Rizky Pratama Suhud

1. Judul Laporan

Laporan pengujian keamanan VM RickdiculouslyEasy

2. Pendahuluan

2.1 Deskripsi Proyek

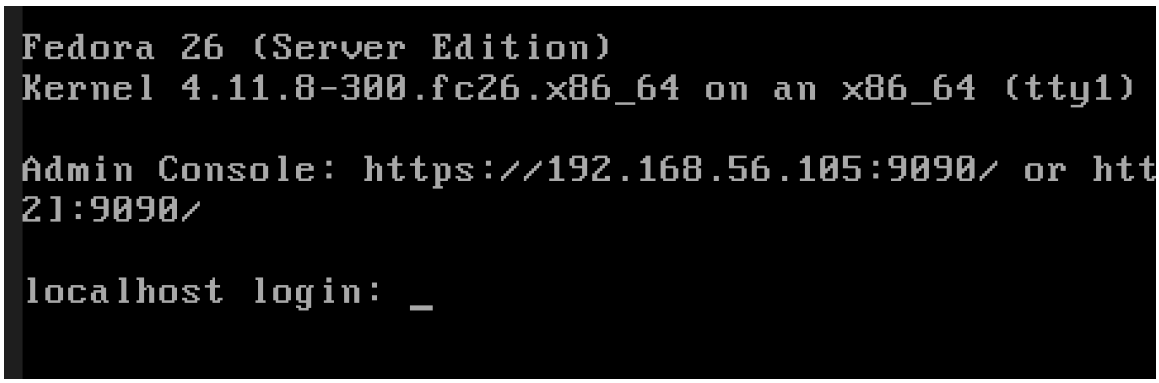
Pengujian keamanan ini dilakukan terhadap virtual machine *RickdiculouslyEasy* dengan tujuan menilai tingkat keamanan layanan yang berjalan serta menemukan flag sebagai indikator keberhasilan eksploitasi. Pengujian dilakukan di lingkungan laboratorium terisolasi dengan izin dari pihak penyelenggara; kegiatan bersifat etis dan tidak berdampak pada sistem produksi.

Ruang lingkup pengujian mencakup tahap reconnaissance, scanning, enumerasi, identifikasi kerentanan, eksploitasi, dan pengumpulan flag. Tahapan dilakukan sistematis dengan penggunaan alat bantu yang sesuai standar pengujian penetrasi.

Hasil pengujian menemukan beberapa kerentanan signifikan, termasuk FTP anonymous access, weak SSH credentials, dan directory listing pada web server. Nmap efektif untuk pemetaan layanan, sementara hydra digunakan untuk brute-force autentikasi dan alat lainnya untuk eksploitasi layanan tertentu.

2.2 Gambaran VM

VM: Rickdiculouslyeasy. Target berisi layanan ftp, http, ssh, dan cockpit yang digunakan untuk latihan eksploitasi dan pengumpulan flag.



2.3 Tujuan

Menemukan flag yang tersembunyi di vm dan mendokumentasikan proses eksploitasi untuk penilaian dan pembelajaran (total poin 130).

2.4 Alat yang Digunakan

Tabel 1Tools yang digunakan

Tool	Fungsi	Contoh Command
Nmap	Scanning & enumeration	nmap -A -p- 192.168.56.105
Nikto	Pemeriksaan kerentanan web	nikto -h http://192.168.56.105
Hydra	Brute force SSH / otentikasi	hydra -l user -P wordlist ssh://host -s 22222
Netcat	Uji koneksi, interaksi TCP	nc 192.168.56.105 13337
Binwalk	Analisis berkas (data tertanam)	binwalk Safe_Password.jpg
unzip	Ekstraksi arsip	Unzip journal.txt.zip

3. langkah-langkah eksekusi

- Langkah 1 : Footprinting

Deskripsi: verifikasi konektivitas dan identifikasi awal terhadap VM *RickdiculouslyEasy*.

Command: ping 192.168.56.105

traceroute 192.168.56.105

whois 192.168.56.105

dig 192.168.56.105

nslookup 192.168.56.105

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.56.105
PING 192.168.56.105 (192.168.56.105) 56(84) bytes of data.
64 bytes from 192.168.56.105: icmp_seq=1 ttl=64 time=0.193 ms
64 bytes from 192.168.56.105: icmp_seq=2 ttl=64 time=0.204 ms
64 bytes from 192.168.56.105: icmp_seq=3 ttl=64 time=0.183 ms
64 bytes from 192.168.56.105: icmp_seq=4 ttl=64 time=0.224 ms
64 bytes from 192.168.56.105: icmp_seq=5 ttl=64 time=0.229 ms
^C
── 192.168.56.105 ping statistics ──
5 packets transmitted, 5 received, 0% packet loss, time 4099ms
rtt min/avg/max/mdev = 0.183/0.206/0.229/0.017 ms

┌──(kali㉿kali)-[~]
└─$ traceroute 192.168.56.105
traceroute to 192.168.56.105 (192.168.56.105), 30 hops max, 60 byte packets
 1  192.168.56.105 (192.168.56.105)  0.206 ms  0.167 ms  0.103 ms

┌──(kali㉿kali)-[~]
└─$ whois 192.168.56.105
getaddrinfo(whois.arin.net): Temporary failure in name resolution

┌──(kali㉿kali)-[~]
└─$ dig 192.168.56.105
;; UDP setup with 10.0.2.3#53(10.0.2.3) for 192.168.56.105 failed: network unreachable.
;; no servers could be reached
;; UDP setup with 10.0.2.3#53(10.0.2.3) for 192.168.56.105 failed: network unreachable.
;; no servers could be reached
;; UDP setup with 10.0.2.3#53(10.0.2.3) for 192.168.56.105 failed: network unreachable.
;; UDP setup with fd17:625c:f037:2::3#53(fd17:625c:f037:2::3) for 192.168.56.105 failed: network unreachable.
;; no servers could be reached

┌──(kali㉿kali)-[~]
└─$ nslookup 192.168.56.105
;; UDP setup with 10.0.2.3#53(10.0.2.3) for 105.56.168.192.in-addr.arpa. failed: network unreachable.
;; no servers could be reached
;; UDP setup with 10.0.2.3#53(10.0.2.3) for 105.56.168.192.in-addr.arpa. failed: network unreachable.
;; no servers could be reached
;; UDP setup with 10.0.2.3#53(10.0.2.3) for 105.56.168.192.in-addr.arpa. failed: network unreachable.
;; UDP setup with fd17:625c:f037:2::3#53(fd17:625c:f037:2::3) for 105.56.168.192.in-addr.arpa. failed: network
chable.
;; no servers could be reached
```

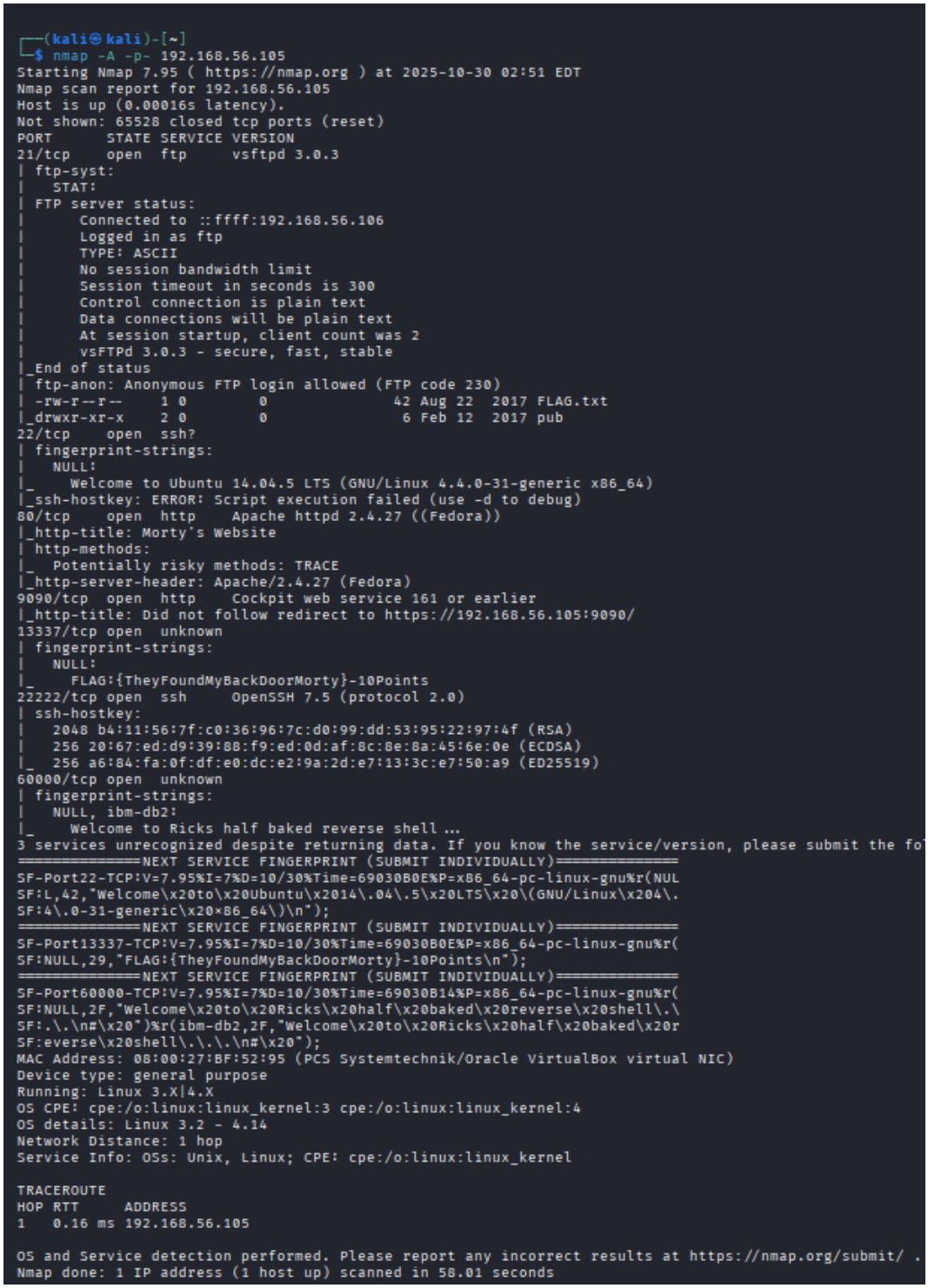
Gambar 1 Footprinting

Pada tahap footprinting, dilakukan verifikasi konektivitas dan identifikasi awal terhadap VM *RickdiculouslyEasy*. Hasil perintah ping 192.168.56.105 menunjukkan host aktif dengan respon stabil (rata-rata waktu 0.178 ms dan tanpa packet loss), menandakan target berada dalam jaringan lokal yang dapat dijangkau. Perintah traceroute juga mengonfirmasi bahwa host berada dalam satu lompatan jaringan (1 hop), sehingga target termasuk dalam lingkungan lab internal. Sementara itu, perintah whois, dig, dan nslookup tidak memberikan hasil karena alamat IP 192.168.56.105 merupakan alamat private yang tidak terdaftar pada layanan publik DNS/WHOIS. Berdasarkan hasil ini, tahap footprinting dinyatakan berhasil memastikan konektivitas dan konteks jaringan sebelum dilanjutkan ke tahap scanning

- Langkah 2: scanning port dengan nmap

Deskripsi: lakukan scanning untuk mengidentifikasi port terbuka dan layanan yang berjalan pada ip target.

Command: nmap -A -p- 192.168.56.105



Hasil pemindaian menggunakan *Nmap* dengan opsi -A terhadap host 192.168.56.105 mendeteksi beberapa port terbuka, yaitu port 21 (FTP) yang menjalankan layanan *vsftpd 3.0.3* dengan akses anonim yang diizinkan serta file *FLAG.txt* dapat diakses, port 22 dan 22222 yang

Gambar 2 Hasil scanning nmap menunjukkan port terbuka dan layanan yang terdeteksi

menjalankan layanan *OpenSSH*, port 80 yang menjalankan *Apache HTTPD 2.4.27 (Fedora)*, port 9090 untuk layanan *Cockpit web service*, serta port 60000 yang menampilkan banner *Rick's half baked reverse shell*. Informasi tambahan menunjukkan sistem operasi yang digunakan adalah *Ubuntu 14.04.5 LTS* (Linux kernel 4.4.0-31-generic) dan perangkat berjalan di lingkungan virtual *Oracle VirtualBox*.

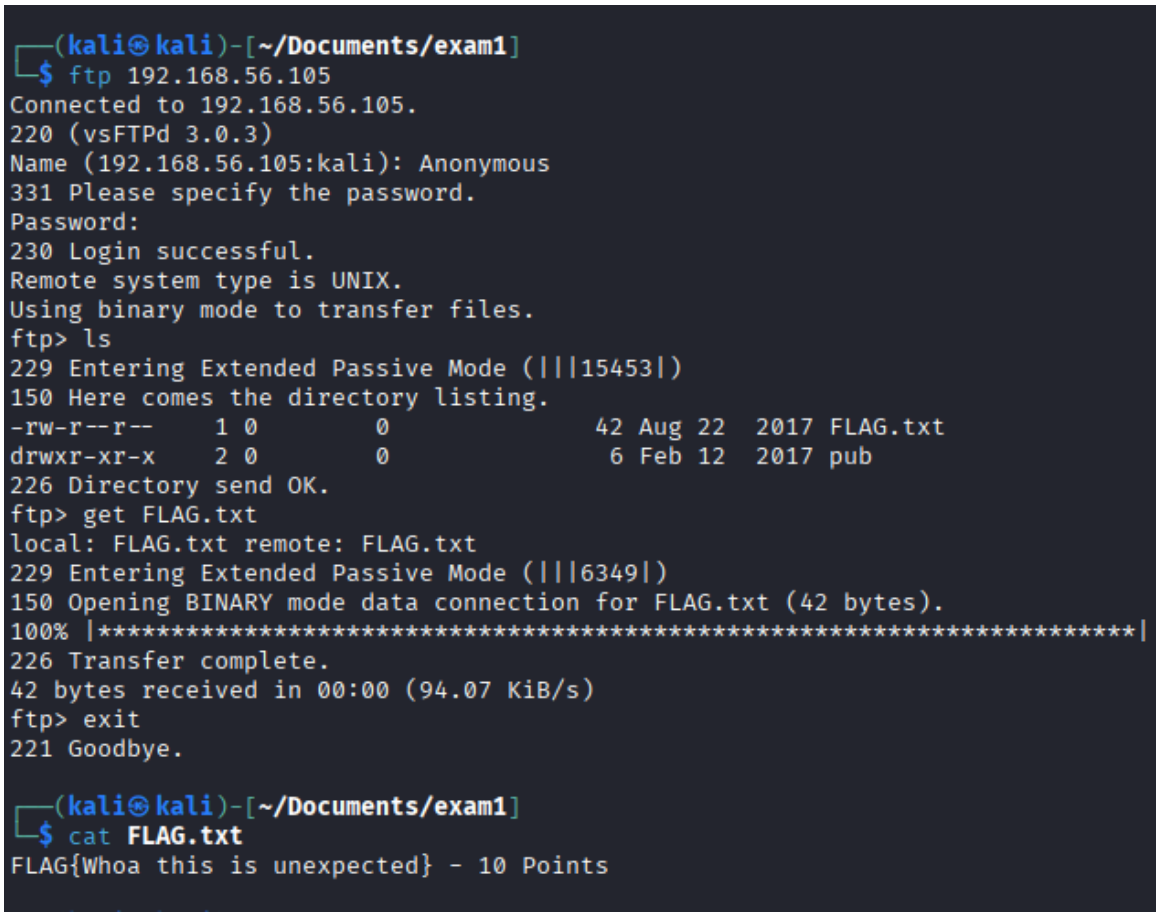
Tabel 2 Hasil Scanning Nmap

Port	Status	Service	Keterangan
21	Open	FTP	vsftpd 3.0.3. Anonymous login diizinkan. Ditemukan file **FLAG.txt** (isi: *FLAG.txt* ditemukan di direktori utama FTP).
22	Open	SSH	OpenSSH 7.5 (Ubuntu 14.04.5 LTS)
80	Open	HTTP	Apache 2.4.27 (Fedora) – Website utama ("Morty's Website"). Potensi eksploitasi melalui metode HTTP TRACE.
9090	Open	HTTP	Cockpit web service 161 atau versi sebelumnya
13337	Open	Unknown	Custom service – menampilkan **FLAG**
22222	Open	SSH	OpenSSH 7.5 (protocol 2.0)
60000	Open	Unknown	Service tidak terdeteksi. Banner: "Welcome to Ricks half baked reverse shell …". Indikasi reverse shell service.

- Langkah 3: Pemeriksaan FTP

Deskripsi: menguji akses anonim dan mengunduh FLAG.txt

Command contoh: ftp 192.168.56.105



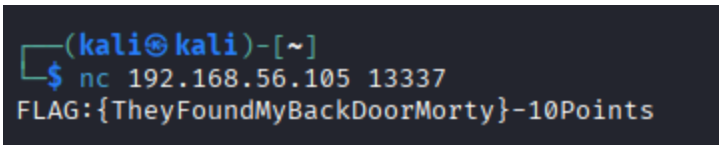
Gambar 3 Menguji akses anonim

Gambar 3 Menunjukkan proses pengujian akses anonim pada layanan FTP dengan alamat IP 192.168.56.105. Dari hasil pengujian, terlihat bahwa server FTP mengizinkan login menggunakan akun anonim tanpa memerlukan kata sandi. Setelah berhasil masuk, dilakukan pemeriksaan direktori dan ditemukan file *FLAG.txt*. File tersebut kemudian diunduh menggunakan perintah get FLAG.txt, dan setelah dibuka dengan perintah cat FLAG.txt, diperoleh isi file berupa *flag* yang menandakan keberhasilan eksploitasi akses anonim pada server FTP.

- Langkah 4: Eksploitasi dan pengumpulan flag

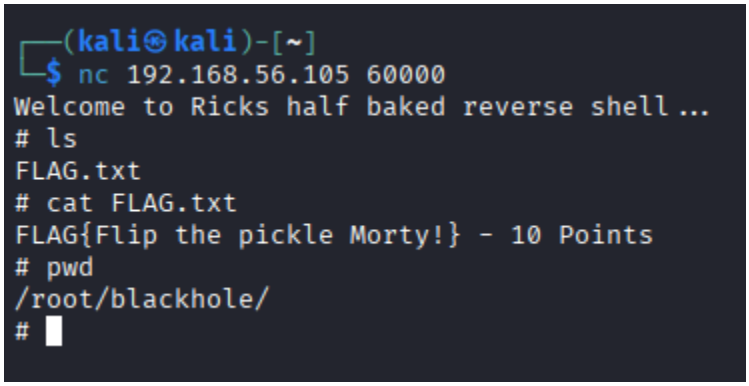
Deskripsi: Pada tahap ini, layanan yang teridentifikasi selama pemindaian (scanning) diuji untuk memperoleh akses dan flag. Eksploitasi dilakukan terhadap dua layanan yang mencurigakan: port 13337 (custom service yang menampilkan flag) dan port 60000 (service yang memberikan shell). Teknik yang dipakai adalah koneksi TCP langsung menggunakan netcat (bind shell/backdoor). Semua aktivitas dicatat sebagai bukti.

Command contoh: nc 192.168.56.105 13337



Gambar 4 Koneksi ke port 13337 menampilkan flag {TheyFoundMyBackDoorMorty} – 10 Points.

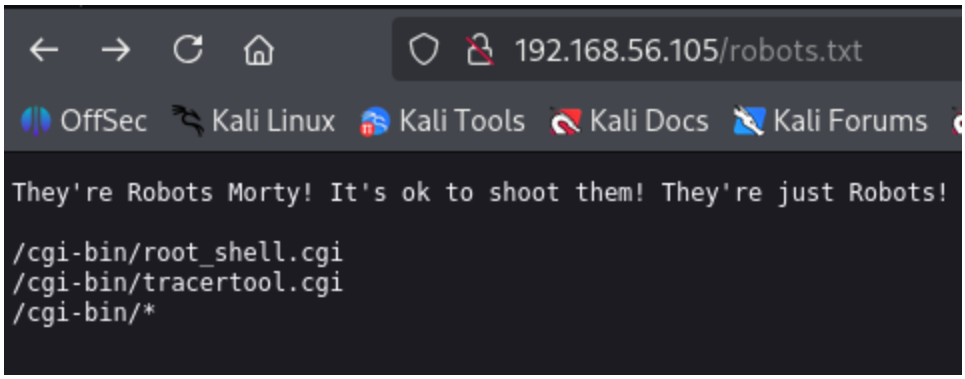
Perintah nc 192.168.56.105 13337 membuka koneksi TCP ke port 13337 dan server langsung mengembalikan teks FLAG:{TheyFoundMyBackDoorMorty}-10Points, menunjukkan bahwa pada port tersebut terdapat layanan sederhana yang menampilkan flag tanpa perlu autentikasi atau interaksi lebih lanjut.



Gambar 5 Koneksi TCP ke port 60000 membuka shell dengan hak istimewa root; flag ditemukan di direktori root.

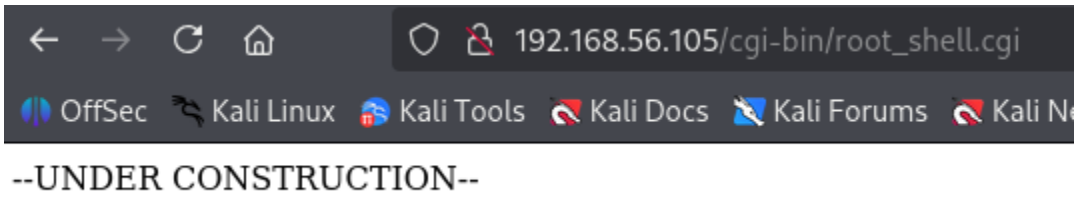
Perintah nc 192.168.56.105 60000 berhasil membuka koneksi ke layanan pada port 60000 yang menampilkan banner *"Welcome to Ricks half baked reverse shell ..."* dan langsung memberikan prompt # sehingga perintah shell bisa dijalankan hasilnya flag FLAG{Flip the pickle Morty!} - 10 Points ditemukan di /root/blackhole/ dan prompt # menandakan shell berjalan dengan hak istimewa root, yang mengindikasikan adanya backdoor atau bind-shell pada host tersebut yang mendengarkan koneksi masuk

Setelah akses awal berhasil diperoleh melalui koneksi TCP ke layanan backdoor pada port 13337 dan 60000 (menampilkan flag dan shell root), fokus eksploitasi dialihkan ke permukaan web untuk menemukan jalur akses tambahan yang dapat dimanfaatkan untuk eskalasi atau pivoting. Pemeriksaan cepat pada http://192.168.56.105/robots.txt mengungkap entri yang menunjuk ke skrip di /cgi-bin, antara lain /cgi-bin/root_shell.cgi dan /cgi-bin/tracertool.cgi, sehingga langkah selanjutnya difokuskan pada verifikasi langsung path-path tersebut untuk menilai apakah endpoint dapat diakses atau mengeksekusi perintah tanpa autentikasi. Hasil verifikasi ini akan menjadi dasar pengujian aplikasi web lanjutan, misalnya pengujian parameter untuk RCE, pemeriksaan autentikasi, dan validasi input, sebelum pemindaian otomatis seperti Nikto dijalankan untuk enumerasi yang lebih luas.



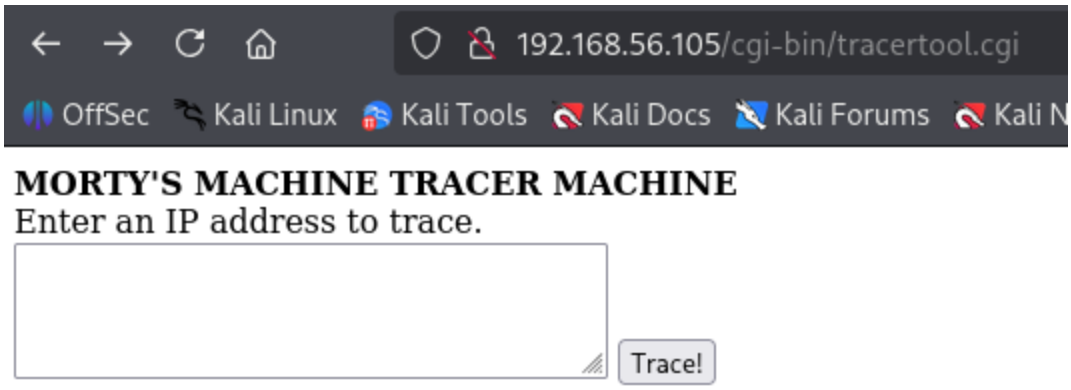
Gambar 6 Isi /robots.txt mengungkap path sensitif di /cgi-bin, termasuk /cgi-bin/root_shell.cgi dan /cgi-bin/tracertool.cgi.

Setelah menemukan entri di /robots.txt, setiap path di /cgi-bin diakses langsung melalui browser untuk menilai respons dan kemungkinan eksekusi skrip. Untuk tiap endpoint dibuka URL-nya (http://192.168.56.105/cgi-bin/root_shell.cgi dan http://192.168.56.105/cgi-bin/tracertool.cgi). Jika halaman menampilkan output, banner, atau form yang memungkinkan input, endpoint ditandai untuk pengujian lanjutan; jika muncul status error atau halaman kosong, endpoint dianggap tidak dapat diakses.



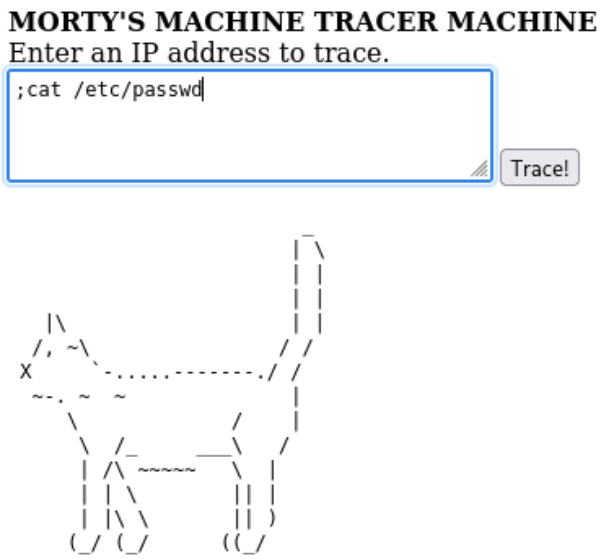
Gambar 7 Tampilan halaman /cgi-bin/root_shell.cgi yang menunjukkan status *Under Construction*.

Hasil pemeriksaan langsung terhadap direktori /cgi-bin, khususnya pada path /cgi-bin/root_shell.cgi, menunjukkan tampilan pesan *"--UNDER CONSTRUCTION--"* tanpa adanya elemen interaktif atau respons dinamis. Berdasarkan verifikasi lanjutan, tidak ditemukan aktivitas, input field, maupun fungsi eksekusi perintah pada halaman tersebut. Dengan demikian, endpoint ini dikategorikan sebagai **nonaktif** dan tidak menunjukkan indikasi kerentanan atau celah eksploitasi yang dapat dimanfaatkan.



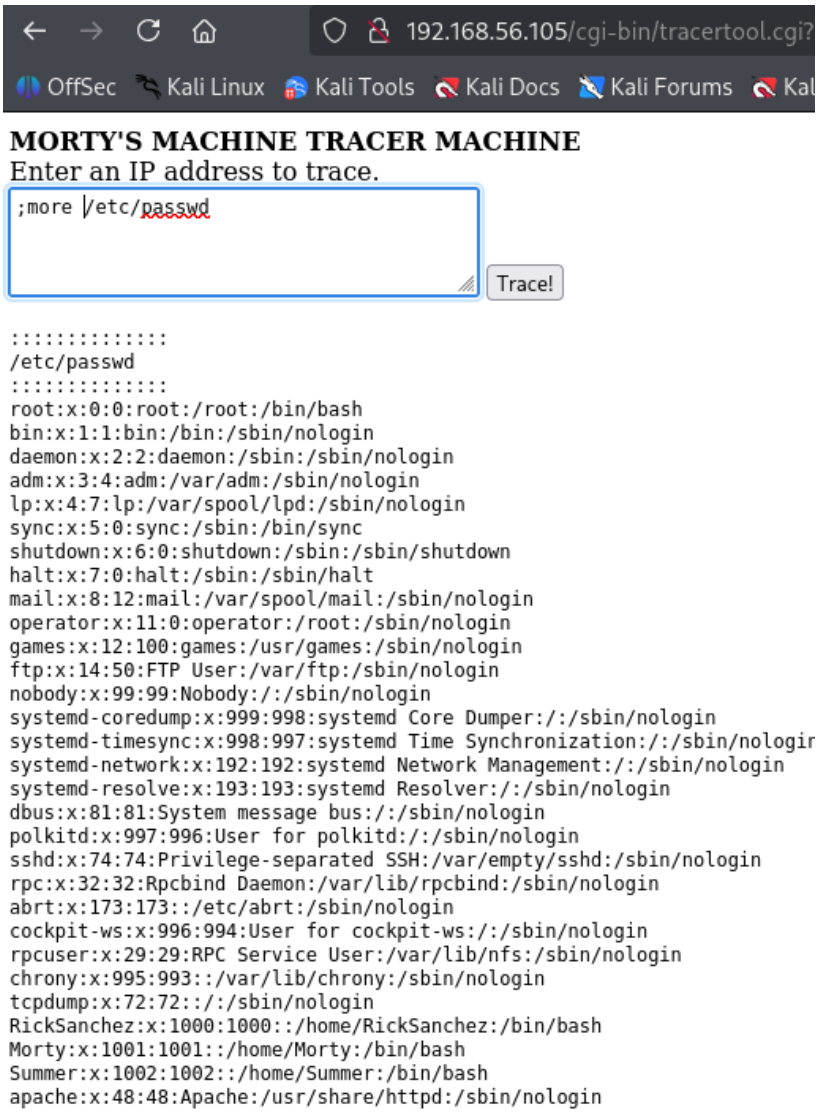
Gambar 8 Tampilan halaman /cgi-bin/tracertool.cgi yang berfungsi untuk melakukan traceroute berdasarkan input alamat IP

Hasil pemeriksaan pada direktori /cgi-bin menunjukkan bahwa halaman /cgi-bin/tracertool.cgi aktif dan menampilkan form sederhana dengan judul *"Morty's Machine Tracer Machine"*. Halaman ini menyediakan kolom input untuk memasukkan alamat IP serta tombol Trace! untuk menjalankan proses pelacakan rute jaringan (*traceroute*). Berdasarkan tampilannya, fungsi ini bekerja dengan mengirimkan data input ke server untuk diproses, kemudian menampilkan hasilnya kembali kepada pengguna. Fitur semacam ini perlu diperhatikan karena jika tidak memiliki validasi input yang baik, dapat berpotensi menimbulkan risiko Command Injection apabila input tidak disanitasi dengan benar, karena perintah yang dikirimkan ke sistem dapat dimanipulasi untuk mengeksekusi instruksi tambahan. Oleh karena itu, halaman ini dicatat sebagai endpoint aktif yang relevan untuk diuji lebih lanjut pada tahap eksploitasi web berikutnya.



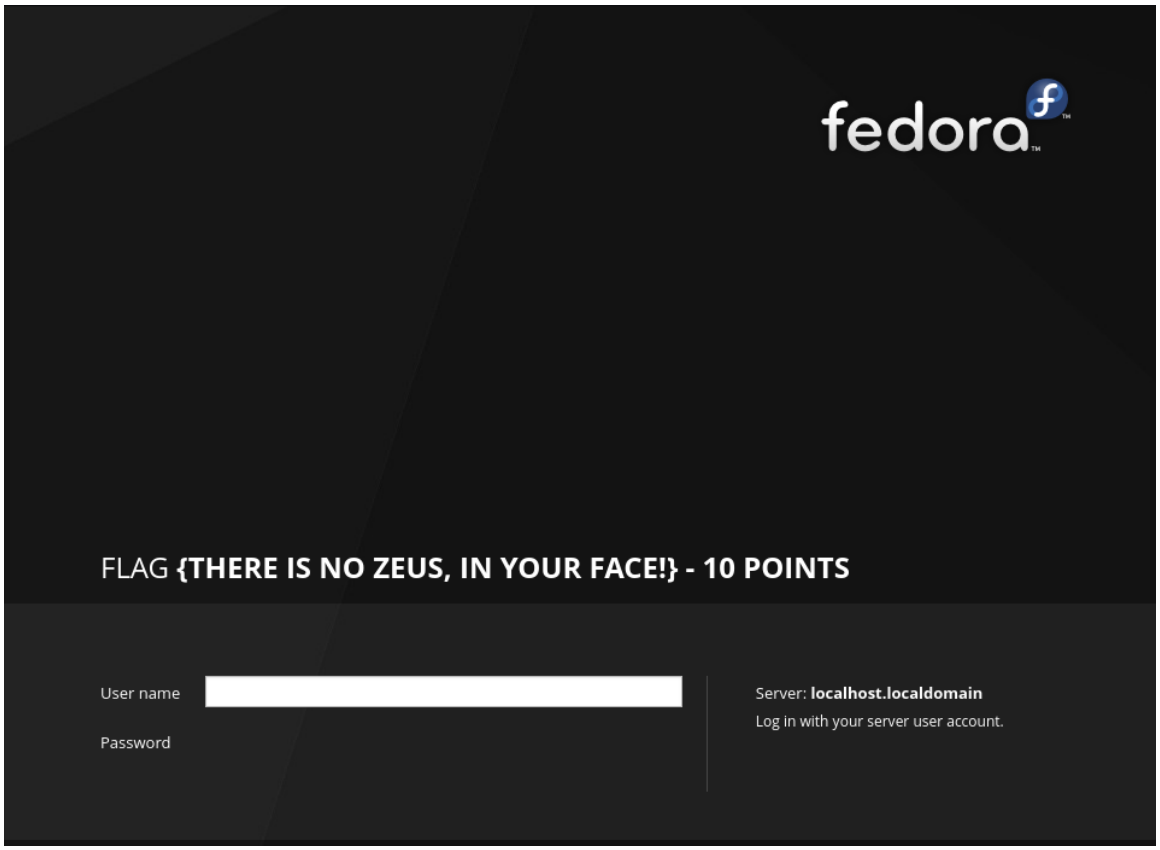
Gambar 9 Input terdeteksi melakukan *command injection* pada /cgi-bin/tracertool.cgi; contoh payload ;cat /etc/passwd menghasilkan output di halaman.

Pengujian pada /cgi-bin/tracertool.cgi mengungkap adanya kerentanan *command injection* ketika payload ;cat /etc/passwd dimasukkan dan server menampilkan isi file tersebut, sehingga dapat disimpulkan bahwa input pengguna diteruskan ke shell di sisi server tanpa validasi yang memadai. File /etc/passwd dipilih sebagai bukti karena umumnya dapat dibaca oleh proses non-privileged dan memberikan konfirmasi cepat bahwa perintah benar-benar dieksekusi. Selain itu isi file ini (daftar akun sistem dan shell default) berguna untuk tahap enumerasi lebih lanjut namun tidak mengekspos kata sandi karena kredensial tersimpan terpisah di /etc/shadow. Penggunaan /etc/passwd sebagai proof-of-concept dianggap non-destruktif dan etis dibandingkan mencoba akses ke file sensitif lainnya. Temuan ini menandakan perlunya perbaikan validasi input, pembatasan hak eksekusi skrip CGI, dan audit lebih lanjut.



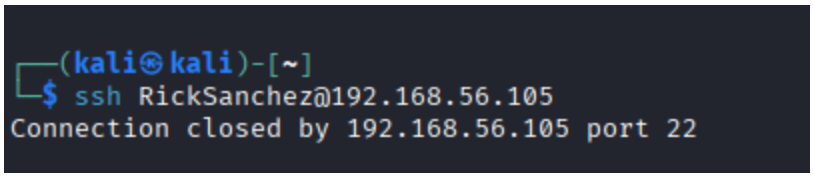
Gambar 10 Payload ;more /etc/passwd pada /cgi-bin/tracertool.cgi menampilkan isi file /etc/passwd, mengonfirmasi terjadinya command injection.

Saat memasukkan input more /etc/passwd ke form pada /cgi-bin/tracertool.cgi, server mengeksekusi perintah tersebut dan menampilkan isi file /etc/passwd di halaman. Keluaran ini mengonfirmasi bahwa parameter input tidak disanitasi dan dapat dipasangkan ke shell di sisi server, sehingga memungkinkan eksekusi perintah arbitrary. Isi /etc/passwd yang tampil memperlihatkan daftar akun sistem (mis. root, system users, serta akun pengguna biasa seperti RickSanchez dan Morty) dan berfungsi sebagai bukti non-destruktif bahwa perintah berhasil dijalankan tanpa mengungkapkan kata sandi (kata sandi biasanya disimpan terpisah di /etc/shadow). Temuan ini menandakan risiko serius terhadap kerahasiaan dan integritas sistem.



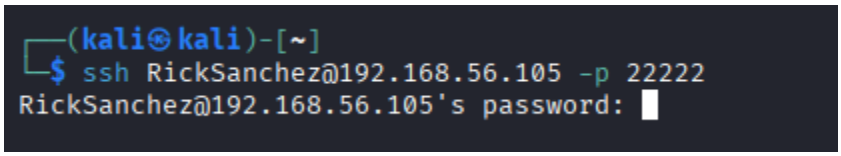
Gambar 11 Pemeriksaan layanan pada port 9090 menghasilkan penemuan flag yang ditampilkan pada antarmuka web.

Kemudian pemeriksaan dilanjutkan dengan mengakses ke layanan yang berjalan pada port 9090 melalui browser dan endpoint administratifnya diperiksa untuk mencari konten sensitif; halaman menampilkan flag FLAG {THERE IS NO ZEUS, IN YOUR FACE!} - 10 POINTS. Setelah pemeriksaan lebih lanjut terhadap endpoint, direktori terkait, dan potensi fungsi administrasi, tidak ditemukan temuan atau data tambahan yang relevan di layanan tersebut. Seluruh interaksi dicatat dan hasil tangkapan layar disimpan sebagai bukti untuk dokumentasi.



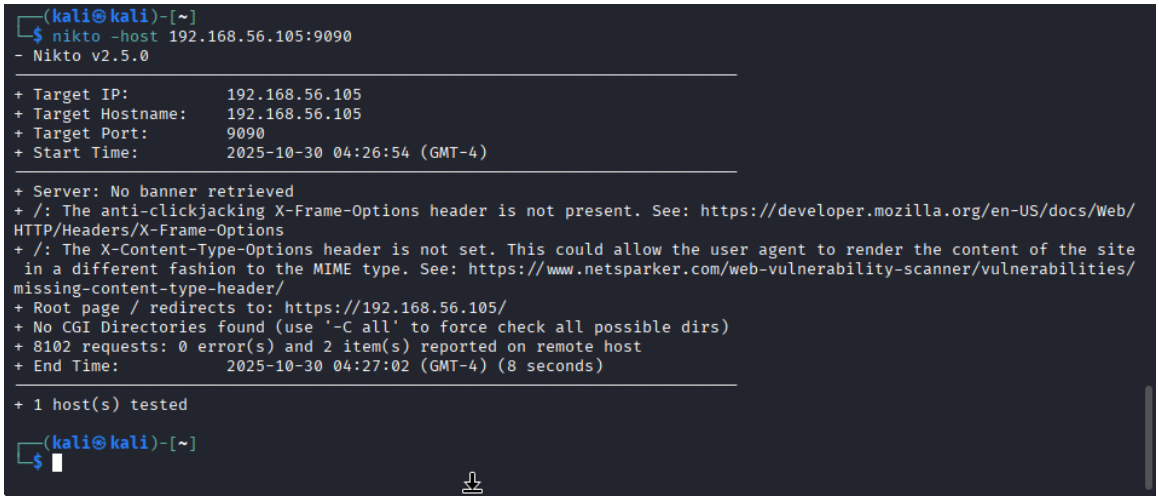
Gambar 12 Percobaan SSH ke RickSanchez@192.168.56.105 gagal dengan pesan *"Connection closed by 192.168.56.105 port 22"*.

Koneksi SSH ke akun *RickSanchez* pada port standar (22) ditutup segera oleh host target, sehingga autentikasi tidak sempat dilanjutkan. Hal ini konsisten dengan temuan sebelumnya (user *RickSanchez* terdaftar di /etc/passwd) dan menunjukkan bahwa meskipun akun ada, akses SSH melalui port 22 saat ini tidak tersedia atau dibatasi. Penyebab yang mungkin meliputi konfigurasi sshd yang menolak koneksi dari host/klien tertentu, aturan firewall atau layanan yang menutup koneksi, mekanisme proteksi yang memutus koneksi setelah percobaan, atau port 22 dikonfigurasi sebagai trap/non-interaktif.



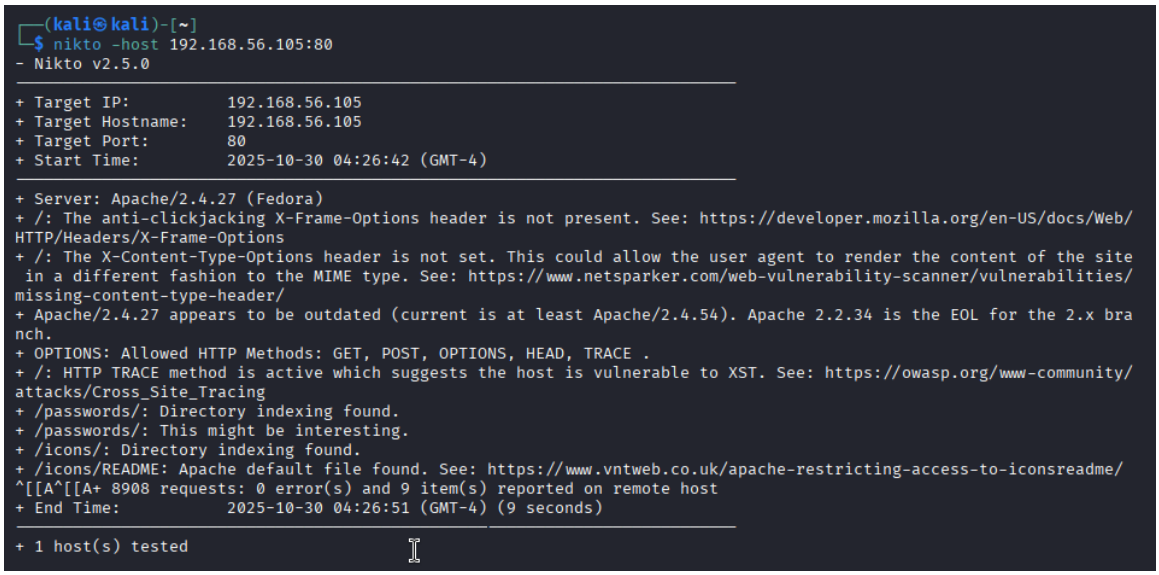
Gambar 13 Koneksi SSH ke RickSanchez@192.168.56.105 pada port 22222 meminta kata sandi (autentikasi SSH diminta).

Setelah upaya koneksi ke port standar (22) ditutup oleh host, percobaan koneksi selanjutnya diarahkan ke port alternatif 22222 yang sebelumnya terdeteksi sebagai layanan SSH. Koneksi ke ssh RickSanchez@192.168.56.105 -p 22222 menghasilkan prompt kata sandi, menandakan bahwa daemon OpenSSH pada port tersebut mendengarkan dan memproses negosiasi autentikasi. Kejadian ini mengkonfirmasi bahwa port 22222 merupakan jalur SSH aktif sementara port 22 tidak memberikan akses interaktif. Langkah berikutnya harus berupa percobaan autentikasi yang sah (menggunakan kredensial yang diizinkan atau kredensial yang ditemukan secara legal selama pengujian) dan/atau pemeriksaan log autentikasi pada host target jika akses diperoleh untuk menentukan kebijakan pembatasan koneksi.



Gambar 14 Hasil pemindaian Nikto terhadap 192.168.56.105:9090, temuan header keamanan yang hilang dan root page redirect ke HTTPS.

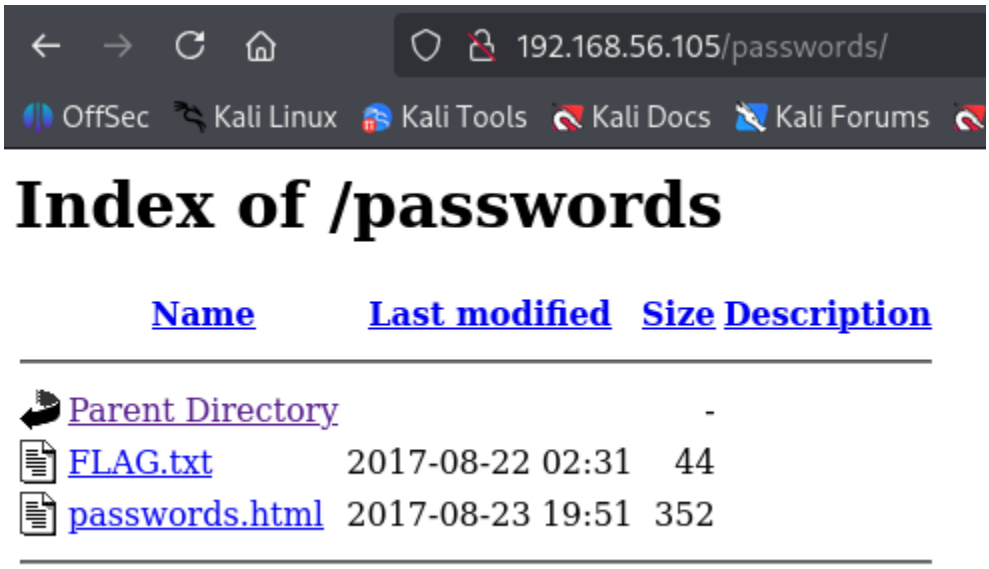
Nikto dijalankan terhadap layanan pada port 9090 untuk mendeteksi konfigurasi web berisiko dan endpoint default. Pemindaian menunjukkan beberapa temuan penting: server tidak mengembalikan banner (tidak ada informasi versi jelas), halaman root melakukan *redirect* ke HTTPS, serta header keamanan penting *X-Frame-Options* dan *X-Content-Type-Options* tidak diset. Kekurangan header ini berpotensi membuka celah seperti *clickjacking* atau pemrosesan konten yang tidak aman oleh user agent. Nikto tidak melaporkan direktori CGI pada pemindaian default (menyarankan penggunaan opsi yang lebih agresif -C all untuk memaksa pemeriksaan semua kemungkinan), namun temuan manual sebelumnya terhadap /robots.txt dan /cgi-bin/tracertool.cgi sudah membuktikan adanya skrip CGI yang perlu diuji lebih lanjut.



Gambar 15 Hasil pemindaian Nikto terhadap 192.168.56.105:80, ditemukan kelemahan konfigurasi server Apache dan direktori yang berpotensi sensitif.

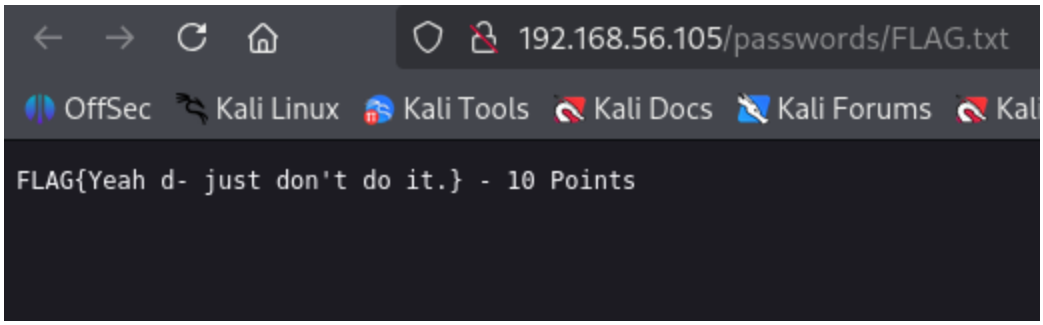
Pemindaian menggunakan Nikto pada port **80** menunjukkan bahwa server web menjalankan *Apache/2.4.27 (Fedora)* dengan beberapa kelemahan konfigurasi keamanan. Header keamanan *X-Frame-Options* dan *X-Content-Type-Options* tidak diset, yang dapat membuka peluang serangan *clickjacking* dan penyalahgunaan MIME type. Selain itu, metode HTTP *TRACE* diaktifkan, sehingga host berpotensi rentan terhadap serangan *Cross-Site Tracing (XST)*.

Nikto juga mengidentifikasi keberadaan beberapa direktori dan file yang menarik untuk ditinjau lebih lanjut, seperti /passwords/ dan /icons/README, yang bisa mengandung informasi sensitif atau konfigurasi default Apache. Versi Apache yang digunakan juga terdeteksi sudah usang dan direkomendasikan untuk diperbarui ke versi yang lebih baru karena versi lama sering kali memiliki kerentanan yang telah diketahui. Temuan ini menunjukkan bahwa konfigurasi web server perlu diperkuat, termasuk dengan menonaktifkan metode HTTP yang tidak diperlukan, memperbarui versi Apache, dan menambahkan header keamanan.



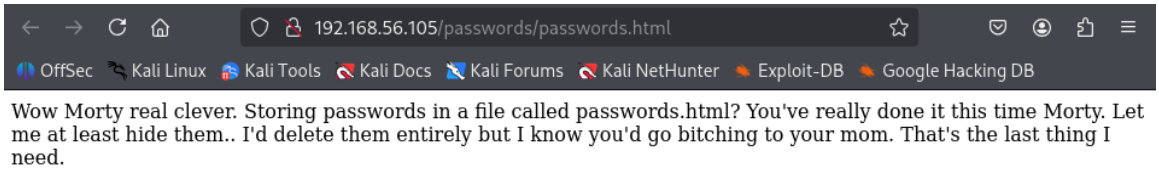
Gambar 16 Indeks direktori /passwords menampilkan file FLAG.txt dan passwords.html, menandakan directory listing aktif dan eksposur berkas sensitif.

Akses ke http://192.168.56.105/passwords/ mengembalikan halaman indeks direktori yang memperlihatkan dua berkas: FLAG.txt dan passwords.html. Keberadaan indeks direktori (directory listing) berarti server mengizinkan daftar isi folder ditampilkan publik, sehingga setiap berkas dalam direktori tersebut dapat diakses tanpa autentikasi. Hal ini berisiko karena file passwords.html berpotensi berisi informasi sensitif (mis. daftar akun atau kredensial) dan FLAG.txt merupakan bukti eksfiltrasi/akses; keduanya seharusnya tidak tersedia untuk publik.



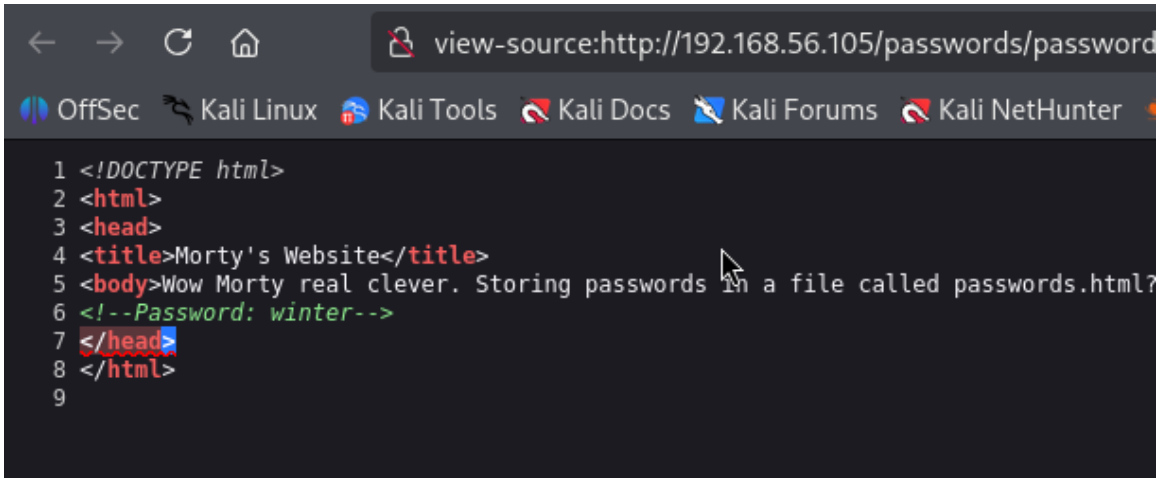
Gambar 17 Isi file FLAG.txt pada direktori /passwords berhasil ditampilkan melalui browser.

Setelah mengakses direktori /passwords, file FLAG.txt dibuka langsung melalui URL http://192.168.56.105/passwords/FLAG.txt. File tersebut berisi pesan *FLAG{Yeah d- just don't do it.} - 10 Points*, yang menjadi bukti bahwa file dapat diakses tanpa autentikasi. Hasil ini menunjukkan bahwa server web tidak memiliki mekanisme pembatasan akses atau izin file yang memadai untuk melindungi konten sensitif dalam direktori publik. Kondisi ini menegaskan kelemahan konfigurasi *directory listing* dan *file permission* pada web server, di mana file internal seharusnya tidak dapat diakses secara langsung oleh pengguna eksternal. Temuan ini dikategorikan sebagai *information disclosure vulnerability* dan direkomendasikan agar akses ke direktori sensitif dinonaktifkan serta file seperti FLAG.txt dipindahkan keluar dari direktori publik (/var/www/html).



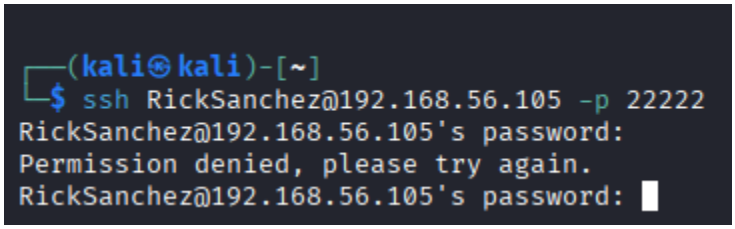
Gambar 18 Isi file passwords.html pada direktori /passwords menampilkan pesan teks tanpa data kredensial yang terlihat.

Ketika file passwords.html diakses melalui URL http://192.168.56.105/passwords/passwords.html, halaman menampilkan pesan berbentuk teks yang bersifat sindiran terhadap praktik penyimpanan kata sandi secara tidak aman. Tidak terdapat informasi kredensial yang terlihat langsung di halaman. Namun, isi pesan ini mengindikasikan bahwa data sensitif kemungkinan "disembunyikan" di dalam file tersebut (misalnya melalui *HTML comments*, *encoding*, atau *inline script*), sehingga perlu dilakukan pemeriksaan kode sumber halaman untuk memastikan tidak ada informasi tersembunyi.



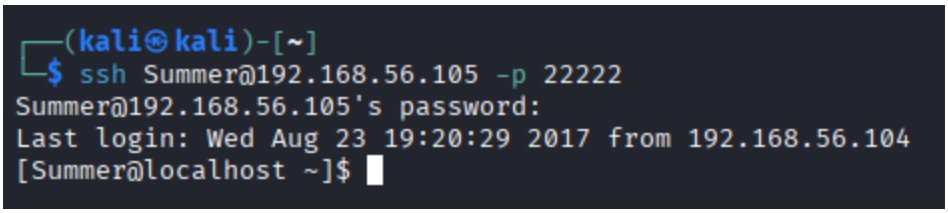
Gambar 19 Kode sumber passwords.html memperlihatkan komentar tersembunyi berisi teks "Password: winter".

Pemeriksaan kode sumber halaman passwords.html melalui fitur *View Source* mengungkap adanya komentar HTML (<!--Password: winter-->) yang berisi informasi kredensial tersembunyi. Meskipun komentar ini tidak terlihat secara langsung di tampilan halaman, kontennya tetap dapat diakses dengan mudah oleh siapa pun yang membuka *source code* menggunakan browser atau alat analisis web. Temuan ini mengonfirmasi bahwa pengembang menyembunyikan data sensitif di dalam elemen komentar HTML, yang merupakan praktik tidak aman karena komentar tetap bersifat publik dan dapat diambil oleh mesin pencari maupun penyerang. Informasi seperti ini dapat dimanfaatkan untuk percobaan login, terutama karena sebelumnya telah teridentifikasi pada sistem.



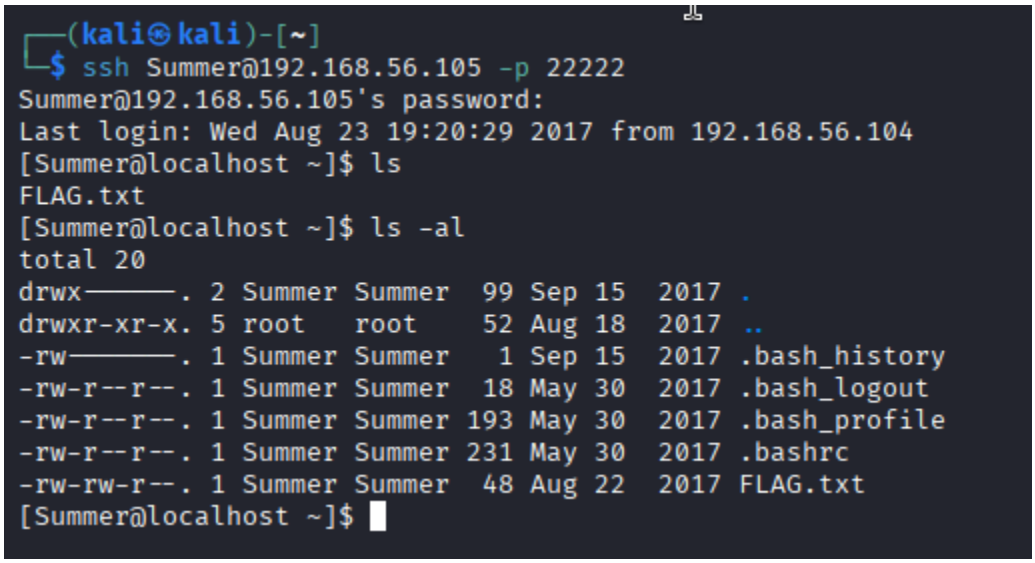
Gambar 20 Percobaan login SSH ke akun RickSanchez pada port 22222 menggunakan kata sandi yang ditemukan gagal dengan pesan *Permission denied*.

Setelah menemukan kredensial tersembunyi pada file passwords.html, dilakukan percobaan autentikasi SSH ke akun RickSanchez melalui port alternatif 22222 menggunakan perintah ssh RickSanchez@192.168.56.105 -p 22222. Namun, proses login gagal dan server menampilkan pesan *Permission denied, please try again*. Hal ini menunjukkan bahwa kata sandi yang digunakan (dalam hal ini winter) tidak valid.



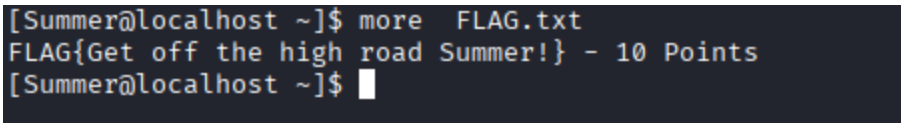
Gambar 21 Koneksi SSH sukses sebagai user Summer ke 192.168.56.105 pada port 22222, ditandai prompt [Summer@localhost ~]$.

Berdasarkan tema kata kunci yang ditemukan sebelumnya (kata sandi bertema *musim*), percobaan autentikasi terhadap akun Summer berhasil pada port alternatif 22222, menghasilkan shell interaktif dengan prompt [Summer@localhost ~]$. Pesan *Last login* menunjukkan jejak login terakhir dan sumbernya (192.168.56.104), yang berguna untuk korelasi aktivitas. Saat ini akses berada pada konteks user biasa (bukan root), sehingga langkah enumerasi awal yang direkomendasikan meliputi pemeriksaan identitas dan hak, isi direktori home (ls -la /home/Summer), file konfigurasi SSH (~/.ssh/authorized_keys), kemampuan sudo (sudo -l), serta pencarian temuan atau kredensial tersimpan (mis. file konfigurasi, skrip, atau history).



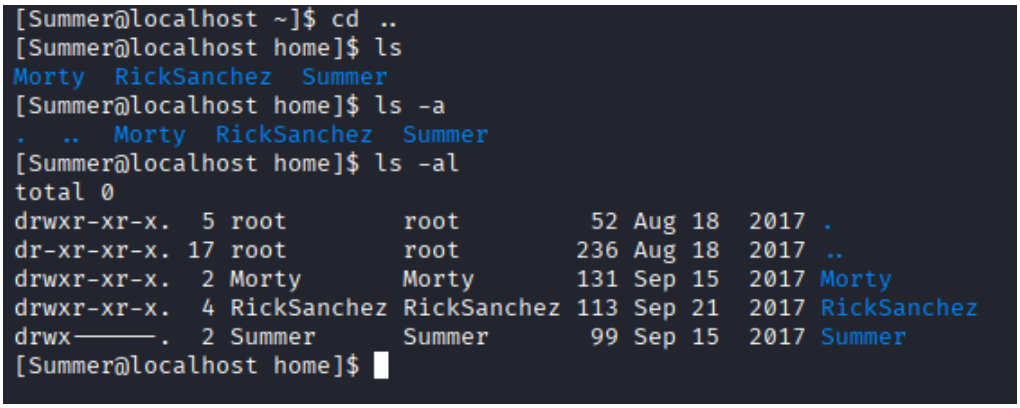
Gambar 22 SSH berhasil sebagai user Summer pada port 22222; berkas FLAG.txt terlihat di home directory /home/Summer.

Koneksi SSH ke Summer@192.168.56.105 pada port 22222 berhasil dan perintah verifikasi dijalankan: ls menampilkan FLAG.txt, sedangkan ls -al memperlihatkan isi direktori home beserta metadata file. Daftar file menunjukkan keberadaan berkas konfigurasi shell (.bash_history, .bash_logout, .bash_profile, .bashrc) dan FLAG.txt yang dimiliki oleh user Summer (permission tipikal rw-r--r--). Entri parent (..) dimiliki oleh root, yang menunjukkan direktori parent berada di bawah kepemilikan root. Temuan ini mengonfirmasi akses interaktif dengan konteks user biasa; langkah berikutnya yang bisa dilakukan adalah membuka dan mendokumentasikan isi FLAG.txt untuk verifikasi bukti akses.



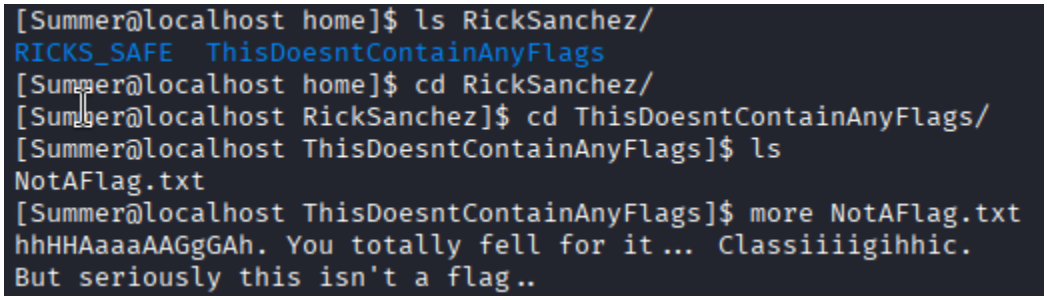
Gambar 23 Isi FLAG.txt pada home user Summer ditampilkan menggunakan more: FLAG{Get off the high road Summer!} - 10 Points.

File FLAG.txt dibuka dengan perintah more dan menampilkan FLAG{Get off the high road Summer!} - 10 Points, yang mengkonfirmasi keberhasilan akses ke akun Summer dan perolehan bukti (flag) yang relevan. Output ini didokumentasikan sebagai bukti akses interaktif; langkah dokumentasi meliputi penyimpanan transcript sesi dan screenshot. Setelah verifikasi flag, dianjurkan dilanjutkan dengan enumerasi non-destruktif untuk mengevaluasi kemungkinan jalur eskalasi atau pivoting.



Gambar 24 Daftar isi direktori /home menampilkan akun pengguna lain: Morty, RickSanchez, dan Summer.

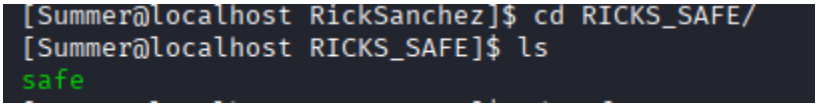
Setelah verifikasi isi home user Summer, direktori kerja dipindahkan ke /home dan dilakukan beberapa listing (ls, ls -a, ls -al) untuk memetakan akun pengguna pada host. Output menunjukkan direktori pengguna Morty, RickSanchez, dan Summer, beserta metadata kepemilikan dan permission dasar; entri parent (..) dimiliki oleh root. Informasi ini menandai potensi jalur enumerasi lanjutan karena direktori home pengguna lain mungkin menyimpan file konfigurasi, history, atau kredensial tersimpan yang dapat membantu eskalasi hak atau pivoting.



Gambar 25 Isi direktori RickSanchez menampilkan subfolder decoy dan file NotAFlag.txt yang berisi pesan bahwa itu bukan flag.

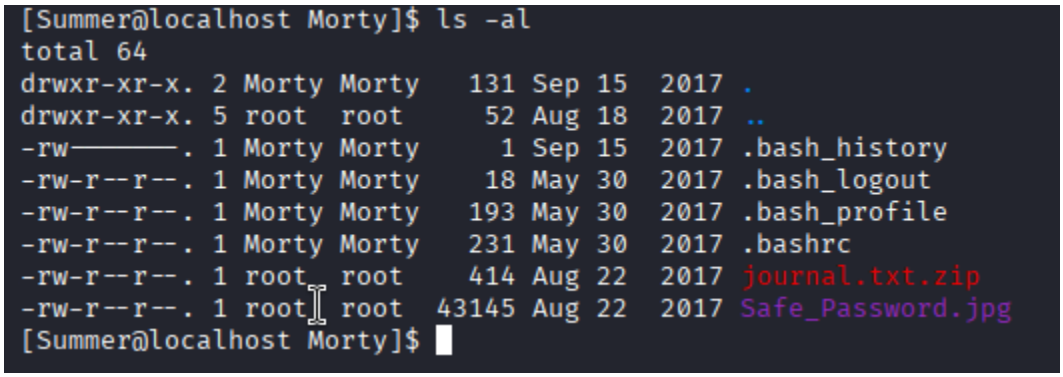
Direktori pengguna RickSanchez berisi dua entri penting: RICKS_SAFE dan ThisDoesntContainAnyFlags. Setelah memasuki ThisDoesntContainAnyFlags, ditemukan file NotAFlag.txt; isi file tersebut menampilkan teks gurauan yang secara eksplisit menyatakan bahwa file tersebut bukan flag (pesan seperti "You totally fell for it … But seriously this isn't a flag."). Temuan ini mengindikasikan upaya untuk menanamkan jebakan atau decoy dalam struktur home user yang dapat mengalihkan perhatian dari temuan nyata.

File NotAFlag.txt dicatat sebagai bukti non-flag. Rekomendasi lanjutan yang layak dilakukan secara non-destruktif: memeriksa direktori RICKS_SAFE untuk temuan lain, menelaah file konfigurasi dan .bash_history pada akun RickSanchez untuk petunjuk kredensial atau perintah penting, serta mengkorelasikan temuan ini dengan hasil enumerasi lainnya sebelum mengambil langkah eskalasi atau pivoting.



Gambar 26 Isi direktori RICKS_SAFE menampilkan satu file bernama safe, yang kemungkinan berfungsi sebagai file atau program penting milik user RickSanchez.

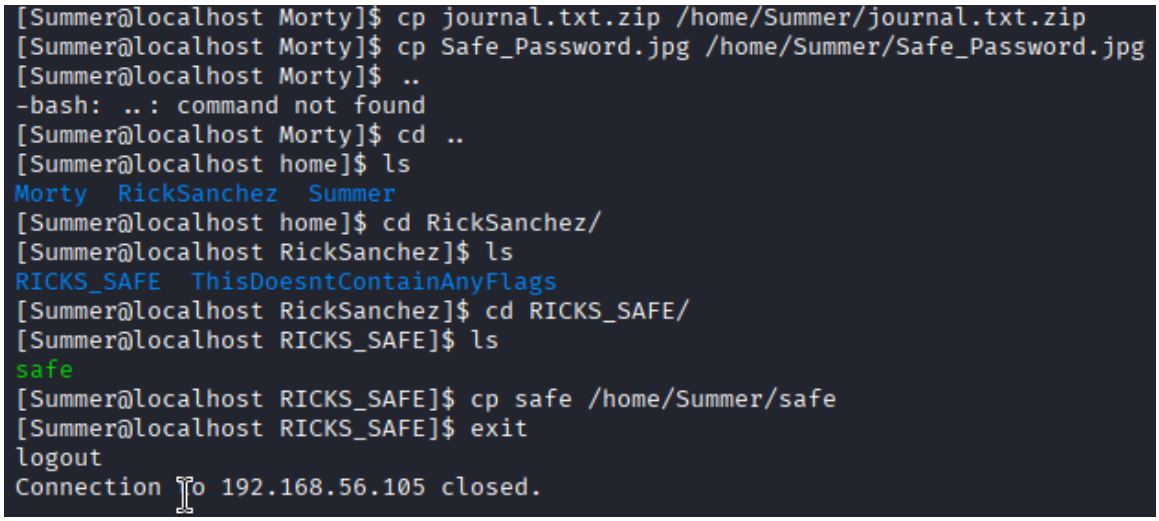
Setelah memasuki direktori /home/RickSanchez, dilakukan penelusuran ke dalam folder **RICKS_SAFE** dan ditemukan satu file bernama **safe**. Berdasarkan konteks sebelumnya, di mana direktori lain hanya berisi file umpan, file ini menjadi kandidat kuat yang patut diperiksa karena kemungkinan berisi informasi penting. Jika file teridentifikasi sebagai executable, dapat dilakukan analisis awal dengan strings safe atau menjalankannya di lingkungan terisolasi untuk melihat fungsinya. Untuk saat ini biarkan file tersebut seperti ini terlebih dahulu; pemeriksaan akan dilanjutkan nanti, dan fokus berikutnya dialihkan pada proses pivoting untuk eksplorasi lebih lanjut.



Gambar 27 Tampilan isi direktori /home/Morty setelah menjalankan perintah ls -al.

Hasil penelusuran pada direktori **/home/Morty** memperlihatkan beberapa file konfigurasi bawaan shell serta dua file mencurigakan bernama **journal.txt.zip** dan **Safe_Password.jpg**. Kedua file tersebut dimiliki oleh user *root*, bukan oleh user *Morty*, yang mengindikasikan adanya kemungkinan akses atau transfer data dari user dengan hak istimewa. File **journal.txt.zip**

berpotensi berisi arsip teks terenkripsi atau terkompresi, sedangkan **Safe_Password.jpg** mungkin menyimpan informasi penting yang disamarkan sebagai gambar.



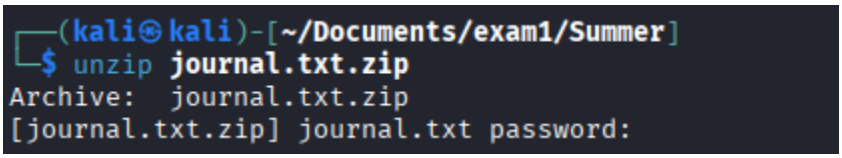
Gambar 28 Penyalinan file-file kandidat bukti ke direktori **/home/Summer** menggunakan perintah cp.

Pada tangkapan layar terlihat perintah **cp journal.txt.zip /home/Summer/journal.txt.zip** dan **cp Safe_Password.jpg /home/Summer/Safe_Password.jpg** yang dijalankan oleh user *Morty*, dilanjutkan dengan navigasi ke **/home/RickSanchez/RICKS_SAFE** dan penyalinan file **safe** ke **/home/Summer/safe**; sesi kemudian diakhiri dengan exit dan koneksi ke target ditutup. Langkah ini berfungsi untuk mengonsolidasikan salinan file yang akan dianalisis lebih lanjut di akun *Summer* guna menghindari perubahan pada lokasi asal dan memfasilitasi pemeriksaan terkontrol.

Gambar 29 Penyalinan salinan bukti dari host target ke mesin analis menggunakan scp -P 22222.

Pada tahap ini dilakukan proses pemindahan file hasil akuisisi, yaitu **journal.txt.zip**, **Safe_Password.jpg**, dan **safe**, dari host target menuju mesin analis melalui protokol **SCP (Secure Copy)** pada port 22222. Langkah ini bertujuan untuk memindahkan seluruh file yang telah

dikumpulkan ke lingkungan analisis yang terisolasi, sehingga pemeriksaan dapat dilakukan tanpa mengubah kondisi asli sistem target serta menjaga integritas data yang diperoleh.

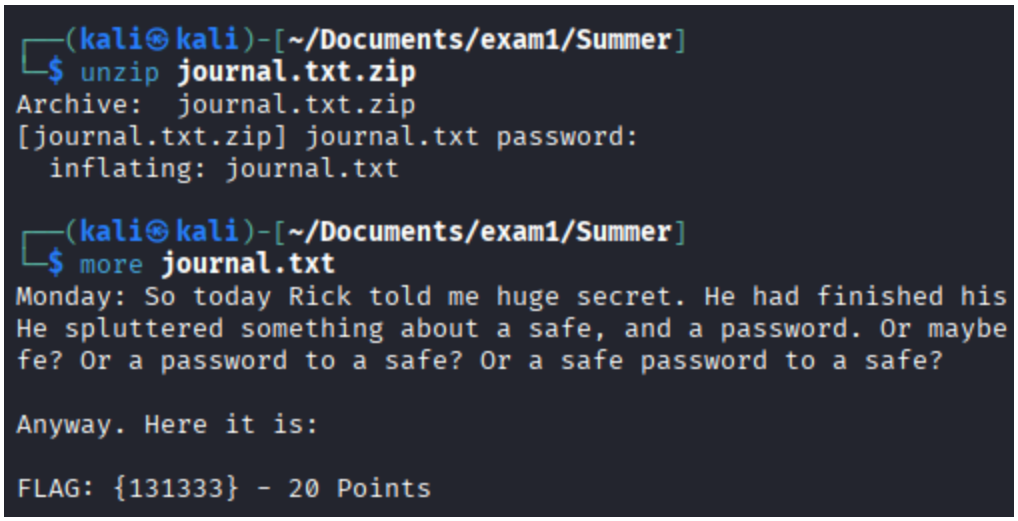


Gambar 30 Percobaan ekstraksi berkas terarchivasi **journal.txt.zip** menggunakan unzip journal.txt.zip.

Percobaan ekstraksi dihentikan sementara karena arsip dilindungi kata sandi; nilai hash salinan arsip dicatat untuk menjaga integritas bukti sebelum upaya pemulihan dilakukan. Ditemukan pula file gambar **Safe_Password.jpg** yang diunduh dari host target dan diperlakukan sebagai kandidat sumber kata sandi. Gambar tersebut berpotensi mengandung kata sandi secara eksplisit, tersimpan di metadata (EXIF), atau menyembunyikan data melalui teknik steganografi. Analisis lanjutan akan dilakukan pada salinan kerja menggunakan **binwalk** untuk mencari data tertanam atau temuan tersembunyi yang berpotensi mengungkap kata sandi.

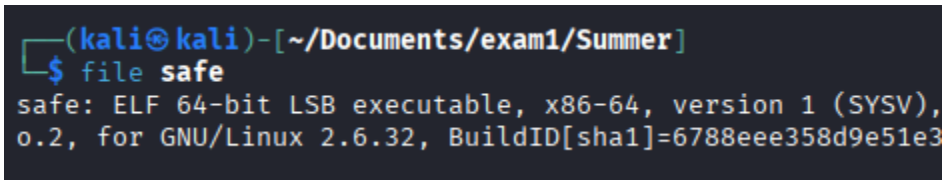
Gambar 31 Hasil analisis file **Safe_Password.jpg** menggunakan perintah binwalk.

Hasil analisis menunjukkan bahwa selain data gambar berformat JPEG dan TIFF, ditemukan informasi tambahan berupa *Unix path* /home/Morty/journal.txt.zip dengan keterangan **Password: Meeseek**. Temuan ini mengindikasikan bahwa file **Safe_Password.jpg** menyimpan data tersembunyi yang berisi kata sandi untuk membuka arsip **journal.txt.zip**, sehingga langkah selanjutnya adalah menggunakan kata sandi tersebut untuk mengekstraksi isi arsip dan melanjutkan proses analisis.



Gambar 32 Proses ekstraksi dan pembacaan isi arsip **journal.txt.zip** menggunakan kata sandi hasil temuan dari **Safe_Password.jpg**.

Setelah arsip **journal.txt.zip** berhasil diekstraksi menggunakan kata sandi **Meeseek**, diperoleh file **journal.txt** yang berisi catatan teks dengan petunjuk terkait "safe" serta kata sandi. Di akhir isi file ditemukan flag berupa **{131333}**, yang menandakan hasil temuan valid dari tahap analisis ini dan mengonfirmasi keterkaitan antara arsip terenkripsi dengan file gambar sebelumnya.



Gambar 33 Hasil identifikasi file **safe** menggunakan perintah file safe.

File **safe** terdeteksi sebagai **ELF 64-bit LSB executable** untuk sistem Linux arsitektur x86-64, yang menunjukkan bahwa file tersebut merupakan program biner yang dapat dijalankan secara langsung. Berdasarkan hasil ini, langkah selanjutnya adalah menjalankan file menggunakan perintah ./safe, karena prefiks ./ digunakan untuk mengeksekusi berkas biner yang berada di direktori kerja saat ini.

Gambar 34 Eksekusi file **safe** tanpa argumen menggunakan perintah ./safe.

Hasil eksekusi menampilkan pesan yang menunjukkan bahwa program tidak dapat dijalankan tanpa *command line arguments* dan membutuhkan input tambahan untuk berfungsi dengan benar. Berdasarkan keluaran tersebut, langkah selanjutnya adalah menjalankan ulang file **safe** dengan menambahkan argumen yang sesuai untuk menguji fungsionalitas dan menganalisis hasil output yang dihasilkan.

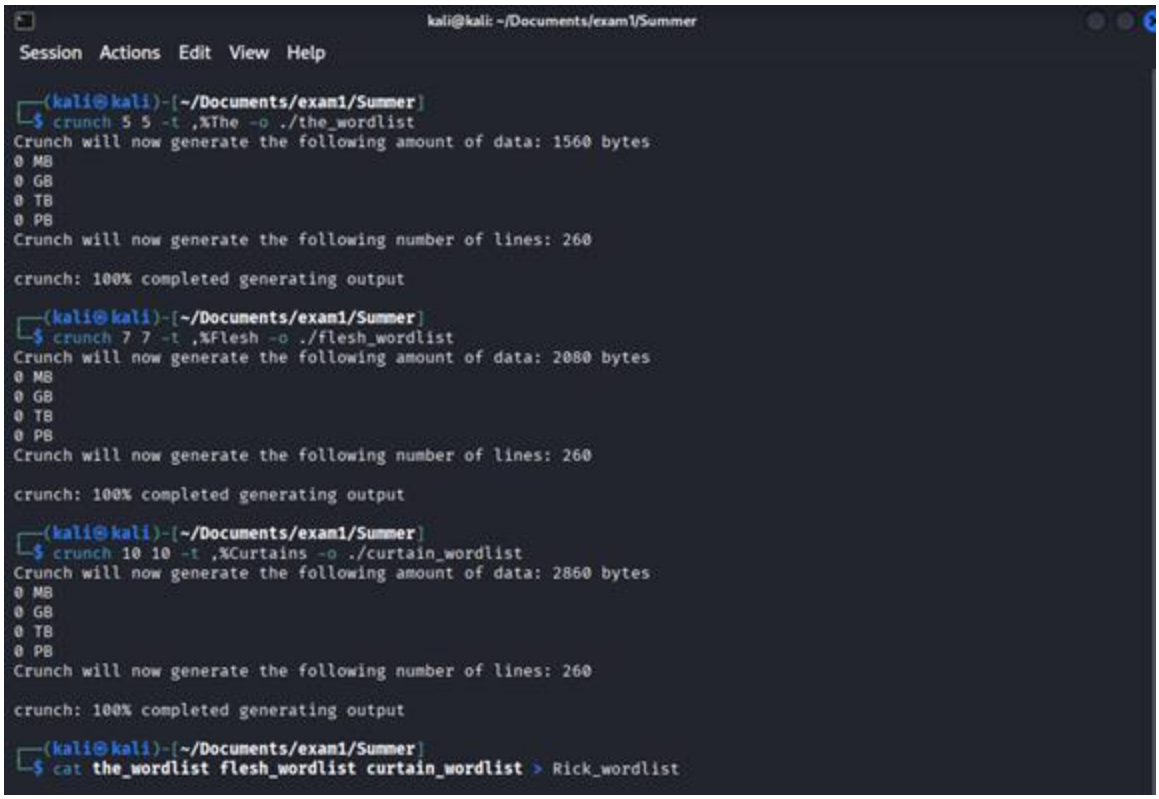
```
┌──(kali㉿kali)-[~/Documents/exam1/Summer]
└─$ ./safe 131333
decrypt:        FLAG{And Awwwaaaaayyyy we Go!} - 20 Points

Ricks password hints:
 (This is incase I forget.. I just hope I don't forget how to write a script to generate potential passwords. Also,
 sudo is wheely good.)
Follow these clues, in order


1 uppercase character
1 digit
One of the words in my old bands name.
```

Gambar 35 Eksekusi file **safe** dengan argumen 131333 berdasarkan temuan flag sebelumnya.

Berdasarkan hasil analisis sebelumnya, ditemukan angka **131333** yang diduga memiliki keterkaitan dengan file **safe**. Ketika angka tersebut digunakan sebagai argumen pada perintah ./safe 131333, program berhasil dijalankan dan menampilkan flag **{And Awwaaaaayyyy we Go!}** serta beberapa petunjuk tambahan terkait kata sandi milik Rick. Hasil ini mengonfirmasi bahwa nilai **131333** berfungsi sebagai parameter yang valid untuk menjalankan dekripsi pada program **safe** dan membuka akses terhadap informasi berikutnya.

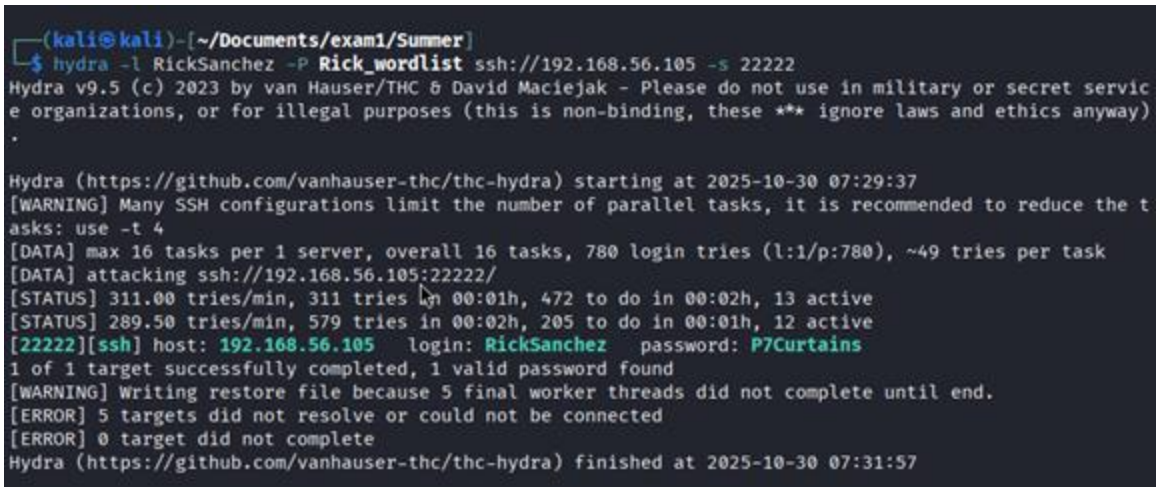
Gambar 36 Pencarian informasi mengenai nama band lama yang disebut dalam keluaran program **safe**.

Mengacu pada *Ricks password hints* (satu huruf kapital, satu digit, salah satu kata dari nama band), pola kandidat sandi disusun sebagai [A-Z][0-9](The|Flesh|Curtains) mis. A0Flesh. Kandidat diuji berurutan pada salinan kerja. Kandidat sandi digenerasi menjadi wordlist pada salinan kerja. Wordlist ini kemudian digunakan oleh hydra untuk mencoba kredensial terhadap layanan yang menerima autentikasi pada host target.



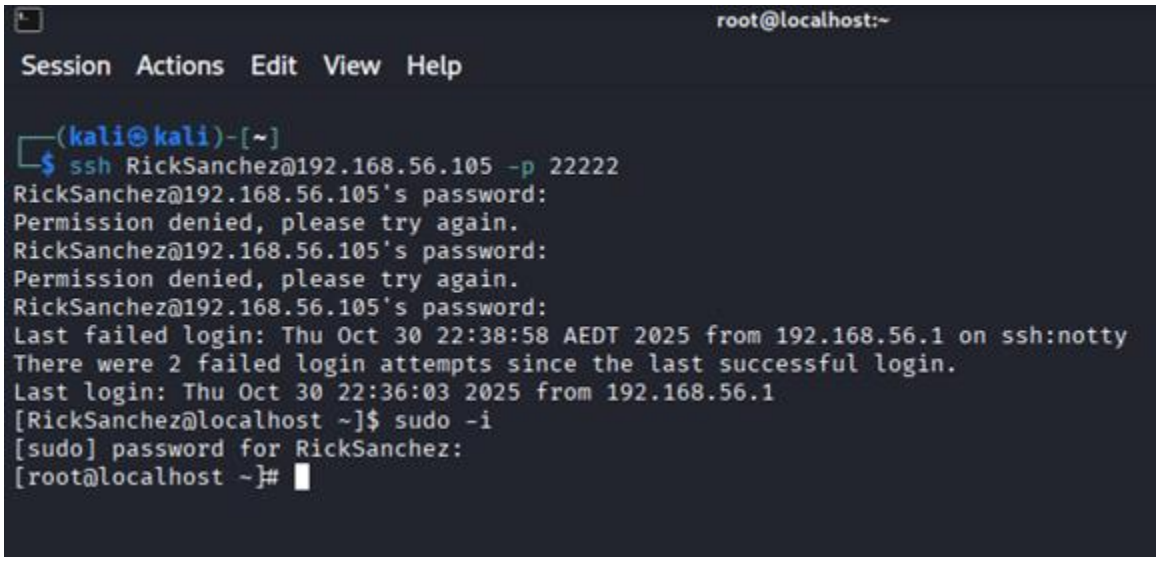
Gambar 37 Pembuatan wordlist berbasis pola kata sandi menggunakan crunch.

Pada salinan kerja dibuat tiga daftar terpisah untuk tiap kata pada nama band (The, Flesh, Curtains) dengan pola yang memenuhi petunjuk: satu huruf kapital di depan diikuti satu digit lalu kata band. Setiap daftar menghasilkan 260 entri sesuai kombinasi huruf A–Z dan angka 0–9; ketiga daftar tersebut kemudian digabungkan menjadi satu file Rick_wordlist untuk memudahkan pengujian otomatis. Pendekatan ini dipilih untuk menjaga keteraturan dan auditabilitas wordlist sehingga setiap entri dapat ditelusuri kembali ke pola asalnya. Langkah selanjutnya adalah menggunakan Rick_wordlist dengan alat uji kredensial hydra terhadap layanan yang menerima autentikasi.



Gambar 38 Percobaan brute-force otentikasi SSH terhadap host **192.168.56.105:22222** menggunakan hydra dengan user **RickSanchez** dan wordlist **Rick_wordlist**.

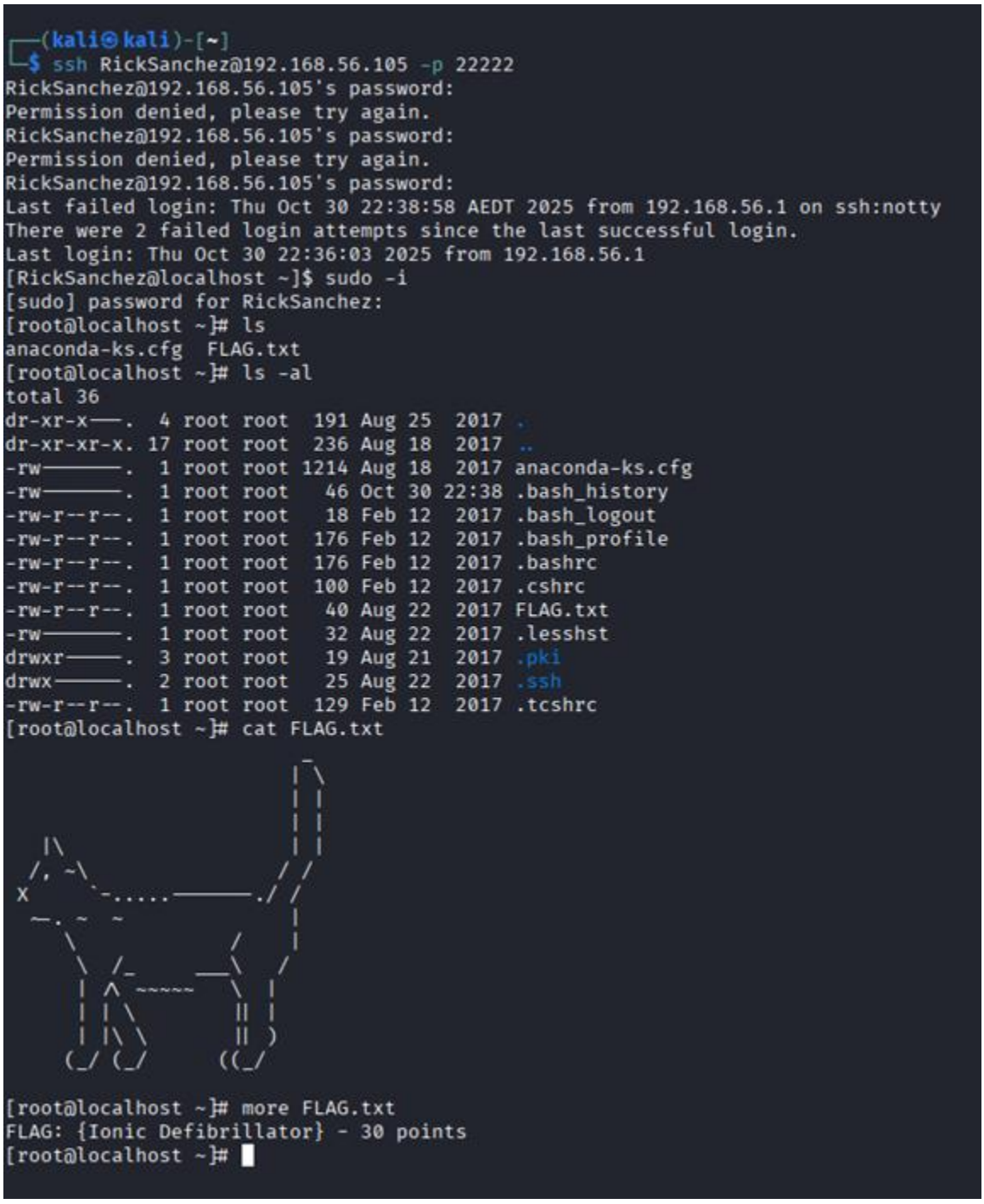
Langkah yang dilakukan adalah menjalankan perintah **hydra -l RickSanchez -P Rick_wordlist ssh://192.168.56.105 -s 22222** untuk mencoba setiap entri pada Rick_wordlist sebagai kata sandi terhadap akun RickSanchez pada layanan SSH di port 22222. Opsi -l menetapkan nama pengguna yang diuji, -P menunjuk file wordlist yang berisi kandidat kata sandi, dan -s menentukan port SSH non-standar. Hasil eksekusi menunjukkan keberhasilan menemukan kredensial valid **RickSanchez:P7Curtains**, proses dihentikan.



Gambar 39 Koneksi SSH ke 192.168.56.105 pada port 22222 dan eskalasi hak istimewa menjadi root menggunakan sudo -i.

Percobaan login dilakukan dengan perintah **ssh RickSanchez@192.168.56.105 -p 22222**; terlihat beberapa percobaan password gagal yang disebabkan oleh kesalahan pengetikan oleh penguji, kemudian autentikasi sukses dan sesi shell untuk user RickSanchez berhasil diperoleh.

Selanjutnya dijalankan **sudo -i** dan autentikasi sudo berhasil sehingga prompt berubah menjadi root, menandakan eskalasi hak istimewa.



Gambar 40 Akses SSH, eskalasi hak, dan pembacaan file flag di direktori root.

Pada langkah ini dilakukan koneksi SSH ke 192.168.56.105 (ssh RickSanchez@192.168.56.105 -p 22222), setelah beberapa percobaan password yang gagal autentikasi berhasil dan diperoleh shell RickSanchez. Perintah sudo -i dijalankan untuk beralih ke

shell login root, ls -al memperlihatkan keberadaan FLAG.txt di direktori root, dan cat FLAG.txt menampilkan seluruh isi file yang ternyata berisi ASCII art. Hal ini terjadi karena perintah cat menampilkan isi file secara langsung ke layar tanpa jeda atau pemformatan, sehingga seluruh karakter pembentuk ASCII art tercetak sekaligus dan memenuhi tampilan terminal. Untuk membaca konten dengan lebih teratur digunakan more FLAG.txt, yang menampilkan isi file secara bertahap dan memudahkan pembacaan baris flag FLAG: {Ionic Defibrillator} - 30 points.

4. kesimpulan

4.1 Jumlah Flag dan Total Poin

- Total flag ditemukan: **9**
  - FLAG{Whoa this is unexpected} - 10 pts
  - FLAG{TheyFoundMyBackDoorMorty} -10 pts
  - FLAG{Flip the pickle Morty!} - 10 pts
  - FLAG{There is no Zeus, in your face!} - 10 pts
  - FLAG{Yeah d- just don't do it.} - 10 pts
  - FLAG{Get off the high road Summer!} - 10 pts
  - FLAG{131333} - 20 pts
  - FLAG{And Awwwaaaaayyyy we Go!} - 20 pts
  - FLAG{Ionic Defibrillator} - 30 pts
- Total skor: **130 poin**

4.2 Analisis Efektivitas Tools

- Nmap: Efektif untuk pemetaan layanan dan identifikasi versi serta port.
- Nikto: Membantu menemukan kelemahan konfigurasi web (header keamanan hilang, directory listing, dsb.).
- Hydra: Efektif untuk uji kredensial berbasis wordlist terhadap layanan jaringan (SSH) bila pola password diketahui.
- Netcat: Berguna untuk interaksi layanan sederhana dan pembuktian konsep (flag pada port TCP).
- Binwalk / unzip : Berguna pada analisis berkas terselubung dan pemulihan kata sandi.

4.3 Tantangan dan Solusi Teknis

No	Tantangan Teknis	Penyebab	Solusi
1	Akses layanan web tidak menampilkan hasil eksploit	Mekanisme input filter aktif pada CGI	Melakukan pengujian manual dengan karakter payload yang dimodifikasi agar sesuai pola filter
2	Gagal melakukan ekstraksi file terenkripsi	Arsip dilindungi kata sandi	Melakukan analisis metadata

3	Output terminal tidak terbaca saat membaca file flag	File berisi ASCII art yang panjang	Menggunakan more atau less untuk menampilkan isi file secara terkontrol
4	Ketidaksesuaian IP atau jaringan antar VM	Pengaturan adaptor jaringan tidak sinkron antara host dan target	Menggunakan konfigurasi kombinasi NAT dan Host-Only Adapter untuk memastikan konektivitas terisolasi dan stabil

4.4 Rekomendasi Tindakan Keamanan

Penentuan tingkat prioritas (High, Medium, Low) pada tabel berikut didasarkan pada kombinasi antara dampak (impact) terhadap sistem dan kemungkinan eksploitasi (likelihood), mengacu pada pendekatan CVSS v3.1 yang disederhanakan.

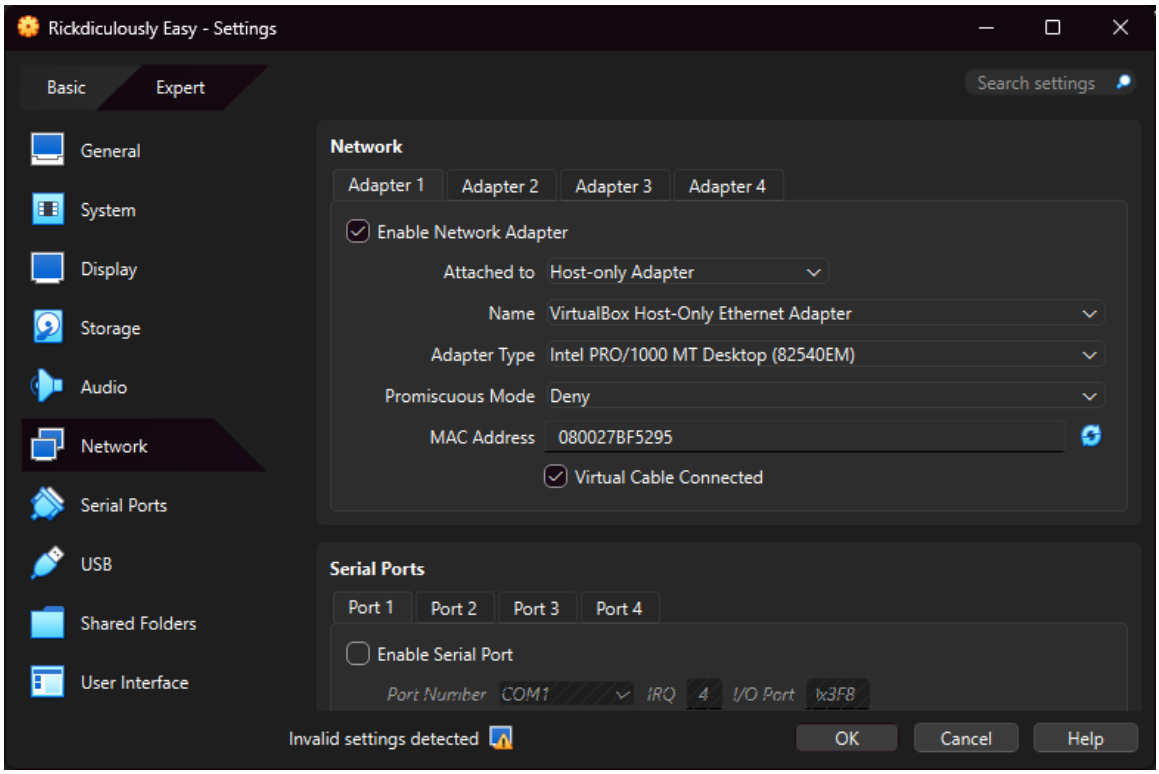
Tabel 3 Rekomendasi tindakan keamanan berdasarkan hasil pengujian VM *RickdiculouslyEasy*

Kerentanan	Rekomendasi	Prioritas	Alasan
FTP anonymous access	Nonaktifkan login anonim dan batasi akses hanya untuk user terotorisasi. Gunakan enkripsi (FTPS/SFTP) untuk mencegah penyadapan.	High	Akses anonim memungkinkan siapa pun membaca atau mengunggah file tanpa autentikasi, berpotensi membocorkan data sensitif atau menyebabkan modifikasi sistem tanpa izin.
Weak SSH credential	Terapkan kebijakan password kompleks (kombinasi huruf besar, kecil, angka, simbol) dan nonaktifkan password login, ganti dengan autentikasi berbasis SSH key. Aktifkan rate limiting dan fail2ban.	High	Kredensial lemah mudah ditebak atau dibobol dengan brute-force, dapat memberi akses penuh ke sistem seperti yang ditemukan dalam pengujian (akses SSH berhasil dengan hydra).
Directory listing HTTP	Nonaktifkan directory browsing pada konfigurasi server web (mis. Options -Indexes di Apache). Pastikan file sensitif tidak disimpan di direktori publik.	Medium	Directory listing dapat memperlihatkan struktur dan nama file sensitif yang bisa dimanfaatkan untuk serangan lanjutan, namun tidak secara langsung memberikan akses eksekusi.

Unsecured Cockpit service (port 9090)	Aktifkan HTTPS dengan sertifikat valid, batasi akses Cockpit hanya dari jaringan internal atau host administrator, dan ubah port default bila perlu.	Medium	Layanan administratif tanpa enkripsi dapat dieksploitasi untuk pencurian kredensial melalui sniffing, namun akses tetap memerlukan autentikasi awal.
File sensitif tanpa permission yang sesuai	Perbaiki permission file dan pastikan hanya user terkait yang memiliki akses terhadap file penting seperti flag atau arsip terenkripsi.	Medium	File dengan izin baca untuk semua user dapat menyebabkan kebocoran informasi internal, tetapi tetap memerlukan akses lokal ke sistem.
Biner tanpa verifikasi keamanan (ELF)	Pastikan hanya executable terpercaya yang disimpan di sistem, aktifkan AppArmor atau SELinux untuk membatasi hak eksekusi program tidak dikenal.	Medium	File executable tidak terverifikasi dapat disalahgunakan untuk eskalasi hak akses, namun eksploitasi membutuhkan kondisi tertentu seperti izin eksekusi.
Tidak ada audit log yang aktif	Aktifkan logging dan audit (mis. rsyslog, auditd) untuk melacak aktivitas user dan akses sistem secara rutin.	Low	Tidak berdampak langsung terhadap keamanan sistem, tetapi menyebabkan sulitnya pelacakan insiden bila terjadi pelanggaran.
Kurangnya pembatasan sudo	Batasi hak sudo hanya untuk perintah yang diperlukan, gunakan sudoers dengan granularitas perintah dan aktifkan logging perintah sudo.	Medium	Hak sudo terlalu luas meningkatkan risiko kesalahan konfigurasi dan eskalasi hak, meskipun masih memerlukan akses user sah.

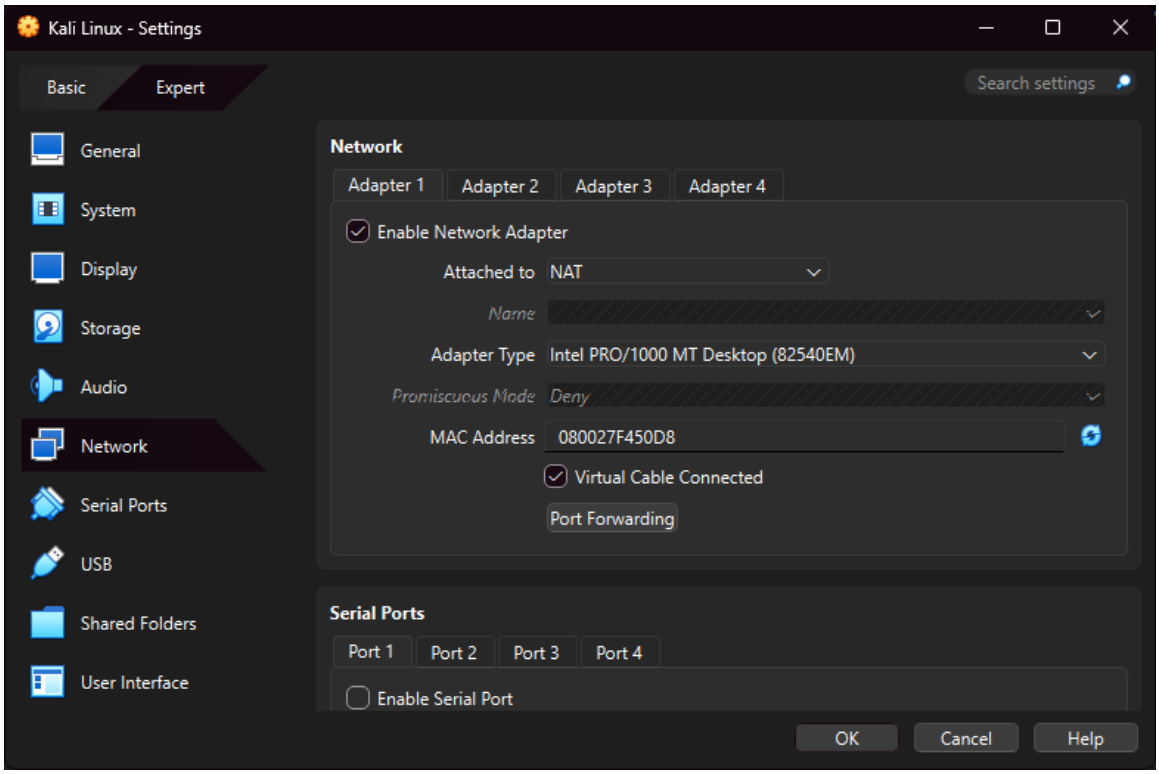
5. lampiran

5.1 Konfigurasi VM

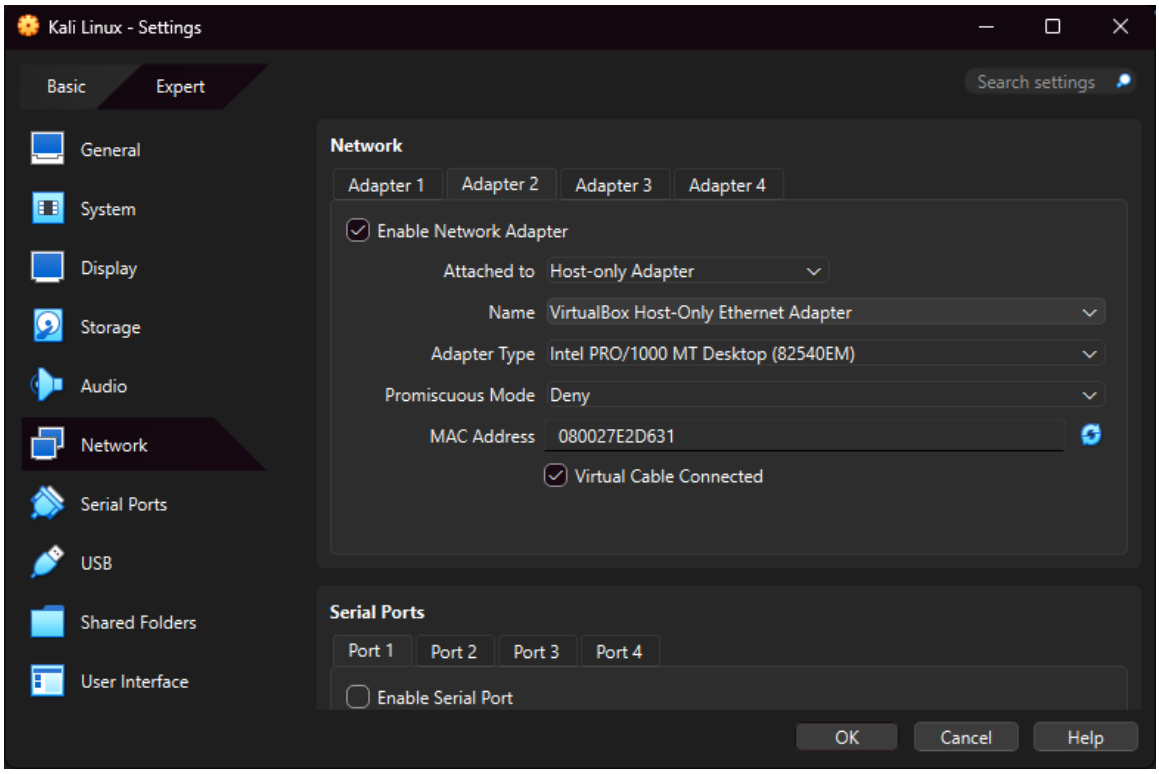


Gambar 41 Konfigurasi VM RickdiculouslyEasy

VM target dikonfigurasi menggunakan mode **Host-Only Adapter** agar sepenuhnya terisolasi dari jaringan eksternal dan hanya dapat diakses oleh mesin penguji melalui jaringan internal VirtualBox. Pengaturan ini memastikan proses pengujian berlangsung aman tanpa risiko koneksi keluar ke internet atau interaksi dengan perangkat jaringan lain di luar lingkungan uji.



Gambar 42 Konfigurasi Network Adapter 1 VM Kali Linux



Gambar 43 Konfigurasi Network Adapter 2 Kali Linux

Sementara itu, VM Kali Linux menggunakan dua adapter jaringan: NAT dan Host-Only Adapter. Mode NAT memungkinkan Kali Linux tetap terhubung ke internet untuk melakukan pembaruan sistem, mengunduh tool tambahan, atau mengakses repositori eksternal, sedangkan Host-Only digunakan untuk membangun komunikasi langsung dengan VM target dalam satu segmen jaringan tertutup. Kombinasi kedua mode ini memberikan fleksibilitas bagi penguji untuk tetap memiliki konektivitas eksternal tanpa mengekspos sistem target ke jaringan publik.