配置示例01（Telnet暴力破解）

2020年1月29日 21:35

目录

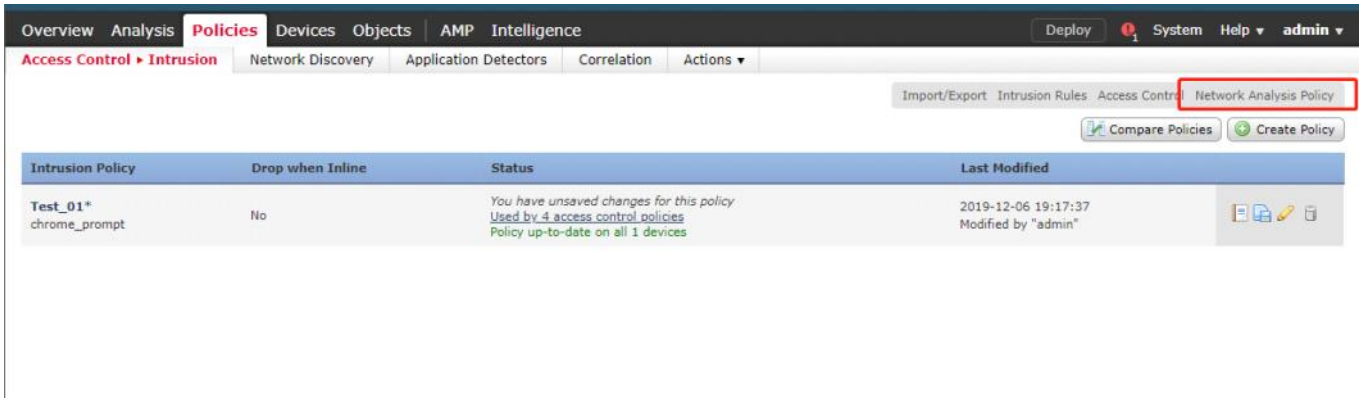
1：配置NGIPS前提

- 参考威胁防御策略介绍里的最佳时间进行设计配置
- 把FTD配置为NGIPS（下一代入侵防御系统）可以引用3个不同的安全策略
  - 网络分析策略
  - 入侵策略
  - 访问控制策略

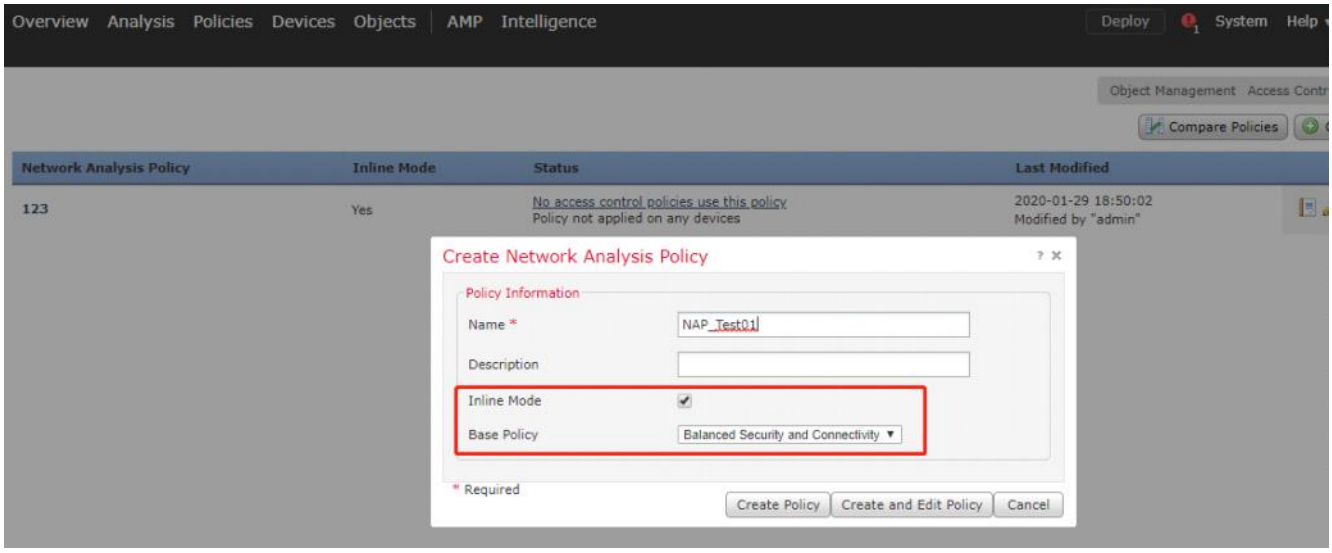
2：配置NAP（网络分析策略）

- 配置NAP需要打开网络分析策略配置页面。
- FMC没有直接的菜单，需要在ACP/IPS策略配置页面点击配置

2.1：创建一条NAP网络分析策略



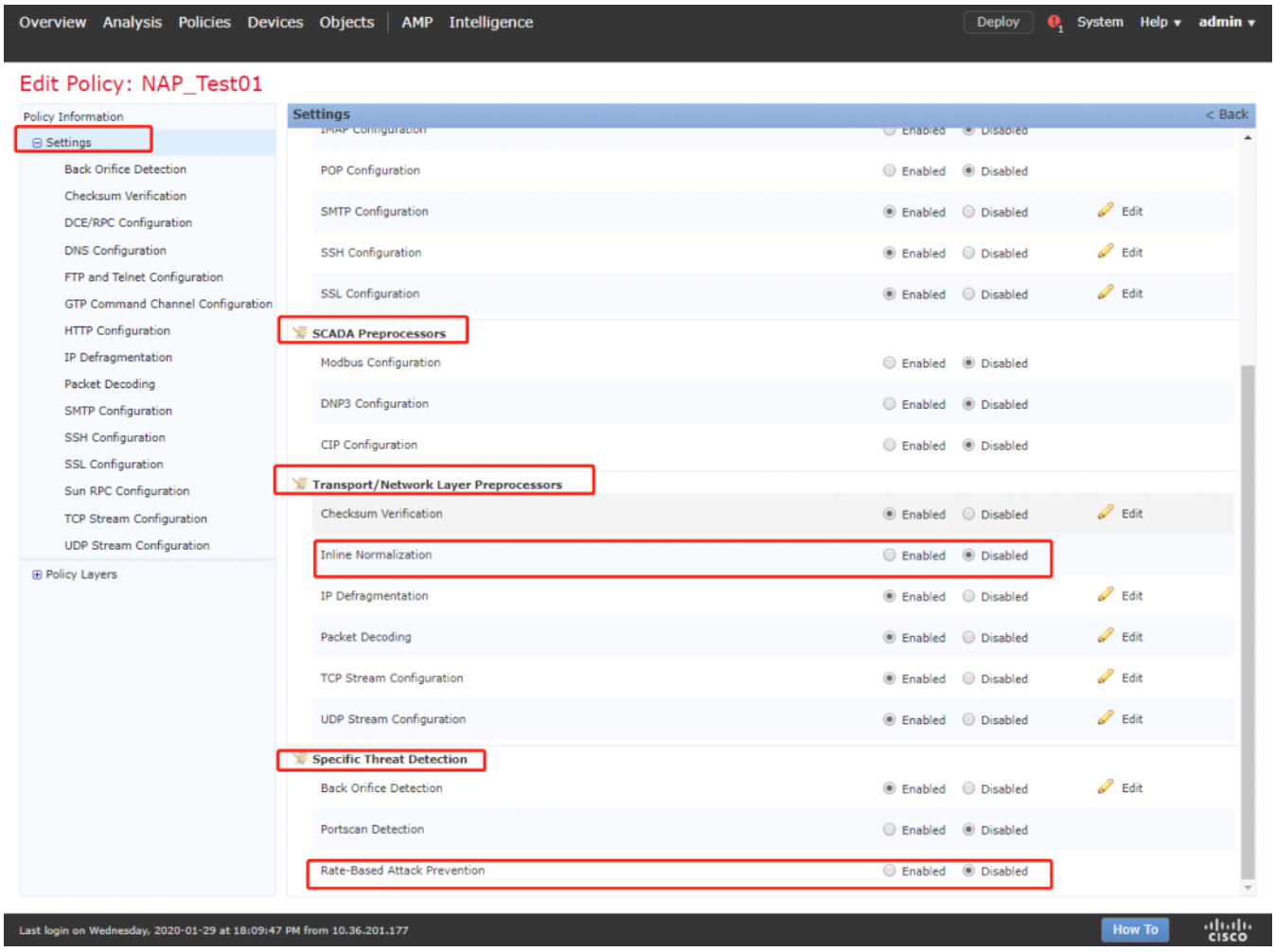
该策略选择 均衡安全和连通信，这类策略提供较好的系统性能且不会放过重要的漏洞检测

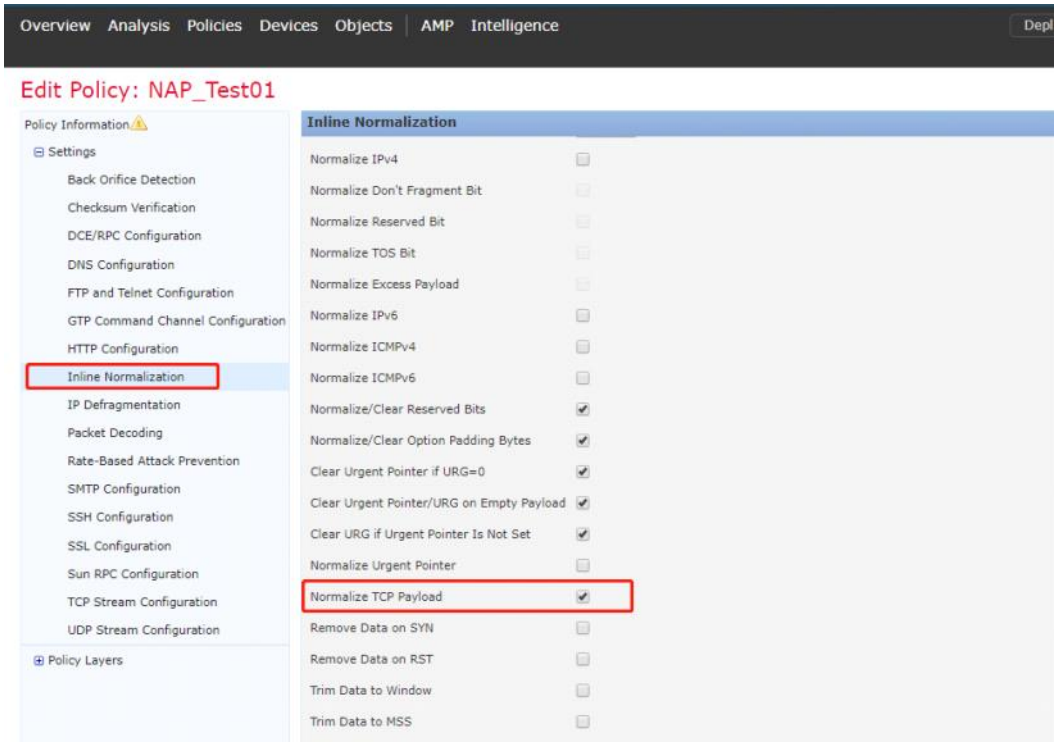


2.2：（可选）更改NAP策略设置

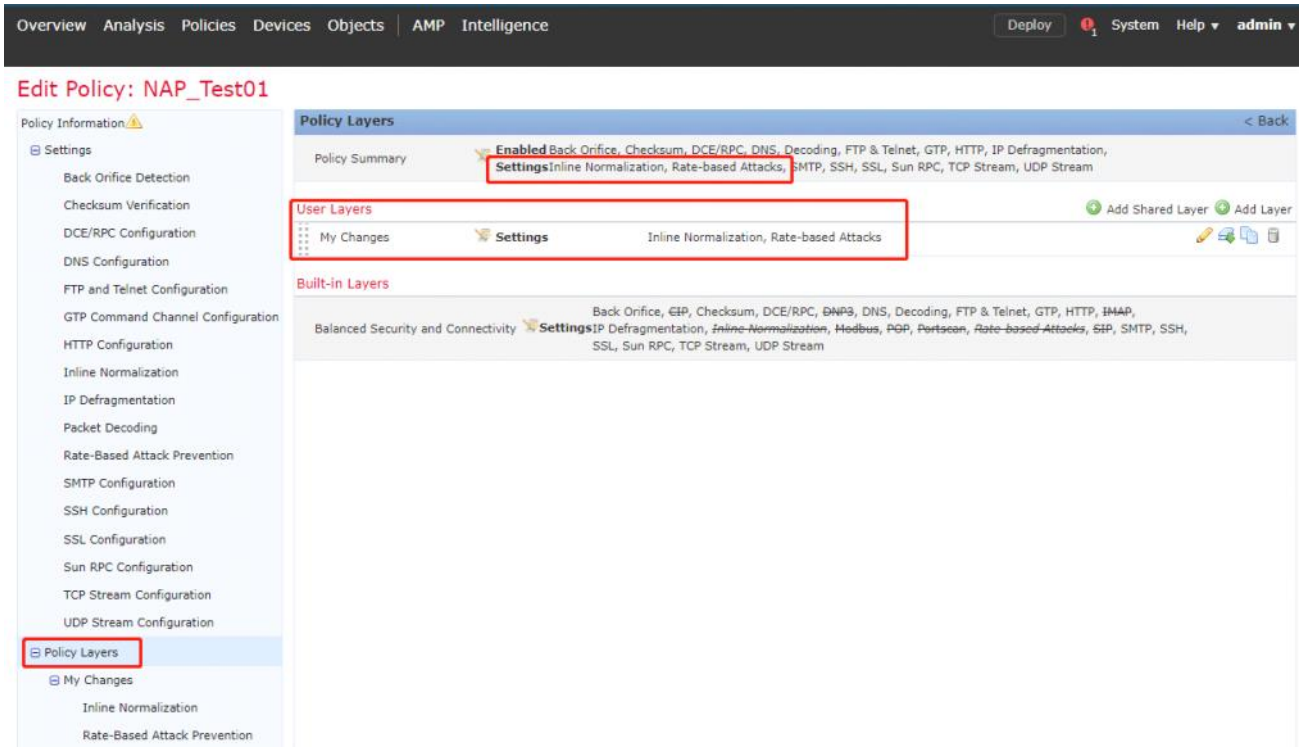
在NAP策略的settings里可以选择打开哪些类型的哪些预处理器

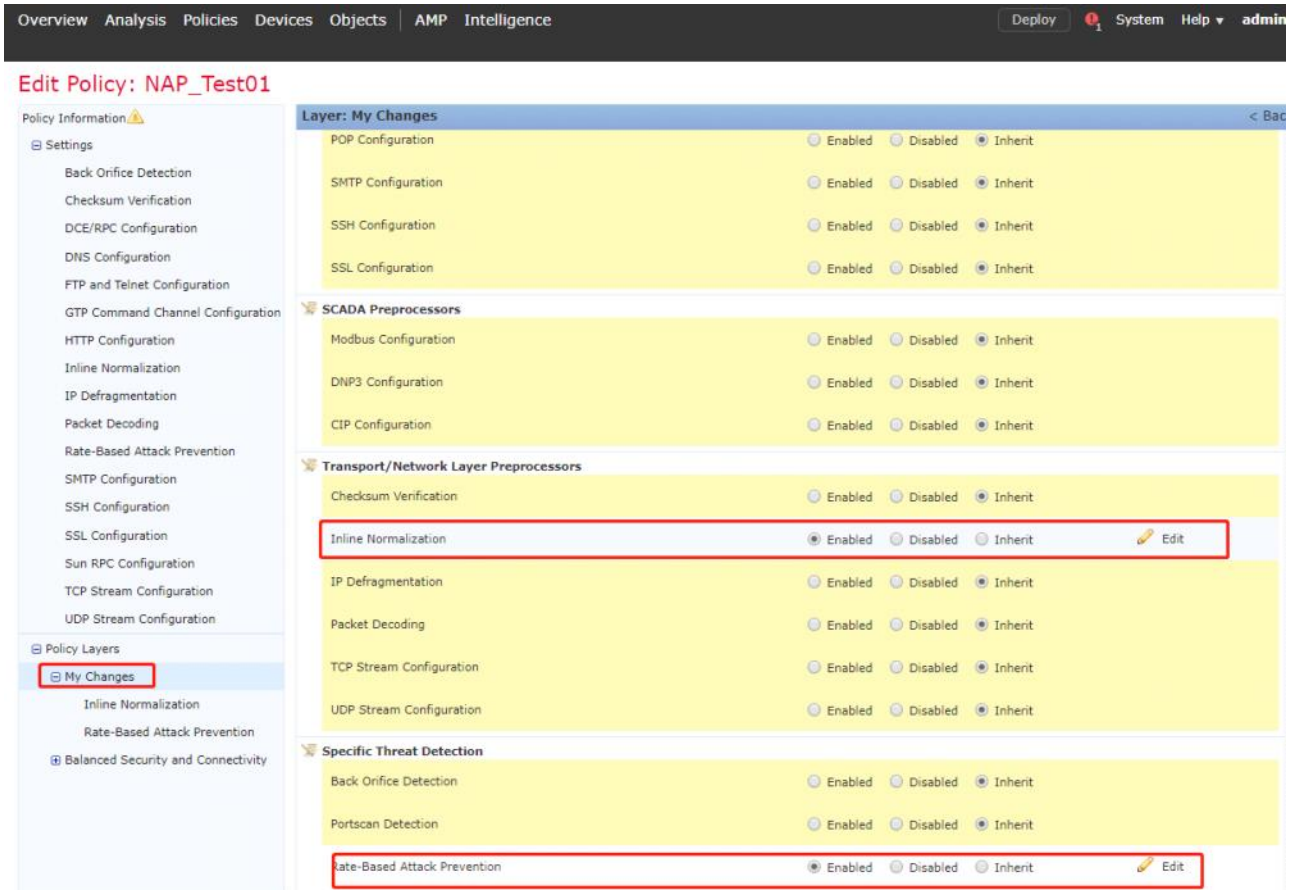
2.2.1：如下图，我们打开了inline Normalization（在线规范化）预处理器，并选择Normalize TCP Payload（规范化TCP负载）选项，这样可以确保TCP重传数据的一致性



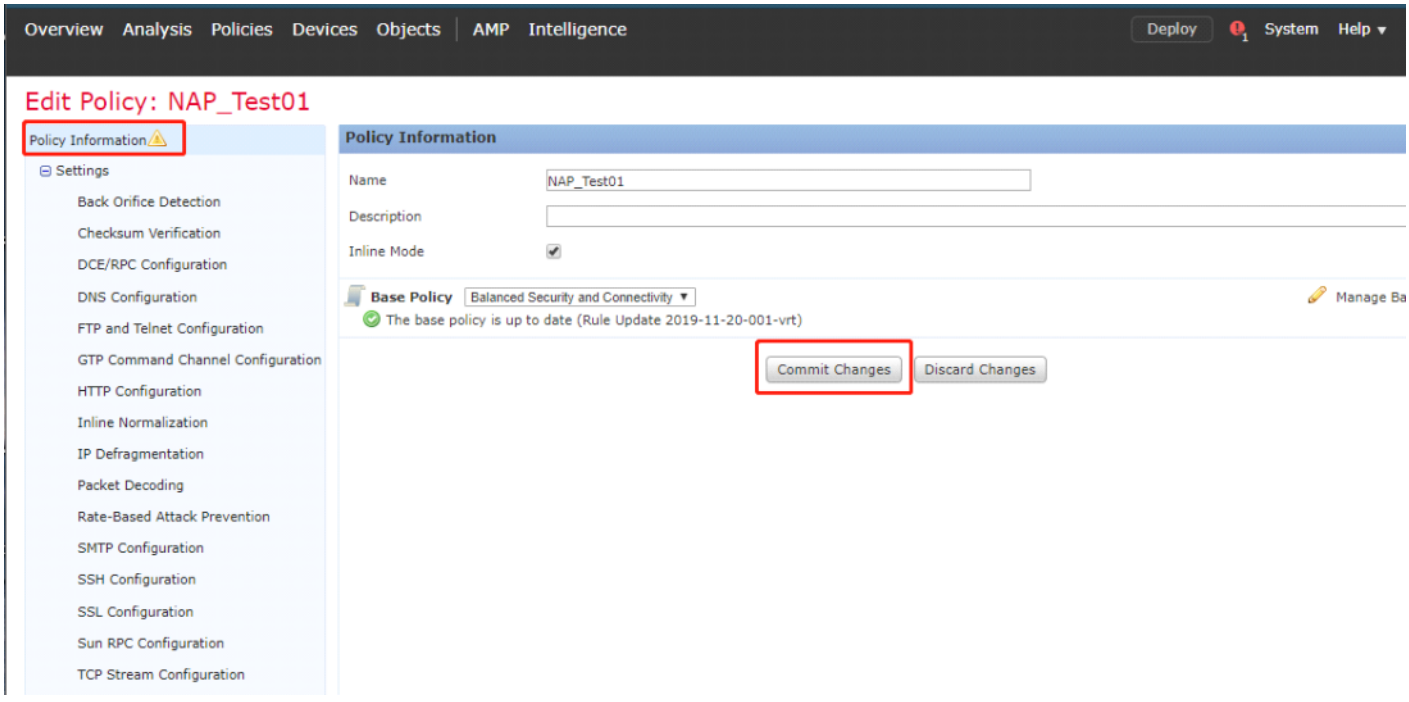


2.2.2：可以看到用户打开了哪些预处理器，做了哪些设置变动





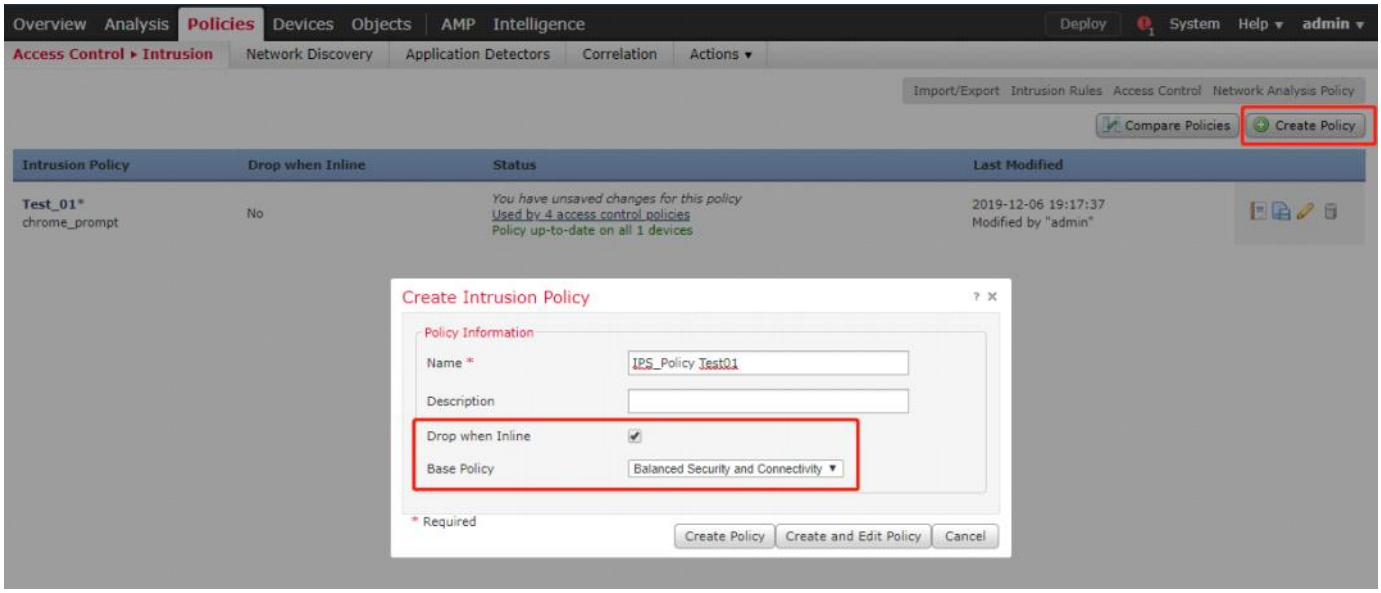
2.2.3：返回策略信息页面，点击执行变更（这个和写完ACP的SAVE一样）



3：配置入侵防御策略

3.1：创建一个IPS策略。

- **Drop when inline=默认启用：**当FTD部署为inline/routed/transparent模式时，开启该功能后可以丢弃数据包。但如果FTD部署为inline-tap/passive模式时，该选项不会影响数据流
- **选择均衡安全和连通信作为基本策略：**该策略提供良好的系统性能并不会漏掉关键和最新漏洞检测



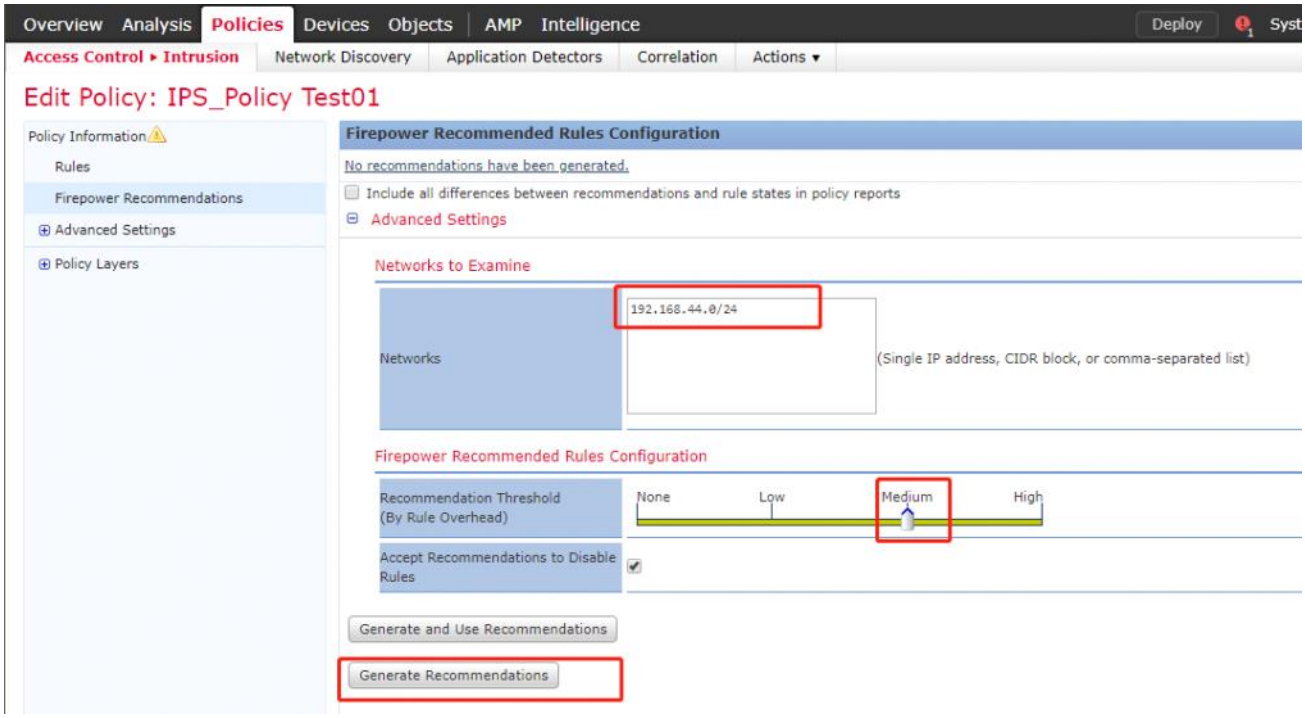
3.2：开启Firepower推荐特性（慎用可选）☆

- 该特性默认禁用，因为该功能会消耗额外的系统资源分析网络发现数据与其相关联的漏洞，并生成相应的推荐设置。
- 但是若安全性需求还是存在的，则可以开启Firepower推荐特性，根据网络中运行的操作系统/服务/应用来建议启用或禁用IPS规则

备注：在大部分网络主机生成流量并且被FTD设备发现后，生成并使用Firepower推荐特性。如果没等待网络发现成功发现网络中所有的主机和应用就开启该特性，那么FTD就会推荐禁用很多IPS规则，这种效果就很差

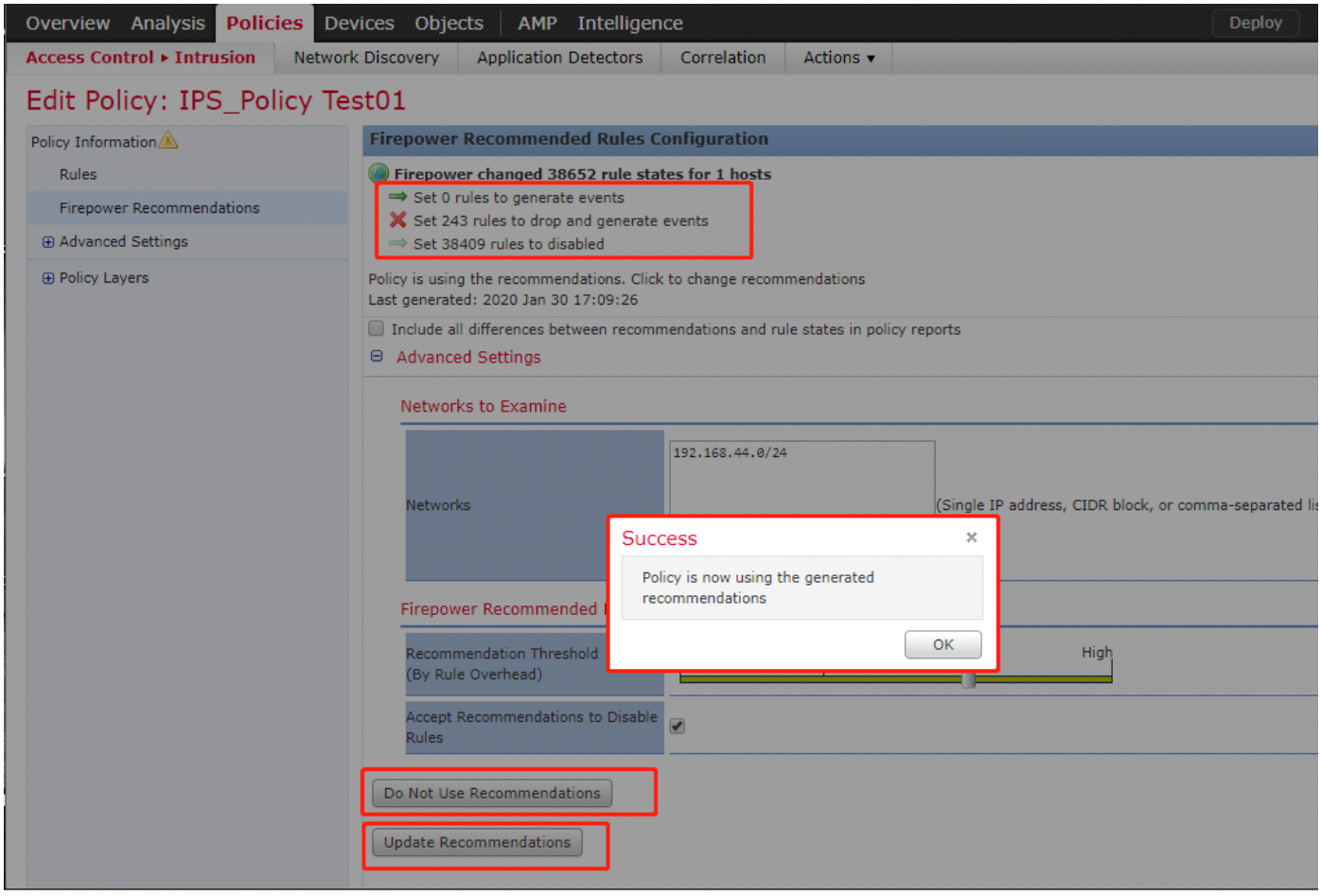
3.2.1：进入IPS策略编辑，Firepower推荐特性

- 填写根据某个网段的流量，并根据网络主机上运行的应用，服务和操作系统推荐入侵规则
- 并且选择推荐中等/低的开销，这样保证系统性能
- 点击Generate and use recommedations（生成并使用推荐）

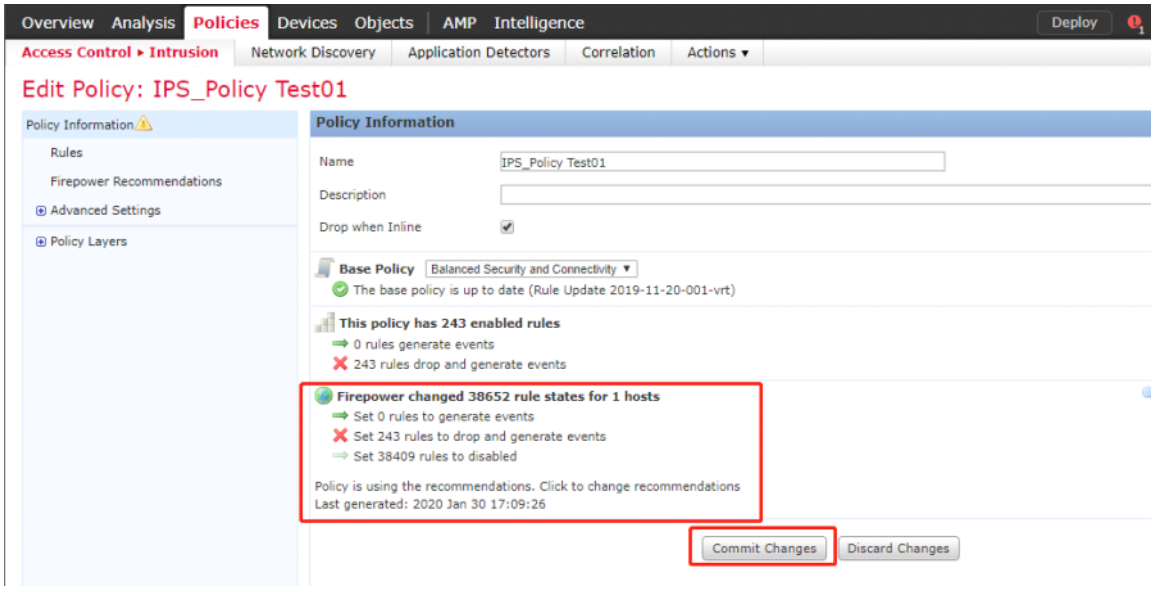


3.2.2：提示推荐的IPS规则已经应用，可以选择不应用或者填写新的IP参数/系统性能登记进行推荐的IPS规则更新

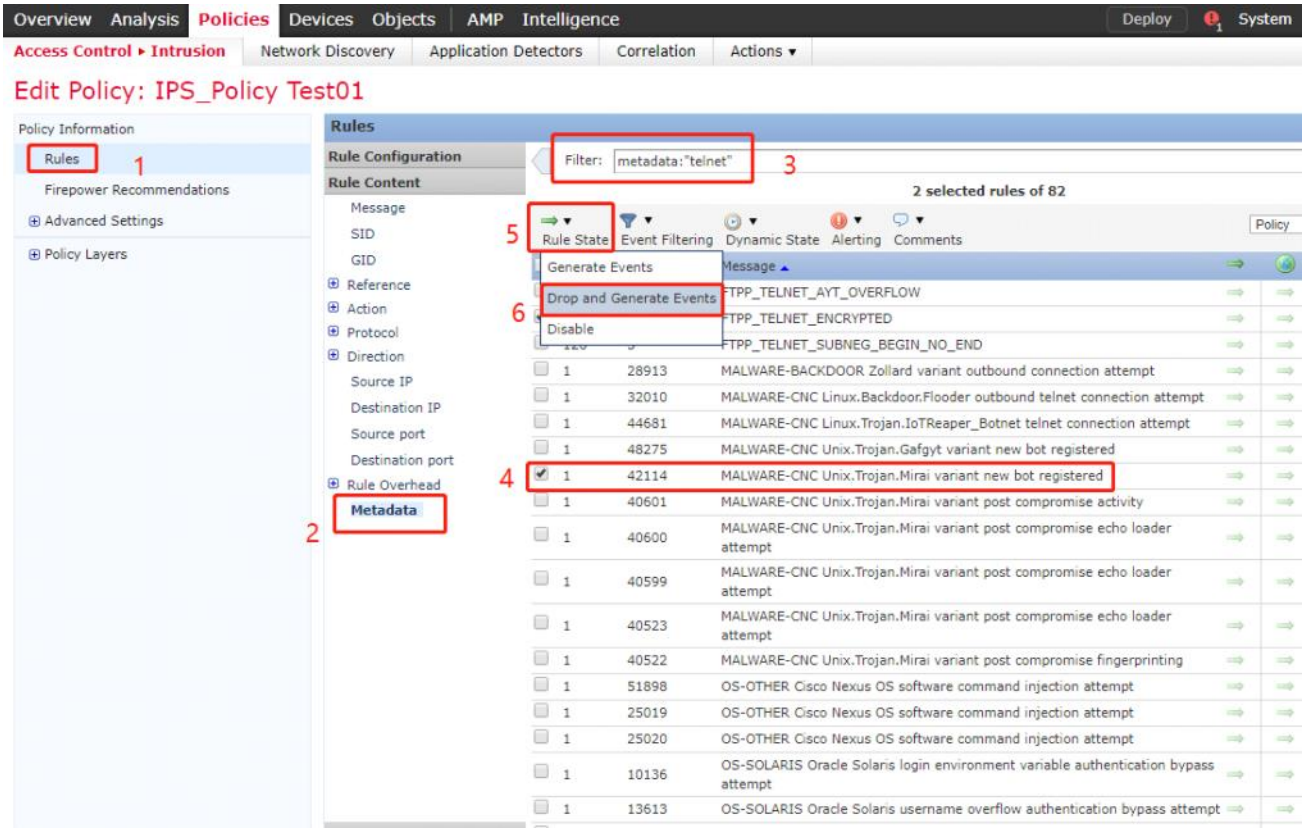
- 0个规则用来生成事件不block流量
- 243个规则用来丢弃并生成事件
- 这些规则时根据这个IP段的主机生成的。



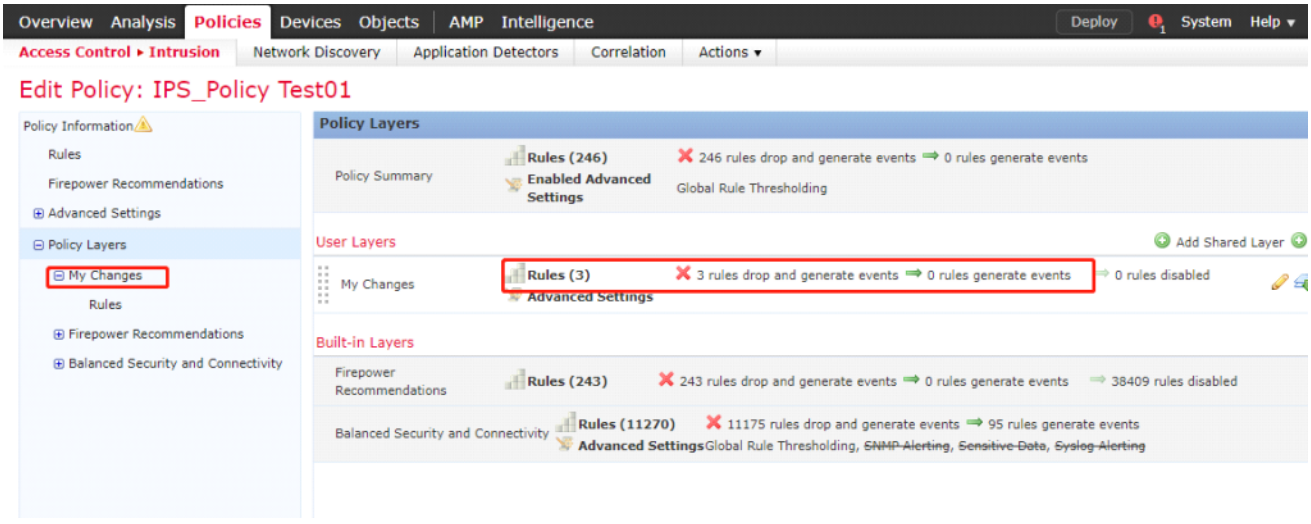
3.2.3：Policy information界面执行变更



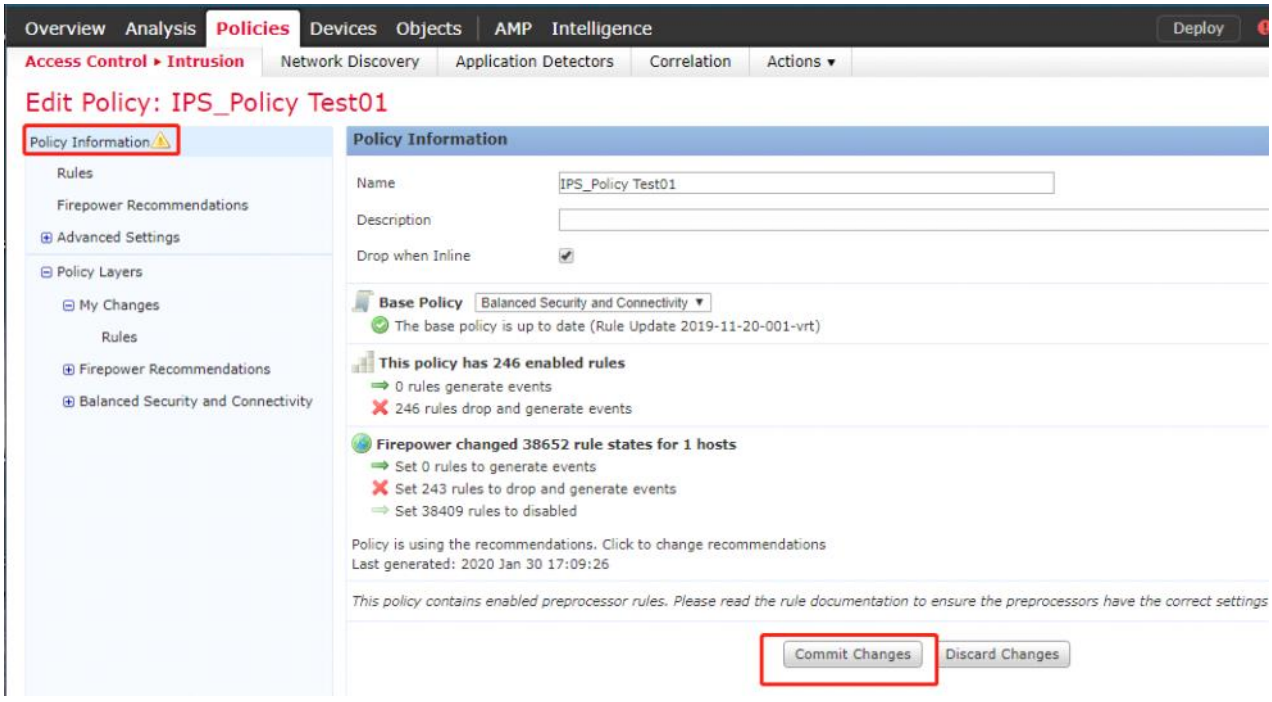
3.3：（可选）如何开启指定的IPS规则



可以看到change中有三个rule被开启并且drop流量了。



保存应用

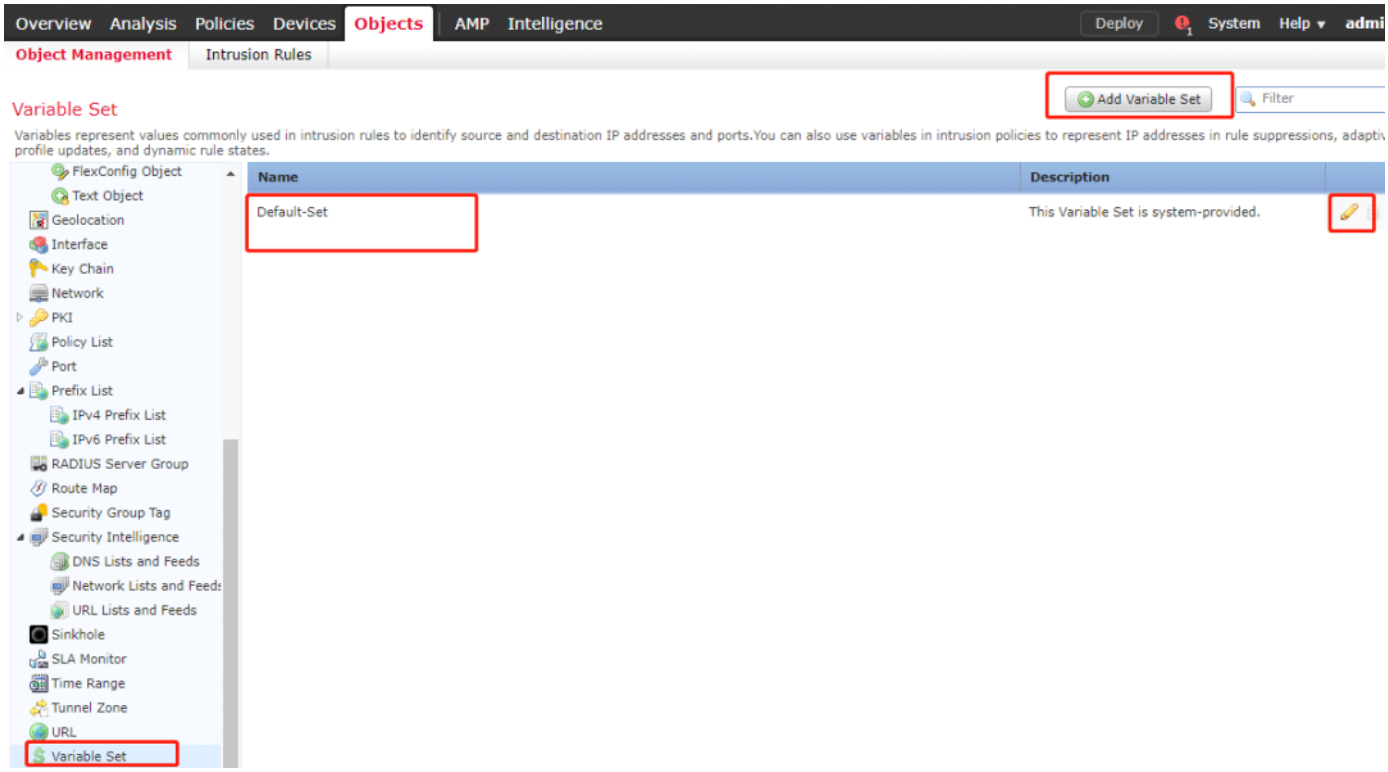


3.4：设置变量（可选+☆☆

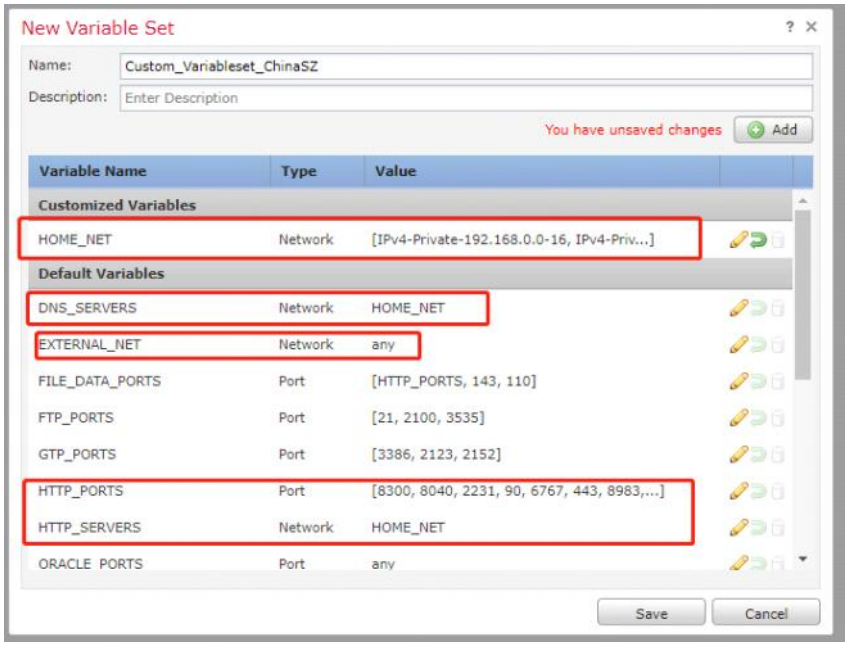
入侵防御策略的重要步骤是定义变量的值，Firepower没有强制客户需要配置
这些变量的定义，意味着IPS基于这些值来审核流量的威胁

- 比如DNS的标准端口，有哪些IP是DNS服务器，有哪些IP是WEB服务器
- 不是WEB服务器的IP提供了http/https服务则会报警异常或丢弃流量。取决于IPS规则的定义了。
- 最经常出现的场景比如是，客户更改了标准的https服务器端口？或者SSLVPN端口等。

你可以编辑默认的变量集，也可以创建自定义的变量集



这里可以设置Home-Net=内部网络IP段，以及DNS服务器地址，HTTP服务器地址和http的可用端口，以及外部IP地址段等信息，IPS依据这些信息进行流量恶意判决。



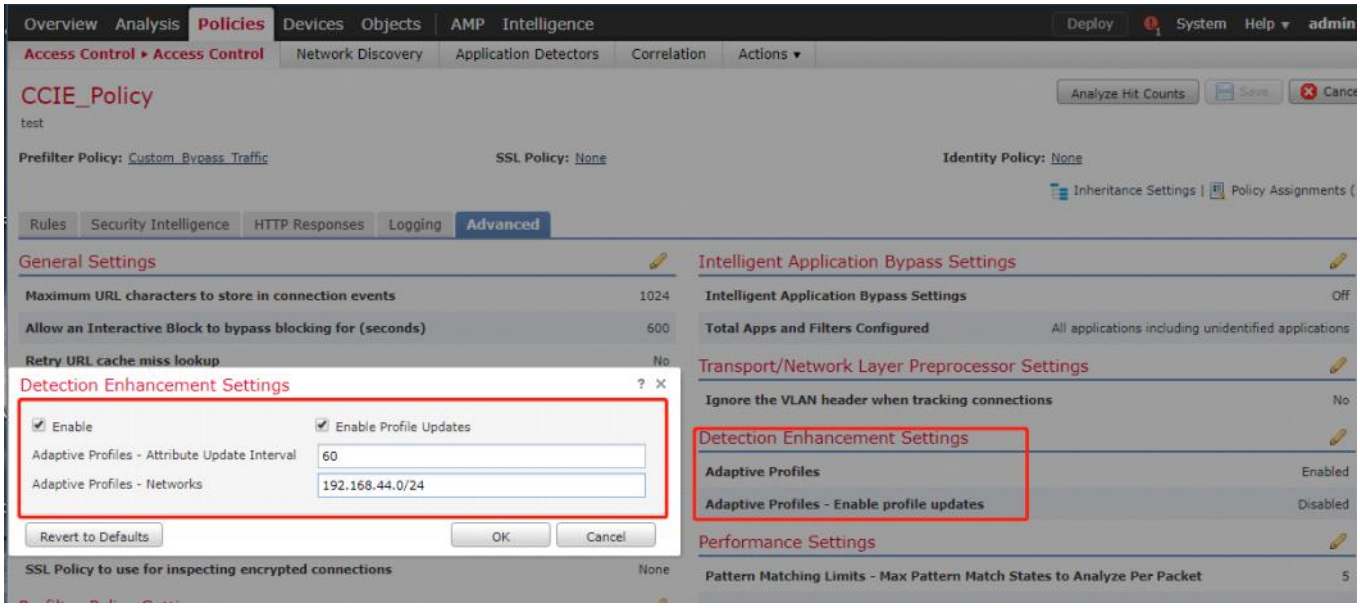
4：配置ACP访问控制策略

2-3已经配置了网络分析策略和入侵检测策略，但是它们不生效，必须要在ACP上调用才行

4.1：配置ACP的Adaptive profiles

ACP上可以开启自适应配置文件和Enable profiles updates特性，这些特性可以智能的让FTD在相关流量上应用已经启用的IPS规则。

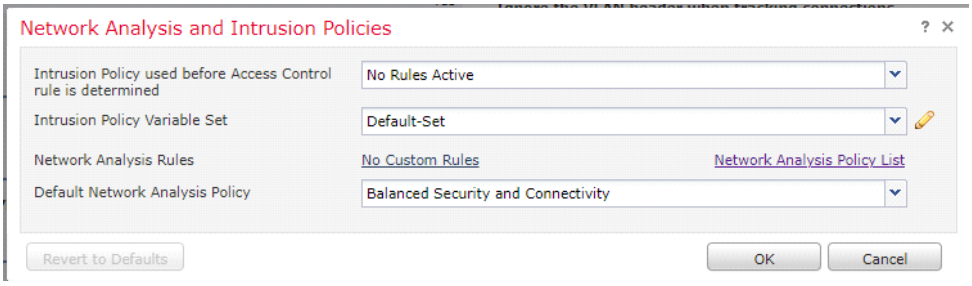
填写的IP段意味着对哪些流量进行只能的入侵规则应用更新



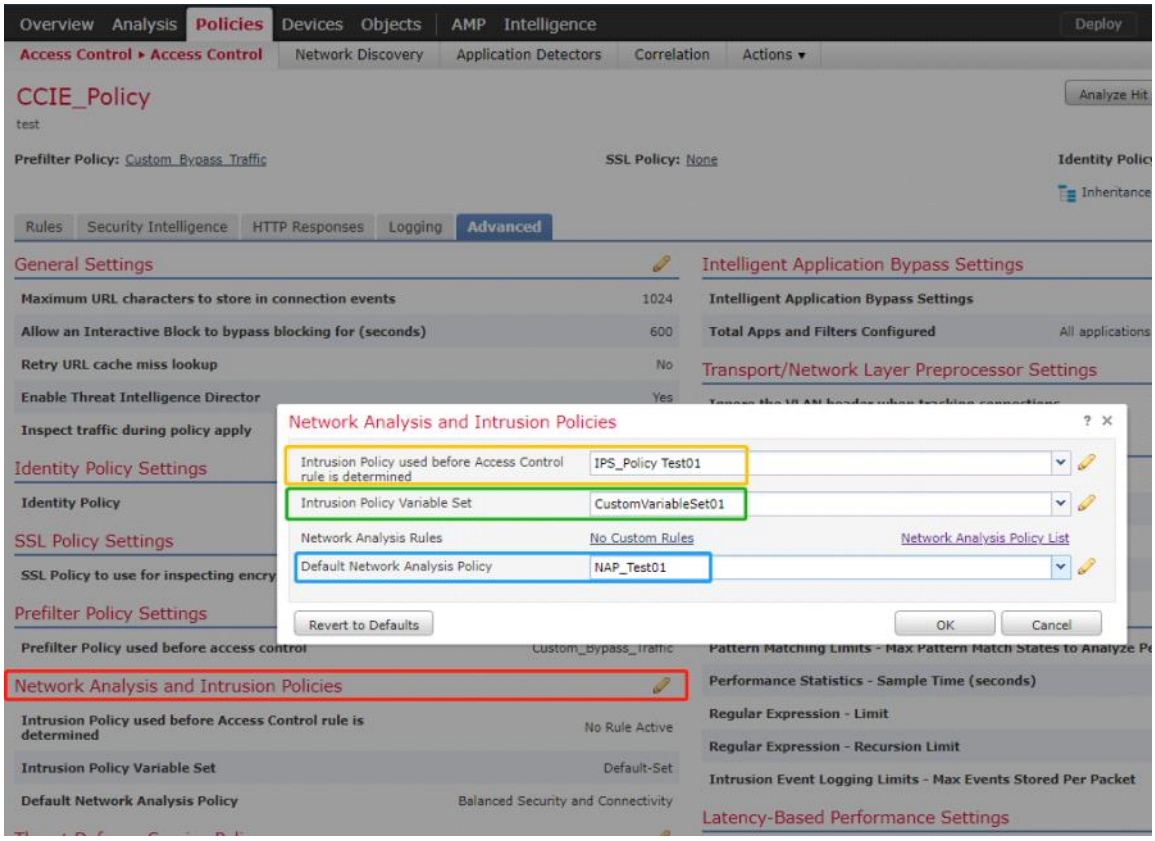
4.2：（可选）引用策略至ACP高级选项

- ACP策略在高级选项中引用NAP网络发现策略和IPS策略，并引用自定义的变量集。

- 注意：这里的调用是整条ACP策略的调用，意思是：所有流量在匹配ACR的IPS策略之前先通过ACP的IPS策略执行一遍
- 可以看到默认配置是没有IPS规则的，开了这个更影响性能和连接性

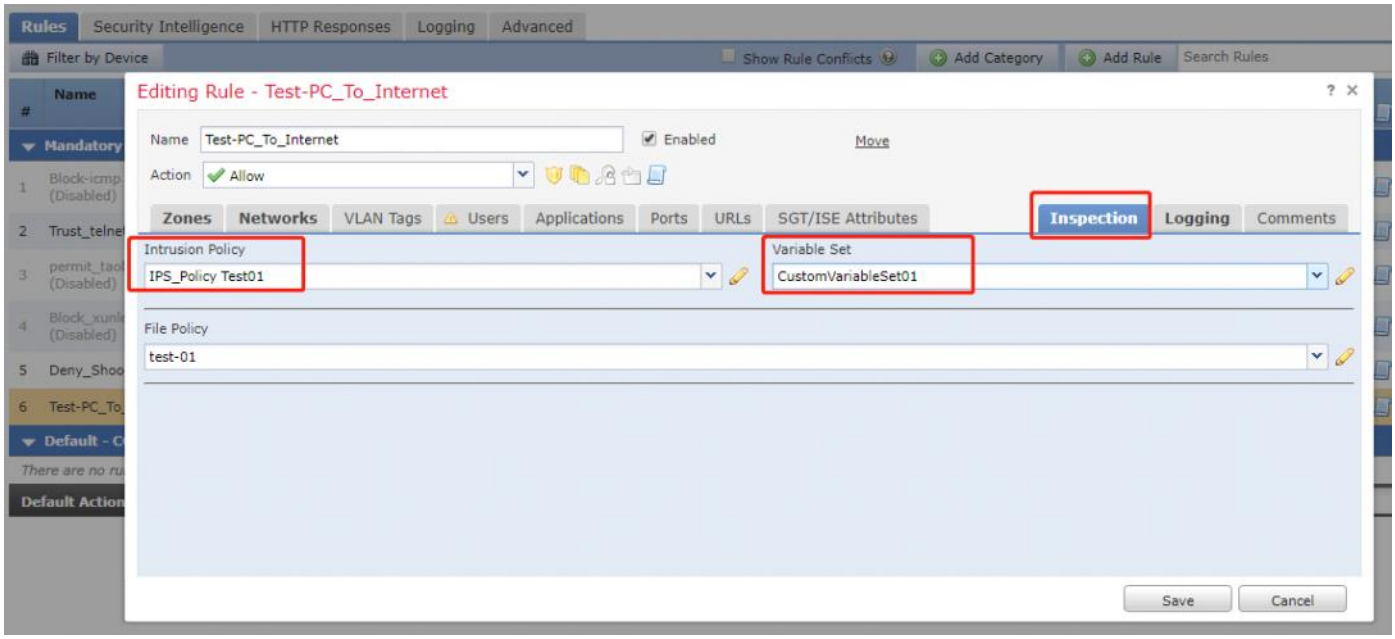


- 但是如果自定义了NAP策略，则必须要在这里调用，因为ACR里面没法调用NAP策略，只可以调用IPS策略和变量集



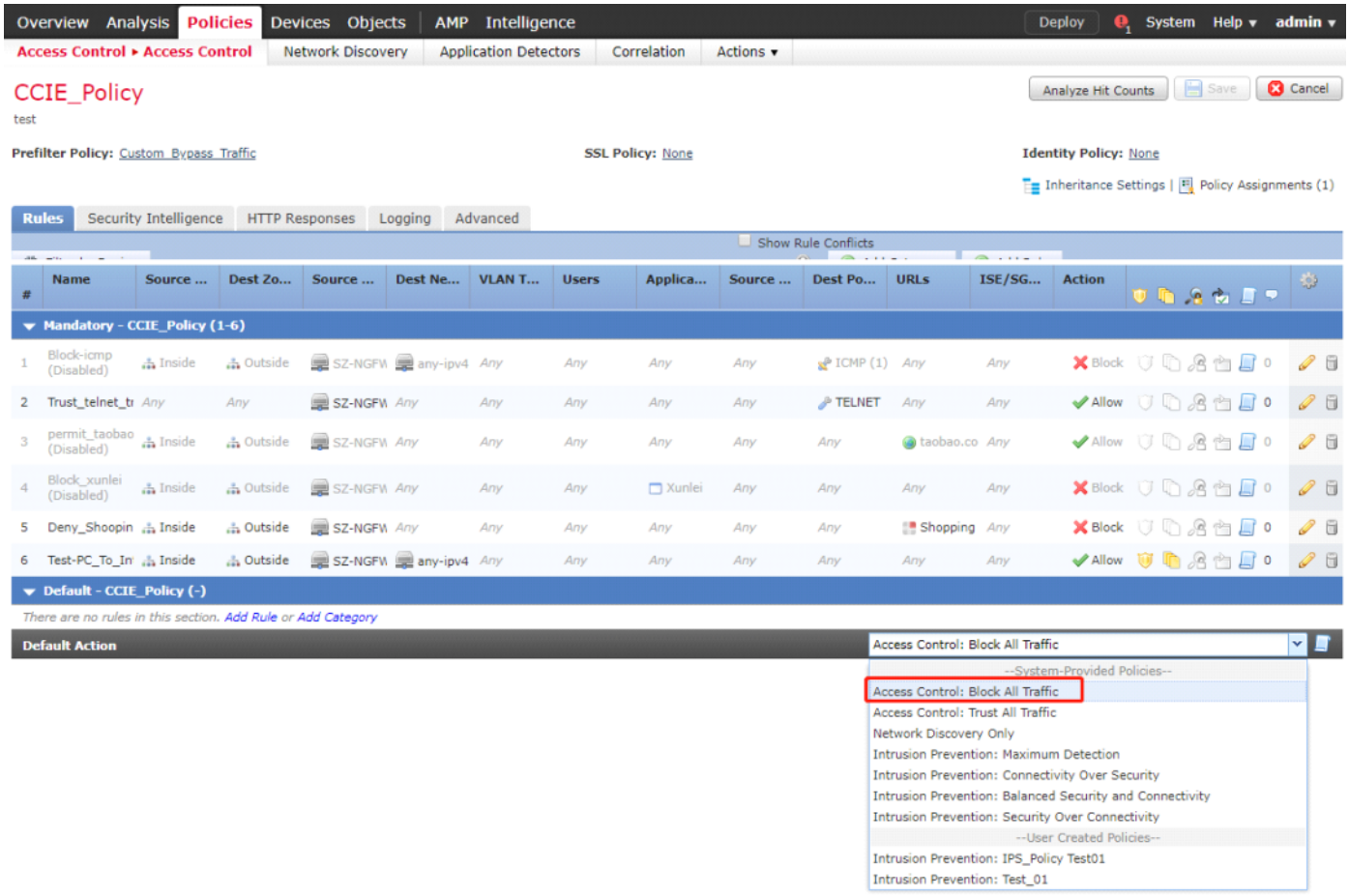
4.3：引用策略到ACR访问控制规则（☆一定要配）

在ACR中可以精细化的指定每条ACR调用哪个IPS策略以及哪些变量，但是没法定义NAP策略

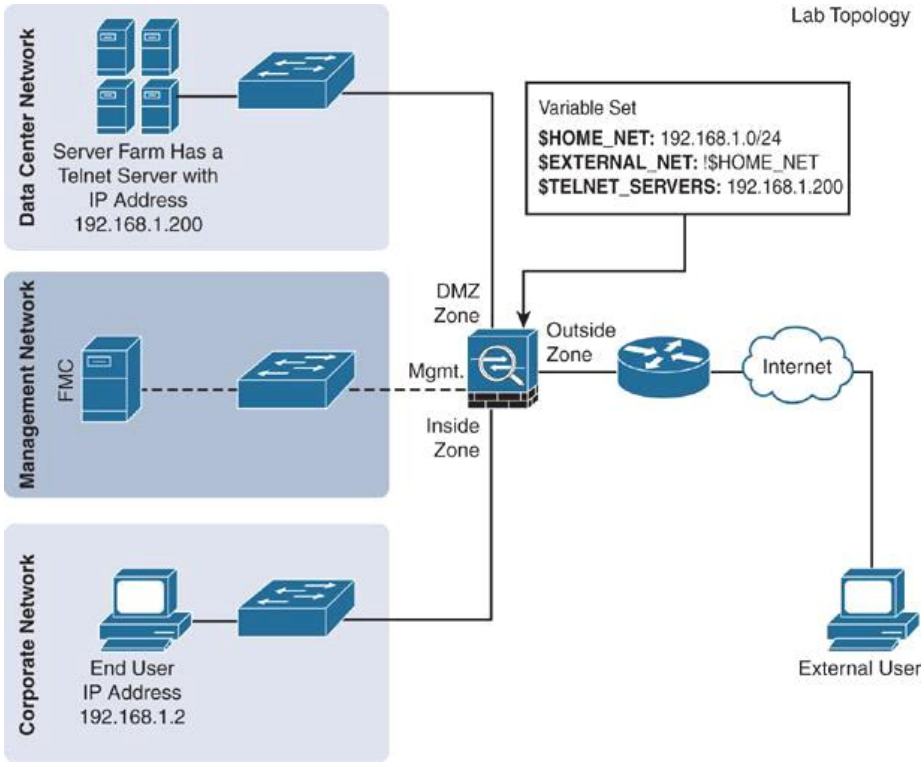


4.4：引用一个IPS规则作为流量不匹配ACR规则时的默认行为

这里一般都是直接block all traffic，你也可以选择放行所有流量通过IPS判定流量安全。。但不推荐



5：验证&分析

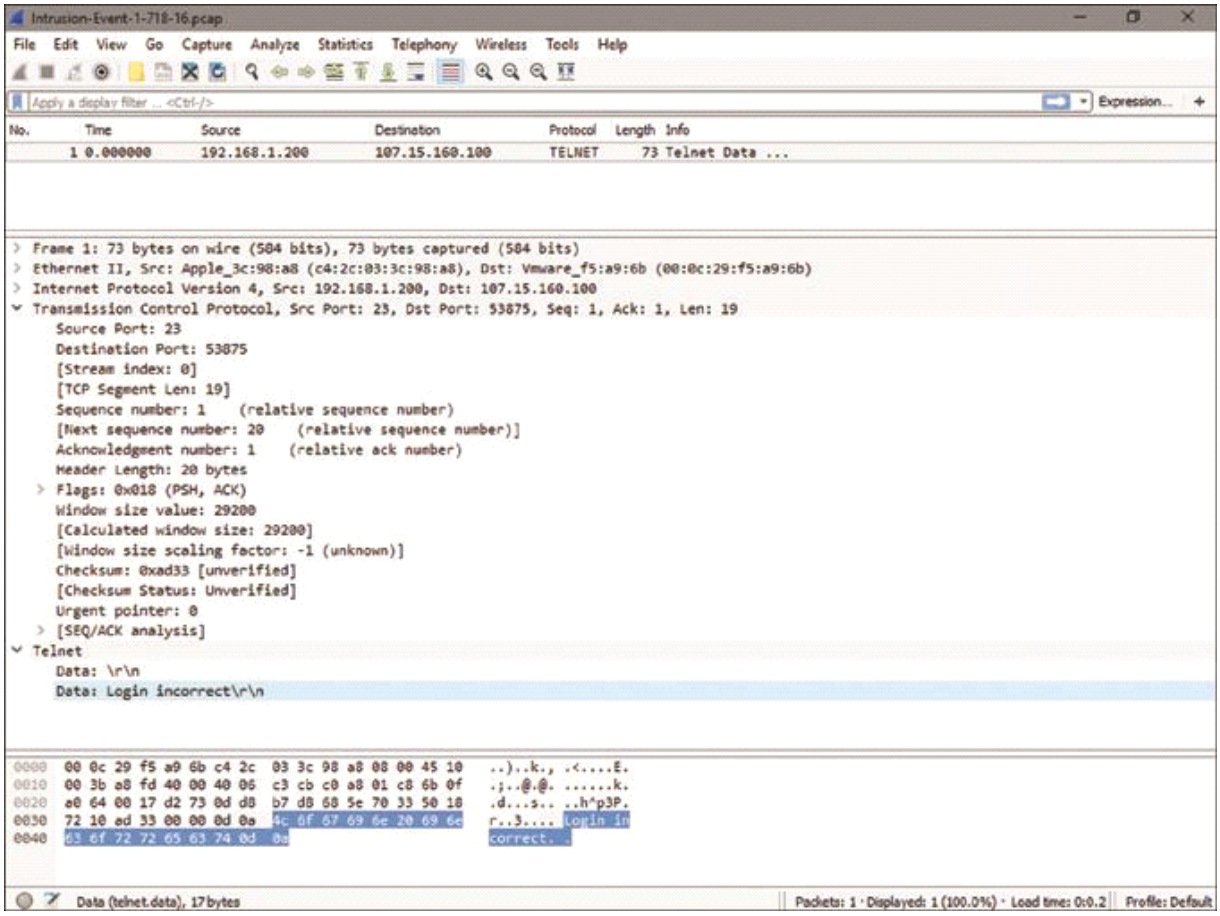


5.1：验证IPS规则1：718，这个规则语法如下

```
alert tcp $TELNET_SERVERS 23 -> $EXTERNAL_NET any (msg:"PROTOCOL-TELNET login incorrect"; flow:to_client, established;
content:"Login incorrect";
metadata: ruleset community, service telnet; classtype: bad-unknown; sid:718;
rev:16; )
```

- 当Telnet服务器没有通过client的登陆认证，并（由于错误的登陆凭据）在有效负载上用“login incorrect”（登陆错误）消息作为对client响应时，FTD设备会触发这条规则预防潜在的暴力破解攻击。
- 并且如果精确定义了变量集，这条IPS规则只会适用于去往EXTERNAL_NET的Telnet流量，而不适用于去往HOME_NET的telnet流量。（意思是可以通过不同的变量集来精细化匹配IPS规则的适用性）

可以看到，Telnet登陆错误的数据包会有一个响应字段，login incorrect，这个字段会被Snort规则1：718检测到注意，因为telnet是明文的，如果是SSH看不到。所以SSH可能需要搭配SSL卸载使用。。



如果从外部网络主机连接Telnet服务器，并输入有效的登陆凭据，则可以正常登陆服务器。
如果输入无效的凭据，则服务器返回包含“login incorrect ”的数据包
如下展示了成功了错误登陆信息

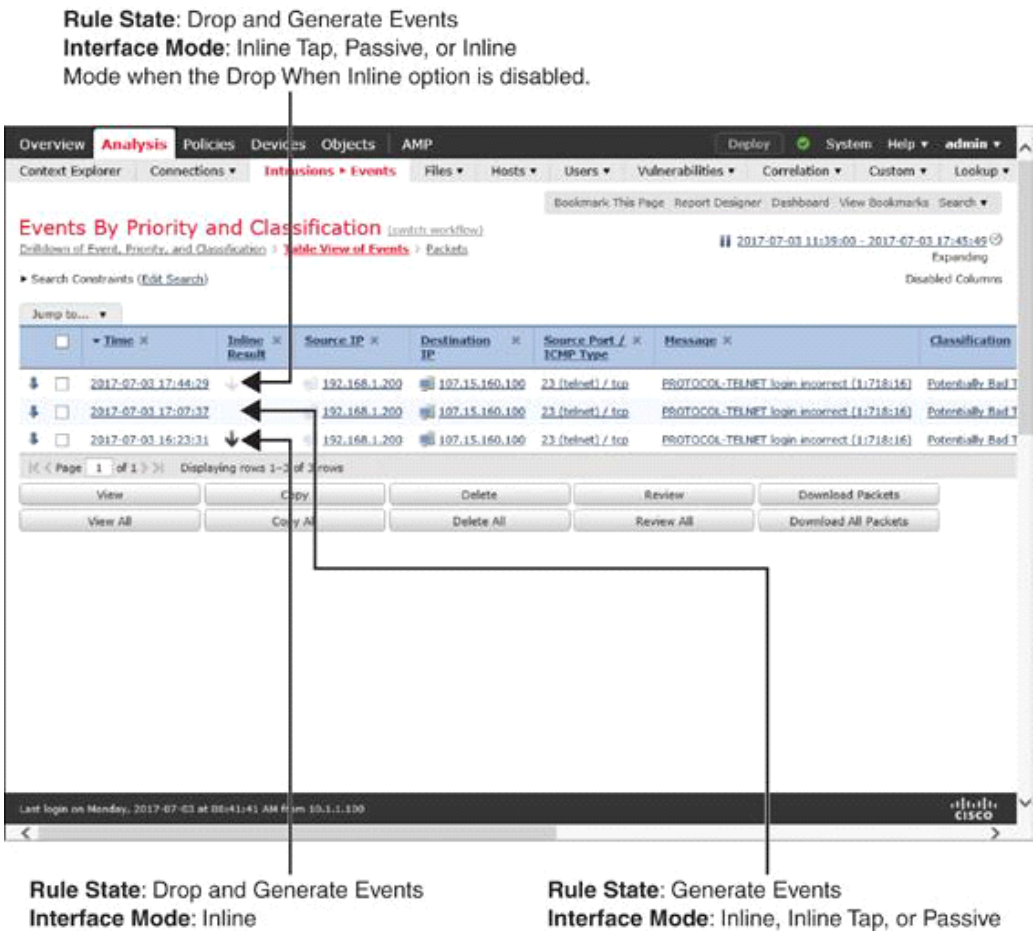
```
external-user@Fedora:~$ telnet 192.168.1.200
Trying 192.168.1.200... Open
Connected to 192.168.1.200. Ubuntu login: internal-user
Password: ********
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-81-generic x86_64)
internal-user@Ubuntu:~$
! When a login attempt is unsuccessful

external-user@Fedora:~$ telnet 192.168.1.200
Trying 192.168.1.200... Open
Connected to 192.168.1.200.
Ubuntu login: internal-user
Password: <incorrect_password>
Login incorrect
Ubuntu login:
```

备注：有的服务器返回的失败消息可能是loging failed，有另一个Snort规则与之匹配1：492.如果返回结果是可以自定义的，那么客户也可以自定义Snort规则匹配字段即可。

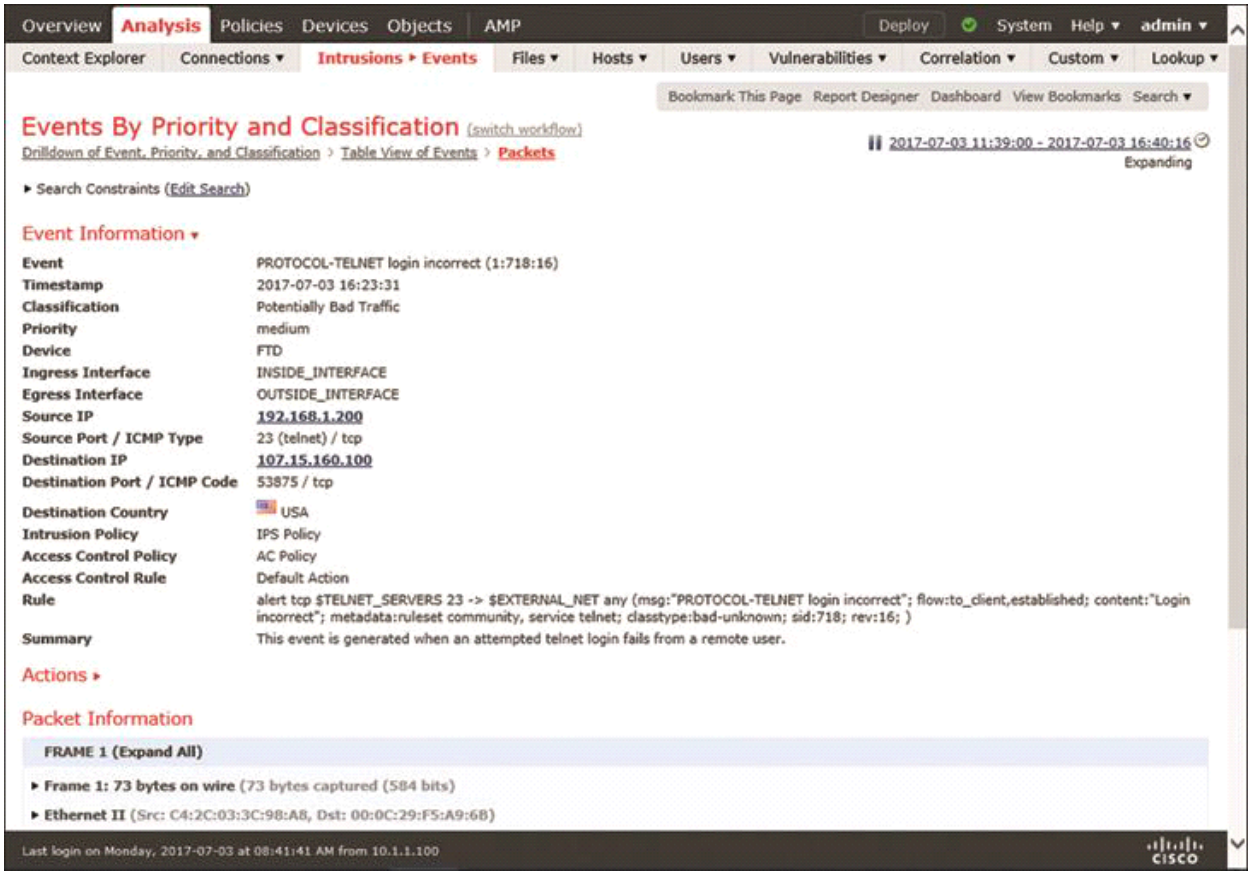
5.2：分析

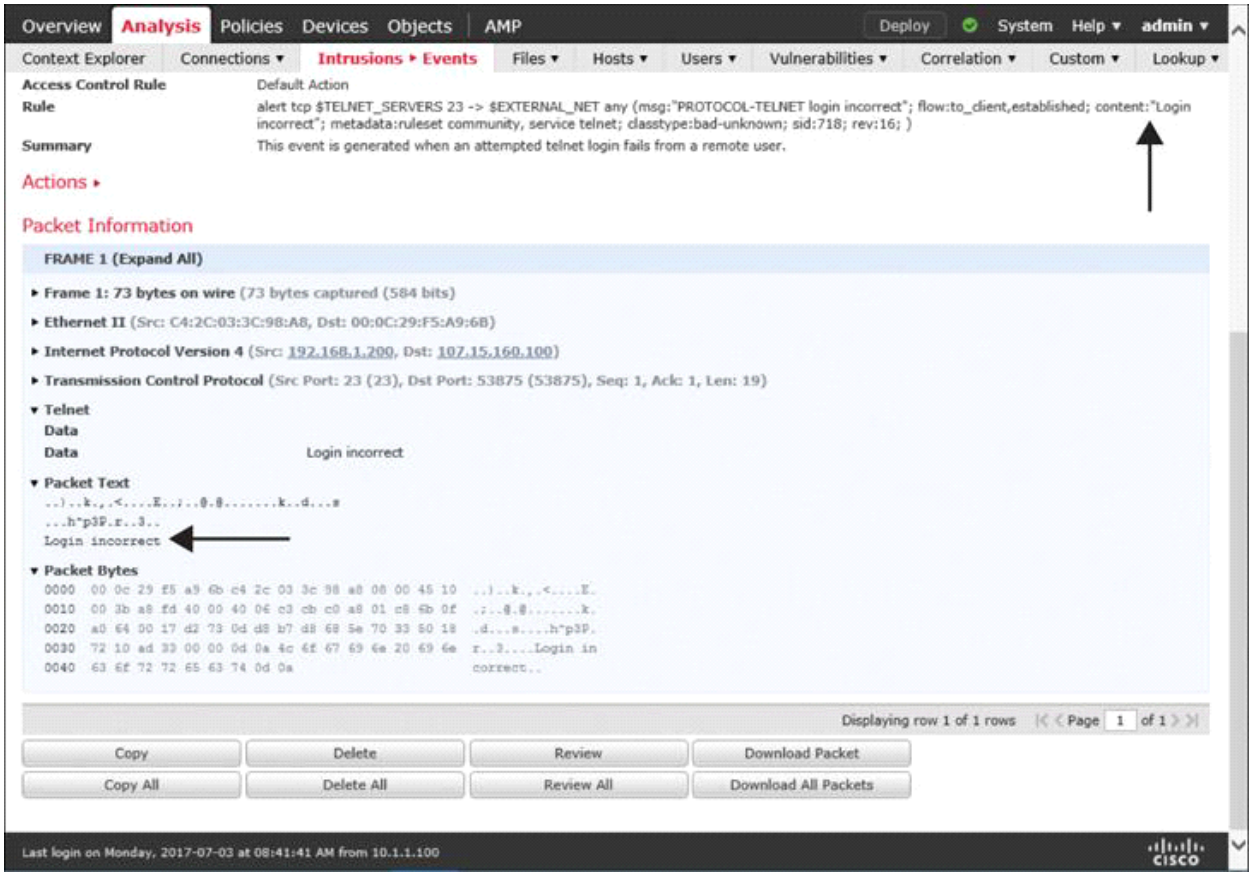
根据Snort规则状态，策略设置和接口模式，Firepower系统对相同的Telnet流量采取不同的行为，可以为同一个Snort规则找到不同类型的入侵事件。

- 比如接口模式=inlinemode，入侵策略=drop when inline，snort rule action = drop and generate events，那么FTD阻塞匹配数据包，并显示已丢弃深灰色向下箭头
- 如果入侵策略没有选择drop when inline，或者接口模式=inline tap/passive mode，则FMC显示would have dropped（本应丢弃事件）浅灰色向下箭头

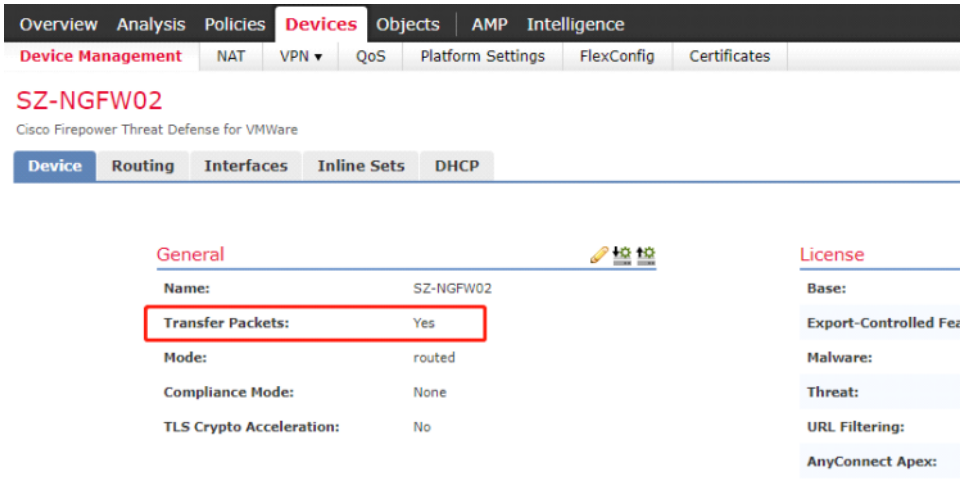


进一步分析入侵事件，单技每一列向下的箭头即可打开追踪入侵事件中的数据和数据包，用以确认事件是否为误报





默认设备开启了FTD捕获Snort规则匹配数据包的特性，如果不希望存储，则可以关闭（不推荐关闭）



6：TroubleShooting

分析数据包追踪数据，确定数据包是否被snort阻塞，或被其他组件阻塞

6.1：利用Capture工具在发起telnet流量前进行捕获

```
> capture telnet_inside trace interface INSIDE_INTERFACE match tcp any any eq 23
```

6.2：实时查看捕获情况

```
> show capture capture telnet_inside type raw-data trace interface INSIDE_INTERFACE
```

```
[Capturing - 0 bytes]
  match tcp any any eq telnet
```

6.3：发起telnet，输入错误凭据，查看抓到的数据包

```
> show capture telnet_inside 119 packets captured
     1: 20:23:21.086802 107.15.160.100.53875 > 192.168.1.200.23: S 1751019501:1751019501(0) win 4128 <mss 1460>
     2: 20:23:21.087229 107.15.160.100.53875 > 192.168.1.200.23: S 1751019501:1751019501(0) win 4128 <mss 1460>
     3: 20:23:21.087565 192.168.1.200.23 > 107.15.160.100.53875: S 232306554:232306554(0) ack 1751019502 win 29200 <mss 1460>
     4: 20:23:21.087702 192.168.1.200.23 > 107.15.160.100.53875: S 232306554:232306554(0) ack 1751019502 win 29200 <mss 1460>
     5: 20:23:21.089717 107.15.160.100.53875 > 192.168.1.200.23: . ack 232306555 win 4128
```

6.4：追踪第一个包数据

```
> show capture telnet_inside packet-number 1 trace
     Phase: 7
     Type: EXTERNAL-INSPECT
     Subtype:
     Result: ALLOW
     Config:
     Additional Information:
     Application: 'SNORT Inspect'

     Phase: 8
     Type: SNORT
     Subtype:
     Result: DROP
     Config:
     Additional Information:
     Snort Verdict: (block-packet) drop this packet
```

可以看到，被Snort block了

6.5：Snort统计丢弃的数据包信息

```
> show asp drop

Frame drop:

  Snort requested to drop the frame (snort-drop)                5

  FP L2 rule drop (l2_acl)                                      1

Last clearing: 20:23:14 UTC Jul 3 2017 by enable_1
```

可以看到用户输入telnet错误凭据后，snort请求丢弃五个数据帧