给我的网关部署了 Endlessh

用 Endlessh 做 SSH tarpit, 并通过 dstnat 把蜜罐端口重定向.

---

1. Endlessh

Endlessh 是一个 SSH tarpit[1]. 它不提供真正的登录服务, 在 SSH 握手阶段慢速发送永远不会结束的 banner, 例如:

```
debug1: Local version string SSH-2.0-OpenSSH_10.3
debug1: kex_exchange_identification: banner line 0: *PTqea>;2,6iXD-?>34IW*
debug1: kex_exchange_identification: banner line 1: e
debug1: kex_exchange_identification: banner line 2: n>{o%].A/
debug1: kex_exchange_identification: banner line 3: ' )'#c6^E<(%yJAe*IIX]Kz,?:
```

我在集群里跑的是 `endlessh-go`:

```yaml
image: shizunge/endlessh-go:latest
args:
  - -interval_ms=1000
  - -logtostderr
  - -v=1
  - -enable_prometheus
  - -host=[::]
  - -port=22
  - -prometheus_host=[::]
  - -proxy_protocol_enabled
```

此外, 为了更方便地看到校园网内的攻击行为 (校园网把 22 端口的校外访问 ban 了, 因此连接 22 端口的一定来自校内), 我部署了两份, 一份侦听 22, 一份侦听 23. 后面用 nftables 把剩余的端口全部打进了 23.

2. 目的地址重定向

另一个有用的部分是边界上的 `dstnat`. 扫描器不一定只扫 22, 也可能扫一串常见或随机端口. 为了看到所有这些 SSH 连接, 正如前面所说, 我把不是实际服务的 TCP 端口全部重定向了.

```
set tunet_real_tcp {
  type inet_service
  flags constant
  elements = { 22, 23, 80, 443 }
}

chain dstnat {
  type nat hook prerouting priority dstnat
  policy accept
  iifname tunet meta l4proto tcp tcp dport != @tunet_real_tcp redirect to :23
}
```

同时配置 nginx:

```
stream {
    server {
        listen 443;
        listen [::]:443;
        proxy_pass traefik:443;
        proxy_protocol on;
    }
    server {
        listen 443 udp;
        listen [::]:443 udp;
        proxy_pass traefik:443;
        proxy_protocol on;
    }
    server {
        listen 23;
        listen [::]:23;
        proxy_pass endlessh:24;
        proxy_protocol on;
    }
}
```

Bibliography

[1] *Endlessh: An SSH Tarpit*, https://nullprogram.com/blog/2019/03/22/.