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, 而是在 SSH 握手阶段慢速发送永远不会结束的 banner. 对扫描器来说, 这个端口看起来像 SSH. 对服务端来说, 它只是在消耗对方的连接时间.

我在集群里跑的是 `endlessh-go`. 配置很少, 重点是监听 SSH 端口, 把日志打到 stderr, 并打开 Prometheus metrics:

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

为了避免和真实 SSH 混在一起, 我保留了一个面向校园网的 endlessh 实例. 它走单独的 TCP 入口, 这样日志和指标可以分开看.

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. 与其为每个端口单独观察, 可以把不是实际服务的 TCP 端口重定向到同一个入口.

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

chain dstnat {
  type nat hook prerouting priority dstnat - 1
  policy accept
```

```
  iifname tunet meta l4proto tcp tcp dport != @tunet_real_tcp redirect to :23
}
```

同时配置 nginx:

```
cat /etc/nginx/nginx.conf
user www-data;
worker_processes auto;
worker_cpu_affinity auto;
pid /run/nginx.pid;
error_log /var/log/nginx/error.log;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 2048;
}

stream {
    server {
        listen 443;
        listen [::]:443;
        proxy_pass svc.t.adamanteye.cc:443;
        proxy_protocol on;
    }
    server {
        listen 443 udp;
        listen [::]:443 udp;
        proxy_pass svc.t.adamanteye.cc:443;
        proxy_protocol on;
    }
    server {
        listen 23;
        listen [::]:23;
        proxy_pass svc.t.adamanteye.cc:24;
        proxy_protocol on;
    }
}
```

这样做之后, 真实服务仍然按原来的端口暴露, 扫到其他端口的连接会被收束到同一处.

Bibliography

[1] “Endlessh: an SSH Tarpit.” [Online]. Available: https://nullprogram.com/blog/2019/03/22/