NIPS

功能简介

- 实现了对常见攻击流量的检测和防御，规则与代码分离，各项参数可以在配置文件中配置，规则和配置文件均为，支持前置规则，阈值预警。
- 可以根据预警级别，对特定ip进行一定时间的封禁或对丢弃特定流量包，具备主动响应和ips功能。
- 并对命中的规则，触发预警的规则，主动响应情况和主动丢包情况分别进行了日志记录。
- 提供了交互配置模式，便于非专业人员修改配置文件或增减规则。配置文件和规则文件均为json格式。
- 默认规则文件中，提供了常见http，mysql，redis，ssh，icmp，dns攻击流量的检测，以及一些常见cve漏洞攻击流量的检测

使用说明

基础使用

初次启动

```
python3 nids.py
```

初次使用需要初始化ip白名单，此处可以填入ip列表，在白名单中的ip将不受主动响应和ips模块的限制

```
[root@centos1 nids_v9]# python3 nids.py
配置文件加载成功
检查规则id是否重复
规则id无重复,开始加载规则文件
规则文件加载成功
白名单主机未初始化,请输入当前ids检测的白名单ip列表,ip之间用逗号分隔:
192.168.248.172
ip白名单初始化为192.168.248.172,重启后生效
```

普通启动

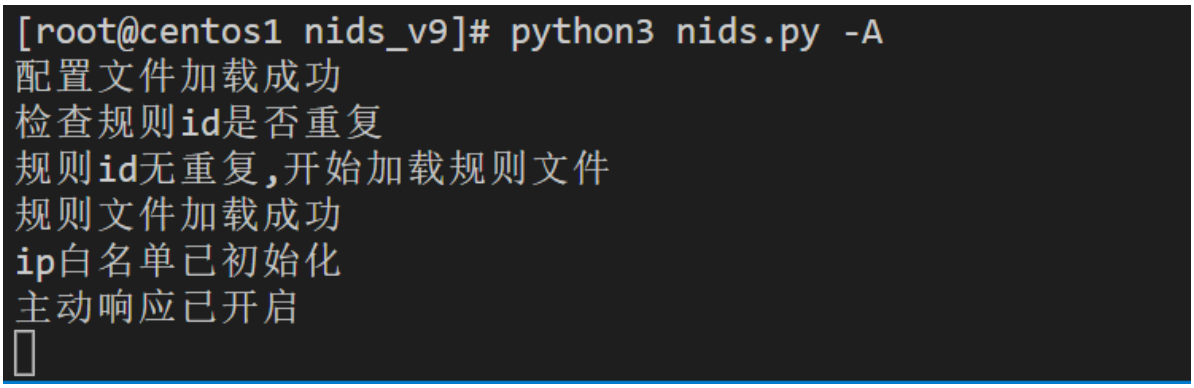
```
python3 nids.py
```

ip白名单初始化之后，就可以正常检测流量了，其中主动响应和ips功能是默认关闭的，如下，可以在配置文件中进行配置，或通过交互模式配置

携带额外参数启动

开启主动响应模块

```
python3 nids.py -A
```



开启NFQ ips模块

```
python3 nids.py -N
```

```
^C[root@centos1 nids_v9]# python3 nids.py -N
配置文件加载成功
检查规则id是否重复
规则id无重复,开始加载规则文件
规则文件加载成功
ip白名单已初始化
nfq已开启
```

进入交互配置模式

```
1 python3 nids.py -I
```

```
-----------------------------------
------------NIDS交互模式------------

-----你可以在这里进行交互式配置-----
---------重新启动NIDS生效-----------

****************注意****************
***错误的配置会导致程序无法正常启动***
如果无法正常启动,请执行reset.py重置配置
************************************

1.修改过滤条件
2.开启主动响应
3.关闭主动响应
4.添加规则
5.删除规则
6.恢复初始配置
7.修改协议端口
8.修改ip白名单
9.退出
-----------------------------------
请输入你的选项:
```

交互配置模式

可以交互式的对配置文件和规则文件进行修改

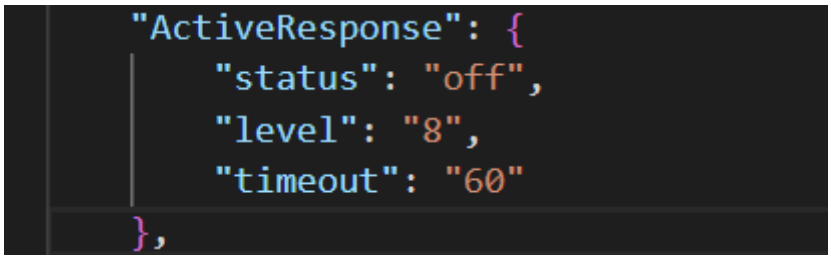
修改过滤条件

可以修改监听流量的协议和端口类型，对应配置文件中的filter节点，支持BPF语法。默认如下：

注意，只有在这里进行了监听，且配置了相应规则的协议，才会被正常检测

主动响应选项

开启和关闭，对应了规则文件中的ActiveResponse节点，默认关闭



开启时，默认8级及以上的规则触发主动响应，封禁对应ip60秒，这两个参数仅支持在配置文件直接修改

增加规则

在指定规则文件中添加规则，有详细的引导过程。对于专业用户，建议直接手动在规则文件中添加，注意，规则中的中文会被unicode编码，不影响最终的使用。

```
[root@centos1 nids_v8]# python3 nids.py -I
配置文件加载成功
检查规则id是否重复
规则id无重复,开始加载规则文件
规则文件加载成功
ip白名单已初始化
-----------------------------------
------------NIDS交互模式------------

-----你可以在这里进行交互式配置-----
---------重新启动NIDS生效-----------

****************注意****************
***错误的配置会导致程序无法正常启动***
如果无法正常启动,请执行reset.py重置配置
************************************

1.修改过滤条件
2.开启主动响应
3.关闭主动响应
4.添加规则
5.删除规则
6.恢复初始配置
7.修改协议端口
8.修改ip白名单
9.退出
-----------------------------------
请输入你的选项:
4
当前配置中，有如下规则文件
规则"icmp":./rules/icmp.json
规则"tcp":./rules/tcp.json
规则"http_req":./rules/http_req.json
规则"http_resp":./rules/http_resp.json
规则"mysql_req":./rules/mysql_req.json
规则"mysql_resp":./rules/mysql_resp.json
规则"ssh":./rules/ssh.json
规则"redis":./rules/redis.json
规则"dns":./rules/dns.json
请输入要添加规则的规则文件路径(Q退出):
./rules/http_req.json
当前增加规则的文件:./rules/http_req.json
已将当前规则文件备份至./rules/http_req.json.back
请输入新规则的名称:
test_rule
请输入规则基础选项,以下各项为必填项,不能为空:

规则id(r_id)
规则预警等级(level)
规则预警信息(alert)
规则匹配正则(regex)
反向匹配正则标志(regex_not)
预警标志(do_alert)

注意:规则id为任意整数,需要在所有规则中全局唯一
```

```
请输入新规则的id(r_id):
999
请输入规则预警等级(level):
6
请输入规则预警信息(alert):
test alert
请输入规则匹配正则(regex):
php://filter|php%3A%2F%2Ffilter|php://input|php%3A%2F%2Finput|phar://
%3A%2F%2F
是否进行反向正则匹配? y

该规则是否屏蔽预警? y

是否配置可选的额外选项?  y
y
前置规则选项(match_id)
前置规则有效时间段(match_time)
当前规则阈值匹配次数(freq_num)
当前规则阈值匹配时间段(freq_time)
当前规则阈值匹配屏蔽时间(freq_noalert)
规则描述(check_info)
注意:
match_time需要在match_id有效时才有效
freq_num,freq_time,freq_noalert需要在三个选项均进行配置的情况下才会生

请输入前置规则的id(match_id),如果没有前置规则,请忽视该项

请输入阈值匹配次数(freq_num),如果不进行阈值匹配,请忽略该项

请输入规则描述(check_info),如果不添加规则描述,请忽略该项
info
规则描述配置成功
新规则内容如下,将添加至./rules/http_req.json:
{'r_id': '999', 'level': '6', 'alert': 'test alert', 'regex': 'php://
%2Finput|phar://|phar%3A%2F%2F|zip://|zip%3A%2F%2F|data://|data%3A%2F
_id': '', 'match_time': '', 'freq_num': '', 'freq_time': '', 'freq_no
请确认新规则是否正确? y
y
成功将新规则添加至./rules/http_req.json


当前配置中，有如下规则文件
规则"icmp":./rules/icmp.json
规则"tcp":./rules/tcp.json
规则"http_req":./rules/http_req.json
规则"http_resp":./rules/http_resp.json
规则"mysql_req":./rules/mysql_req.json
规则"mysql_resp":./rules/mysql_resp.json
规则"ssh":./rules/ssh.json
规则"redis":./rules/redis.json
规则"dns":./rules/dns.json
请输入要添加规则的规则文件路径(Q退出):
Q
已退出修改规则模块
----------------------------------
```

```
------------NIDS交互模式------------

-----你可以在这里进行交互式配置-----
---------重新启动NIDS生效-----------

*****************注意****************
***错误的配置会导致程序无法正常启动***
如果无法正常启动,请执行reset.py重置配置
************************************

1.修改过滤条件
2.开启主动响应
3.关闭主动响应
4.添加规则
5.删除规则
6.恢复初始配置
7.修改协议端口
8.修改ip白名单
9.退出
-------------------------------------
请输入你的选项:
9
```


```
    "test_rule": {
        "r_id": "999",
        "level": "6",
        "alert": "test alert",
        "regex": "php://filter|php%3A%2F%2Ffilter|php://input|php%3A%2F%2Finput|phar://|phar%3A%2F%2F|zip://|zip%3A%2F%2F|
        data://|data%3A%2F%2F",
        "regex_not": "0",
        "do_alert": "1",
        "match_id": "",
        "match_time": "",
        "freq_num": "",
        "freq_time": "",
        "freq_noalert": "",
        "check_info": "info"
    }
}
```


删除规则

删除指定规则文件中特定id的规则，规则id不存在时不会进行任何操作

```
--------------------------------------
请输入你的选项:
5
当前配置中，有如下规则文件
规则"icmp":./rules/icmp.json
规则"tcp":./rules/tcp.json
规则"http_req":./rules/http_req.json
规则"http_resp":./rules/http_resp.json
规则"mysql_req":./rules/mysql_req.json
规则"mysql_resp":./rules/mysql_resp.json
规则"ssh":./rules/ssh.json
规则"redis":./rules/redis.json
规则"dns":./rules/dns.json
请输入要删除规则的规则文件路径(Q退出):
./rules/http_req.json
当前删除规则的文件:./rules/http_req.json
已将当前规则文件备份至./rules/http_req.json.back
请输入需要删除的规则id:
999
成功删除./rules/http_req.json中,id为999的规则
```

恢复初始配置

会调用同目录下的do_reset.py，会重置配置文件

修改协议端口

修改配置配置文件中的协议对应的端口，这个配置决定了特定协议的规则会作用于哪些端口。目前支持http，mysql，ssh，redis，dns五种协议

```
    "protocol_port": {
        "tcp": {
            "http": "80;8080;8888",
            "mysql": "3306",
            "ssh": "22",
            "redis": "6379"
        },
        "udp": {
            "dns": "53"
        }
    },
```

```
7
请输入对应协议的端口列表,多个端口之间用;分隔
默认支持对如下协议进行端口设置:
http/tcp,mysql/tcp,ssh/tcp,redis/tcp,dns/udp

请输入http的端口列表:
80;8080;8888
请输入mysql的端口列表:
3306;3307
请输入ssh的端口列表:
22
请输入dns的端口列表:
53
请输入redis的端口列表:
6379;6890
新的协议端口对应关系如下:
{'tcp': {'http': '80;8080;8888', 'mysql': '3306;3307', 'ssh': '22', 'redis': '6379;6890'}, 'udp': {'dns': '53'}},是否确认修改? y
y
协议端口修改成功
```

端口配置出现冲突或非数字时，会提示

```
7
请输入对应协议的端口列表,多个端口之间用;分隔
默认支持对如下协议进行端口设置:
http/tcp,mysql/tcp,ssh/tcp,redis/tcp,dns/udp

请输入http的端口列表:
80;8080;8888;8888
请输入mysql的端口列表:
3306;8888
请输入ssh的端口列表:
22
请输入dns的端口列表:
53
请输入redis的端口列表:
6379
出现重复端口或端口号不合法,修改协议端口失败
```

修改ip白名单

可以覆盖修改ip白名单

```
请输入你的选项:
8
当前的ip白名单列表为:
192.168.248.172
请输入新的ip白名单列表,ip之间用逗号隔开

注意:新的ip白名单会覆盖旧的ip白名单:
192.168.248.172,192.168.248.1
修改ip白名单成功,重启后生效
```

检测攻击流量功能演示

普通规则预警

规则如下，为检测sql注入规则


```
"http_sql_injection": {
    "r_id": "2",
    "level": "8",
    "regex": ".+(((@@|%40%40)(datadir|version))|((user|database|version)(\\(\\)|%28%29))|(union.+select.+)|(select.+from.
    +(information_schema|mysql)\\.\\w+.+(group|where|limit)*)|updatexml|extractvalue|load_file|into.+outfile).+",
    "regex_not": "0",
    "alert": "\u89e6\u53d1\u89c4\u5219ID:2,\u7591\u4f3cSQL\u6ce8\u5165",
    "do_alert": "1",
    "match_id": "",
    "match_time": "",
    "freq_num": "",
    "freq_time": "",
    "freq_noalert": "",
    "check_info": "SQL\u6ce8\u5165\u89c4\u5219"
},
```


测试payload

```
updatexml
```

预警信息

```
2023-04-10 11:30:41  [id:2|level:8]  触发规则ID:2,疑似SQL注入  192.168.248.1:7833<->192.168.248.172:80
```

日志

```
2023-04-10 11:30:41  [id:2|level:8]  触发规则ID:2,疑似SQL注入  192.168.248.1:7833<->192.168.248.172:80
```

前置规则预警

规则如下

```
    "http_AntSword-chr-1": {
        "r_id": "17",
        "level": "1",
        "regex": "post[\\s\\S]+content-length:\\s*\\d{4,}[\\s\\S]+(\\.ChR\\(\\d+\\)){100,}",
        "regex_not": "0",
        "alert": "触发规则ID:17,疑似蚁剑-chr编码流量-1",
        "do_alert": "0",
        "match_id": "",
        "match_time": "",
    "freq_num": "",
    "freq_time": "",
    "freq_noalert": "",
        "check_info": "蚁剑-chr-1"
    },
    "http_AntSword-chr-2": {
        "r_id": "18",
```

```
        "level": "5",
        "regex": "ChR\\(105\\)\\.ChR\\(110\\)\\.ChR\\
(105\\)\\.ChR\\(95\\)\\.ChR\\(115\\)\\.ChR\\(101\\)\\.ChR\\
(116\\)\\.ChR\\(40\\)\\.ChR\\(34\\)\\.ChR\\(100\\)\\.ChR\\
(105\\)",
        "regex_not": "0",
        "alert": "触发规则ID:18,疑似蚁剑-chr编码流量",
        "do_alert": "1",
        "match_id": "17",
        "match_time": "10",
    "freq_num": "1",
    "freq_time": "5",
    "freq_noalert": "10",
        "check_info": "蚁剑-chr-2"
    },
```

使用蚁剑连接webshell

规则18的前置规则为规则17，当命中规则17的10秒内，命中规则18，则会预警

```
2023-04-10 12:42:36  [id:18|level:5]  触发规则ID:18,疑似蚁剑-chr编码流量  192.168.248.1:11815<->192.168.248.172:80
```

在check.csv中可以看到命中情况

```
2023-04-10 12:42:36,17,1,"触发规则ID:17,疑似蚁剑-chr编码流量-1",192.168.248.1,11815,192.168.248.172,80,1
2023-04-10 12:42:36,18,5,"触发规则ID:18,疑似蚁剑-chr编码流量",192.168.248.1,11815,192.168.248.172,80,1
```

日志中只记录预警的规则

```
2023-04-10 12:42:36  [id:18|level:5]  触发规则ID:18,疑似蚁剑-chr编码流量  192.168.248.1:11815<->192.168.248.172:80
```

阈值规则预警

规则如下

```
"redis_login_bf": {
    "r_id": "500",
    "level": "5",
    "regex": "\\$\\d+[\\s\\S]+AUTH[\\s\\S]+\\$\\d+[\\s\\S]+\\d+",
    "regex_not": "0",
    "alert": "触发规则ID:500,连续出现redis登录，疑似redis登录爆破",
    "do_alert": "1",
    "match_id": "",
    "match_time": "",
    "freq_num": "5",
    "freq_time": "20",
    "freq_noalert": "60",
    "check_info": "redis登录爆破"
  }
```

该规则匹配的是redis的认证特征，这里作为示例，当20秒内命中5次以上该规则时，进行预警，然后60秒内不再预警

```
2023-04-10 12:51:02  [id:500|level:5]  触发规则ID:500,连续出现redis登录，疑似redis登录爆破  192.168.248.149:41328<->
192.168.248.172:6379
```

check.csv中查看命中情况

命中时，is_hint为0，当到达阈值时，is_hint为1，触发预警

```
logtime,r_id,r_level,r_alert,ip_src,port_src,ip_dst,port_dst,is_hit
2023-04-10 12:50:47,500,5,"触发规则ID:500,连续出现redis登录，疑似redis登录爆破",192.168.248.149,41328,192.168.248.172,6379,0
2023-04-10 12:50:50,500,5,"触发规则ID:500,连续出现redis登录，疑似redis登录爆破",192.168.248.149,41328,192.168.248.172,6379,0
2023-04-10 12:50:53,500,5,"触发规则ID:500,连续出现redis登录，疑似redis登录爆破",192.168.248.149,41328,192.168.248.172,6379,0
2023-04-10 12:51:00,500,5,"触发规则ID:500,连续出现redis登录，疑似redis登录爆破",192.168.248.149,41328,192.168.248.172,6379,0
2023-04-10 12:51:02,500,5,"触发规则ID:500,连续出现redis登录，疑似redis登录爆破",192.168.248.149,41328,192.168.248.172,6379,1
```

日志中仅记录预警情况

```
2023-04-10 12:51:02  [id:500|level:5]  触发规则ID:500,连续出现redis登录，疑似redis登录爆破  192.168.248.149:41328<->192.168.248.
172:6379
```

主动响应

主动响应模式默认关闭，开启后可以在触发特定级别及以上的预警时，调用iptables封禁触发预警的ip，一段时间后自动解封，包含入站和出站流量。白名单中的ip不会被封禁。

可以在配置文件中如下节点进行配置，默认8级及以上，封禁60秒：

```
"ActiveResponse": {
        "status": "off",
        "level": "8",
        "timeout": "60"
    },
```

或启动时携带`-A`参数(仅当次运行生效)

这里使用如下规则进行演示，测试payload为：`updatexml`

```
"http_sql_injection": {
        "r_id": "2",
        "level": "8",
        "regex": ".+((((@@|%40%40)(datadir|version))|
((user|database|version)(\\(\\)|%28%29))|(union.+select.+)|
(select.+from.+(information_schema|mysql)\\.\\w+.+
(group|where|limit)*)|updatexml|extractvalue|load_file|into.+outfi
le).+",
```

```
        "regex_not": "0",
        "alert": "触发规则ID:2,疑似SQL注入",
        "do_alert": "1",
        "match_id": "",
        "match_time": "",
    "freq_num": "1",
    "freq_time": "5",
    "freq_noalert": "10",
        "check_info": "SQL注入规则"
    },
```

预警信息

```
2023-04-10 13:07:54  [id:2|level:8]  触发规则ID:2,疑似SQL注入  192.168.248.149:43840<->192.168.248.172:80
```

此时该ip的入站和出站都被禁止了

```
[root@centos1 redis-6.2.7]# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  192.168.248.149      0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  0.0.0.0/0            192.168.248.149
```

在active_response.log中记录了主动响应情况

```
168.248.149]被禁止入站60秒
['2023-04-10 13:07:54', '2', '8', '触发规则ID:2,疑似SQL注入', '192.168.248.149', 43840, '192.168.248.172', 80, '1']----源ip:[192.
168.248.149]被禁止入站60秒
['2023-04-10 13:07:54', '2', '8', '触发规则ID:2,疑似SQL注入', '192.168.248.149', 43840, '192.168.248.172', 80, '1']----目的ip:
[192.168.248.149]被禁止出站60秒
['2023-04-10 13:07:54', '2', '8', '触发规则ID:2,疑似SQL注入', '192.168.248.149', 43840, '192.168.248.172', 80, '1']----源ip:[192.
168.248.149]解除入站限制
['2023-04-10 13:07:54', '2', '8', '触发规则ID:2,疑似SQL注入', '192.168.248.149', 43840, '192.168.248.172', 80, '1']----目的ip:
[192.168.248.149]解除出站限制
```

60秒之后，可以看到ip已解封：

```
[root@centos1 redis-6.2.7]# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

NetFilterQueue

NFQ模式默认关闭，可以在配置文件中修改如下节点开启：

```
"netfilterqueue": {
    "status": "off",
    "level": "6"
},
```

默认为6级及以上的预警触发时，丢弃触发规则的流量。目前nfq仅支持http，mysql，ssh，redis，dns和icmp协议。

也可以通过携带`-N`参数启动(仅本次启动生效)

开启后，对应协议的流量均会交由NFQ处理：

```
[root@centos1 redis-6.2.7]# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
NFQUEUE    icmp --  0.0.0.0/0            0.0.0.0/0            NFQUEUE num 1
NFQUEUE    udp  --  0.0.0.0/0            0.0.0.0/0            udp dpt:53 NFQUEUE num 1
NFQUEUE    tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:6379 NFQUEUE num 1
NFQUEUE    tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:3306 NFQUEUE num 1
NFQUEUE    tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:8888 NFQUEUE num 1
NFQUEUE    tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:8080 NFQUEUE num 1
NFQUEUE    tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:80 NFQUEUE num 1

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
NFQUEUE    icmp --  0.0.0.0/0            0.0.0.0/0            NFQUEUE num 1
NFQUEUE    udp  --  0.0.0.0/0            0.0.0.0/0            udp spt:53 NFQUEUE num 1
NFQUEUE    tcp  --  0.0.0.0/0            0.0.0.0/0            tcp spt:6379 NFQUEUE num 1
NFQUEUE    tcp  --  0.0.0.0/0            0.0.0.0/0            tcp spt:3306 NFQUEUE num 1
NFQUEUE    tcp  --  0.0.0.0/0            0.0.0.0/0            tcp spt:8888 NFQUEUE num 1
NFQUEUE    tcp  --  0.0.0.0/0            0.0.0.0/0            tcp spt:8080 NFQUEUE num 1
NFQUEUE    tcp  --  0.0.0.0/0            0.0.0.0/0            tcp spt:80 NFQUEUE num 1
```

此处使用如下规则进行演示：

```
"http_sql_injection": {
        "r_id": "2",
        "level": "8",
        "regex": ".+((((@@|%40%40)(datadir|version))|
((user|database|version)(\\(\\)|%28%29))|(union.+select.+)|
(select.+from.+(information_schema|mysql)\\.\\w+.+
(group|where|limit)*)|updatexml|extractvalue|load_file|into.+outfi
le).+",
        "regex_not": "0",
        "alert": "触发规则ID:2,疑似SQL注入",
        "do_alert": "1",
        "match_id": "",
        "match_time": "",
    "freq_num": "",
```

```
    "freq_time": "",
    "freq_noalert": "",
        "check_info": "SQL注入规则"
    },
```

预警信息

```
2023-04-10 13:23:12  [id:2|level:8]  触发规则ID:2,疑似SQL注入  192.168.248.1:14145<->192.168.248.172:80
2023-04-10 13:23:12  [id:2|level:8]  触发规则ID:2,疑似SQL注入  192.168.248.1:14145<->192.168.248.172:80
2023-04-10 13:23:13  [id:2|level:8]  触发规则ID:2,疑似SQL注入  192.168.248.1:14145<->192.168.248.172:80
2023-04-10 13:23:13  [id:2|level:8]  触发规则ID:2,疑似SQL注入  192.168.248.1:14145<->192.168.248.172:80
```

可以在nfq.log中查看nfq情况

```
2023-04-10 13:23:12  [id:2|level:8]  触发规则ID:2,疑似SQL注入  192.168.248.1:14145<->192.168.248.172:80--触发nfq,已drop
2023-04-10 13:23:13  [id:2|level:8]  触发规则ID:2,疑似SQL注入  192.168.248.1:14145<->192.168.248.172:80--触发nfq,已drop
2023-04-10 13:23:13  [id:2|level:8]  触发规则ID:2,疑似SQL注入  192.168.248.1:14145<->192.168.248.172:80--触发nfq,已drop
```