EK…

Here was something different GREAT! So I got hold of two samples of sundown using flash exploit. Okay Flash .. heaps of exploits there but we will come back to that .. lets see what this little guy does.

Sundown was delivering through a email campaign leading to a gate, landing page and payload delivery.

No.	Time	Source	Destination	Protocol	Length	Info
3	1.681707023	192.168.5.56	192.168.5.1	DNS	72	Standard query 0x33d5 A www.bing.com
4	1.854162595	192.168.5.1	192.168.5.56	DNS	164	Standard query response 0x33d5 A www.bing.com CNAME www-bing-com.a-0001.a-msedge.net CNAME a-0001.a-msedge.net A 204.79.197.2…
5	1.854162595	192.168.5.56	204.79.197.200	TCP	66	49187 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
6	1.854162595	204.79.197.200	192.168.5.56	TCP	66	80 → 49187 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
7	1.854162595	192.168.5.56	204.79.197.200	TCP	60	49187 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
8	1.854162595	192.168.5.56	204.79.197.200	HTTP	587	GET /favicon.ico HTTP/1.1
9	1.854162595	204.79.197.200	192.168.5.56	TCP	54	80 → 49187 [ACK] Seq=1 Ack=534 Win=30336 Len=0
10	2.415146916	204.79.197.200	192.168.5.56	TCP	296	[TCP segment of a reassembled PDU]
11	2.615026548	192.168.5.56	204.79.197.200	TCP	60	49187 → 80 [ACK] Seq=534 Ack=243 Win=65456 Len=0
12	2.615309276	204.79.197.200	192.168.5.56	HTTP	354	HTTP/1.1 200 OK (PNG)
13	2.840173795	192.168.5.56	204.79.197.200	TCP	60	49187 → 80 [ACK] Seq=534 Ack=543 Win=65156 Len=0
14	23.689117266	fe80::89db:9dc8:341…	ff02::c	SSDP	181	M-SEARCH * HTTP/1.1
15	23.690054030	192.168.5.56	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1

Here we can see

- 192.168.5.56 (Win7VM my lab machine)
- 192.158.5.1(Kali:MITM proxy&Wireshark capture)
- 204.79.197.200 (Markmonitor)
- 239.255.255.250 (IANA) Internet Assigned Numbers Authority
- Some ipv6 traffic after that.

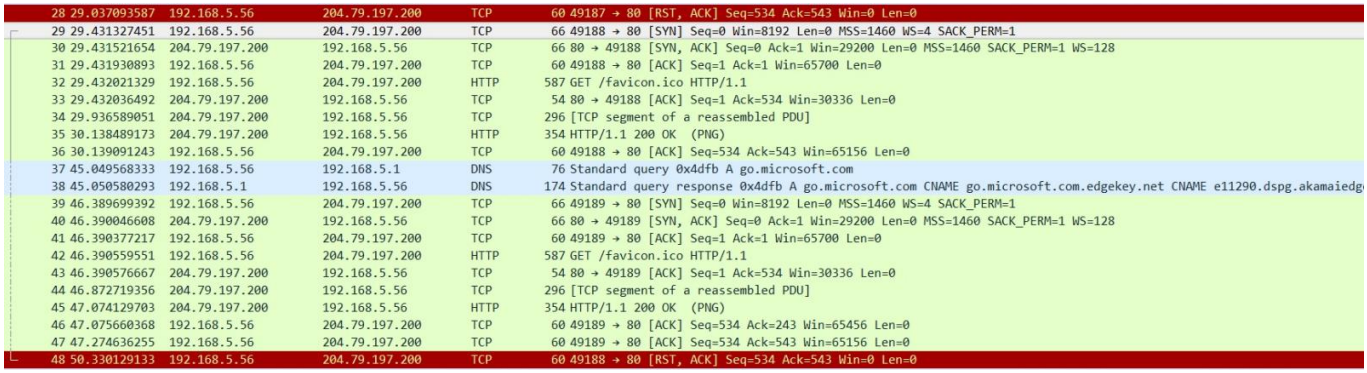
The EK points to DNS which Looks like the good guys redirected already, nice work.

We then see some TCP packets occurring, no8 we see a favicon.ico been requested which is a small icon file (no12 .PNG type)

We then see a heap of SSDP packets(no14+) which is foundation plug and play (local network discovery)

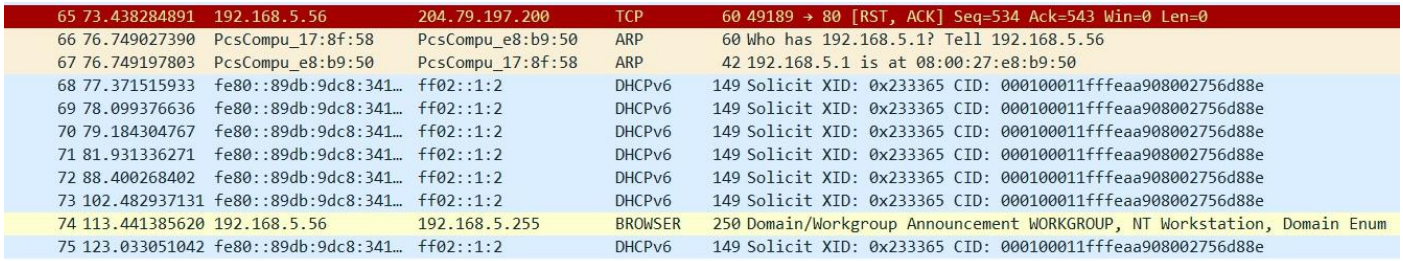
Further down I see another DNS request for an akamaiedge (conference hosting site) session before the Markmonitor sends a RESET ACK request relating to the ICO file mentioned above.

Beyond that it’s a repeat so to me it looks like the good guys are already blocking the EK from hitting its landing page.



28	29.037093587	192.168.5.56	204.79.197.200	TCP	60 49187 → 80 [RST, ACK] Seq=534 Ack=543 Win=0 Len=0
29	29.431327451	192.168.5.56	204.79.197.200	TCP	66 49188 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
30	29.431521654	204.79.197.200	192.168.5.56	TCP	66 80 → 49188 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
31	29.431930893	192.168.5.56	204.79.197.200	TCP	60 49188 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
32	29.432021329	192.168.5.56	204.79.197.200	HTTP	587 GET /favicon.ico HTTP/1.1
33	29.432036492	204.79.197.200	192.168.5.56	TCP	54 80 → 49188 [ACK] Seq=1 Ack=534 Win=30336 Len=0
34	29.936589051	204.79.197.200	192.168.5.56	TCP	296 [TCP segment of a reassembled PDU]
35	30.138489173	204.79.197.200	192.168.5.56	HTTP	354 HTTP/1.1 200 OK (PNG)
36	30.139091243	192.168.5.56	204.79.197.200	TCP	60 49188 → 80 [ACK] Seq=534 Ack=543 Win=65156 Len=0
37	45.049568333	192.168.5.56	192.168.5.1	DNS	76 Standard query 0x4dfb A go.microsoft.com
38	45.050580293	192.168.5.1	192.168.5.56	DNS	174 Standard query response 0x4dfb A go.microsoft.com CNAME go.microsoft.com.edgekey.net CNAME e11290.dspg.akamaiedg
39	46.389699392	192.168.5.56	204.79.197.200	TCP	66 49189 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
40	46.390046608	204.79.197.200	192.168.5.56	TCP	66 80 → 49189 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
41	46.390377217	192.168.5.56	204.79.197.200	TCP	60 49189 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
42	46.390559551	192.168.5.56	204.79.197.200	HTTP	587 GET /favicon.ico HTTP/1.1
43	46.390576667	204.79.197.200	192.168.5.56	TCP	54 80 → 49189 [ACK] Seq=1 Ack=534 Win=30336 Len=0
44	46.872719356	204.79.197.200	192.168.5.56	TCP	296 [TCP segment of a reassembled PDU]
45	47.074129703	204.79.197.200	192.168.5.56	HTTP	354 HTTP/1.1 200 OK (PNG)
46	47.075660368	192.168.5.56	204.79.197.200	TCP	60 49189 → 80 [ACK] Seq=534 Ack=243 Win=65456 Len=0
47	47.274636255	192.168.5.56	204.79.197.200	TCP	60 49189 → 80 [ACK] Seq=534 Ack=543 Win=65156 Len=0
48	50.330129133	192.168.5.56	204.79.197.200	TCP	60 49188 → 80 [RST, ACK] Seq=534 Ack=543 Win=0 Len=0

Beyond that I saw some DHCPv6 traffic and the EK still trying to work out the local network



65	73.438284891	192.168.5.56	204.79.197.200	TCP	60 49189 → 80 [RST, ACK] Seq=534 Ack=543 Win=0 Len=0
66	76.749027390	PcsCompu_17:8f:58	PcsCompu_e8:b9:50	ARP	60 Who has 192.168.5.1? Tell 192.168.5.56
67	76.749197803	PcsCompu_e8:b9:50	PcsCompu_17:8f:58	ARP	42 192.168.5.1 is at 08:00:27:e8:b9:50
68	77.371515933	fe80::89db:9dc8:341…	ff02::1:2	DHCPv6	149 Solicit XID: 0x233365 CID: 000100011fffeaa908002756d88e
69	78.099376636	fe80::89db:9dc8:341…	ff02::1:2	DHCPv6	149 Solicit XID: 0x233365 CID: 000100011fffeaa908002756d88e
70	79.184304767	fe80::89db:9dc8:341…	ff02::1:2	DHCPv6	149 Solicit XID: 0x233365 CID: 000100011fffeaa908002756d88e
71	81.931336271	fe80::89db:9dc8:341…	ff02::1:2	DHCPv6	149 Solicit XID: 0x233365 CID: 000100011fffeaa908002756d88e
72	88.400268402	fe80::89db:9dc8:341…	ff02::1:2	DHCPv6	149 Solicit XID: 0x233365 CID: 000100011fffeaa908002756d88e
73	102.482937131	fe80::89db:9dc8:341…	ff02::1:2	DHCPv6	149 Solicit XID: 0x233365 CID: 000100011fffeaa908002756d88e
74	113.441385620	192.168.5.56	192.168.5.255	BROWSER	250 Domain/Workgroup Announcement WORKGROUP, NT Workstation, Domain Enum
75	123.033051042	fe80::89db:9dc8:341…	ff02::1:2	DHCPv6	149 Solicit XID: 0x233365 CID: 000100011fffeaa908002756d88e

So…. Now what.

Well I asked myself okay the packet capture did not give me a lot. What else can I learn…

We know this EK is using a flash exploit, its using SWF filetype (Shockwave Flash) but what if I actually wanted to know which SWF exploit it was…

Look in the SWF

what you mean "look in the SWF"

I mean open him up

Its not a tin of baked beans! What do you mean "open him up"

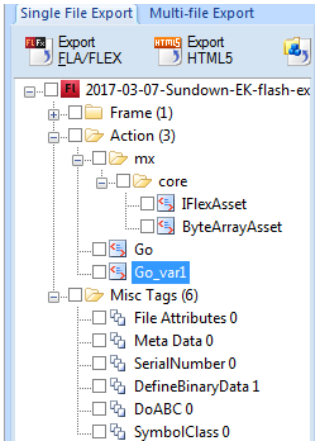
You know what I mean

That's a bit strong ain't it. I don't know about this.

So Decompiler in hand I opened it up looking to try identify the particular exploit it was using. I was drawn first to function “Go” and this line (loc_7) which looked like an overflow attack and in general I was getting a feel for a lot of data been written to memory locations.

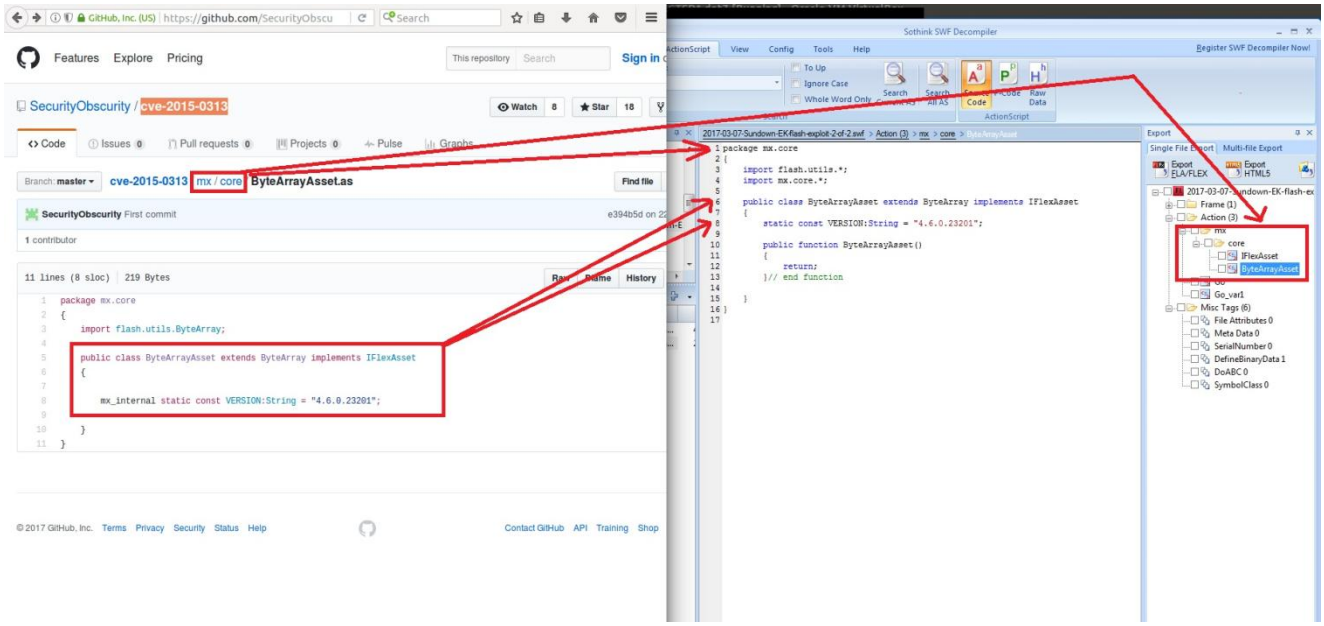
Go_var1 was set to execute GO and also extend the ByteArrayAsset … interesting..

```
package
{
    import mx.core.*;

    public class Go_var1 extends ByteArrayAsset
    {

        public function Go_var1()
        {
            return;
        }// end function

    }
}
```



So this sent me down a path of looking for overflow attacks against byteArrayAssets under the mx core …

OH …. Look a github page showing how CVE-2015-0313 works…



Researching CVE-2015-0313 (Use after free) refers to bytearray been freed from an actionscript worker which can FILL THE MEMORY and notify the main thread to corrupt the new contents. Which then allows for remote code execution.

I concluded at this point that CVE-2015-0313 was been used.

Thankyou

MBYX

No SWF's were harmed in the making, bonus points for the movie reference ☺