12_security02zabbix 监控报警机制+zabbix 进阶操作

一 Zabbix 监控报警机制

1.1 Zabbix 监控报警机制-基本概念

自定义的监控项默认不会自动报警

首页也不会提示错误

需要配置触发器与报警动作才可以自动报警

1.2 Zabbix 监控报警机制-触发器

1.2.1 创建触发器

触发器(trigger)

表达式,如果内存不足 300M,用户超过 30 个等

当错发条件发生后,会导致一个触发事件

触发时会执行某个动作

动作(Action)

触发器的条件被触发后的行为

可以是发送邮件,也可以是重启某个服务等

1.2.2 触发器表达式

Expression 表达式:触发异常的条件

{<server>:<key>.<function>(<parameter>)}<operator><constant>

{主机:key.函数(参数)}<表达式>常数

{web1:system.cpu.load[all,avg1].last(0)}>5 #0 表示最新数据

如果 web1 主机最新的 CPU 平均负载值大于 5,则触发器状态 Problem

{vfs.fs.size[/,free].max(5m)}<10G #5m 表示最近 5 分钟

根分区,最近 5 分钟的最大容量小于 10G,则状态进入 Problem

{vfs.file.cksum[/etc/passwd].diff(0)}>0 #0 为最新数据

最新一次校验/etc/passwd 如果与上一次有变化，则状态进入 Problem

1.2.3 配置触发器

通过配置->模板,选择模板点击后面的 Triggers->Create trigger

名称 cfq1;严重性 灾难;

表达式->添加->监控项 ATMP:mon_web100_user;功能->下拉三角->最新 T 值>N->N 值输入 24->插入,生成表达式;

最后点击最下面的添加

强烈建议使用英文创建

删除触发器时,必须先停用触发器,再勾选该触发器,点击删除

1.3 设置邮件服务器 192.168.2.5

zabbixserver ~]# ss -antulp | grep 25 #检查邮件端口

zabbixserver ~]# yum -y install postfix

```
abbixserver ~]# systemctl start postfix
zabbixserver ~]# ss -antulp | grep 25  #再次检查邮件端口

zabbixserver ~]# echo "127.0.0.1 zabbixserver" >> /etc/hosts
      #添加域名解析

zabbixserver ~]# yum list | grep -i mail
zabbixserver ~]# yum -y install mailx.x86_64
```

安装完邮件后,测试发送邮件

1.3.1 创建 Media

通过 Administration（管理）-->Media Type（报警媒体类型）-->选择 Email（邮件）-->报警媒介类型

SMTP 服务器 localhost;SMTP 电邮 root@localhost

点击更新,返回报警媒介类型页面,启用 Email

在 Administration（管理）-->Users（用户）中找到选择 admin 账户

点击报警媒介标签页,报警媒介->添加,打开新的报警媒介页面

收件人 zabbix@localhost,点击添加

返回报警媒介页面,点击更新

1.3.2 创建 Action

通过 Configuration（配置）-->Actions（动作）-->Create action（创建动作），

动作标签页:名称 act1,新的触发条件 触发器名称 似 cfq1,添加

操作标签页:操作->新的->操作类型 发送消息->发送到用户,添加,选择 Admin->添加->点击最下面的添加

1.3.3 效果测试

1.3.3.1 web1 上添加用户,直到数量>24

```
web1 ~]# wc -l /etc/passwd
25 /etc/passwd
```

1.3.3.2 监控服务器上切换到用户 zabbix,收邮件

```
abbixserver ~]# su - zabbix
zabbixserver ~]$ mail    #查看 zabbix 的邮件
```

1.3.3.3 登录监控页面查看状态

监测中-仪表板-问题

排错思路

检查触发器表达式和配置,若错误删除重建

检查邮件服务器,检查 Media,Action

二 Zabbix 进阶操作-自动发现

2.1 概述

2.1.1 自动发现(Discovery)

当 Zabbix 需要监控的设备越来越多,手动添加监控设备越来越有挑战,此时,可以考虑使用自动发现功能

2.1.2 自动发现可以实现

自动 发现主机\添加主机\添加主机到组\连接模板等

2.2 自动发现流程

2.2.1 创建自动发现规则

登录监控页面,配置-自动发现-创建发现规则

名称 自定义 fxrule1;IP 范围 192.168.2.1-254 ;更新间隔 1m;

检查-新的,检查类型 HTTP,端口 80,添加;点击最下方添加,回到自动发现规则页面

2.2.2 创建 Action 动作(发现主机后自动执行什么动作)

配置-动作-(事件源:自动发现)创建动作

动作标签页:名称 act2;新的触发条件 主机 IP 地址=192.168.2.1-254,添加

2.2.3 通过动作,执行添加主机,链接模板到主机等操作

操作标签页:操作-新的;操作类型 添加到主机群组;主机群组 选择 Linux servers,添加

操作标签页:操作-新的;操作类型 与模板关联;模板 选择 ATMP,添加

最后点击最下面的添加

修改 192.168.2.200 的配置文件,创建用户 zabbix,并重启 zabbix_agentd

```
web2 ~]# vim /usr/local/etc/zabbix_agentd.conf
93 Server=127.0.0.1,192.168.2.5
134 ServerActive=192.168.2.5:10051
web2 local]# useradd zabbix

web2 ~]# killall -9 zabbix_agentd
zabbix_agentd: no process found
web2 ~]# zabbix_agentd
```

三 Zabbix 进阶操作-主被动监控

3.1 概述

3.1.1 主动和被动都是对被监控主机而言的

3.1.2 默认 zabbix 采用的是被动监控

被动监控:server 向 agent 发起连接,发送监控 key,agent 接收请求,响应监控数据

主动监控:agent 向 server 发起连接,agent 请求需要检查的监控项目列表,server 响应 agent 发送一个 items 列表,agent 确认收到监控列表,TCP 连接完成,会话关闭,agent 开始周期性的收集数据

区别:server 不用每次需要数据都连接 agent,agent 会自己收集数据并处理数据,server 仅需要保存数据即可

当监控主机达到一定量级后,Zabbix 服务器会越来越慢

此时,可以考虑

3.2 添加被监控新主机(192.168.2.201-web3)

```
room9pc01 ~]$ scp -r /linux-soft/03/Zabbix/
root@192.168.2.1:/root

web3 ~]# yum -y install gcc pcre-devel
web3 ~]# cd Zabbix/
web3 Zabbix]# tar -xf zabbix-3.4.4.tar.gz
web3 Zabbix]# cd zabbix-3.4.4/
web3 zabbix-3.4.4]# ./configure --enable-agent
web3 zabbix-3.4.4]# make && make install
```

```
web3 ~]# useradd zabbix

web3 ~]# zabbix_agentd

web3 ~]# ss -ntulp | grep 10050
```

3.3 修改配置文件

```
web3 ~]# vim /usr/local/etc/zabbix_agentd.conf

93 #Server=127.0.0.1    #注释该行

118 StartAgents=0    #禁止被动监控(有服务进程没有端口)

134 ServerActive=192.168.2.5:10051    #监控服务器取消 127.0.0.1

145 Hostname=web201    #告诉监控服务器,是谁发的数据,一定要和 zabbix 服
务器配置的监控主机名一致(后续会设置)

183 RefreshActiveChecks=120  #默认 120 秒检测一次


web3 ~]# killall -9 zabbix_agentd

web3 ~]# ss -ntulp | grep :10050  #修改配置文件后,检测不到此端口

web3 ~]# ps -C zabbix_agentd    #能检测到 zabbix_agentd 进程
```

3.4 201 上安装网站服务

```
web3 ~]# yum -y install httpd

web3 ~]# systemctl restart httpd
```

```
web3 ~]# ss -ntulp | grep 80
```

检查 2.5 的监控页面的自动检查是否检测到该机.因为 10050 端口被禁用,能自动检测到该机,但不能监控.

3.5 添加被监控主机

配置-主机-创建主机->主机名称 web201->群组 linux servers->IP 地址 0.0.0.0->端口 0->最下面的添加\更新

3.6 克隆模板

配置->模板,选择 Template OS Linux,点击,打开后,模板名称 ATmplates OS Linux,可见名称 ATmplates OS Linux->群组 templates->点击全克隆->添加

配置->模板,点击 ATmplates OS Linux->监控项->全选->批量更新->勾选类型,下拉选择 zabbix 客户端(主动式)->更新->返回页面后,点击类型,停用无(主动式)的 3 项

3.7 调用克隆的监控模板监控 201

配置->主机->点击 201->模板标签页->链接指示器,选择群组 templates,勾选 ATmplates OS Linux->选择->返回页面点击添加->添加\更新

返回主机页面后,web201 的状态为已启用,ZBX 为灰色

3.6 验证效果

监测中->图形->群组 Linux servers,主机 web201,图形 cpu load

四 Zabbix 进阶操作-拓扑图与聚合图形

4.1 拓扑图

4.2 聚合图形

五 自定义监控案例

5.1 问题

沿用前面的练习，使用自定义 key 监控常用监控项目，实现以下目标：

监控 Nginx 状态

监控网络连接状态

5.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：监控 Nginx 服务状态

1）准备环境，部署 nginx 软件

安装 nginx 软件，开启 status 模块

```
[root@zabbixclient_web1 nginx-1.12.2]# ./configure \
> --with-http_stub_status_module
[root@zabbixclient_web1 nginx-1.12.2]# make && make install
```

```
web1 ~]# cat /usr/local/nginx/conf/nginx.conf
… …
location /status {
            stub_status on;
      }
… …
web1 ~]# curl  http://192.168.2.100/status
Active connections: 1
server accepts handled requests
10 10 3
Reading: 0 Writing: 1 Waiting: 0
```

2）自定义监控 key

语法格式：

UserParameter=key,command

UserParameter=key[*],<command>

key 里的所有参数，都会传递给后面命令的位置变量

如：

UserParameter=ping[*],echo $1

ping[0]， 返回的结果都是 0

ping[aaa]， 返回的结果都是 aaa

注意：被监控端修改配置文件，注意要允许自定义 key 并设置 Include!

创建自定义 key

```
web1 ~]# vim /usr/local/etc/zabbix_agentd.conf.d/nginx.status
UserParameter=nginx.status[*],/usr/local/bin/get_nginx_status.sh $1
web1 ~]# killall zabbix_agentd
web1 ~]# zabbix_agentd
```

自定义监控脚本（仅供参考，未检测完整状态）

```
web1 ~]# vim /usr/local/bin/get_nginx_status.sh
#!/bin/bash
case $1 in
active)
  curl -s http://192.168.2.100/status |awk '/Active/{print $NF}';;
waiting)
  curl -s http://192.168.2.100/status |awk '/Waiting/{print $NF}';;
accepts)
  curl -s http://192.168.2.100/status |awk 'NR==3{print $2}';;
esac
web1 ~]# chmod +x  /usr/local/bin/get_nginx_status.sh
```

测试效果：

```
web1 ~]# zabbix_get  -s 127.0.0.1 -k get_nginx.status[accepts]
```

登陆 Zabbix 监控 Web，创建监控项目 item，点击 Configuration（配置）-->Hosts(主机)，点击主机后面的 items（项目），点击 Create item（创建项目）。修改项目参数如图-36 所示。

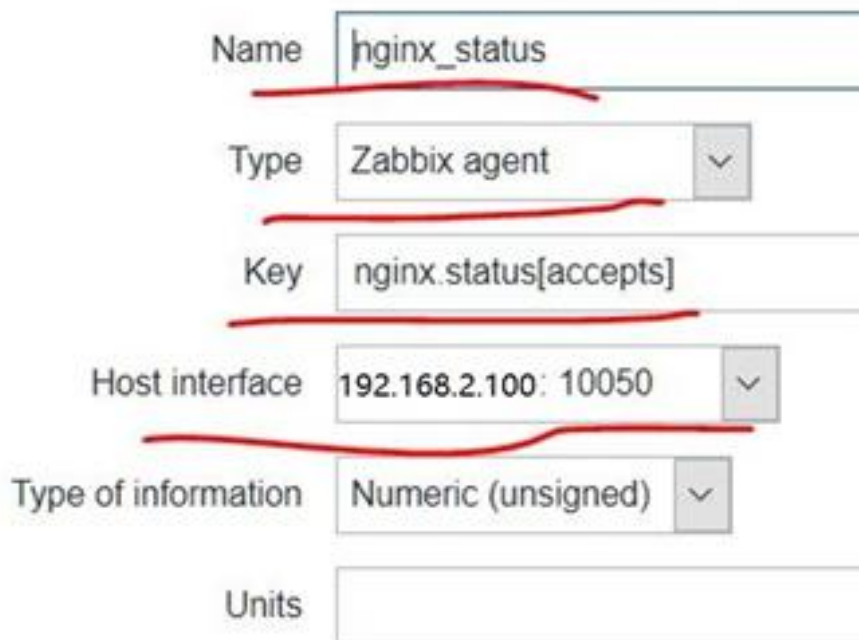

图-36

步骤二：监控网络连接状态

1）了解 TCP 协议

熟悉 TCP 三次握手，参考图-37。

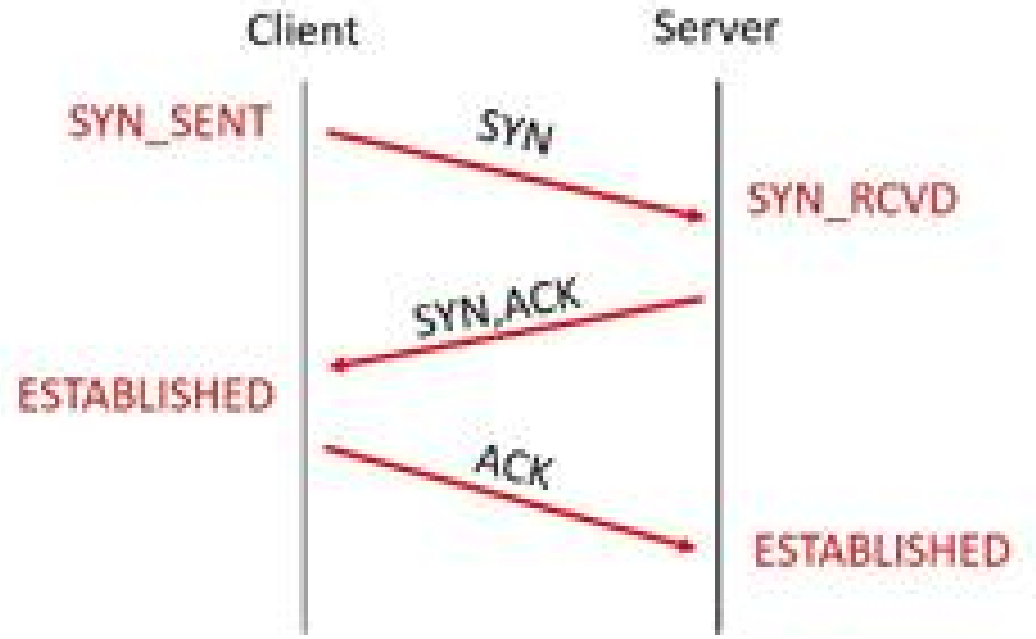

图-37

熟悉 TCP 连接的四次断开，参考图-38。

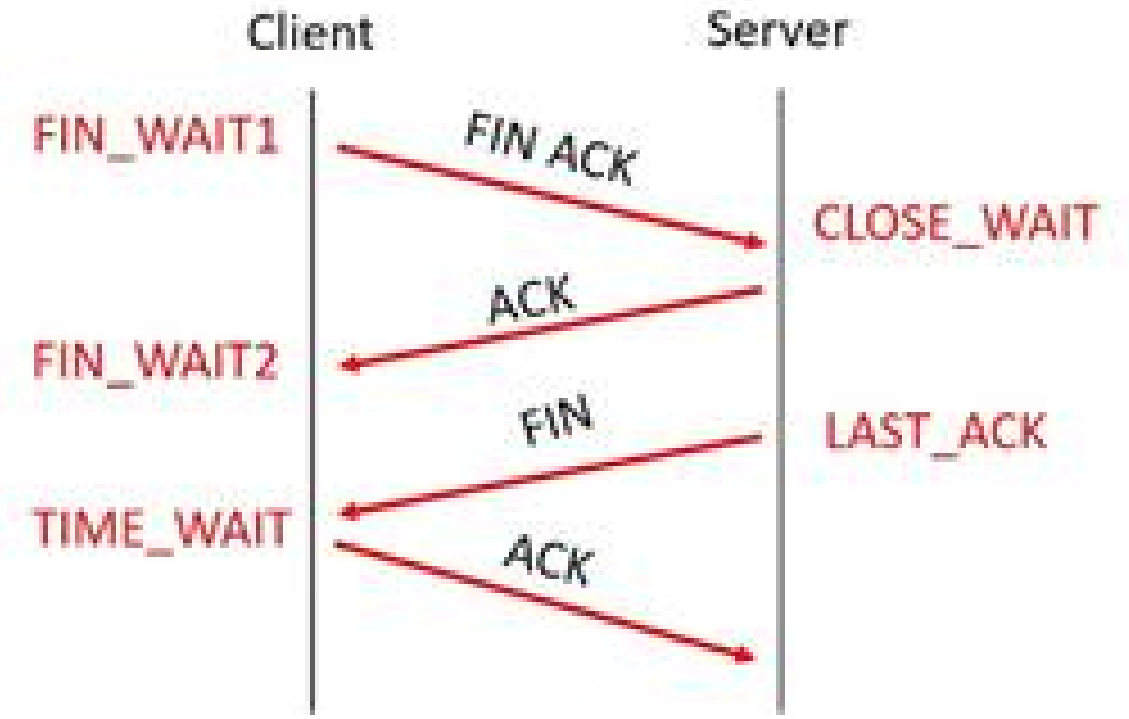

图-38

2）查看网络连接状态

模拟多人并发连接

```
web1 ~]# ab -c 1000 -n 100000 http://192.168.2.100/
```

步骤二：监控网络连接状态

1）了解 TCP 协议

查看网络连接状态，仔细观察、分析第二列的数据

```
web1 ~]# ss -antup
```

//-a 显示所有

//-t 显示 TCP 连接状态

//-u 显示 UDP 连接状态

//-n 以数字形式显示端口号和 IP 地址

//-p 显示连接对应的进程名称

3）创建自定义 key

注意：被监控端修改配置文件，注意要允许自定义 key 并设置 Include。

```
web1 ~]# vim /usr/local/etc/zabbix_agentd.conf.d/net_status
UserParameter=net_status[*],/usr/local/bin/net_status.sh $1


web1 ~]# killall zabbix_agentd
web1 ~]# zabbix_agentd
```

自定义监控脚本（仅供参考，未检测完整状态）

```
web1 ~]# vim /usr/local/bin/net_status.sh
#!/bin/bash
```

```
case $1 in
estab)
      ss -antp | awk 'BEGIN{x=0}/^ESTAB/{x++} END{print x}';;
close_wait)
      ss -antp | awk 'BEGIN{x=0}/^CLOSE-WAIT/{x++} END{print x}';;
time_wait)
      ss -antp | awk 'BEGIN{x=0}/^TIME-WAIT/{x++} END{print x}';;
esac
```

```
web1 ~]# chmod +x  /usr/local/bin/net_status.sh
```

测试效果：

```
web1 ~]# zabbix_get  -s 127.0.0.1 -k net_status[time-wait]
```

4）监控 netstatus

在监控服务器，添加监控项目 item，Configuration-->Hosts 点击主机后面的 items

点击 Create item，如图-39 所示。

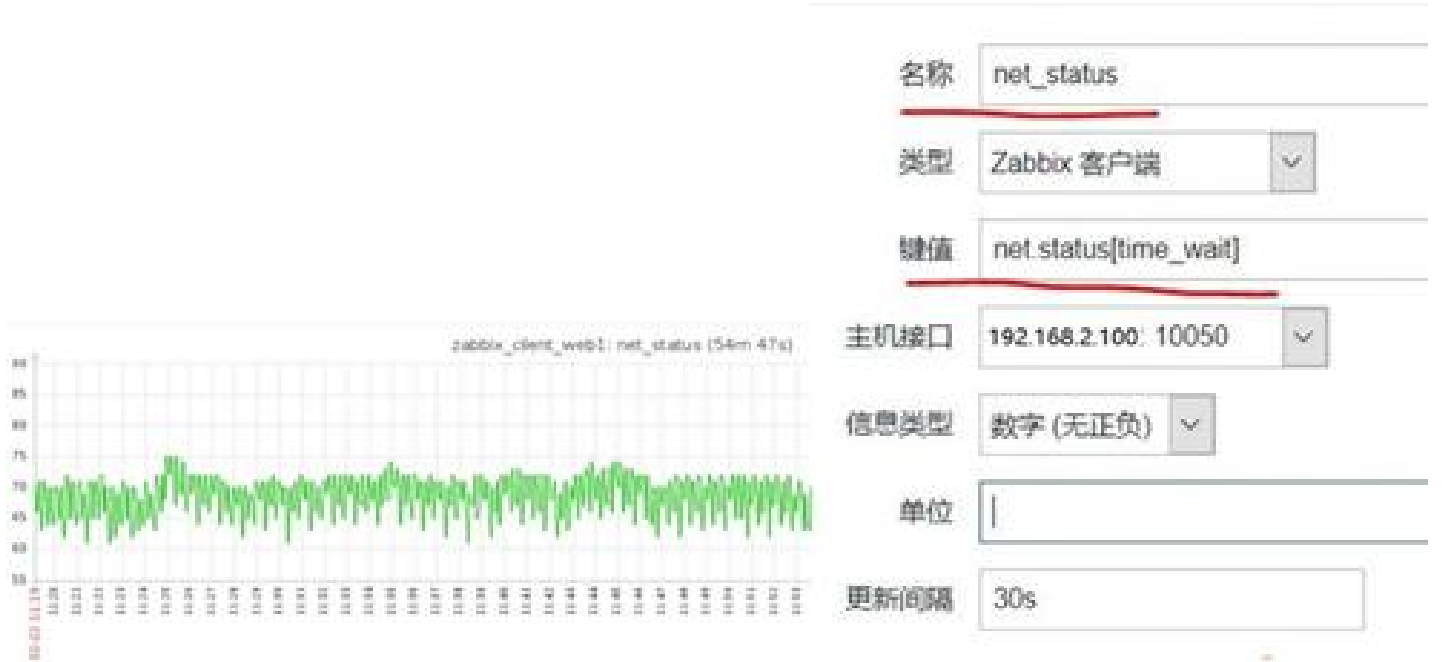

图-39

监控案例

监控 nginx

192.168.2.100 运行 nginx 服务,并支持查看状态信息

编写脚本,并给脚本添加执行权限

把脚本定义为命令

重启 zabbix_agentd 服务

测试命令

监控 nginx 状态

在监控页面做如下配置

创建新的监控模板　ATMP2

创建新的应用集 nginx_status

创建新的监控项　　名称和对应的命令

now_link_num　　get_nginx_status[Active]

his_sum_num　　get_nginx_status[accepts]

waiting_nginx_num　　get_nginx_status[waiting]

调用新闯将的模板监控主机 100

查看监控数据

监控网络连接状态

在监控页面做如下配置

使用监控模板　ATMP2

创建新的应用集 net_status

创建新的监控项　　名称和对应的命令

estab_link　　net_status[estab]

time_wait_num　　net_status[time_wait]

close_wait_num　　net_status[close_wait]