Heap Exploitation

USE AFTER FREE AND META DATA ATTACKS
CAFFIX

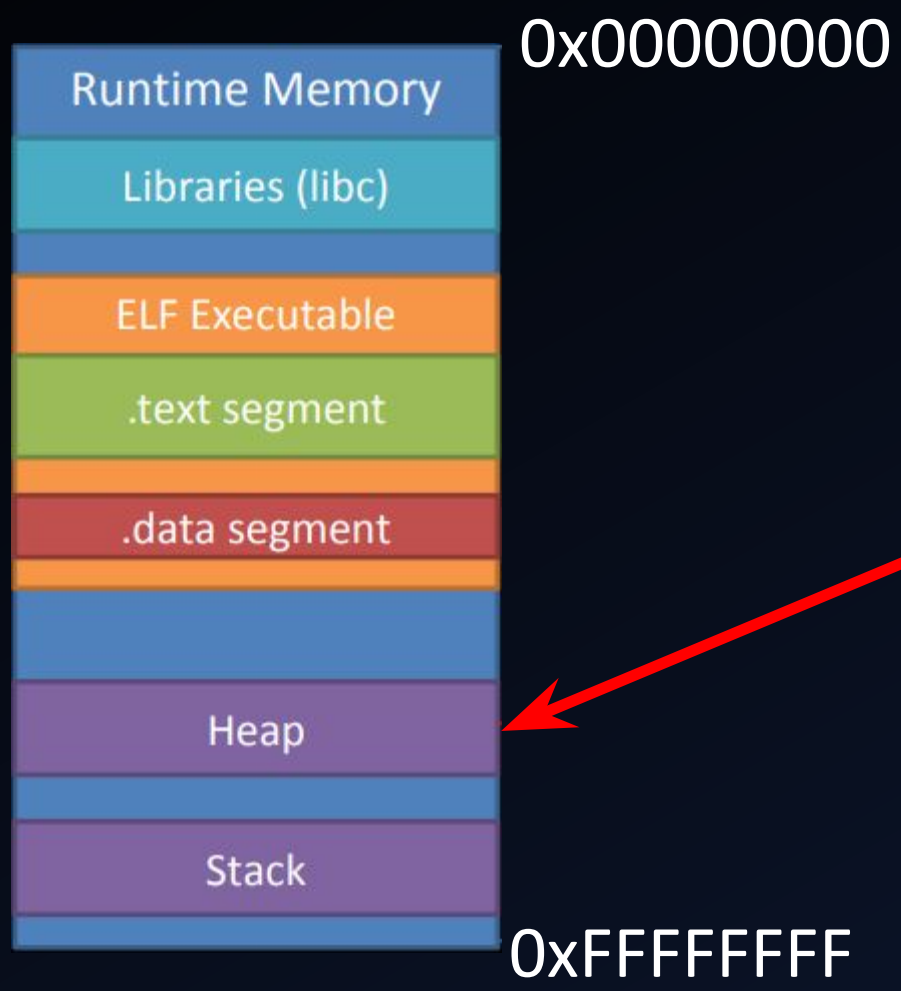
Modern Binary Exploitation – Heap Overview Slides

- These first slides are stolen from RPI's modern binary exploitation course
- More information at http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/17/10_lecture.pdf

The Heap

- The heap is a pool of memory used for dynamic allocations at runtime
  - malloc() grabs memory on the heap
  - free() releases memory on the heap

The Heap



It’s just another segment in runtime memory

Heap vs Stack

HEAP

- Dynamic memory allocations at runtime
- Objects, big buffers, structs, persistence larger things
- Slower, Manual
  - Done by the programmer
  - Malloc/calloc/recalloc/free
  - New/delete

STACK

- Fixed memory allocations known at compile time
- Local variables, return addresses, function args
- Fast, Automatic
  - Done by the compiler
  - Abstracts away any concept of allocating/de-allocating

Heap Implementations

- Tons of different heap implementations
  - dlmalloc
  - ptmalloc
  - tcmalloc
  - jemalloc
  - nedmalloc
  - hoard
- Some applications even create their own heap implementations!

Heap Implementations

- Ubuntu 16.04 currently has glibc 2.23
  - It uses ptmalloc2
  - Very fast, low fragmentation, thread safe

Know Thy Heap

- Everyone uses the heap (dynamic memory) but few usually know much about its interals
- Do you even know the cost of your mallocs?

Malloc Trivia

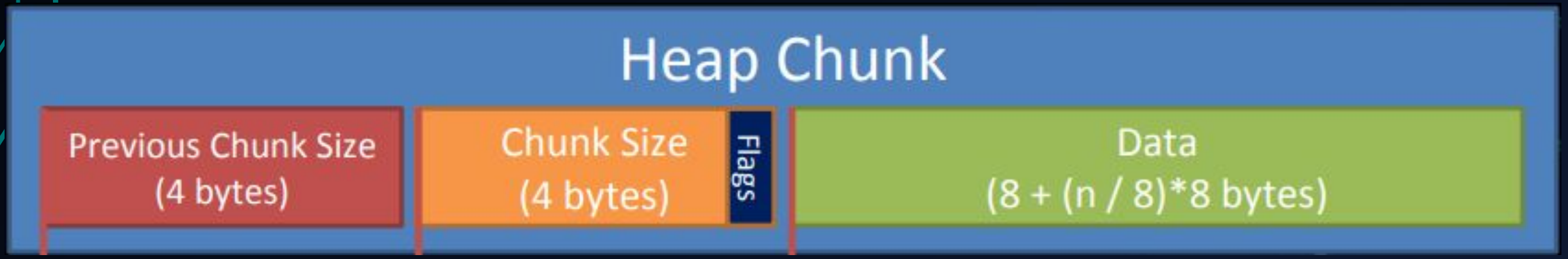
- malloc(32);
  - 40 bytes
- malloc(4);
  - 16 bytes
- malloc(20);
  - 24 bytes
- malloc(0);
  - 16 bytes

How many bytes on the heap are your malloc chunks really taking up?

How many did you get right?
Maybe one? Right?

Heap Chunks

- unsigned int * buffer = NULL;
- buffer = malloc(0x100);



*(buffer-2) *(buffer-1) *buffer

Heap Chunks

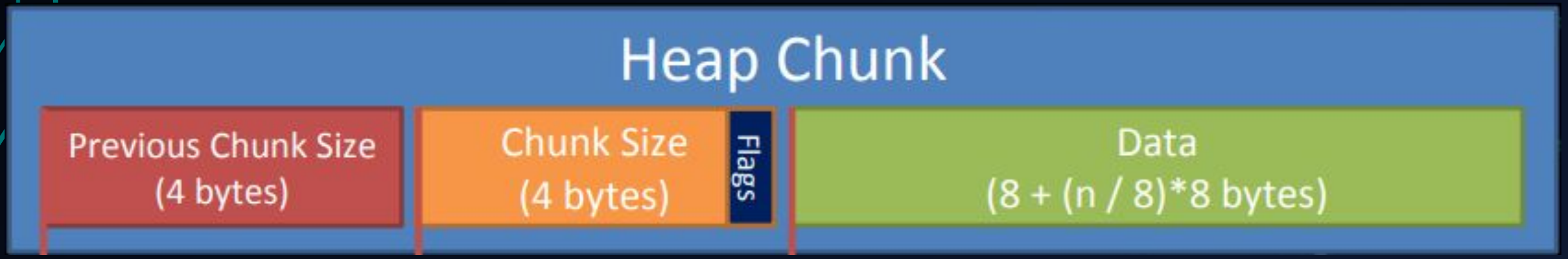
- Flags
  - Because of byte alignment, the lower 3 bits of the chunk size field would always be zero. Instead they are used for flag bits.

0x01 PREV_INUSE – set when previous chunk is in use
0x02 IS_MMAPPED – set if chunk was obtained with mmap()
0x04 NON_MAIN_ARENA – set if chunks belongs to a thread arena

*(buffer-2) *(buffer-1) *buffer

Heap Chunks – In Use

- Heap chunks exist in two states
  - In use(malloc'd)
  - free-'d



*(buffer-2) *(buffer-1) *buffer

Heap Chunks - Freed free(buffer);

- Forward pointer
  - A pointer to the next freed chunk
- Backwards Pointer
  - A pointer to the previous freed chunk

*(buffer-2) *(buffer-1) *buffer

Use After Free

Google

use after free site:exploit-db.com

All | Shopping | News | Videos | Images | More | Settings | Tools

Past month | Sorted by relevance | All results | Clear

Linux Kernel - VMA Use-After-Free via Buggy vmacache_flush_all ...
https://www.exploit-db.com/exploits/45497/
Sep 26, 2018 - Linux Kernel - VMA **Use-After-Free** via Buggy vmacache_flush_all() Fastpath Local Privilege Escalation. CVE-2018-17182. Local exploit for Linux platform.

WebKit - 'WebCore::SVGTextLayoutAttributes::context' Use-After-Free
https://www.exploit-db.com/exploits/45488/
Sep 25, 2018 - WebKit - 'WebCore::SVGTextLayoutAttributes::context' **Use-After-Free**. CVE-2018-4318. Dos exploit for Multiple platform. Tags: **Use After Free** (UAF)

[PDF] CVE-2017-11176: A step-by-step Linux Kernel exploitation - Exploit-DB
https://www.exploit-db.com/.../45551-cve-2017-11176-a-step-by-step-linux-kernel-expl...
Oct 2, 2018 - A reallocation strategy will be presented to turn the **use-after-free** into an ... The technics exposed here are a common way to exploit a **use-after-free** in the Linux ...

[PDF] CVE-2017-11176: A step-by-step Linux Kernel exploitation - Exploit-DB
https://www.exploit-db.com/.../45549-cve-2017-11176-a-step-by-step-linux-kernel-e...
Oct 2, 2018 - To reduce memory leaks in the kernel and to prevent **use-after-free**, most Linux data structures embed a. "ref counter". The refcounter itself is represented with an ...

Microsoft Edge - Sandbox Escape - Exploit-DB
https://www.exploit-db.com/exploits/45502/
Sep 27, 2018 - The host then redirects the request to the broker after checking whether the ... "file://" scheme and it's a folder, IE will just open it using ShellExecuteExW. ... Microsoft Edge - CMarkup::EnsureDeleteCFState **Use-After-Free** (MS15-125) · Skylined.

Basic Use after Free

- char * a = malloc(256);
- char * b = malloc(256);
- strncpy(a, “Hello”,5);
- free(a);
- printf(“%s\n”, a);

Hello

Basic Use after Free

- char * a = malloc(256);
- char * b = malloc(256);
- strncpy(a, “Hello”,5);
- free(a);
- char * c = malloc(256);
- strncpy(c, “Bye!!”,5);
- printf(“%s\n”, a);

Bye!!

Called a dangling pointer in your CS 367 class

Use after free

```c
typedef struct UAFME {
    void (*vulnfunc)();
} UAFME;
```

```c
void good(){
    printf("I AM GOOD :)\n");
}
```

```c
void bad(){
    printf("I AM BAD >:|\n");
}
```

```
int main(int argc, const char * argv[]){
    printf("[i] Allocating a chunk malloc1 holding a UAFME struct\n");
    UAFME *malloc1 = m                    eof(UAFME));
    malloc1->vulnfunc    good;
    printf("[+] UAFME  truct i itialized with size: %lu\n", sizeof(UAFME));
    printf("[i] good at %p\n", good);
    printf("[i] bad at %p\n", bad);

    printf("[i]  alling malo 1's vulnfunc: \n");
        malloc1 >vulnfunc();

    pr  ntf( [i] Free ng malloc1\n");
    fr  e(malloc1);
    pr                ating a chunk malloc2 with 24(Assuming 64bit) byte siz
    // See why malloc(0) reserves 24+8 bytes in 64bit at:
    //  https://sensepost.com/blog/2017/painless-intro-to-the-linux-userlan

    printf("[i] Setting mallo 2 to bad's pointer\n");
    *malloc2 = (long)bad;

    printf( [i] Now calling malloc1 vulnfunc again...\n");
    // Here is where the use-after-free happens
    // as we are using the free malloc1 which
    // was populated with a pointer to bad
    s from %p and malloc2 refs from %p\n", &malloc1,
    malloc1->vulnfunc();
}
```

```
[i] Allocating a chunk malloc1 holding a UAFME struct
[+] UAFME struct initialized with size: 8
[i] good at 0x5618f542e78a
[i] bad at 0x5618f542e79d
[i] Calling malloc1's vulnfunc:
I AM GOOD :)
[i] Freeing malloc1
[i] Allocating a chunk malloc2 with 24(Assuming 64bit) byte size
[i] Setting malloc2 to bad's pointer
[i] Now calling malloc1 vulnfunc again...
[i] malloc1 refs from 0x7ffec33a5838 and malloc2 refs from 0x7ffec33a5840
I AM BAD >:|
```

Challenge Problem: Use After Free

- nc tctf.competitivecyber.club 9999
  - Or just go to the site and see it at https://tctf.competitivecyber.club

Heap metadata attacks

How2Heap

- Fastbin attack
- Unlink
- House of spirit
- Poison null byte
- House of lore
- Overlapping chunk
- House of force
- Unsorted bin attack
- House of orange
- tcache poisoning

Glibc 2.26

Metadata Corruption

- Heap metadata corruption based exploits are usually very involved and require more intimate knowledge of heap internals
- Each type listed on the slide before pretty much has an entire write up on exploiting one tiny edge case of malloc and links for those are on the next slide.

Glibc Metadata Corruption

- http://acez.re/ctf-writeup-hitcon-ctf-2014-stkof-or-modern-heap-overflow/
- http://wapiflapi.github.io/2014/11/17/hacklu-oreo-with-ret2dl-resolve/
- http://phrack.org/issues/66/10.html
- https://dl.packetstormsecurity.net/papers/attack/MallocMaleficarum.txt
- https://sensepost.com/blog/2017/painless-intro-to-the-linux-userland-heap/

Questions?

- Complaints?

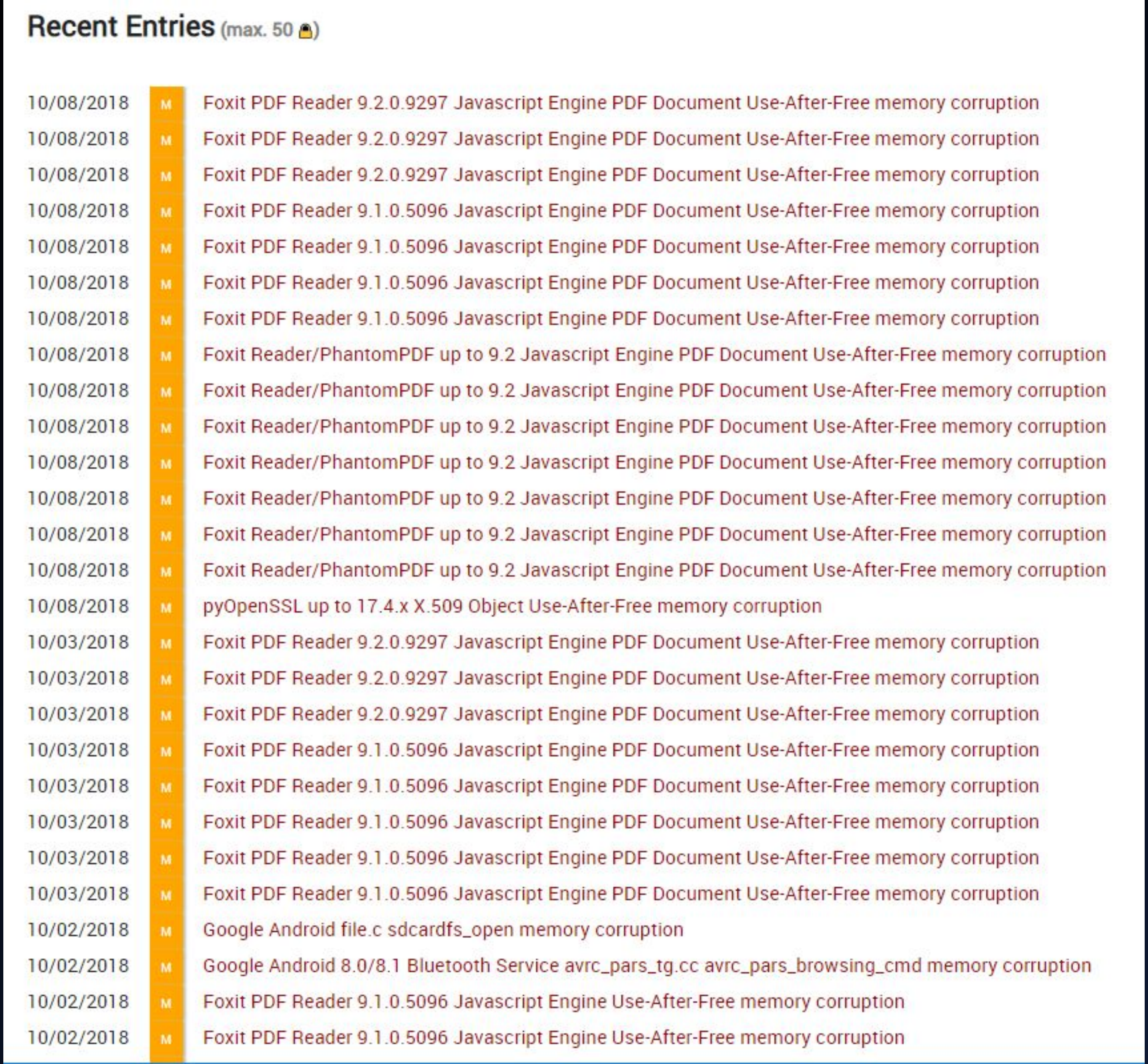
Backup slides

Painless Intro to Heap

- https://sensepost.com/blog/2017/painless-intro-to-the-linux-userland-heap/

Use after free - Sensepost

- https://sensepost.com/blog/2017/linux-heap-exploitation-intro-series-used-and-abused-use-after-free/



Recent Entries (max. 50 🔒)

10/08/2018	M	Foxit PDF Reader 9.2.0.9297 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	Foxit PDF Reader 9.2.0.9297 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	Foxit PDF Reader 9.2.0.9297 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	Foxit PDF Reader 9.1.0.5096 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	Foxit PDF Reader 9.1.0.5096 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	Foxit PDF Reader 9.1.0.5096 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	Foxit PDF Reader 9.1.0.5096 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	Foxit Reader/PhantomPDF up to 9.2 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	Foxit Reader/PhantomPDF up to 9.2 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	Foxit Reader/PhantomPDF up to 9.2 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	Foxit Reader/PhantomPDF up to 9.2 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	Foxit Reader/PhantomPDF up to 9.2 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	Foxit Reader/PhantomPDF up to 9.2 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	Foxit Reader/PhantomPDF up to 9.2 Javascript Engine PDF Document Use-After-Free memory corruption
10/08/2018	M	pyOpenSSL up to 17.4.x X.509 Object Use-After-Free memory corruption
10/03/2018	M	Foxit PDF Reader 9.2.0.9297 Javascript Engine PDF Document Use-After-Free memory corruption
10/03/2018	M	Foxit PDF Reader 9.2.0.9297 Javascript Engine PDF Document Use-After-Free memory corruption
10/03/2018	M	Foxit PDF Reader 9.2.0.9297 Javascript Engine PDF Document Use-After-Free memory corruption
10/03/2018	M	Foxit PDF Reader 9.1.0.5096 Javascript Engine PDF Document Use-After-Free memory corruption
10/03/2018	M	Foxit PDF Reader 9.1.0.5096 Javascript Engine PDF Document Use-After-Free memory corruption
10/03/2018	M	Foxit PDF Reader 9.1.0.5096 Javascript Engine PDF Document Use-After-Free memory corruption
10/03/2018	M	Foxit PDF Reader 9.1.0.5096 Javascript Engine PDF Document Use-After-Free memory corruption
10/03/2018	M	Foxit PDF Reader 9.1.0.5096 Javascript Engine PDF Document Use-After-Free memory corruption
10/02/2018	M	Google Android file.c sdcardfs_open memory corruption
10/02/2018	M	Google Android 8.0/8.1 Bluetooth Service avrc_pars_tg.cc avrc_pars_browsing_cmd memory corruption
10/02/2018	M	Foxit PDF Reader 9.1.0.5096 Javascript Engine Use-After-Free memory corruption
10/02/2018	M	Foxit PDF Reader 9.1.0.5096 Javascript Engine Use-After-Free memory corruption