

V3

Defensive | Security



Linuxhackingid.org

Serangan yang Sering Terjadi

Ada 8 Serangan yang sering terjadi pada saat ini, yaitu;

1. Malware
2. Phishing
3. Spear Phishing
4. MITM
5. Password Attack
6. Drive by Attack
7. DDoS
8. Data Breach
9. OWASP TOP 10

Malware

Malware adalah aplikasi yang melakukan aktivitas berbahaya pada perangkat, jaringan, atau sistem. Aktivitas dapat mengakibatkan kerusakan data, memperoleh akses tidak sah atas informasi rahasia atau manipulasi data. Kata "malware" terbentuk dengan menjaga esensi dari dua hal yang berbeda kata-kata "**malicious**" dan "**software**". Malware dapat berupa virus, Trojan, spyware, atau ransomware yang menyebabkan kerusakan pada sistem.

Pencegahan: pake Anti-malware, Cek komunikasi jaringan dengan wireshark, download software yang terpercaya, hindari menggunakan software crack, jangan klik link yang mencurigakan, disable autorun.

Phishing

Phishing adalah ancaman berbasis pesan (terutama di E-Mail) yang menipu pengguna akhir agar mengekspos data pribadi atau memikat untuk mengunduh perangkat lunak berbahaya dengan mengikuti tautan yang dikirim bersama email. Email dirancang sedemikian rupa untuk membuat penerima percaya sumber dan percaya entitas pengirim. Setelah penerima dibujuk ke dalam konten, mereka dibujuk untuk memberikan informasi pribadi atau dipaksa untuk mendownload malware ke komputer target. Contoh umumnya siii adalah email yang berisi pemberitahuan pemenang hadiah dan bank permintaan detail untuk tindak lanjut dalam menerima kemenangan. Jadii hati-hati yahh

Phishing

Pencegahan: menggunakan software internet security, karena sudah ada scan phishing, melihat url target, jangan memasukkan informasi beharga seperti username dan password pada form url pihak ketiga, apabila ada yang mengirim di email, jangan juga percaya dengan scam atau ancaman yang ada dan lakukan konfirmasi ke pihak terkait.

Spear Phishing

Ini adalah bentuk phishing yang lebih klasik yang melibatkan penyusup untuk mendapatkan informasi tentang korban dan memposisikan dirinya sebagai entitas yang diketahui dan dipercaya oleh target. Target utama dalam kasus ini adalah individu yah gays, bukan sekelompok orang. Penyerang juga mengamati dan melakukan studi lengkap pada korban mereka melalui media sosial dan data lain yang tersedia untuk umum agar terlihat lebih otentik gitu. Ini seringkali merupakan langkah pertama untuk menghilangkan penghalang keamanan siber suatu organisasi karena kan ini bersifat pasif bukan aktif.

Spear Phishing

Pencegahan: melihat email yang dipakai seseorang, menggunakan 2FA, jika anda mendapatkan email untuk merubah setting akun bank atau dll, sebelumnya konfirmasi terlebih dahulu ke pihak bank apakah ini betul informasinya.

Man In The Middle Attack

MITM sendiri jadi me-intercept komunikasi antara 2 client, nah setelah itu peretas memodifikasi data berarti data yang dikirimkan tidak utuh. MITM dapat menghilangkan aspek *Confidential* dan *Integrity* gays. Serangan ini sering digunakan oleh militer untuk menciptakan kebingungan bagi kubu lawan. Ini membutuhkan tiga pemain — korban (pengirim), pengguna sasaran (penerima), dan "pria di tengah," yang mengganggu komunikasi tanpa kesadaran pihak yang berkomunikasi.

Man In The Middle Attack

Pencegahan: mengecek arp untuk mendeteksi arp spoof, melihat traffic dengan wireshark, force protokol HTTPS, VPN.

Password Attack

Karena kata sandi adalah mekanisme yang paling umum digunakan untuk mengotentikasi pengguna ke sistem informasi, mendapatkan kata sandi adalah pendekatan serangan yang umum dan efektif. Akses ke kata sandi seseorang dapat diperoleh dengan melihat sekeliling meja orang tersebut, “sniffing” koneksi ke jaringan untuk memperoleh kata sandi yang tidak terenkripsi, menggunakan Soceng/Social Engineering, mendapatkan akses ke basis data kata sandi atau Bruteforce.

Password Attack

Pencegahan: Menggunakan password yang unik, untuk membuat password yang unik bisa digenerate <https://passwordsgenerator.net>, setup akun dengan multi authentication, untuk pengguna android bias menggunakan aplikasi “google auhenticator” tersedia secara gratis di playstore.

Eavesdropping Attack

Serangan ini disebut mengintip atau “sniffing” yang dimana saat penyerang mencari komunikasi jaringan yang tidak aman yang dikirim melalui jaringan.

Pencegahan: menggunakan VPN saat dalam berada di jaringan public, seperti di coffee, bandara, dikampus, dl.

Distribute Deniel of Service

DDoS itu serangan yang digunakan untuk menghilangkan aspek ketersedian atau availability pada CIA Triad. Jika DDoS ini diluncurkan maka service atau system target menjadi down atau sibuk dengan lalu lintas; itu hanya crash atau melarang pengguna untuk merespon, mengakibatkan kegagalan akses untuk pengguna yang diautomasi. Serangan ini memakan waktu serta terbukti mahal karena sumber daya dan layanannya tidak dapat diakses.

Distribute Deniel of Service

Pencegahan: melihat traffic wireshark, menggunakan physical security , implementasi pelindung ddos seperti Nexusguard, block malicious IP

Data Breach

Ini mengacu pada pencurian informasi oleh penyusup berbahaya. Tujuan umum serangan semacam itu melibatkan pencurian identitas, kebutuhan untuk bertindak sebagai pelapor.

<https://news.linuxhackingid.org/Mazafaka-Hacked>

Pencegahan: Lakukan vulnerability scanning, melakukan pentest, jangan upload file penting pada web server yang diakses secara publik, menggunakan enkripsi pada file penting, update software-software, implementasi multi-factor authentication.

OWASP TOP 10 - 2020

OWASP 10 adalah daftar kerentanan yang paling umum. Daftar kerentanan itu terdiri dari:

1. Injection
2. Broken Authentication
3. Sensitive Data Exposure
4. XXE
5. Broken Access Control
6. Security Misconfigurations
7. XSS
8. Insecure Deserialization
9. Using Component with known vulnerabilities
10. Insufficient Logging and Monitoring

1. Injection

Injeksi terjadi ketika penyerang memasukan code untuk mengelabuhi aplikasi agar melakukan tindakan yang illegal. Jenis serangan injeksi yang paling umum adalah SQL Injection.

Contoh: `linuxhackingid.org/akun.php?=-1 UNION SELECT 1, username,email,password FROM Users WHERE id = 1`

Pencegahan: menggunakan firewall atau data filtering, validasi input dari user, sanitasi data dengan limit spesial karakter

2. Broken Authentication

Tipe kerentanan yang dimana mengizinkan seorang penyerang untuk me-capture atau mem-bypass authentication yang ada pada web application.

Contoh:

linuxhackingid.org/salescourses;jsessionid=78HB7AJDHY638KMN62HVCIA?dest=Indonesia

Pencegahan: Aplikasi Web harus mengakhiri sesi web yang tidak aktif. Aplikasi Web harus mengeluarkan ID baru saat user login berhasil.

3. Sensitive Data Exposures

Ini terjadi Web Application tidak cukup untuk melindungi informasi sensitive agar tidak ditemukan ke penyerang. Sensitive Data Exposure Ini dapat mencakup informasi tentang username dan password database yang digunakan, kartu kredit/cc, dll

Pencegahan: disable caching pada respon yang ada data sensitive, enkripsi semua data sensitive, dll

4. XXE (XML External Entity Attack)

Jenis serangan yang mem-parsing input XML pada web application.

Pencegahan: Disable DTD adalah cara yang bagus buat prevent vuln XXE.

5. Broken Access Control

Sistem yang mengakses informasi dan fungsionalitasnya. Broken Access Control memungkinkan attacker mem-bypass authorization dan melakukan hal yang biasa oleh admin.

Pencegahan: tolak acces ke fungsionalitas secara default, menggunakan access control list dan role-based authentication.

6. Security Misconfigurations

Security Misconfigurations adalah muncul ketika pengaturan keamanan tidak ditentukan, diterapkan, dan nilai default dipertahankan. Biasanya, ini pengaturan konfigurasi tidak sesuai dengan standar keamanan industri (benchmark CIS, OWASP Top 10 dll) yang sangat penting untuk menjaga keamanan dan mengurangi risiko bisnis.

Security Misconfigurations biasanya terjadi ketika sistem atau administrator database atau pengembang tidak mengkonfigurasi dengan benar kerangka keamanan aplikasi, situs web, desktop, atau server yang mengarah ke jalur terbuka yang berbahaya bagi peretas.

6. Security Misconfigurations

Pencegahan:

- Disable Debugging: Ini sangat penting saat menerapkan ke lingkungan produksi. Anda sebaiknya memberi perhatian khusus pada konfigurasi untuk fitur debugging, dan semuanya harus dinonaktifkan.
- Disable Directory Listing: Pastikan bahwa fitur ini tidak diaktifkan pada aplikasi apa pun yang Anda terapkan dan periksa apakah izin yang tepat telah ditetapkan untuk file dan folder.
- Patch dan Update software: Ini akan membantu melindungi aplikasi dan sistem Anda dari malware dan kerentanan baru yang mungkin belum Anda sadari.

7. XSS

XSS memungkinkan penyerang memasukkan script sisi klien ke dalam laman web yang dilihat oleh pengguna lain. Kerentanan XSS dapat digunakan oleh penyerang untuk bypass kontrol akses.

Contoh:

`https://linuxhackingid.org/?s=<script>alert("Linuxhackingid")</script>`

Pencegahan: Menggunakan encoding, set HTTP Only Flag yang dimana cookie tidak akan dapat diakses melalui JavaScript sisi klien. Lakukan scanning dengan web vuln scanner

8. Insecure Deserialization

Insecure Deserialization adalah saat data yang dapat dikontrol pengguna dinonaktifkan oleh situs web. Hal ini berpotensi memungkinkan penyerang memanipulasi objek berseri untuk meneruskan data berbahaya ke dalam kode aplikasi.

Pencegahan: mengimplementasikan integritas seperti digital signature, Mengisolasi dan menjalankan kode yang deserialisasi di lingkungan dengan hak istimewa rendah jika memungkinkan.

9. Using Component with know vulnerabilities

Adalah kerentanan yang ditemukan di komponen sumber terbuka dan dipublikasikan di NVD, orang keamanan. Dari saat publikasi, kerentanan dapat dieksloitasi oleh peretas yang menemukan dokumentasinya untuk mengexploit kerentanan tersebut.

Pencegahan: Hapus dependensi yang tidak digunakan, fitur, komponen, file, dan dokumentasi yang tidak perlu, update yang memiliki kerentanan, patch management

10. Insufflience Logging and Monitoring

Terjadi ketika peristiwa penting keamanan tidak dimatikan dengan benar, dan sistem tidak dipantau. Kurangnya fungsi tersebut dapat membuat aktivitas berbahaya lebih sulit untuk dideteksi dan pada gilirannya mempengaruhi proses penanganan insiden.

Pencegahan:

- Pastikan semua login, kegagalan kontrol akses, dan kegagalan validasi input sisi server dapat dicatat dengan konteks pengguna yang memadai untuk mengidentifikasi akun yang mencurigakan atau berbahaya, dan ditahan untuk waktu yang cukup untuk memungkinkan analisis forensik tertunda.

10. Insufflience Logging and Monitoring

- Tetapkan atau terapkan respons insiden dan rencana pemulihan yang digunakan oleh solusi manajemen log terpusat.
- Buat pemantauan dan peringatan yang efektif sehingga aktivitas mencurigakan terdeteksi dan ditanggapi secara tepat waktu.
- Tetapkan atau terapkan respons insiden dan rencana pemulihan.

Cyber Security Mekanisme

Keamanan dan privasi telah dianggap sebagai aspek-aspek penting dalam perlindungan sistem komputer, pencurian untuk kerusakan, atau kemungkinan bahaya untuk berkompromi dalam data elektronik, perangkat keras, atau salah perangkat lunak tanpa pengidentifikasi dan mekanisme keamanan, sistem komputasi dapat dianggap usang. Dasar dalam cyber security adalah uji pada aplikasi web, pengujian aplikasi yang dirancang, virus, worm, dan msh banyak lagi bro. Cybersecurity juga merupakan salah satu tools yang hebat untuk kekuatan ekonomi, diplomasi, dan bersenjata untuk waktu yang sangat lama. Ini umumnya merujuk pada kemampuan untuk mengontrol akses ke sistem jaringan dan informasi yang dikandungnya. Ketentuan Cybersecurity adalah badan teknologi, pemrosesan, dan praktik yang dimaksudkan untuk melindungi sumber daya jaringan, perangkat terkait, dan program yang satu ini memainkan peran penting dalam keamanan informasi untuk melindungi sistem dari virus, serangan malware, dll.

Approach Cyber Security

- Identify Threats = Identifikasi dan pahami ancaman dunia maya internal dan eksternal yang disebabkan karena kurangnya kesadaran. Usahakan sii dapetin informasi yang penting dan penyebab tentang ancaman tersebut dengan mengamatinya secara jelas.
- Identify Vulnerability = Dengan menggunakan tautan komunikasi berbeda yang mungkin langsung atau tidak langsung, berikan inventaris pada sistem dan coba kalian pahami konsekuensi yang ditimbulkan akibat ancaman siber. Perlu memahami kapabilitas dan batasan tindakan yang ada.
- Identify Risk = Untuk mengetahui kerentanan yang dieksloitasi oleh ancaman eksternal dan terekspos oleh tindakan yang tidak tepat. Diperlukan tindakan keselamatan dan keamanan.
- Respond and recover from cybersecurity incidents = Siapkan rencana respons untuk pulih dari insiden keamanan siber dan nilai kembali ancaman dan kerentanan. Tentukan efektivitas rencana respons.

Strategis Cyber Security

Tujuan orang Cyber Security adalah CIA Triad.

- Confidential: untuk menjaga kerahasiaan, contoh untuk menjaganya adalah dengan mengimplementasikan enkripsi.
- Integrity: memastikan data yang dikirim dari client ke server bersifat utuh, artinya tidak ada pengurangan atau penambahan data. Contoh untuk menjaga akses ini dengan cara backup atau menggunakan enkripsi
- Availability: ketersediaan informasi atau resource, yang dimana seorang keamanan harus menjamin bahwa resource atau informasi dapat diakses tanpa kendala. Untuk melindunginya, dapat mengimplementasikan physical protection

Challenge Cyber Security

- Network security
- Application security
- Endpoint security
- Data security
- Identity management
- Database and infrastructure security
- Cloud security
- Mobile security
- Disaster recovery/business continuity planning
- End-user education.

Security Challenges

- Network Issue: Jaringan komunikasi yang kamu pilih sangat penting karena keandalan jaringan. Jenis topologi dapat memengaruhi teknologi komunikasi titik-ke-titik untuk beberapa jenis jaringan pribadi.
- Cyber Attack: Karena semua informasi online dan ada kekurangan informasi yang signifikan tentang perlindungan data di bidang ini yang dapat berbahaya bagi praktisi, kasus yang paling dikenal terjadi karena peretas yang menghancurkan data untuk memprotes penggunaan organisme atau pestisida yang dimodifikasi secara genetik.
- Continous Monitoring: Orang harus terus menerus memonitor sistem karena semuanya otomatis. Jika terjadi kesalahan dalam sistem, itu akan menyebabkan lebih banyak kesalahan di sistem lain.

Misalnya, jika ada gangguan listrik, maka sistem akan gagal memperbarui perubahan pada saat itu.

Block Threat SSH Service in Linux

Pada contoh ini, saya menggunakan software linux bernama Fail2ban untuk mempraktikkan judul ini. Sebelum mengimplementasikannya, kita akan mencoba install terlebih dahulu fail2ban.

Attacker: Andrax

Defender: Linux Mint

1. Pada terminal linux mint, install terlebih dahulu Fail2ban.
 - sudo su
 - apt update
 - apt install fail2ban
2. Konfigurasi
 - cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
 - gedit /etc/fail2ban/jail.local

Block Threat SSH Service in Linux

Pada line ke 92, 101, 105, 108 dan 288 bisa kalian atur sesuai keinginan kamu.

Jika sudah, save lalu start service fail2ban dengan perintah berikut:

- sudo systemctl enable fail2ban
- sudo systemctl start fail2ban
- Sudo systemctl status fail2ban.service (apabila warna hijau, berarti sudah aktif)

Testing Fail2Ban

Sebagai contoh, saya menggunakan Andrax sebagai Offensive dan Linuxmint yang sudah kita konfigurasi Fail2ban sebagai Defensive.



Testing Fail2Ban

Dan pada gambar diatas terlihat, bahwa linuxmint telah me-block serangan dari Andrax (Silahkan lihat terminal linuxmint yang sudah saya mark atau tandain)

Setting Firewall dengan IPTables di Linux

Mengaktifkan Firewall merupakan hal yang penting dilakukan untuk menambah keamanan server. Iptables merupakan firewall yang disertakan di banyak sistem operasi Linux.

Firewall adalah sebuah sistem perangkat lunak atau perangkat keras untuk keamanan jaringan dengan cara menyaring lalu lintas yang masuk atau keluar pada jaringan komputer. Pada sistem operasi berbasis Linux secara default tersedia IPTables sebagai perangkat lunak firewall untuk menyaring paket dan NAT.

Dalam konfigurasi IPTables terdiri dari beberapa table, kemudian table berisi beberapa chain. Chain ada yang tersedia default dan bisa ditambah oleh sysadmin. Chain dapat berisi beberapa rule untuk paket.

Jadi struktur IPTables adalah **IPTables -> Tables -> Chains -> Rules**.

Sekilas Tentang IPTables

Iptables adalah program utilitas ruang pengguna yang memungkinkan administrator sistem untuk mengkonfigurasi aturan filter paket IP firewall Linux kernel, diimplementasikan sebagai modul Netfilter yang berbeda.

Hampir iptables sudah diinstal sebelumnya pada distribusi Linux mana pun. Untuk memperbarui / menginstalnya dalam distribusi debian, cukup ambil paket iptables:

- sudo apt-get update -y
- sudo apt-get install iptables -y

Ada alternatif GUI untuk iptables seperti Firestarter, tetapi iptables tidak terlalu sulit setelah Anda mematikan beberapa perintah. Kamu harus berhati-hati saat mengkonfigurasi aturan iptables, terutama jika Anda menggunakan SSH di server, karena satu perintah yang salah dapat mengunci Anda secara permanen hingga diperbaiki secara manual di mesin fisik. Soo hati-hati yah gays sebaiknya backup terlebih dahulu.

Tables dan Chains

IPTables memiliki 4 built-in tables

1. Filter Table

Filter adalah default table untuk IPTables. Jika sysadmin tidak mendefinisikan table sendiri, digunakanlah filter table. Filter table memiliki built-in chains:

- INPUT chain** : Untuk menyaring paket yang menuju ke server.
- OUTPUT chain** : Untuk menyaring paket yang keluar dari server.
- FORWARD chain** : Untuk menyaring paket yang menuju ke NIC lain dalam sever atau host lain.

Tables dan Chains

2. NAT Table

Chain pada NAT table:

- **PREROUTING chain** : Mengubah paket sebelum routing. Paket ditranslasi setelah paket masuk ke sistem sebelum routing. Ini untuk membantu menerjemahkan alamat IP tujuan (destination IP address) dari paket ke sesuatu yang cocok dengan perutean di server. Ini digunakan untuk DNAT (Destination NAT).
- **POSTROUTING chain**: Mengubah paket setelah routing. Paket ditranslasi ketika paket tersebut meninggalkan sistem. Ini untuk membantu menerjemahkan alamat IP sumber (source IP address) ke sesuatu yang cocok dengan perutean pada destinasi. Ini digunakan untuk SNAT (Source NAT).
- **OUTPUT chain** : NAT untuk paket yang dibuat secara lokal di server.

Tables dan Chains

3. Mangle Table

Mangle table adalah untuk pengubahan paket khusus. Ini mengubah bit QOS di header TCP. Chain pada Mangle table:

- PREROUTING chain
- OUTPUT chain
- FORWARD chain
- INPUT chain
- POSTROUTING chain

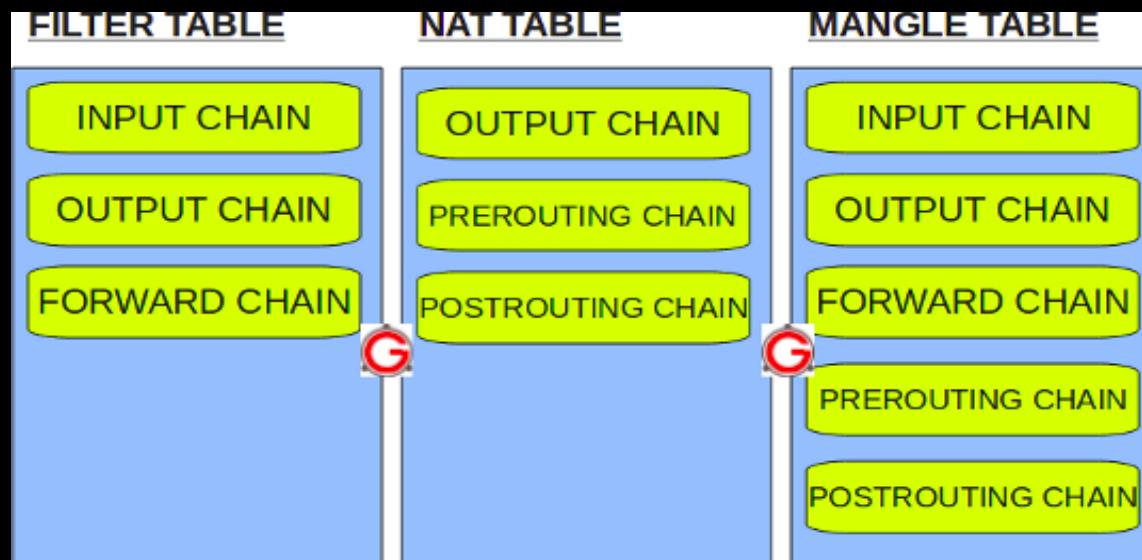
Tables dan Chains

4. Raw Table

Raw table adalah untuk pengecualian konfigurasi. Chain pada Raw table:

- PREROUTING chain
- OUTPUT chain

Diagram berikut menunjukkan tiga tabel penting di iptables.



Tables dan Chains

B. IPTABLES RULES

Berikut ini poin-poin penting yang harus diingat dalam IPTables rules.

- Rule mengandung kriteria dan target.
- Jika kriteria tersebut cocok, menuju ke rule yang ditentukan dalam target atau mengeksekusi nilai-nilai khusus yang disebutkan dalam target.
- Jika kriteria tidak cocok, pindah ke rule berikutnya.

Tables dan Chains

Target Values

Berikut ini value yang dapat dipasangkan pada target.

- **ACCEPT** : Firewall akan menerima paket tersebut.
- **DROP** : Firewall akan menghancurkan paket.
- **QUEUE** : Firewall akan meneruskan paket ke userspace.
- **RETURN** : Firewall akan berhenti mengeksekusi rule berikutnya dalam chain saat ini khusus untuk paket ini. Kontrol akan dikembalikan ke calling chain.

Config IPTables

~~Menampilkan rules dari Filter table~~

```
└# iptables -t filter --list                                CONFIG IPTABLES
Chain INPUT (policy ACCEPT)
target     prot opt source                               destination
target     prot opt source                               destination
target     prot opt source                               anywhere
target     prot opt source                               anywhere
ACCEPT    all  --  anywhere                            anywhere
          ctstate RELATED,ESTABLISHED
ACCEPT    all  --  anywhere                            anywhere
ACCEPT    all  --  anywhere                            anywhere
ACCEPT    all  --  anywhere                            anywhere

Chain FORWARD (policy DROP)
target     prot opt source                               destination
target     prot opt source                               destination
target     prot opt source                               anywhere
target     prot opt source                               anywhere
ACCEPT    all  --  anywhere                            anywhere
          ctstate RELATED,ESTABLISHED
ACCEPT    all  --  anywhere                            anywhere
ACCEPT    all  --  anywhere                            anywhere
ACCEPT    all  --  anywhere                            anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source                               destination

Chain DOCKER (1 references)
target     prot opt source                               destination

Chain DOCKER-ISOLATION-STAGE-1 (1 references)
target     prot opt source                               destination
```

Config IPTables

Menampilkan rules dari Mangle table

```
# iptables -t mangle --list
Chain PREROUTING (policy ACCEPT)
target    prot opt source          destination
Chain INPUT (policy ACCEPT)
target    prot opt source          destination
Chain FORWARD (policy ACCEPT)
target    prot opt source          destination
Chain OUTPUT (policy ACCEPT)
target    prot opt source          destination
Chain POSTROUTING (policy ACCEPT)
target    prot opt source          destination
```

2 ◉

Config IPTables

Menampilkan rules dari NAT table

```
# iptables -t nat --list
Chain PREROUTING (policy ACCEPT)
target    prot opt source          destination
DOCKER    all  --  anywhere        anywhere          ADDRTYPE match dst
-type LOCAL

Chain INPUT (policy ACCEPT)
target    prot opt source          destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source          destination
DOCKER    all  --  anywhere        !127.0.0.0/8      ADDRTYPE match dst
-type LOCAL

Chain POSTROUTING (policy ACCEPT)
target    prot opt source          destination
MASQUERADE all  --  172.17.0.0/16 anywhere

Chain DOCKER (2 references)
target    prot opt source          destination
RETURN   all  --  anywhere        anywhere
```

Config IPTables

Menampilkan rules dari Raw table

```
[# iptables -t raw --list
Chain PREROUTING (policy ACCEPT)
target    prot opt source          destination
Chain OUTPUT (policy ACCEPT)
target    prot opt source          destination
[omitted]
```

Mengubah Default Policy Filter Table

Melihat status policy iptables

```
[# sudo iptables -L | grep policy
Chain INPUT (policy ACCEPT)
Chain FORWARD (policy DROP)
Chain OUTPUT (policy ACCEPT)
```

Mengubah Default Policy Filter Table

Mengubah policy chain dan mengeceknya ulang

```
(root㉿kali)-[~/Documents]$ sudo iptables -L | grep policy
Chain INPUT (policy ACCEPT)
Chain FORWARD (policy DROP)
Chain OUTPUT (policy ACCEPT)

(root㉿kali)-[~/Documents]$ # sudo iptables --policy INPUT DROP
[root@kali ~]# 

(root㉿kali)-[~/Documents]$ # sudo iptables --policy FORWARD DROP
[root@kali ~]# 

(root㉿kali)-[~/Documents]$ # sudo iptables --policy OUTPUT ACCEPT
[root@kali ~]# 

(root㉿kali)-[~/Documents]$ # sudo iptables -L | grep policy
Chain INPUT (policy DROP)
Chain FORWARD (policy DROP)
Chain OUTPUT (policy ACCEPT)
```

Menambah Rules

Pada contoh di bawah ini, IPTables mengijinkan paket ke server untuk protokol ICMP, SSH, HTTP, HTTPS, dan FTP.

Contoh Menambah rule :

```
(root㉿kali)-[~/home/a233sec]
# sudo iptables --policy INPUT ACCEPT
2 ◎

(root㉿kali)-[~/home/a233sec]
# sudo iptables -A INPUT -p icmp -j ACCEPT
2 ◎

(root㉿kali)-[~/home/a233sec]
# sudo iptables -A INPUT -p tcp --dport 21 -j ACCEPT
2 ◎

(root㉿kali)-[~/home/a233sec]
# sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
2 ◎

(root㉿kali)-[~/home/a233sec]
# sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
2 ◎

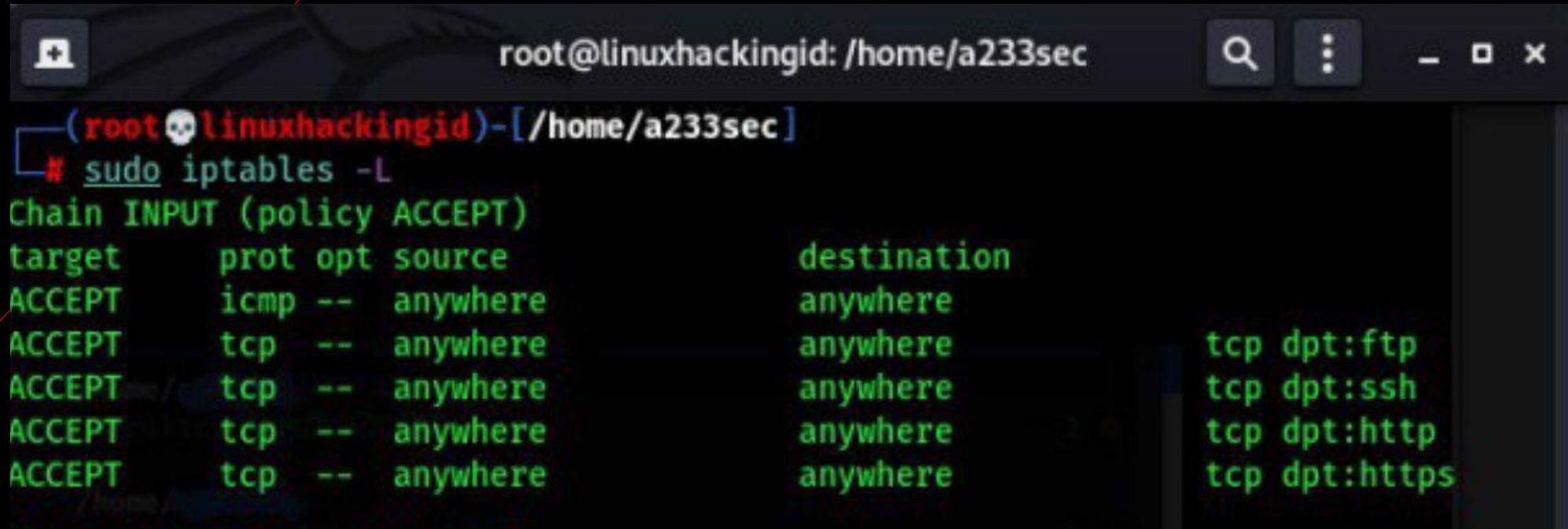
(root㉿kali)-[~/home/a233sec]
# sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
2 ◎
```

Menghapus satu rules, misalnya port 443 (TCP)

```
[root@linuxhackingid]# sudo iptables -D INPUT -p tcp --dport 443 -j ACCEPT
```

```
[root@linuxhackingid]# sudo iptables -F
```

Melihat Semua Rules



```
root@linuxhackingid: /home/a233sec
(root@linuxhackingid)-[ /home/a233sec]
# sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere
tcp dpt:ftp
tcp dpt:ssh
tcp dpt:http
tcp dpt:https
```

Simpan Konfigurasi IPTables

- sudo netfilter-persistent save
- sudo netfilter-persistent reload

Rule IPTables tersimpan di `/etc/iptables/rules.v4`

Edit file rules.v4



```
(root@linuxhackingid)-[~/home/a233sec]
# sudo nano /etc/iptables/rules.v4
```

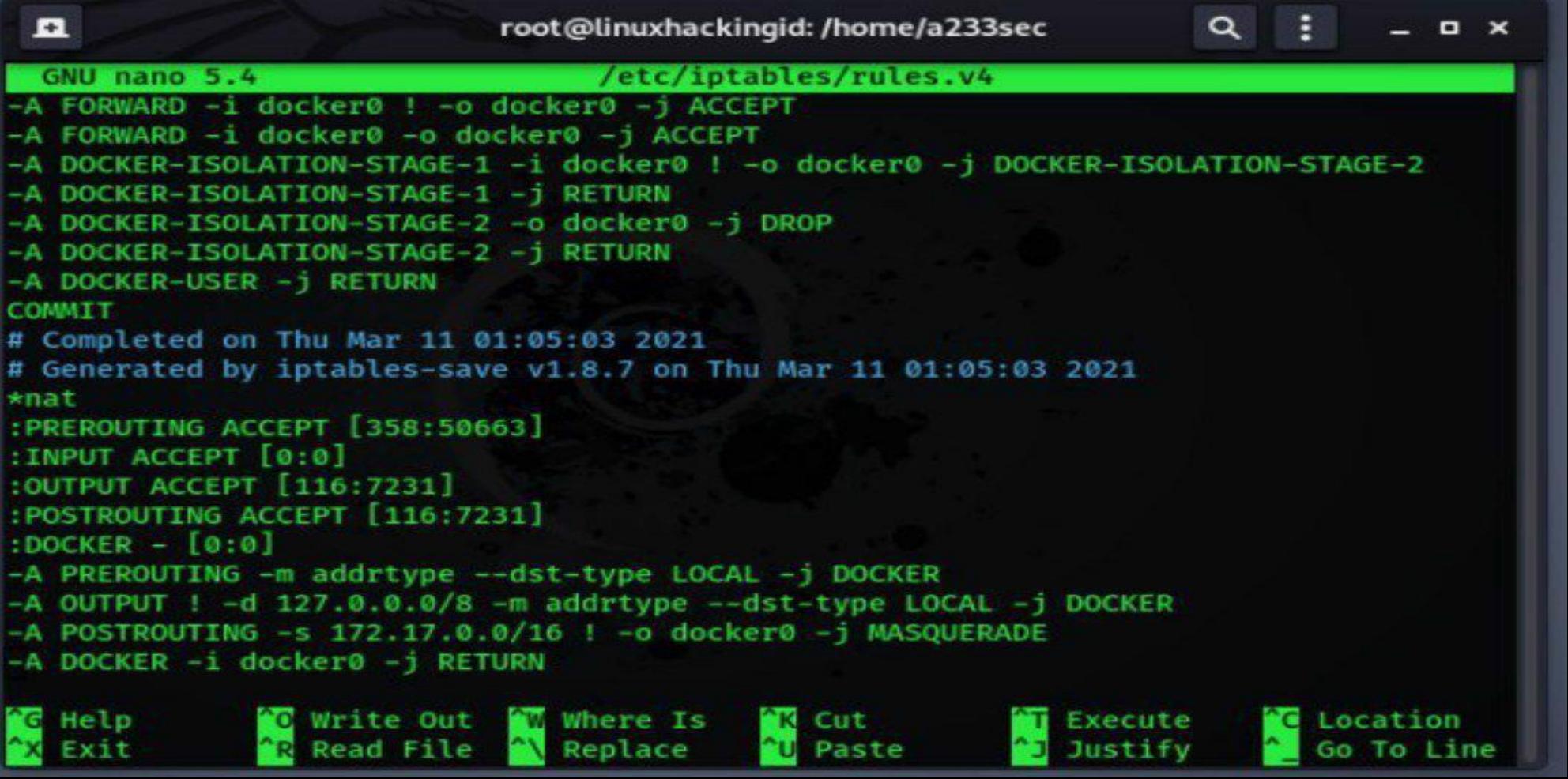
Edit rules.v4

Paste rules di bawah ini.

```
GNU nano 5.4                               /etc/iptables/rules.v4
# Generated by iptables-save v1.8.7 on Thu Mar 11 01:05:03 2021
*filter
:INPUT ACCEPT [2834:2277691]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [3019:388323]
:DOCKER - [0:0]
:DOCKER-ISOLATION-STAGE-1 - [0:0]
:DOCKER-ISOLATION-STAGE-2 - [0:0]
:DOCKER-USER - [0:0]
-A INPUT -p icmp -j ACCEPT
-A INPUT -p tcp -m tcp --dport 21 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A FORWARD -j DOCKER-USER
-A FORWARD -j DOCKER-ISOLATION-STAGE-1
-A FORWARD -o docker0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -o docker0 -j DOCKER
-A FORWARD -i docker0 ! -o docker0 -j ACCEPT
-A FORWARD -i docker0 -o docker0 -j ACCEPT
[l] Read 40 lines ]
```

[^G Help ^C Write Out ^W Where Is ^K Cut
^X Exit ^R Read File ^M Replace ^U Paste ^J Execute ^C Location
^C Go To Line

Edit



```
root@linuxhackingid: /home/a233sec
GNU nano 5.4          /etc/iptables/rules.v4

-A FORWARD -i docker0 ! -o docker0 -j ACCEPT
-A FORWARD -i docker0 -o docker0 -j ACCEPT
-A DOCKER-ISOLATION-STAGE-1 -i docker0 ! -o docker0 -j DOCKER-ISOLATION-STAGE-2
-A DOCKER-ISOLATION-STAGE-1 -j RETURN
-A DOCKER-ISOLATION-STAGE-2 -o docker0 -j DROP
-A DOCKER-ISOLATION-STAGE-2 -j RETURN
-A DOCKER-USER -j RETURN
COMMIT
# Completed on Thu Mar 11 01:05:03 2021
# Generated by iptables-save v1.8.7 on Thu Mar 11 01:05:03 2021
*nat
:PREROUTING ACCEPT [358:50663]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [116:7231]
:POSTROUTING ACCEPT [116:7231]
:DOCKER - [0:0]
-A PREROUTING -m addrtype --dst-type LOCAL -j DOCKER
-A OUTPUT ! -d 127.0.0.0/8 -m addrtype --dst-type LOCAL -j DOCKER
-A POSTROUTING -s 172.17.0.0/16 ! -o docker0 -j MASQUERADE
-A DOCKER -i docker0 -j RETURN

^G Help      ^O Write Out   ^W Where Is
^X Exit      ^R Read File   ^M Replace  ^K Cut
                                         ^U Paste   ^T Execute Justify
                                         ^L Location Go To Line
```

Edit

```
root@linuxhackingid: /home/a233sec
GNU nano 5.4          /etc/iptables/rules.v4

-A DOCKER-ISOLATION-STAGE-1 -j RETURN
-A DOCKER-ISOLATION-STAGE-2 -o docker0 -j DROP
-A DOCKER-ISOLATION-STAGE-2 -j RETURN
-A DOCKER-USER -j RETURN
COMMIT
# Completed on Thu Mar 11 01:05:03 2021
# Generated by iptables-save v1.8.7 on Thu Mar 11 01:05:03 2021
*nat
:PREROUTING ACCEPT [358:50663]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [116:7231]
:POSTROUTING ACCEPT [116:7231]
:DOCKER - [0:0]
-A PREROUTING -m addrtype --dst-type LOCAL -j DOCKER
-A OUTPUT ! -d 127.0.0.0/8 -m addrtype --dst-type LOCAL -j DOCKER
-A POSTROUTING -s 172.17.0.0/16 ! -o docker0 -j MASQUERADE
-A DOCKER -i docker0 -j RETURN
COMMIT
# Completed on Thu Mar 11 01:05:03 2021

^G Help      ^O Write Out   ^W Where Is
^X Exit      ^R Read File  ^M Replace  ^K Cut
                                         ^U Paste   ^T Execute
                                         ^J Justify ^C Location
                                         ^L Go To Line
```

Edit

Simpan file ctrl+o, keluar dari nano ctrl+x. Agar file konfigurasi terload jalankan perintah

```
(root@linuxhackingid)-[~/a233sec]
# sudo nano /etc/iptables/rules.v4

(root@linuxhackingid)-[~/a233sec]
# sudo iptables-restore -t < /etc/iptables/rules.v4

(root@linuxhackingid)-[~/a233sec]
# sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save

(root@linuxhackingid)-[~/a233sec]
# sudo netfilter-persistent reload
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables start
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables start
```

Linux Security

Mengamankan server Linux memang penting untuk melindungi data, atau segala threat lainnya dari tangan para cracker (peretas). Jika kamu seorang hacker atau yang baru belajar hacking malu dong jika linux anda diretas? Nah kebetulan Linuxhackingid punya beberapa cara yang bisa kamu coba nihh.

Enkripsi Komunikasi Data pada Linux

Semua data yang dikirimkan melalui jaringan terbuka harus dienkripsi agar ketika ada unauthorized access yang ingin sniffing, maka si blackhat tidak dapat membaca dalam bentuk plain text. Enkripsi data yang dikirim bisa memungkinkan dengan kata sandi atau menggunakan kunci / sertifikat.

1. Gunakan scp, ssh, rsync, atau sftp untuk transfer file. Anda juga dapat memasang sistem file server jarak jauh atau direktori home Anda sendiri menggunakan sshfs khusus dan alat sekering.
2. GnuPG memungkinkan untuk mengenkripsi dan menandatangani data dan komunikasi Anda, menampilkan sistem manajemen kunci serbaguna serta modul akses untuk semua jenis direktori kunci publik.
3. OpenVPN adalah VPN SSL ringan yang hemat biaya. Pilihan lainnya adalah mencoba tinc yang menggunakan tunneling dan enkripsi untuk membuat jaringan pribadi yang aman antara host di Internet atau LAN pribadi yang tidak aman.

Hindari Service yang Tidak Aman

Di bawah sebagian besar konfigurasi jaringan, nama pengguna, kata sandi, perintah FTP / Telnet / RSH dan file yang ditransfer dapat ditangkap oleh siapa saja di jaringan yang sama menggunakan paket sniffer. Solusi umum untuk masalah ini adalah dengan menggunakan OpenSSH, SFTP, atau FTPS (FTP over SSL), yang menambahkan enkripsi SSL atau TLS ke FTP.

SELinux

Saya sangat merekomendasikan menggunakan Selinux yang menyediakan kontrol akses wajib yang fleksibel/Mandatory Access Control (MAC). Di bawah Standard Linux Discretionary Access Control (DAC), aplikasi atau proses yang berjalan sebagai pengguna (UID atau SUID) memiliki izin pengguna ke objek seperti file, soket, dan proses lainnya. Menjalankan kernel Mac melindungi sistem dari aplikasi berbahaya atau cacat yang dapat merusak atau menghancurkan sistem.

Disable Service yang Tidak Perlu

Nonaktifkan semua layanan dan daemon yang tidak perlu (layanan yang berjalan di latar belakang). Anda perlu menghapus semua layanan yang tidak diinginkan dari start-up sistem. Ketikkan perintah berikut untuk membuat daftar semua layanan yang dimulai saat boot saat dijalankan.

- # chkconfig -list | grep '3:on'
- # service **serviceName** stop
- # chkconfig **serviceName** off

Konfigurasi Iptables

Block FTP Service

- iptables -A INPUT -p tcp --dport 21 -j DROP

Block Dengan IP yang spesifik

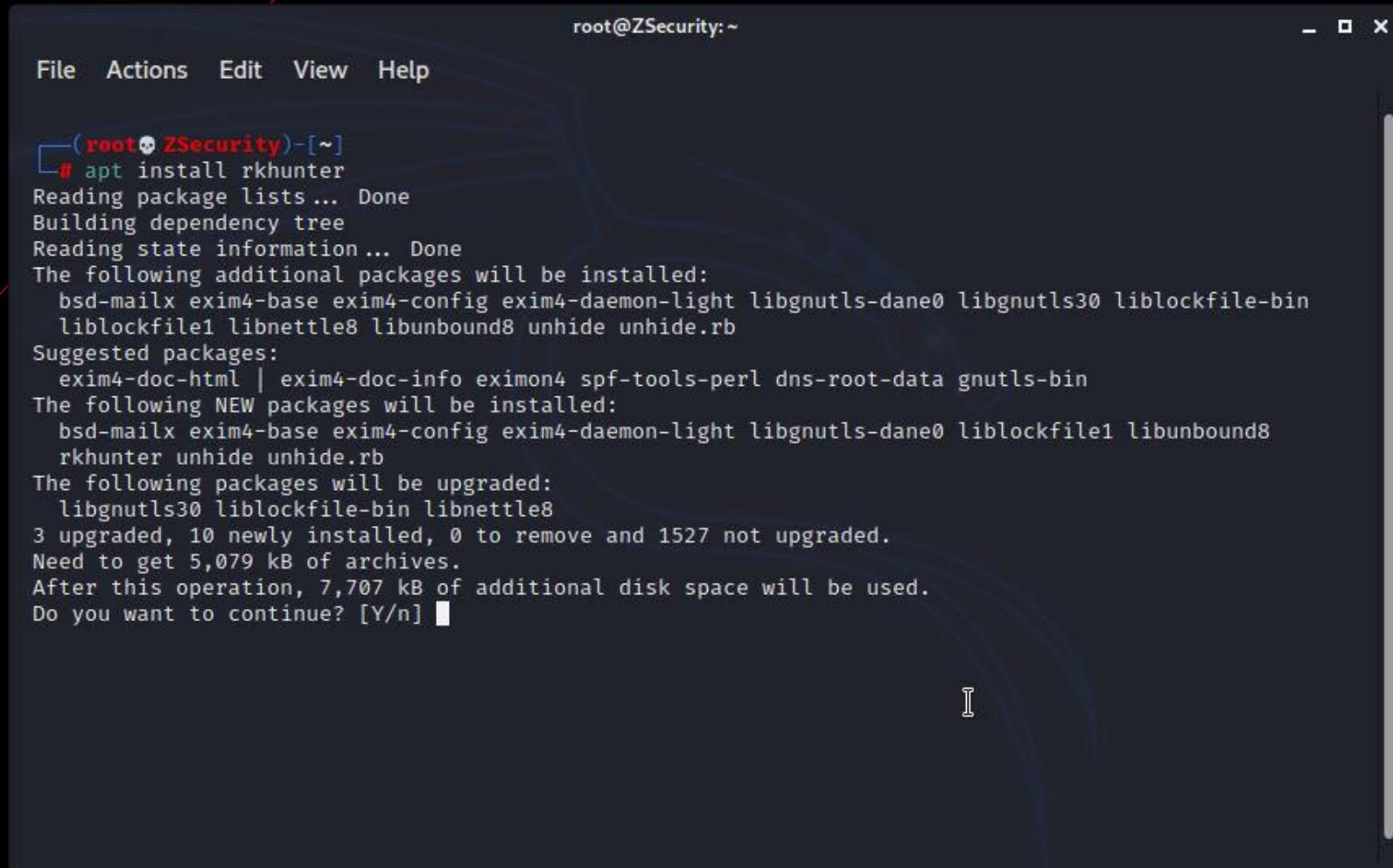
- Iptables -A INPUT -p tcp -s 10.10.10.10 -dport 21 -j DROP

Check Rootkit Linux

Rootkit adalah malicious software yang digunakan untuk privilege escalation yang berisi beberapa malcode untuk mendapatkan akses root. Linuxhackingid akan memberikan cara untuk mendeteksi rootkit pada Linux kamu.

- sudo apt update
- sudo apt install chkrootkit rkhunter -y

Check Rootkit Linux



root@ZSecurity:~

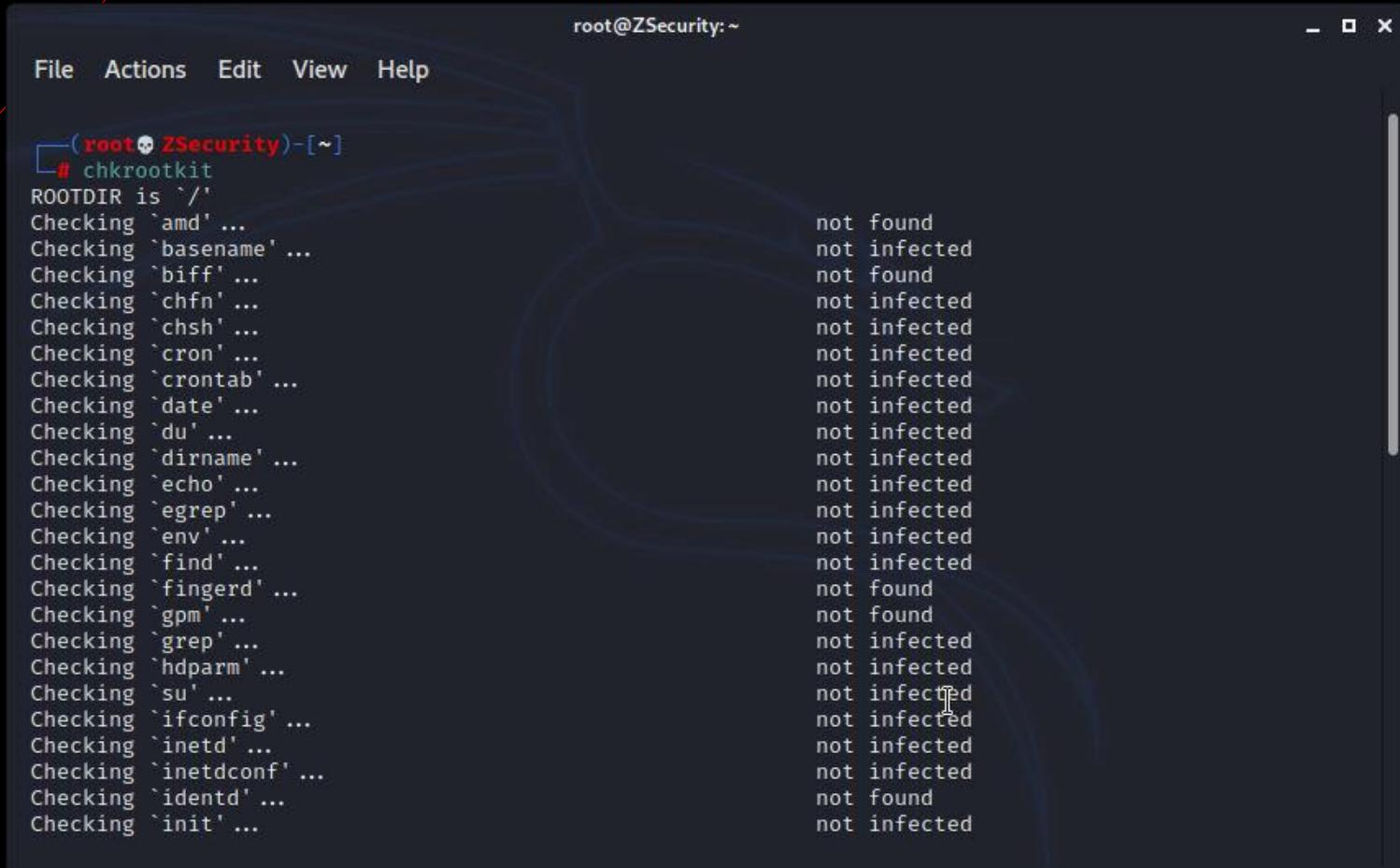
File Actions Edit View Help

```
[root@ZSecurity ~]# apt install rkhunter
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bsd-mailx exim4-base exim4-config exim4-daemon-light libgnutls-dane0 libgnutls30 liblockfile-bin
  liblockfile1 libnettle8 libunbound8 unhide unhide.rb
Suggested packages:
  exim4-doc-html | exim4-doc-info eximon4 SPF-tools-perl dns-root-data gnutls-bin
The following NEW packages will be installed:
  bsd-mailx exim4-base exim4-config exim4-daemon-light libgnutls-dane0 liblockfile1 libunbound8
  rkhunter unhide unhide.rb
The following packages will be upgraded:
  libgnutls30 liblockfile-bin libnettle8
3 upgraded, 10 newly installed, 0 to remove and 1527 not upgraded.
Need to get 5,079 kB of archives.
After this operation, 7,707 kB of additional disk space will be used.
Do you want to continue? [Y/n] ■
```

Secara Default, chkrootkit sudah terinstall jadi saya hanya install rkhunter

Testing Chkrootkit

- sudo chkrootkit

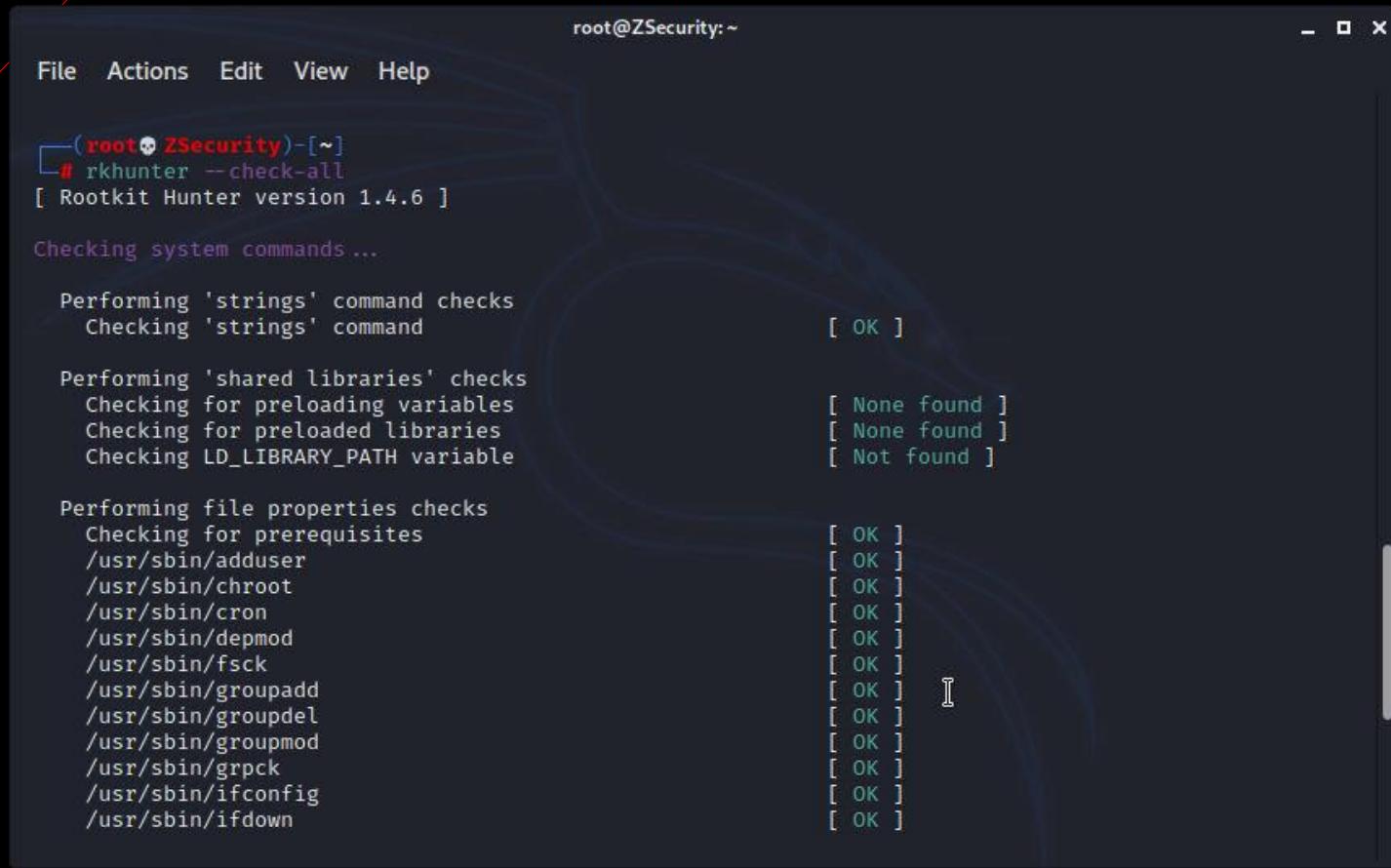


The screenshot shows a terminal window titled "root@ZSecurity:~". The window contains the output of the "chkrootkit" command. The output lists various system binaries being checked, with their status indicating whether they are "not found" or "not infected".

```
root@ZSecurity:~  
File Actions Edit View Help  
[root@ZSecurity ~]# chkrootkit  
ROOTDIR is '/'  
Checking `amd' ... not found  
Checking `basename' ... not infected  
Checking `biff' ... not found  
Checking `chfn' ... not infected  
Checking `chsh' ... not infected  
Checking `cron' ... not infected  
Checking `crontab' ... not infected  
Checking `date' ... not infected  
Checking `du' ... not infected  
Checking `dirname' ... not infected  
Checking `echo' ... not infected  
Checking `egrep' ... not infected  
Checking `env' ... not infected  
Checking `find' ... not infected  
Checking `fingerd' ... not found  
Checking `gpm' ... not found  
Checking `grep' ... not infected  
Checking `hdparm' ... not infected  
Checking `su' ... not infected  
Checking `ifconfig' ... not infected  
Checking `inetd' ... not infected  
Checking `inetdconf' ... not infected  
Checking `identd' ... not found  
Checking `init' ... not infected
```

Testing Rkhunter

- sudo rkhunter -check-all



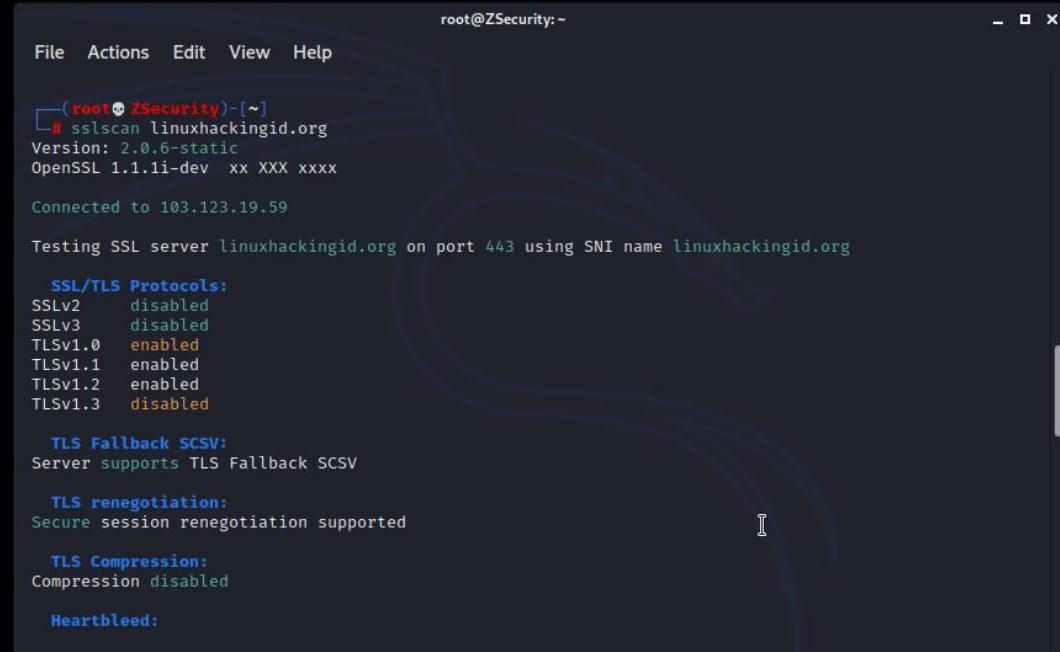
The screenshot shows a terminal window titled "root@ZSecurity:~". The command "rkhunter -check-all" is run, and the output is displayed. The output shows various system checks being performed, such as command checks, shared libraries, file properties, and prerequisites. Most results are marked as "[OK]", while some are "[None found]" or "[Not found]".

```
root@ZSecurity:~  
File Actions Edit View Help  
└─(root💀ZSecurity)─[~]  
# rkhunter --check-all  
[ Rootkit Hunter version 1.4.6 ]  
  
Checking system commands ...  
  
Performing 'strings' command checks  
  Checking 'strings' command [ OK ]  
  
Performing 'shared libraries' checks  
  Checking for preloading variables [ None found ]  
  Checking for preloaded libraries [ None found ]  
  Checking LD_LIBRARY_PATH variable [ Not found ]  
  
Performing file properties checks  
  Checking for prerequisites  
    /usr/sbin/adduser [ OK ]  
    /usr/sbin/chroot [ OK ]  
    /usr/sbin/cron [ OK ]  
    /usr/sbin/depmod [ OK ]  
    /usr/sbin/fsck [ OK ]  
    /usr/sbin/groupadd [ OK ]  
    /usr/sbin/groupdel [ OK ]  
    /usr/sbin/groupmod [ OK ]  
    /usr/sbin/grpck [ OK ]  
    /usr/sbin/ifconfig [ OK ]  
    /usr/sbin/ifdown [ OK ]
```

Scanning SSL

SSL adalah jenis enkripsi yang biasa digunakan oleh website. HTTPS mengindikasikan bahwa website tersebut sudah memiliki cert SSL/TLS. Agar data yang dikirim tidak dalam bentuk plain text, diperlukanlah sebuah SSL/TLS agar dienkripsi.

- `sslscan linuxhackingid.org`



```
root@ZSecurity:~#
└─# sslscan linuxhackingid.org
Version: 2.0.6-static
OpenSSL 1.1.1i-dev xx XXX xxxx

Connected to 103.123.19.59

Testing SSL server linuxhackingid.org on port 443 using SNI name linuxhackingid.org

SSL/TLS Protocols:
SSLv2      disabled
SSLv3      disabled
TLSv1.0    enabled
TLSv1.1    enabled
TLSv1.2    enabled
TLSv1.3    disabled

TLS Fallback SCSV:
Server supports TLS Fallback SCSV

TLS renegotiation:
Secure session renegotiation supported

TLS Compression:
Compression disabled

Heartbleed:
```

Wi-Fi Defensive Security

Wifi adalah yang paling banyak digunakan saat-saat ini karena tidak ribet untuk saling berkomunikasi antar sesama tidak seperti kabel yang harus mencolokan terlebih dahulu. Tetapi dibalik kemudahan ini ada masalah keamanan karena pada dasarnya keamanan selalu berbanding terbalik dengan kenyamanan. Masalah yang paling sering adalah ketika para hacker/skid mencoba untuk memasuki/menjebol keamanan wifi. Tetapi setelah kamu baca PDF ini insya Allah akan mengurangi unauthorized itu. Berikut saya berikan implementasi yang harus dilakukan untuk men-defense keamanan wifi.

1. Enkripsi yang digunakan lemah

Salah satu yang sering hacker/skid menjebol keamanan wifi adalah lemahnya enkripsi yang diimplementasikan oleh router untuk mengenkripsi authentication yang ada. Hindari menggunakan WEP, karena dengan WEP kira-kira 10-20 menit wifi bias jebol. WPA-WPA2 direkomendasikan untuk enkripsi.

2. Passwordnya Lemah

Apabila sudah mengganti enkripsi dengan WPA atau WPA2, password kamu juga harus kuat. Hindari model password seperti ini, “passwordku”, “12345678”, “rumahku”. Jika bingung untuk membuat password, bisa kalian kunjungi ke website ini, <https://passwordsgenerator.net>

3. WPS Aktif

Aktifnya WPS dapat menimbulkan kerentanan bagi para hacker/skid, karena WPS semacam ini bisa di bruteforce attack. Maka dengan mendisable WPS akan mengurangi serangan pada Wi-Fi.

4. Firmware Update

Firmware adalah hal yang terpenting, banyak firmware yang jarang diupdate dan beberapa kerentanan tidak ditambal yang mengakibatkan mudahnya seseorang untuk meretas dalam satu jaringan tersebut. Sebaiknya rutin untuk mengecek update pada router dan pilihlah merek vendor router yang selalu update dan peduli dengan keamanan yang ada.

5. Monitoring Pengguna Wi-Fi

Memonitoring pengguna Wi-Fi ialah mengecek apakah user tersebut memang pengguna kita atau sebaliknya, apabila sebaliknya segera blocking user tersebut dan ganti password dan cek kembali enkripsi yang kita gunakan, takutnya enkripsi yang kita gunakan diubah menjadi low security. Hal ini harus diperhatikan terlebih lagi jika kamu membuka sebuah coffee atau tempat makan yang dimana orang-orang dapat menikmati akses wifi pada usaha kamu. Pisahkan juga antara wifi khusus untuk pelanggan dan khusus untuk pribadi.

Detect Sniffer

Sniffer adalah paket analyzer yang digunakan untuk melihat paket dalam satu jaringan. Secara dasarnya, sniffer berkerja pada mode promiscuous. Kali ini kita akan mendekripsi sniffer dengan bantuan software nmap.

- nmap --script=sniffer-detect 192.168.43.1/24

Jika outputnya seperti ini,

Host script results:

```
|_sniffer-detect: Likely in promiscuous mode (tests:  
"11111111")
```

Kemungkinan ada sniffer

Detect Sniffer

```
root@ZSecurity:~  
File Actions Edit View Help  
root@ZSecurity: ~ x root@ZSecurity: ~ x  
└─(root💀 ZSecurity)-[~]  
# nmap --script=sniffer-detect 192.168.43.1/24  
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-12 05:58 EST  
Nmap scan report for 192.168.43.5  
Host is up (0.0086s latency).  
Not shown: 999 closed ports  
PORT      STATE SERVICE  
53/tcp    open  domain  
MAC Address: 00:0A:F5:58:A8:6C (Airgo Networks)  
  
Host script results:  
|_sniffer-detect: Likely in promiscuous mode (tests: "11111111")  
  
Nmap scan report for 192.168.43.53  
Host is up (0.000015s latency).  
All 1000 scanned ports on 192.168.43.53 are closed  
  
Nmap done: 256 IP addresses (2 hosts up) scanned in 3.73 seconds  
└─(root💀 ZSecurity)-[~]  
#
```

LYNIS



Lynis adalah alat keamanan sumber terbuka yang dapat melakukan pemindaian keamanan sistem secara mendalam untuk mengevaluasi profil keamanan sistem. Karena kesederhanaan dan fleksibilitasnya, Lynis bisa digunakan untuk, vulnscan, Security audit, pentest, dll

LYNIS Usage

Install

- sudo apt update
- sudo apt install lynis

Usage

- sudo lynis audit system



Parrot Terminal

```
File Edit View Search Terminal Help
[x]-[zsecurity@linuxhackingid]-[~]
$ sudo lynis audit system
sudo: unable to resolve host linuxhackingid: Temporary failure in name resolution

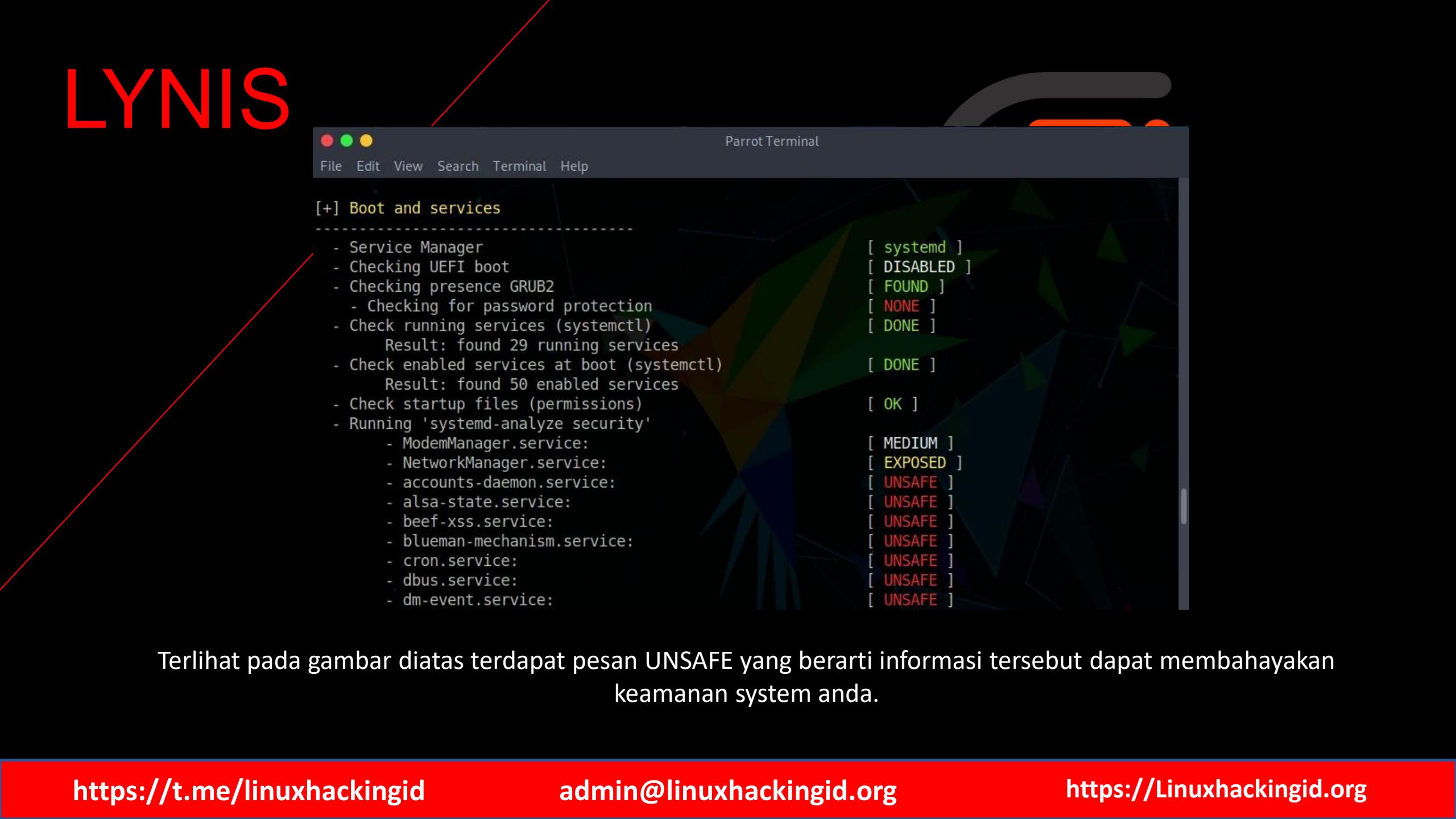
[ Lynis 3.0.0 ]

#####
Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
welcome to redistribute it under the terms of the GNU General Public License.
See the LICENSE file for details about using this software.

2007-2020, CISOfy - https://ciscofy.com/lynis/
Enterprise support available (compliance, plugins, interface and tools)
#####

[+] Initializing program
-----
=====
Exception found!
```

LYNIS



```
Parrot Terminal
File Edit View Search Terminal Help

[+] Boot and services
-----
- Service Manager
- Checking UEFI boot
- Checking presence GRUB2
- Checking for password protection
- Check running services (systemctl)
  Result: found 29 running services
- Check enabled services at boot (systemctl)
  Result: found 50 enabled services
- Check startup files (permissions)
- Running 'systemd-analyze security'
  - ModemManager.service:
  - NetworkManager.service:
  - accounts-daemon.service:
  - alsa-state.service:
  - beef-xss.service:
  - blueman-mechanism.service:
  - cron.service:
  - dbus.service:
  - dm-event.service:

[ systemd ]
[ DISABLED ]
[ FOUND ]
[ NONE ]
[ DONE ]

[ DONE ]

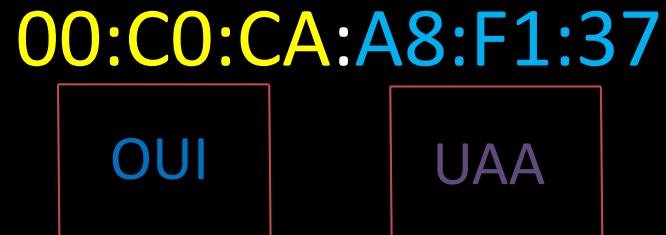
[ OK ]

[ MEDIUM ]
[ EXPOSED ]
[ UNSAFE ]
```

Terlihat pada gambar diatas terdapat pesan UNSAFE yang berarti informasi tersebut dapat membahayakan keamanan system anda.

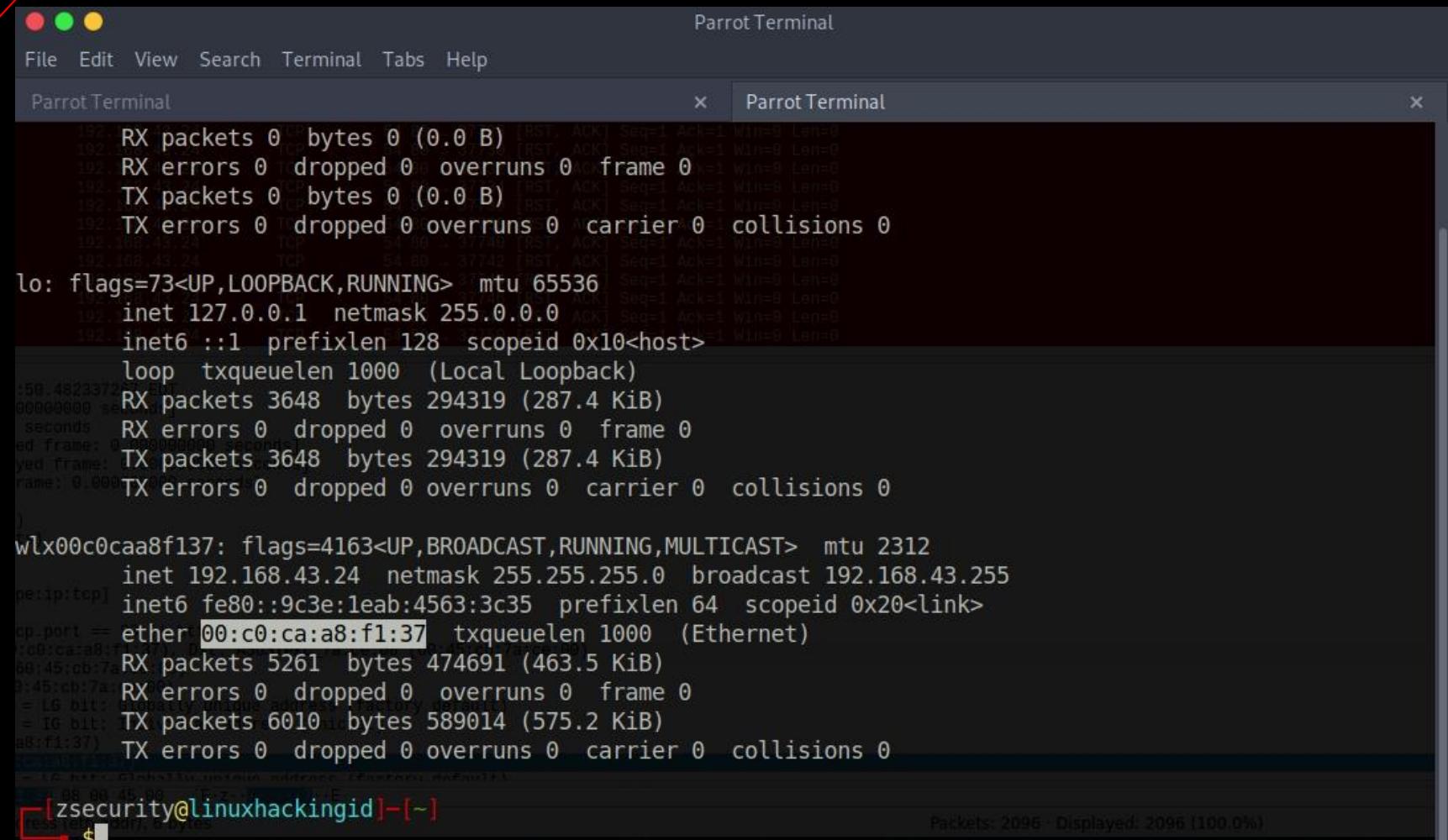
MAC Address Tracking Vendor

MAC Address memiliki 48 bits dan memiliki 6 kolom blok yang masing-masing blok memiliki 2 value. Pada 24 bits pertama disebut dengan OUI atau ini digunakan untuk menentukan vendor yang digunakan oleh user atau penyerang. 24 bits setelah nya sebagai pembeda antara wireless adapter/card satu dengan yang lainnya. Disini saya menggunakan vendor merek Alfa Network yang memiliki MAC Address berikut.



MAC Address Tracking Vendor

Ketik ifconfig untuk melihat MAC Add pada bagian “HW Ether”
00:C0:CA:A8:F1:37

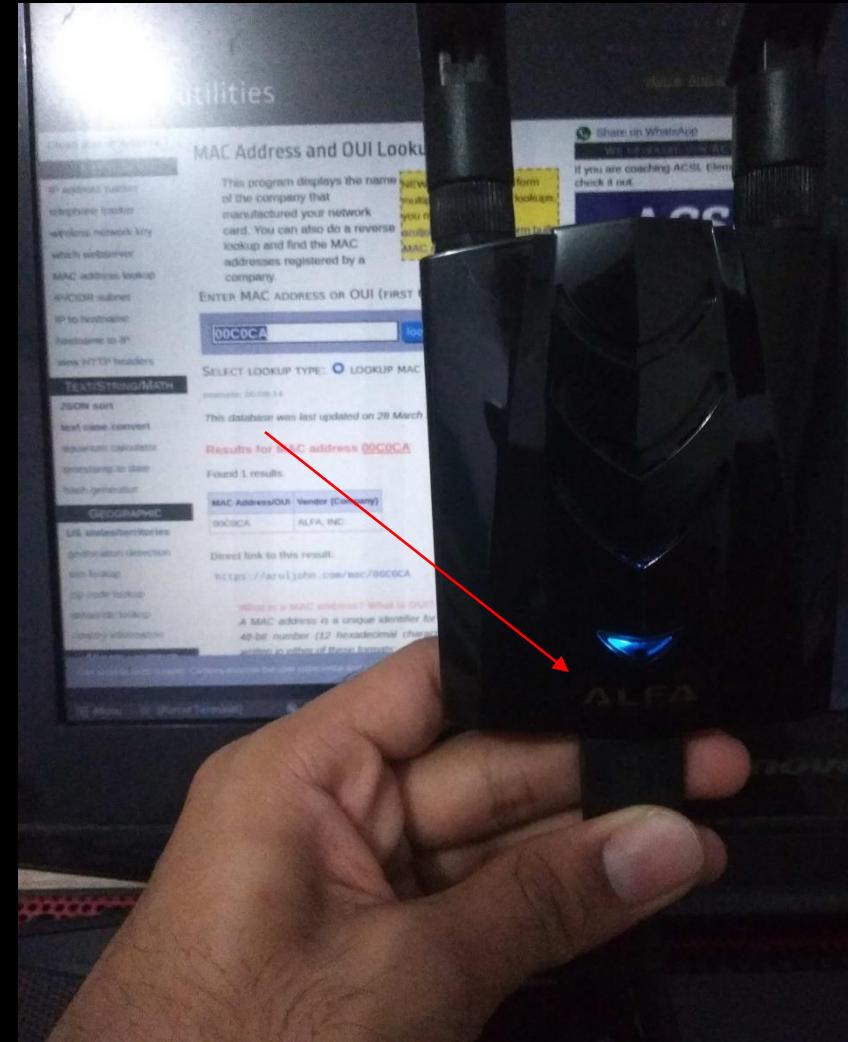


```
Parrot Terminal
File Edit View Search Terminal Tabs Help
Parrot Terminal x Parrot Terminal x
RX packets 0 bytes 0 (0.0 B) [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
RX errors 0 dropped 0 overruns 0 frame 0 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
TX packets 0 bytes 0 (0.0 B) [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
[...]
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 3648 bytes 294319 (287.4 KiB)
seconds RX errors 0 dropped 0 overruns 0 frame 0
TX packets 3648 bytes 294319 (287.4 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
[...]
wlx00c0caa8f137: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 2312
inet 192.168.43.24 netmask 255.255.255.0 broadcast 192.168.43.255
inet6 fe80::9c3e:1aab:4563:3c35 prefixlen 64 scopeid 0x20<link>
ether 00:c0:ca:a8:f1:37 txqueuelen 1000 (Ethernet)
RX packets 5261 bytes 474691 (463.5 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 6010 bytes 589014 (575.2 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
[...]
[zsecurity@linuxhackingid]~[~]
Packets: 2096 Displayed: 2096 (100.0%)
```

MAC ADDRESS Tracking Vendor

<https://aruljohn.com/mac.pl>

Masukan 24 bits pertama ke kolom search dan terlihat bahwa digambar tersebut berhasil mencetak merek vendor dengan MAC Address dan saya sesuaikan dengan merek wireless adapter yang ada pada gambar



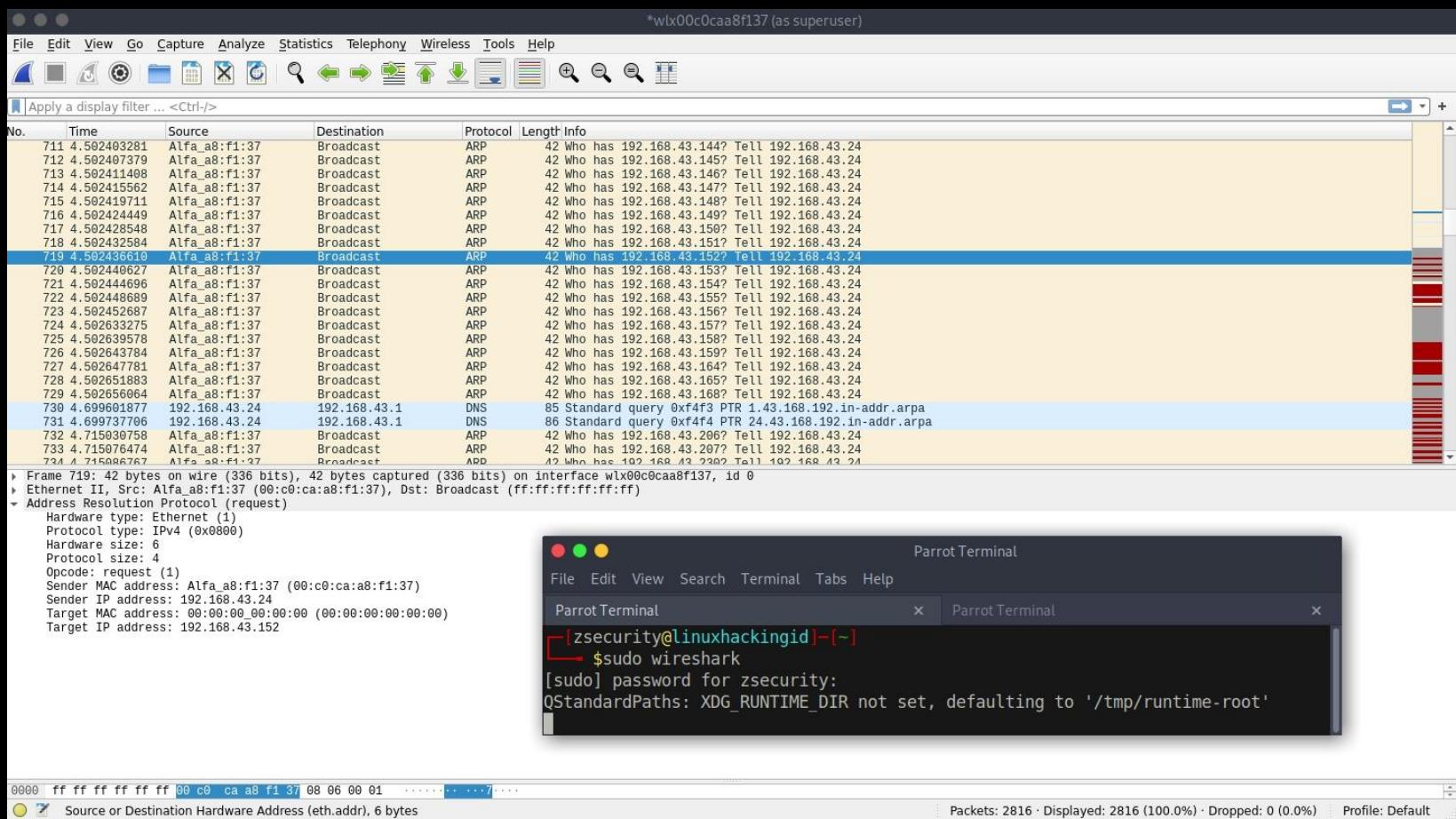
MAC ADDRESS Tracking Vendor

Berikut saya foto dari belakang wireless adapter-nya karena informasi disana lebih jelas

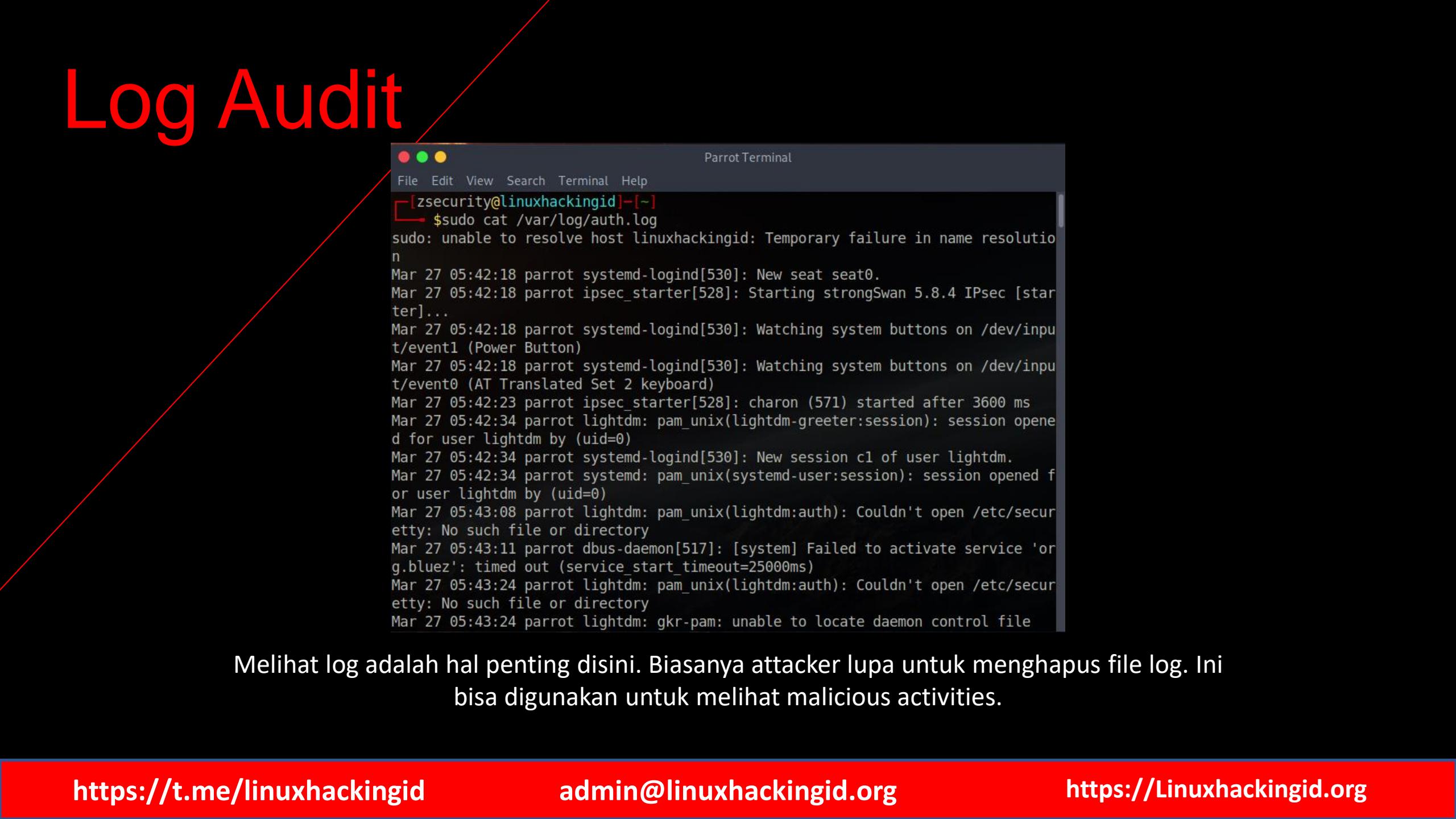


MAC ADDRESS Tracking Vendor

Wireshark pun memiliki list OUI yang bisa me resolve nilai OUI secara otomatis. Dan terlihat di merek vendor di paket list dan detail packet



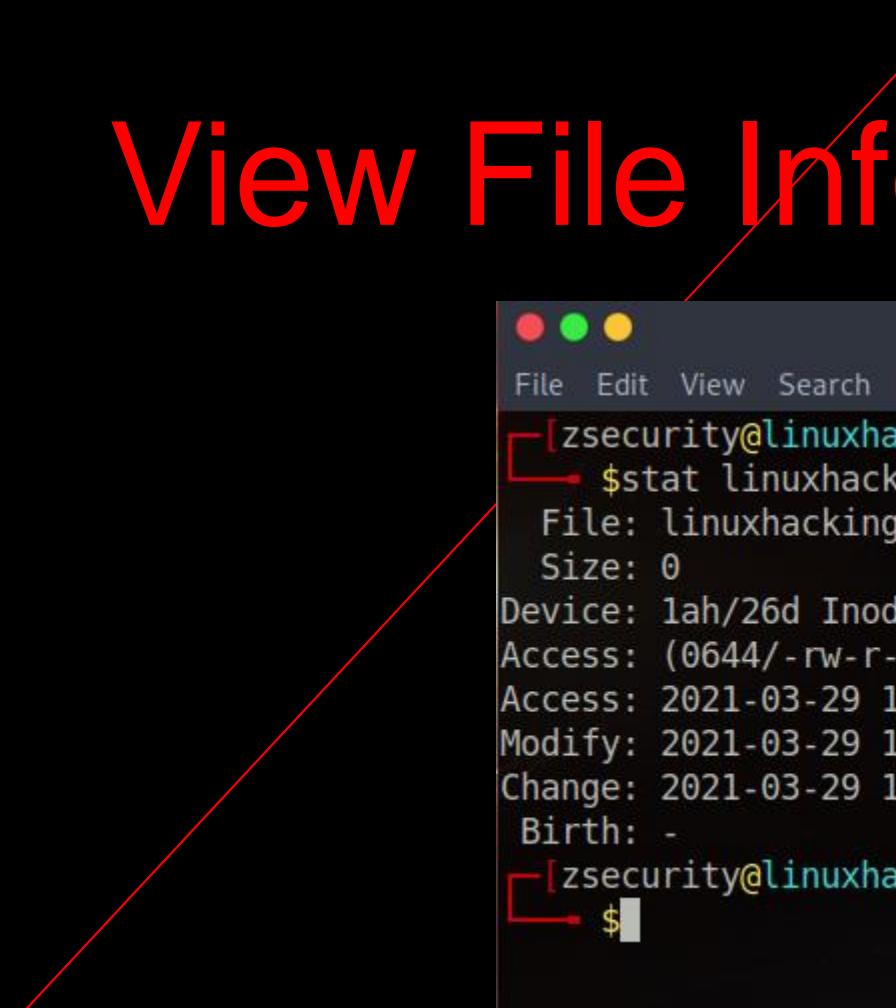
Log Audit



```
Parrot Terminal
File Edit View Search Terminal Help
[zsecurity@linuxhackingid]~
$ sudo cat /var/log/auth.log
sudo: unable to resolve host linuxhackingid: Temporary failure in name resolution
Mar 27 05:42:18 parrot systemd-logind[530]: New seat seat0.
Mar 27 05:42:18 parrot ipsec_starter[528]: Starting strongSwan 5.8.4 IPsec [starter]...
Mar 27 05:42:18 parrot systemd-logind[530]: Watching system buttons on /dev/input/event1 (Power Button)
Mar 27 05:42:18 parrot systemd-logind[530]: Watching system buttons on /dev/input/event0 (AT Translated Set 2 keyboard)
Mar 27 05:42:23 parrot ipsec_starter[528]: charon (571) started after 3600 ms
Mar 27 05:42:34 parrot lightdm: pam_unix(lightdm-greeter:session): session opened for user lightdm by (uid=0)
Mar 27 05:42:34 parrot systemd-logind[530]: New session c1 of user lightdm.
Mar 27 05:42:34 parrot systemd: pam_unix(systemd-user:session): session opened for user lightdm by (uid=0)
Mar 27 05:43:08 parrot lightdm: pam_unix(lightdm:auth): Couldn't open /etc/securetty: No such file or directory
Mar 27 05:43:11 parrot dbus-daemon[517]: [system] Failed to activate service 'org.bluez': timed out (service_start_timeout=25000ms)
Mar 27 05:43:24 parrot lightdm: pam_unix(lightdm:auth): Couldn't open /etc/securetty: No such file or directory
Mar 27 05:43:24 parrot lightdm: gkr-pam: unable to locate daemon control file
```

Melihat log adalah hal penting disini. Biasanya attacker lupa untuk menghapus file log. Ini bisa digunakan untuk melihat malicious activities.

View File Information



```
Parrot Terminal
File Edit View Search Terminal Help
[zsecurity@linuxhackingid]~
$ stat linuxhackingid.txt
  File: linuxhackingid.txt
  Size: 0          Blocks: 0          IO Block: 4096   regular empty file
Device: 1ah/26d Inode: 419369      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1001/zsecurity)  Gid: ( 1001/zsecurity)
Access: 2021-03-29 10:26:34.220216951 -0400
Modify: 2021-03-29 10:26:34.220216951 -0400
Change: 2021-03-29 10:26:34.220216951 -0400
 Birth: -
[zsecurity@linuxhackingid]~
$
```

Melihat kapan aksesnya, permission, modify, dll adalah salah satu teknik untuk Defense. Dengan menggunakan stat namafile bisa melihat informasi tersebut.

RMF-NIST

Proses untuk mengidentifikasi potensi ancaman terhadap organisasi untuk menentukan strategi ,menghilangkan atau meminimalkan dampak risiko ini.

Dapat dihitung dengan:

$$\text{Risk} = \text{Threat} \times \text{Vulnerability}$$

Risk Management Framework

RMF adalah seperangkat kebijakan dan standar keamanan informasi yang dikembangkan oleh pemerintah federal oleh The National Institute of Standards and Technology (NIST).



6 Step RMF

1. Categorize
2. Select
3. Implement
4. Assess
5. Authorize
6. Monitor



Rumus Risk

$$\text{Risk} = \text{Threats} \times \text{Vulnerability}$$

NIST

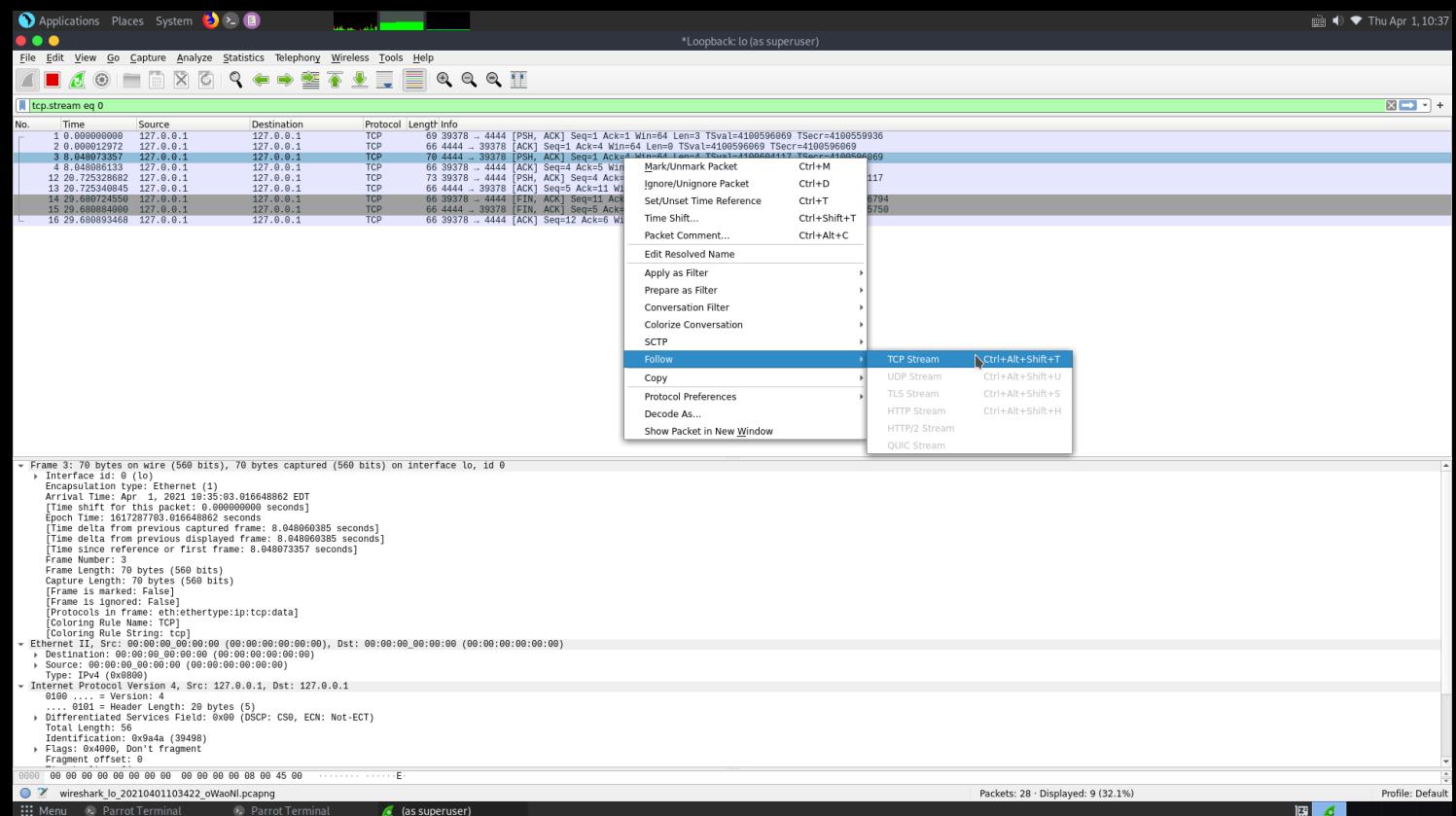
Linuxhackingid.org

Attack Detection Netcat

Pendeteksian Netcat atau sejenisnya biasanya menggunakan wireshark untuk melihat komunikasi yang dikirim maupun diterima. Seperti analisis menggunakan DNSCat misalnya, dapat dideteksi menggunakan wireshark juga akan tetapi komunikasi yang memang betul2 DNS yang asli jelas sekali berbeda, salah satunya pada bagian size yang dikirim oleh DNSCat. Biasanya DNSCat mengirim 300 bytes/paket yang dikirim dan biasanya kan kalo DNS dia A atau AAAA akan tetapi kalau DNSCat dia juga mengirim TXT dan MX yang tidak umum diminta

Attack Detection Netcat

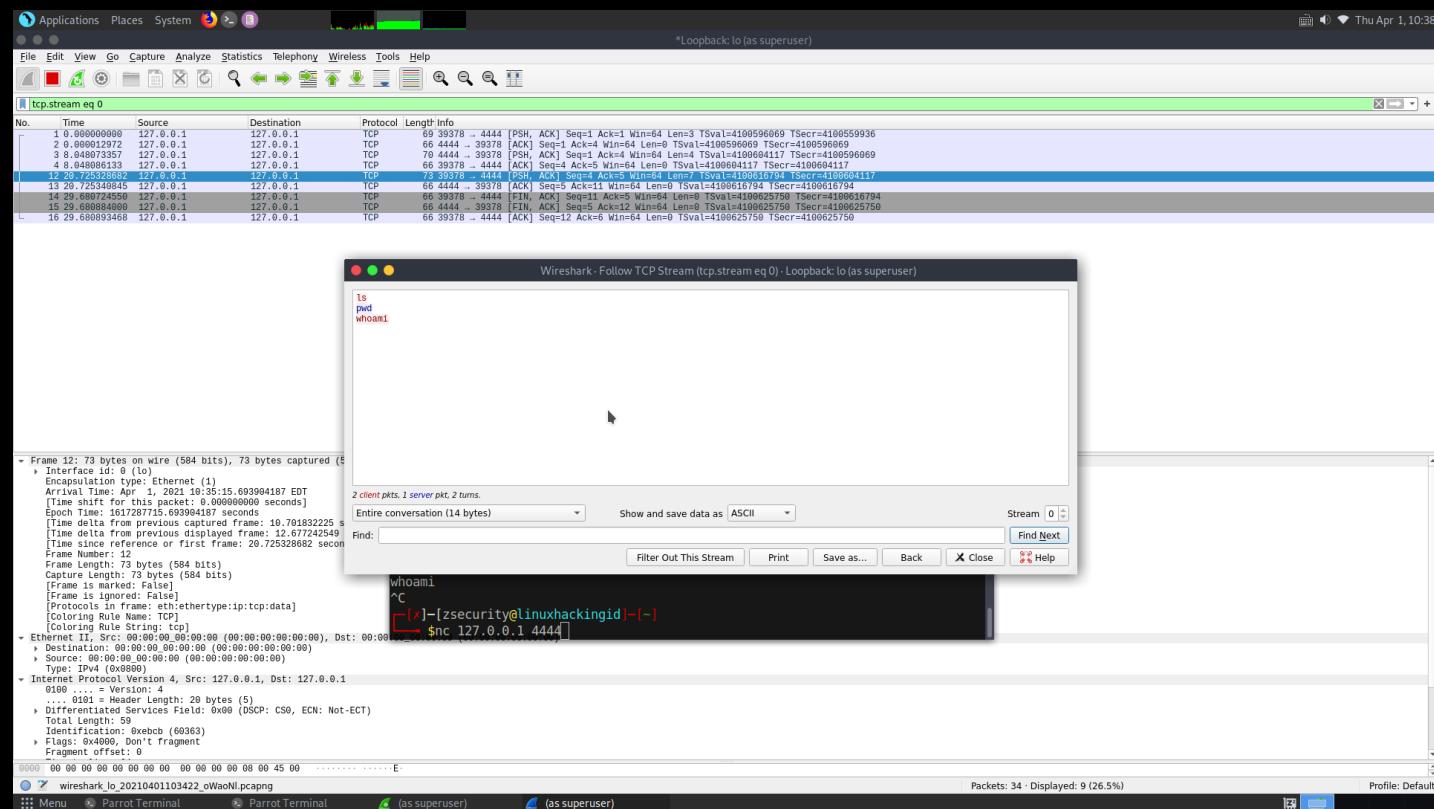
Sekarang kita akan mencoba mendeteksi Netcat. Setelah menjalankan wireshark dalam satu interface yang terhubung ke dalam satu jaringan yang menggunakan netcat, maka lakukan sniffing. Netcat menggunakan TCP sebagai media transmisinya untuk menganalisisnya, klik kanan salah satu paket dan pilih Follow TCP Stream.



Attack Detection Netcat

Berikut isi komunikasi command attacker yang dikirim dari attacker ke korban untuk melakukan gaining access. Dan terlihat pada percobaan ini, client mengirim ls dan whoami ke server machine

Note: warna merah menunjukan client
warna biru menunjukan server



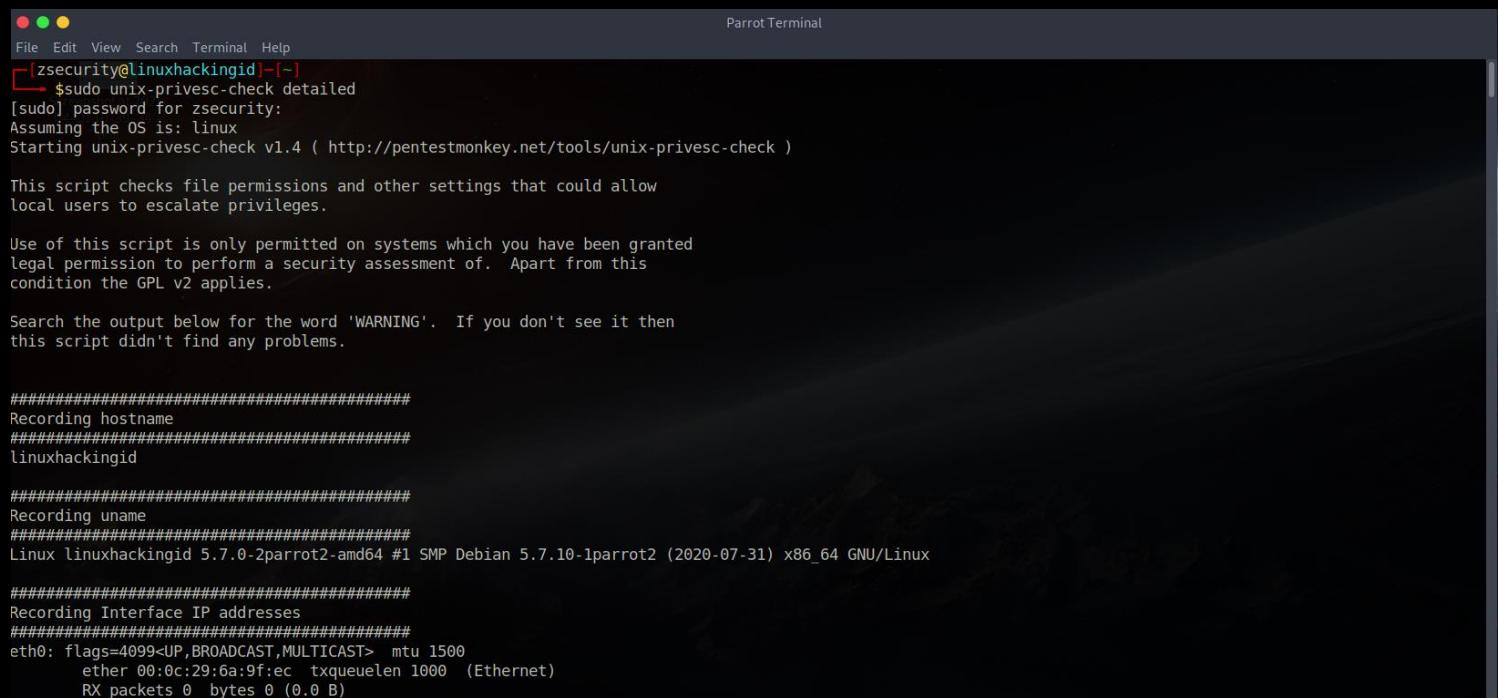
Checking Vulnerable for Privesc

Privilege Escalation (Privesc) cara untuk mengambil akses yang lebih tinggi. Seperti gini, hacker berhasil mendapatkan akses ke system windows anda (standard user), akan tetapi dia butuh mendapatkan akses yang lebih tinggi (Administrator) dibutuhkanlah cara ini untuk mendapatkan Administrator.

Unix-privesc-check salah satu tools untuk mendeteksi vulnerable privesc

Usage:

- sudo unix-privesc-check detailed



```
[zsecurity@linuxhackingid] ~
$ sudo unix-privesc-check detailed
[sudo] password for zsecurity:
Assuming the OS is: linux
Starting unix-privesc v1.4 ( http://pentestmonkey.net/tools/unix-privesc-check )

This script checks file permissions and other settings that could allow
local users to escalate privileges.

Use of this script is only permitted on systems which you have been granted
legal permission to perform a security assessment of. Apart from this
condition the GPL v2 applies.

Search the output below for the word 'WARNING'. If you don't see it then
this script didn't find any problems.

#####
Recording hostname
#####
linuxhackingid

#####
Recording uname
#####
Linux linuxhackingid 5.7.0-2parrot2-amd64 #1 SMP Debian 5.7.10-1parrot2 (2020-07-31) x86_64 GNU/Linux

#####
Recording Interface IP addresses
#####
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      ether 00:0c:29:6a:9f:ec txqueuelen 1000  (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
```

UAC di Win10

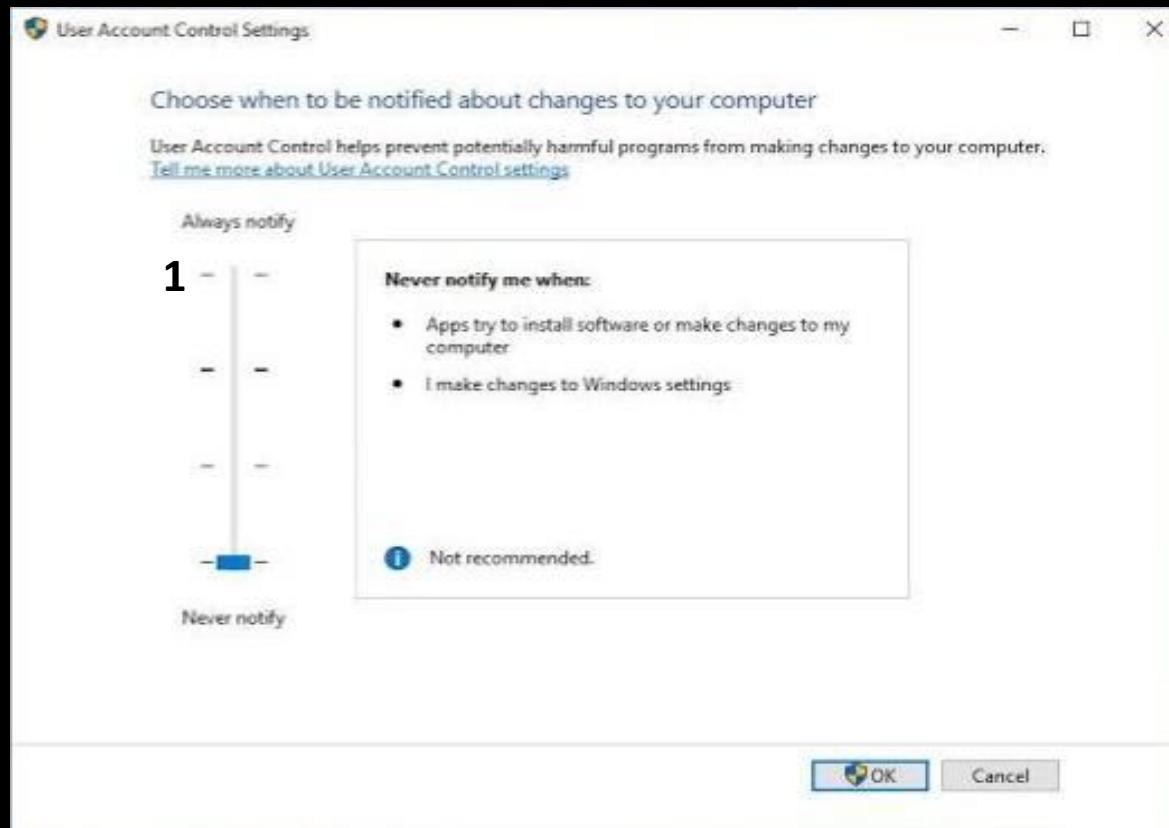
User Access Control untuk mencegah aplikasi atau program membuat perubahan yang di dalam sistem Windows.

1. High — Always Notify

Selalu beri tahu jika aplikasi mencoba menginstal perangkat lunak atau membuat perubahan pada komputer.

Tujuan:

Microsoft merekomendasikan pengaturan ini jika Anda menginstal program baru secara konsisten dan membuka situs web yang berpotensi tidak aman



UAC di Win10

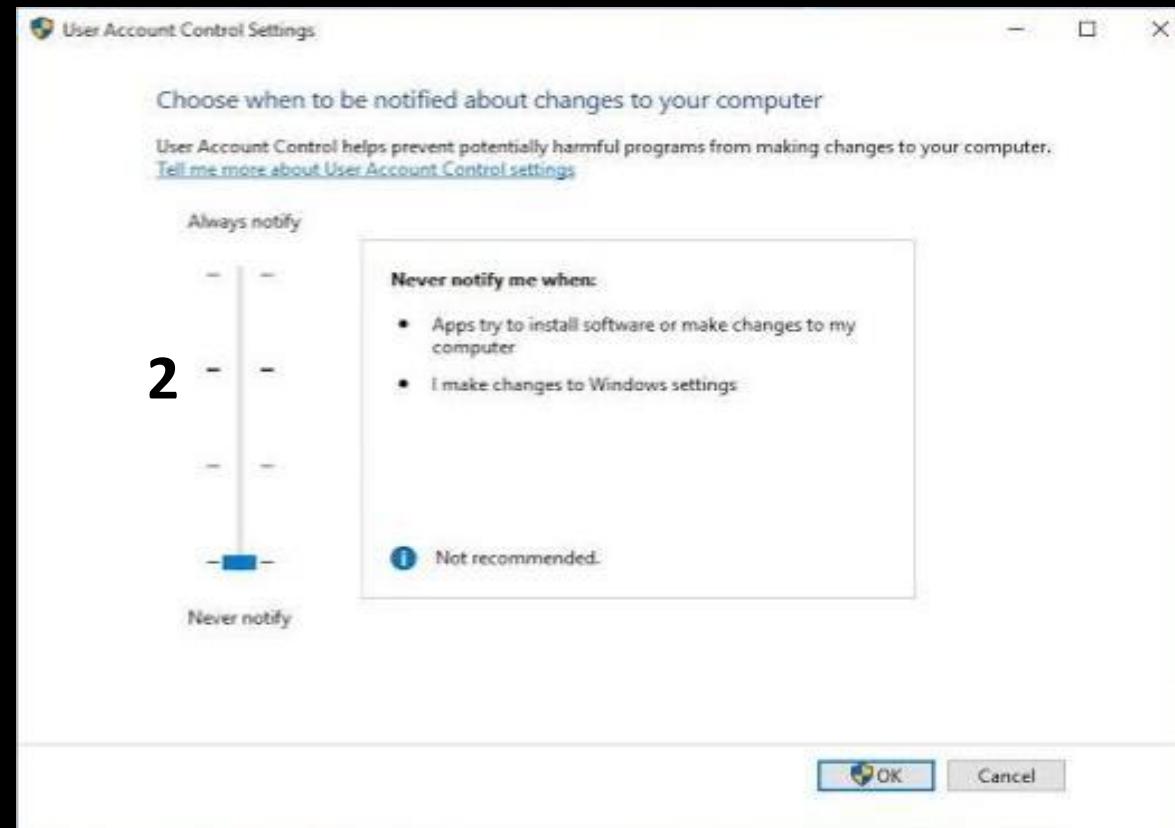
User Access Control untuk mencegah aplikasi atau program membuat perubahan yang di dalam sistem Windows.

2. Medium High (default)

Beri tahu hanya jika aplikasi mencoba melakukan perubahan pada komputer.

Tujuan:

Microsoft menganjurkan pengaturan ini jika Anda memiliki daftar aplikasi tertentu yang Anda jalankan dan situs web yang Anda kunjungi secara teratur



UAC di Win10

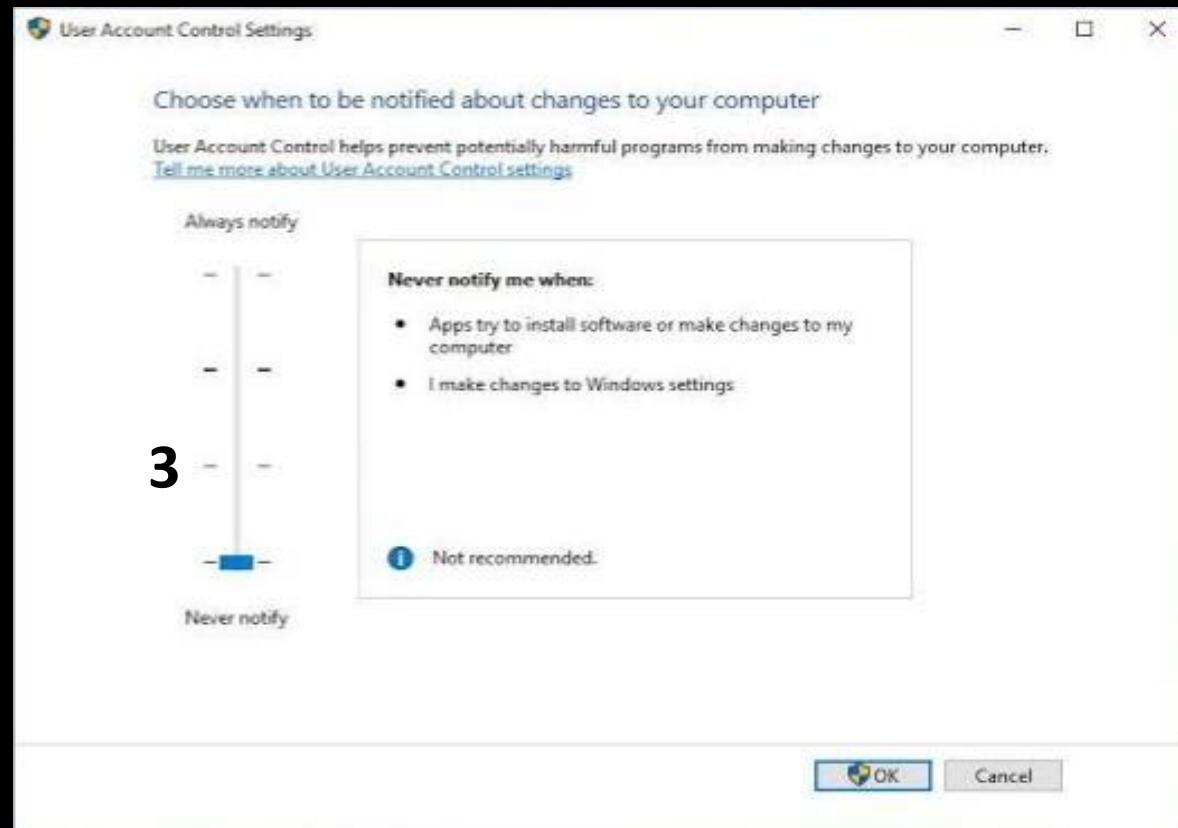
User Access Control untuk mencegah aplikasi atau program membuat perubahan yang di dalam sistem Windows.

3. Medium Low

program yang mana itu belum mendapatkan suatu verifikasi secara resmi dari Microsoft dan tidak mempunyai digital signature

Tujuan:

Microsoft menganjurkan pengaturan ini jika Anda memiliki daftar aplikasi tertentu yang Anda jalankan dan situs web yang Anda kunjungi secara teratur



UAC di Win10

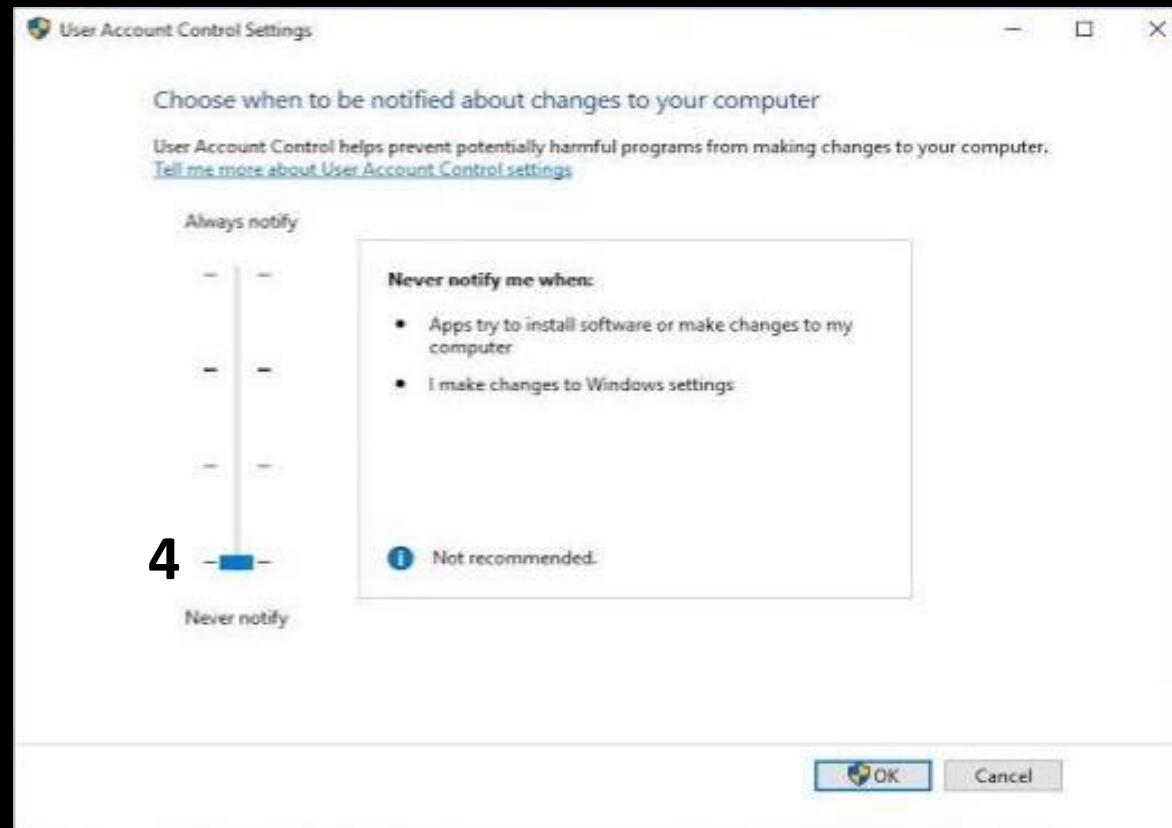
User Access Control untuk mencegah aplikasi atau program membuat perubahan yang di dalam sistem Windows.

4. Low

Jangan pernah beri tahu saya saat aplikasi mencoba menginstal perangkat lunak atau membuat perubahan pada komputer

Tujuan:

Fungsi ini menetapkan UAC serendah mungkin. Meskipun ini secara efektif menonaktifkan UAC, mungkin masih ada perlindungan tertentu yang aktif. Microsoft tidak menganjurkan pengaturan ini jika memungkinkan



Bluetooth Security

1. Update

Cek selalu firmware update

2. Matiin kalau ngga dipake

Jika Bluetooth tidak digunakan, maka matikan Bluetooth itu dan jika ingin menggunakannya maka nyalakan lagi bluetooth nya

3. Limit app permission

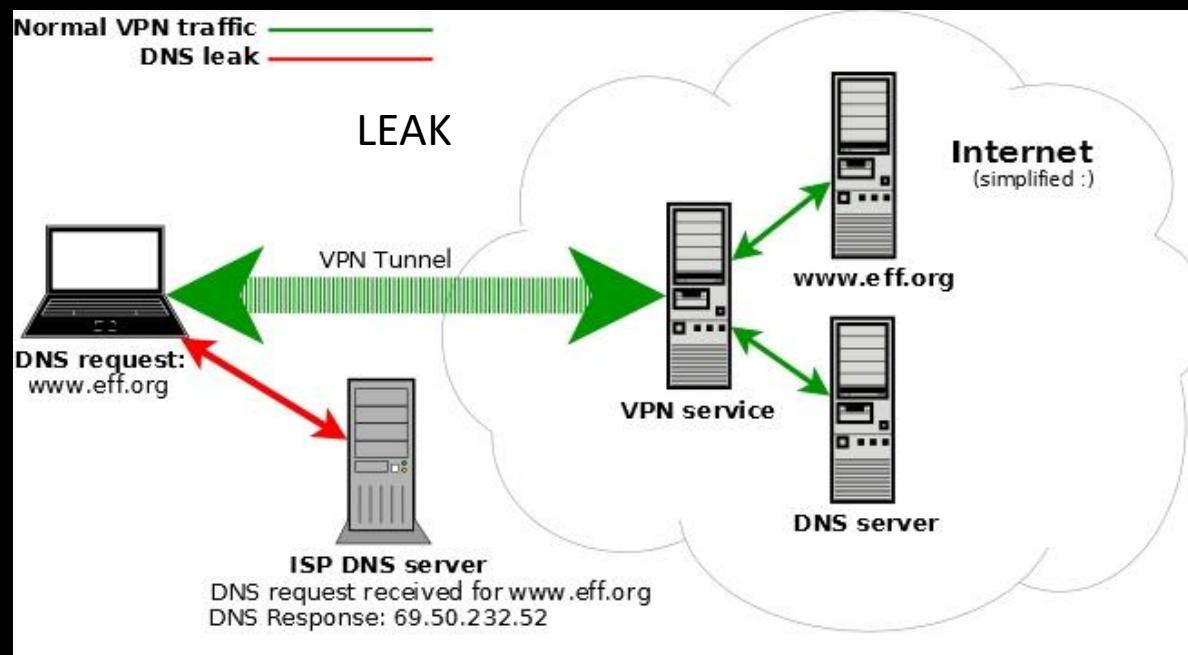
Ada beberapa aplikasi yang mengharuskan untuk menggunakan permission bluetooth,maka cek app permission tersebut di setting pada menu aplikasi di android kalian.

4. Jaga jarak

Jaga jarak disini bukan berarti social Distancing, akan tetapi bluetooth bisa digunakan jika jaraknya dekat dengannya. Maka jika kamu merasa kurang aman dengan lingkungan sekitar kamu, sebaiknya matikan langsung fitur bluetooth itu.

DNS Leak

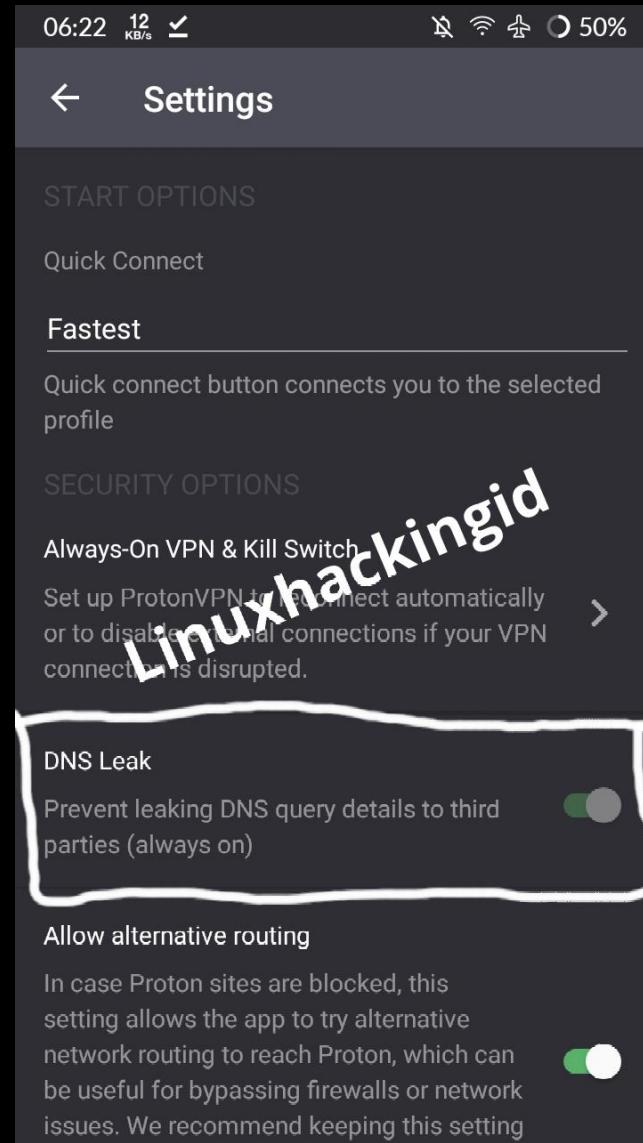
DNS Leak pada kelemahan keamanan yang memungkinkan permintaan DNS diungkapkan ke server DNS ISP, meskipun layanan VPN digunakan untuk mencoba menyembunyikannya. Meskipun terutama menyangkut pengguna VPN, mungkin juga mencegahnya untuk proxy dan pengguna internet langsung (Wikipedia).



DNS Leak-Prevent

VPN yang dapat mencegah DNS Leak

1. ExpressVPN
2. NordVPN
3. Proton VPN
4. Surfshark
5. Cyberghost



ProtonVPN

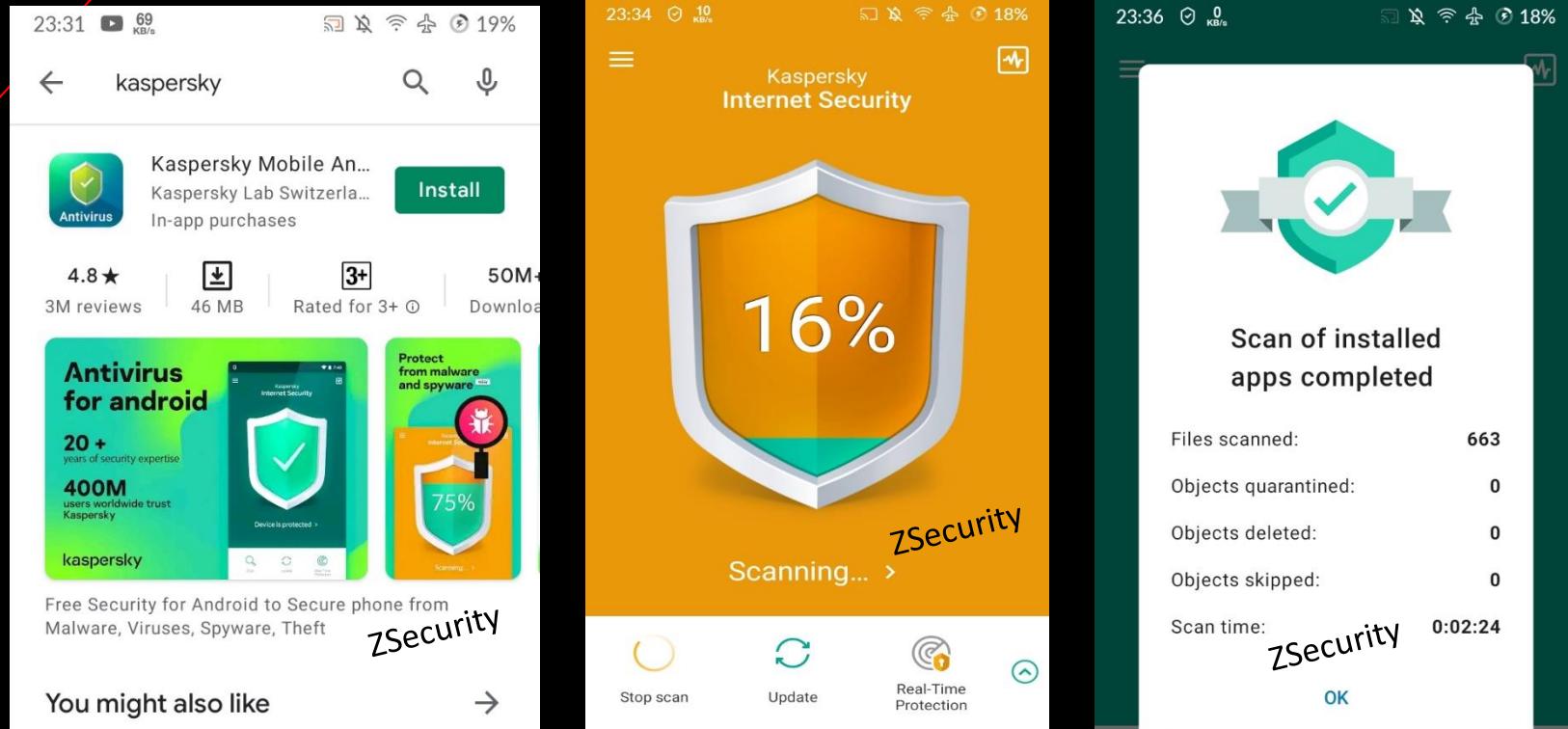
Android Security Topik

Topik Android Security:

1. Antivirus
2. Android Permission
3. ADB (Android Debug Bridge)
4. Port Scanning Local Android
5. Stagefright Detector

Android Antivirus

Untuk mengurangi dampak terinfeksinya android terhadap serangan malware, kita bias menggunakan Antivirus dari Kaspersky yang saya sudah lakukan pengujian dan Kaspersky adalah AV yang menurut saya bagus di device Android.

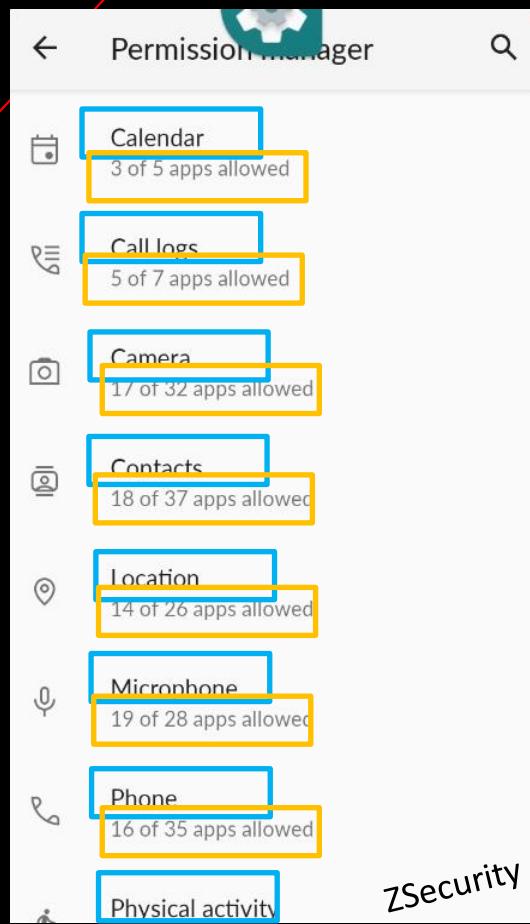


Android Permission

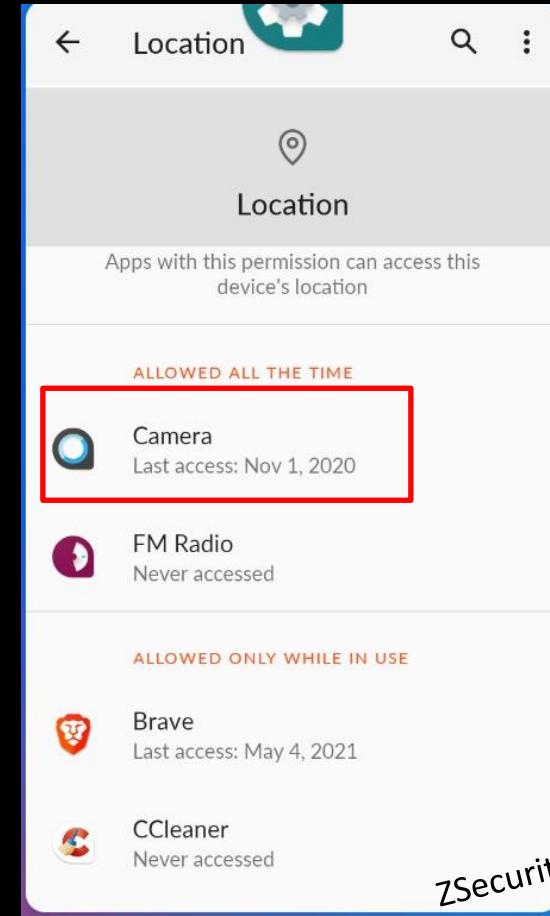
Kita juga harus cek beberapa permission yang digunakan oleh aplikasi yang kita install. Ini adalah salah satu hal yang harus dilakukan. Izinkan permission aplikasi yang memang seharusnya digunakan, contohnya aplikasi kamera hanya izinkan permission “**Camera**” dan “**File Storage**”. Permission **Camera** untuk menggunakan hardware kamera kita, dan **File Storage** untuk menyimpan hasil foto kita di penyimpanan kita.

Android Permission di Settings

Gambar di samping merupakan permission beserta aplikasi yang sudah diizinkan maupun yang belum. Warna kotak biru adalah nama permission-nya, dan warna orange aplikasi yang sudah diizinkan maupun yang belum.



App Permission Manager

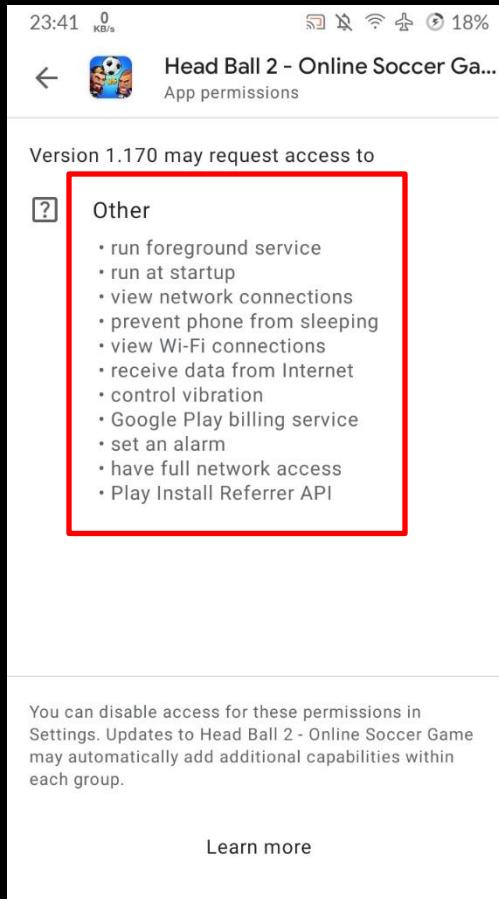
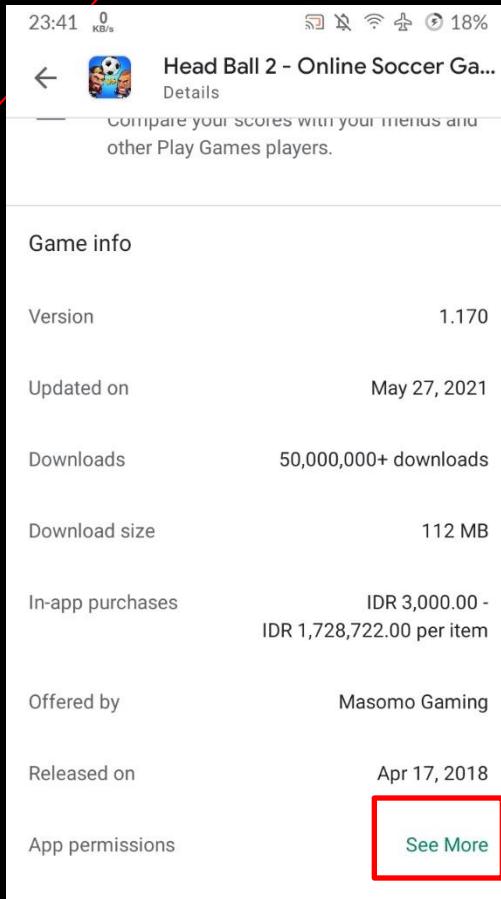


Permission Location Accesed

Terlihat pada gambar disamping bahwa aplikasi "Camera" menggunakan Permission "Location" pada tanggal 1 November 2020

Android Permission di Playstore

Pilih “See More” untuk melihat secara detail permission yang digunakan game tersebut



Ini adalah contoh full permission yang digunakan pada game tersebut

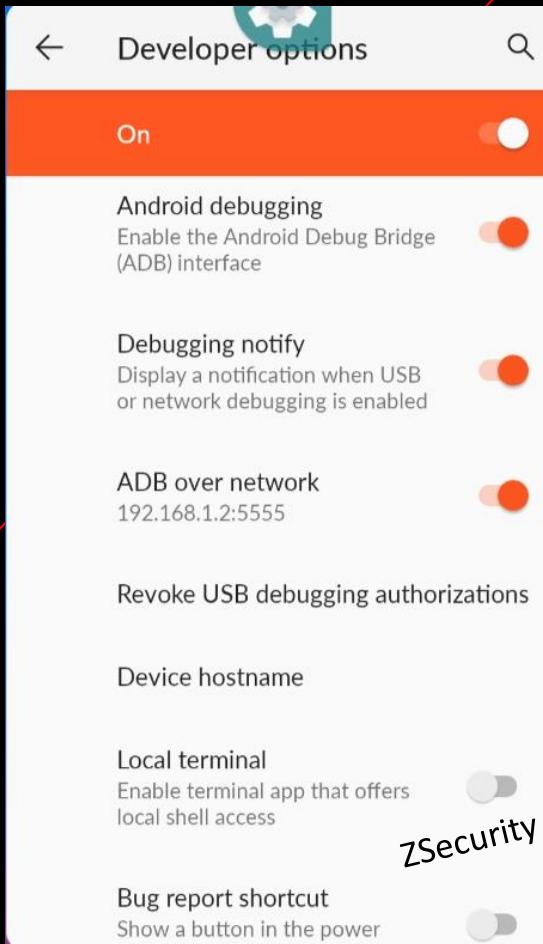
Android Debug Bridge



ADB tools command line yang memungkinkan pengguna untuk berkomunikasi dengan perangkat. Perintah adb memfasilitasi berbagai tindakan perangkat, seperti menginstal dan men-debug aplikasi, dan memberikan akses ke shell Unix yang dapat Anda gunakan untuk menjalankan berbagai perintah di perangkat. Ini adalah program klien-server yang meliputi tiga komponen:

- **Klien**, yang mengirimkan perintah. Klien berjalan pada mesin pengembangan. Kita dapat memanggil klien dari terminal command line dengan mengeluarkan perintah adb.
- **Daemon (adb)**, yang menjalankan perintah di perangkat. Daemon berjalan sebagai proses latar belakang/background di setiap perangkat.
- **Server**, yang mengelola komunikasi antara klien dan daemon. Server berjalan sebagai proses latar belakang pada mesin pengembangan kita.

Android Debug Bridge



ADB Running

The screenshot shows a terminal window titled 'Parrot Terminal'. The user has run the command '\$ sudo adb connect 192.168.1.2:5555' and is connected to the device. They then run '\$ adb shell' and enter a root shell. The terminal shows the following session:

```
$ sudo adb connect 192.168.1.2:5555
already connected to 192.168.1.2:5555
$ adb shell
X00H:/ $ pwd
/
X00H:/ $ whoami
shell
X00H:/ $ su
X00H:/ # whoami
root
X00H:/ #
```

ADB Connect

Android Port Scanning

Dalam Port Scanning kita melihat port yang terbuka di android kita, bisa saja port yang terbuka adalah remote access shell yang sudah dimasuki dan dibuka oleh peretas guna meremote android kalian.

```
23:55 >_ 0 kB/s          39%
$ nmap -p- 127.0.0.1
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06
-07 23:54 WIB
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0061s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE
5555/tcp   open  freeciv

Nmap done: 1 IP address (1 host up) scanned in 62.
47 seconds
$ █
```

ZSecurity

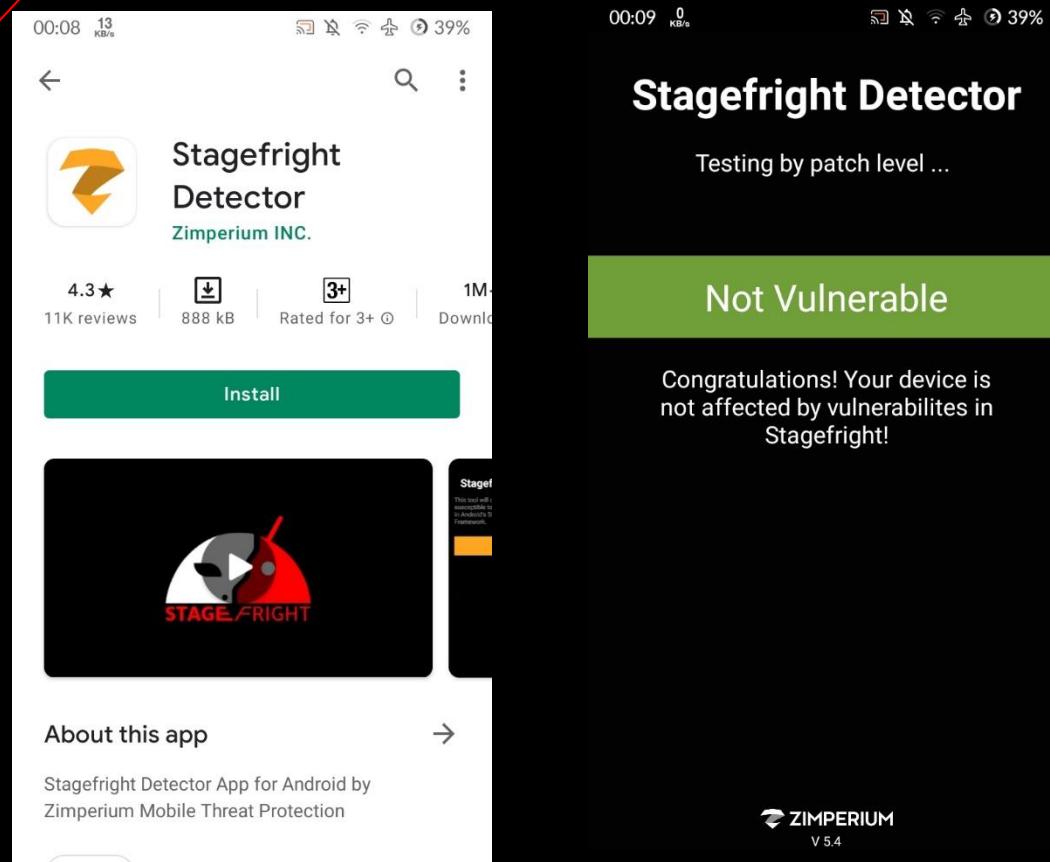
Dan terlihat port 5555 running pada android kita dan default nya itu adalah port ADB yang sudah di bahas pada halaman sebelumnya.

Android Stagefright Vulnerability

Stagefright adalah julukan yang diberikan untuk serangkaian kerentanan yang ditemukan di pemutar media Stagefright yang digunakan oleh perangkat Android. Kerentanan Stagefright membawa implikasi keamanan yang serius: penyerang dapat mengeksplitasinya untuk mengontrol dan mencuri data dari jarak jauh dari perangkat dengan mengirimkan pesan multimedia (MMS) yang dikemas dengan eksploitasi kepada korban. Kerentanan Stagefright mempengaruhi perangkat Android yang menjalankan Froyo 2.2 hingga Lollipop 5.1.1.



Android Stagefright Vulnerability



Kill User Session di Linux

Dalam contoh ini hekel skid koneksi ke Ubuntu server menggunakan SSH

```
zsecurity@linuxhackingid:~$ who
zsecurity  tty1          2021-06-11 16:37
zsecurity  pts/0          2021-06-11 16:42 (192.168.43.173)
zsecurity@linuxhackingid:~$ _
```

Dengan command *who* hanya simple informasi skid

```
zsecurity@linuxhackingid:~$ w
16:42:25 up 7 min,  2 users,  load average: 0.15, 0.27, 0.18
USER   TTY     FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
zsecuri tty1      -           16:37    0.00s  0.32s  0.00s  w
zsecuri pts/0    192.168.43.173  16:42    8.00s  0.05s  0.05s -bash
zsecurity@linuxhackingid:~$ _
```

Dengan command *w* dapat melihat lebih detail hingga proses yang sedang di exec oleh si skid ☺

Kill User Session di Linux

Kill User SSH

```
zsecurity@linuxhackingid:~$ w
 16:45:03 up 9 min,  2 users,  load average: 0.01, 0.15, 0.15
USER   TTY      FROM          LOGIN@    IDLE   JCPU   PCPU WHAT
zsecurit  tty1     -           16:37    3.00s  0.33s  0.00s w
zsecurit pts/0    192.168.43.173  16:42   55.00s  0.05s  0.05s -bash
zsecurity@linuxhackingid:~$ pkill -9 -t pts/0
```

Menggunakan pkill untuk terminate skid user

```
zsecurity@linuxhackingid:~$ pwd
/home/zsecurity
zsecurity@linuxhackingid:~$ whoami
zsecurity
zsecurity@linuxhackingid:~$ Connection to 192.168.43.134 closed.
msfadmin@metasploitable:~$ _
```

Skid udah di terminate session menggunakan pkill

Email Security

1. Password Cycling
2. Secure Login
3. Spam Filtering
4. Email Encryption
5. Education

Password Cycling

Untuk ini sub-bab ini, mengharuskan kepada pengguna untuk menggunakan kata sandi yang unik dan diganti secara berkala, misalnya diganti setiap 1 minggu sekali dan tidak dipake lagi password yang udah di pake pada sebelumnya.

Secure Login

Harus adanya enkripsi yang diterapkan untuk mencegah sniffing atau MITM. Enkripsi juga dapat di decrypt jika menggunakan enkripsi yang lemah.

Filter Spam

Software untuk scanning pesan yang masuk dan dapat memblokir attachment email yang berisi malware.

Email Encryption

OpenPGP memungkinkan untuk mengenkripsi email antara pengirim dan penerima

User Education

Ini adalah social engineering, biasnya mengirim phishing di email dengan iming-iming mendapatkan uang/hadiah. Ada banyak jenis serangan social engineering seperti whaling, spear phishing, Baiting, dll

User Education

Ini adalah social engineering, biasnya mengirim phishing di email dengan iming-iming mendapatkan uang/hadiah. Ada banyak jenis serangan social engineering seperti whaling, spear phishing, Baiting, dll

NIDS

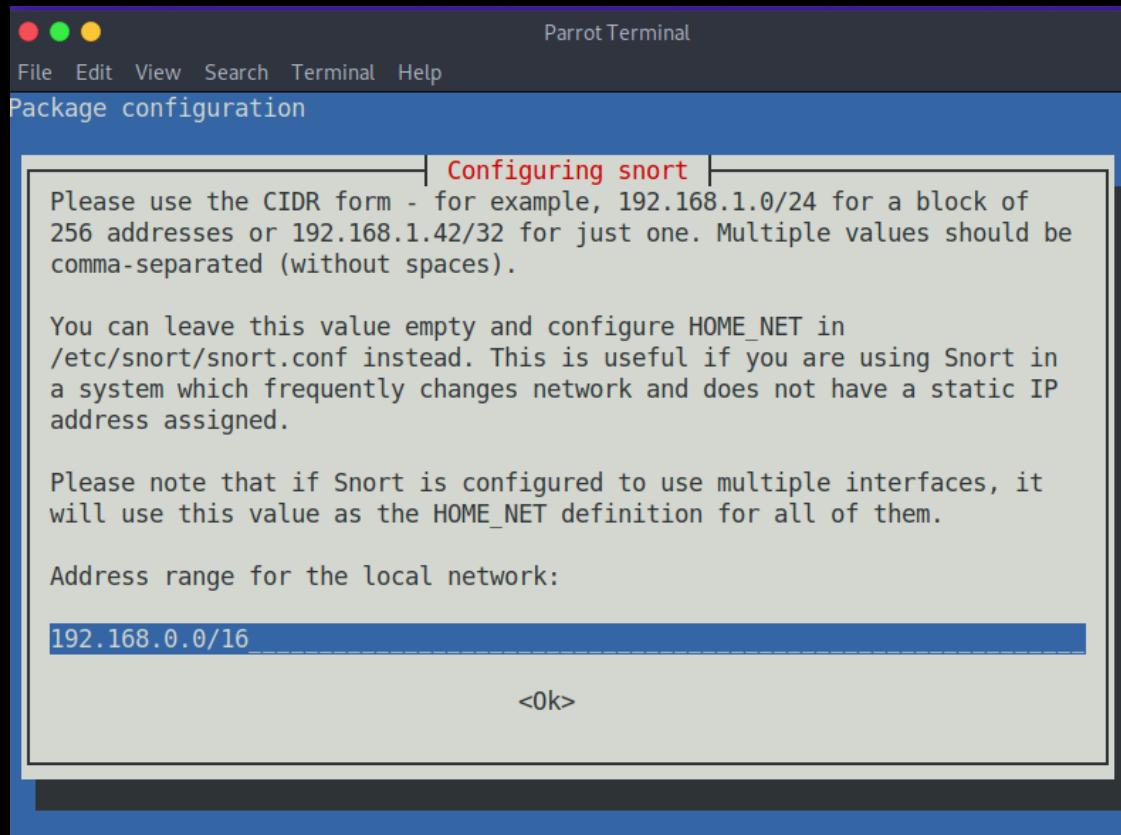
Network Intrusion Detection System untuk mendeteksi adanya serangan yang sudah terkonfigurasi melalui file config yang sudah memiliki beragam config pendektsian, mulai dari serangan ddos hingga dapat mendeteksi tingkah laku malware yang menggunakan jaringan sebagai media berkomunikasi atau penyebarannya.

Snort NIDS

Snort termasuk kedalam kategori NIDS karena dapat mendeteksi melalui jaringan. Snort bertindak sebagai sniffer yang dapat melihat semua paket yang melintasi di jaringan tersebut lalu dicocokan dengan config yang sudah dibuat

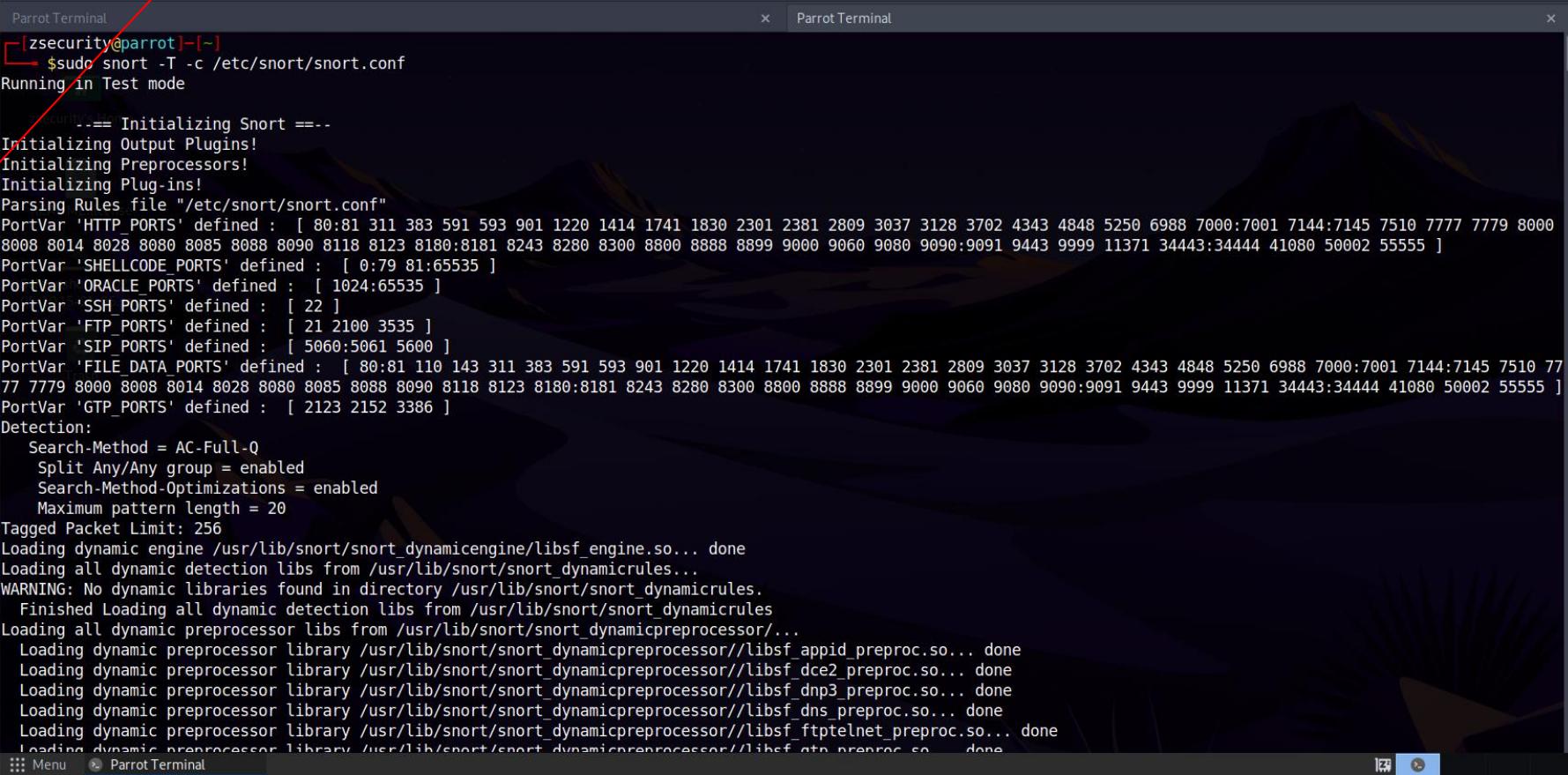
Snort NIDS-Install

```
sudo apt update && apt install snort -y
```



Snort NIDS-Konfigurasi

```
sudo snort -T -c /etc/snort/snort.conf
```



The screenshot shows a terminal window titled "Parrot Terminal" with the command \$ sudo snort -T -c /etc/snort/snort.conf entered. The output of the command is displayed, showing the initialization of Snort, parsing of rules, and loading of dynamic engines and preprocessors.

```
[zsecurity@parrot:~] $sudo snort -T -c /etc/snort/snort.conf
Running in Test mode

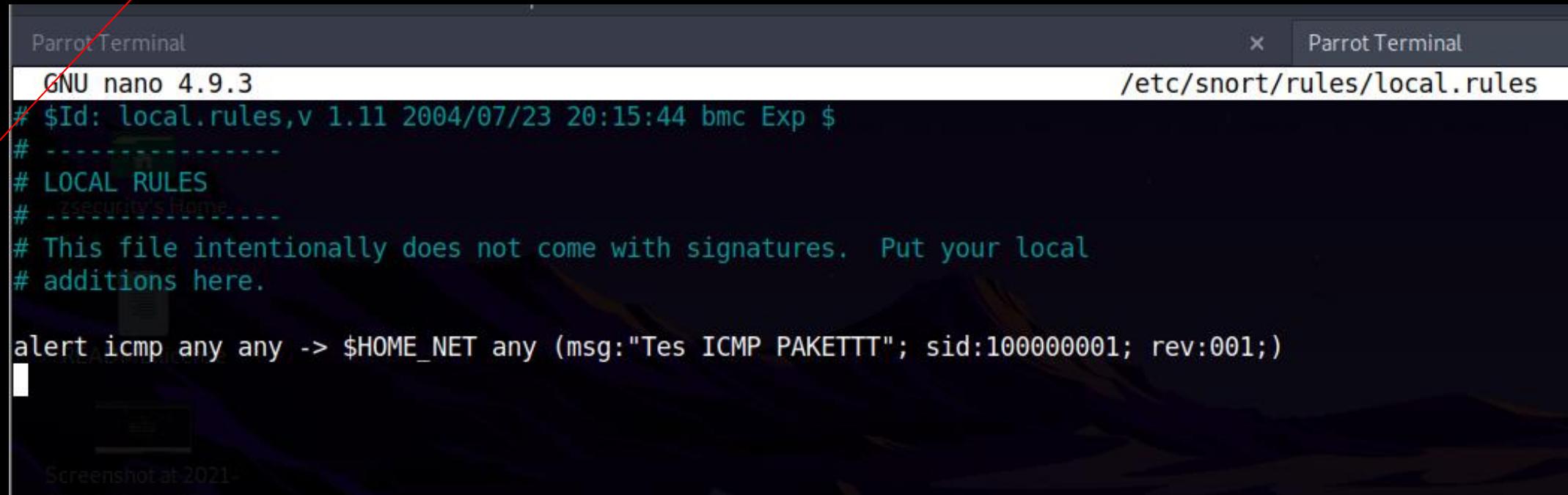
--== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined : [ 80:81 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'SHELLCODE_PORTS' defined : [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined : [ 1024:65535 ]
PortVar 'SSH_PORTS' defined : [ 22 ]
PortVar 'FTP_PORTS' defined : [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined : [ 5060:5061 5600 ]
PortVar 'FILE_DATA_PORTS' defined : [ 80:81 110 143 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 77 77 779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'GTP_PORTS' defined : [ 2123 2152 3386 ]
Detection:
  Search-Method = AC-Full-Q
  Split Any/Any group = enabled
  Search-Method-Optimizations = enabled
  Maximum pattern length = 20
Tagged Packet Limit: 256
Loading dynamic engine /usr/lib/snort/snort_dynamicengine/libsf_engine.so... done
Loading all dynamic detection libs from /usr/lib/snort/snort_dynamicrules...
WARNING: No dynamic libraries found in directory /usr/lib/snort/snort_dynamicrules.
  Finished Loading all dynamic detection libs from /usr/lib/snort/snort_dynamicrules
Loading all dynamic preprocessor libs from /usr/lib/snort/snort_dynamicpreprocessor/...
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_appid_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_dce2_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_dnp3_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_dns_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_ftptelnet_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_gtp_preproc.so... done
```

Snort NIDS-Konfigurasi

```
sudo nano /etc/snort/local.rules
```

1. Masukan konfig berikut untuk mendeteksi paket ICMP

```
alert icmp any any -> $HOME_NET any (msg:"BEBAS";  
sid:100000001; rev:001;)
```



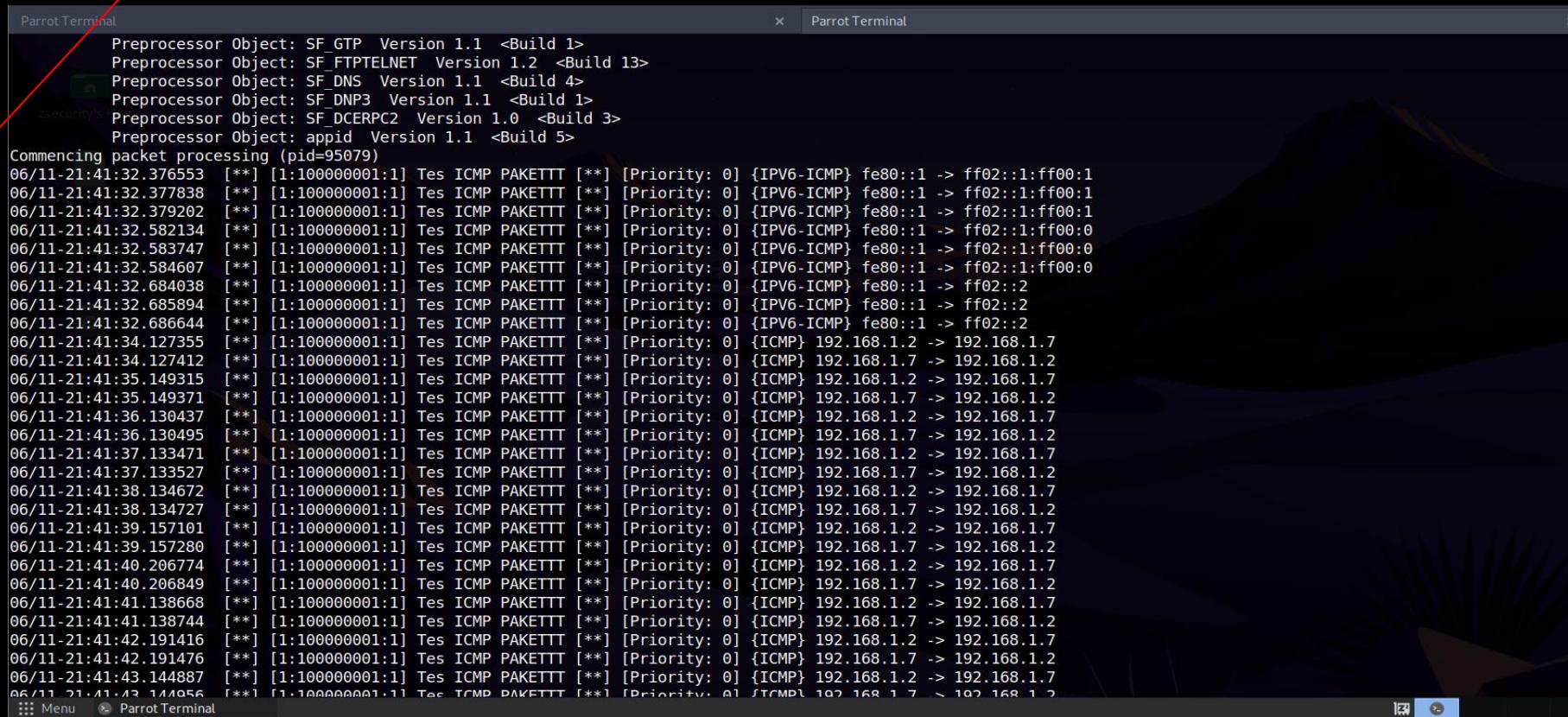
```
Parrot Terminal
GNU nano 4.9.3
/etc/snort/rules/local.rules
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -zsecurity's Home -
# This file intentionally does not come with signatures. Put your local
# additions here.

alert icmp any any -> $HOME_NET any (msg:"Tes ICMP PAKETTT"; sid:100000001; rev:001;)

Screenshot at 2021-
```

Snort NIDS-Testing

```
sudo snort -A console -i <ifacenya> -u snort -g snort -c /etc/snort/snort.conf
```



The screenshot shows a terminal window titled "Parrot Terminal" displaying the output of the Snort NIDS tool. The output indicates that Snort is processing packets from interface "ifacenya" using configuration file "/etc/snort/snort.conf". The log shows numerous ICMP echo requests (pings) being processed, primarily between 192.168.1.2 and 192.168.1.7. The log entries are timestamped and show the source and destination IP addresses and ports for each packet.

```
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Preprocessor Object: appid Version 1.1 <Build 5>
Commencing packet processing (pid=95079)
06/11-21:41:32.376553 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {IPV6-ICMP} fe80::1 -> ff02::1:ff00:1
06/11-21:41:32.377838 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {IPV6-ICMP} fe80::1 -> ff02::1:ff00:1
06/11-21:41:32.379202 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {IPV6-ICMP} fe80::1 -> ff02::1:ff00:1
06/11-21:41:32.582134 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {IPV6-ICMP} fe80::1 -> ff02::1:ff00:0
06/11-21:41:32.583747 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {IPV6-ICMP} fe80::1 -> ff02::1:ff00:0
06/11-21:41:32.584607 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {IPV6-ICMP} fe80::1 -> ff02::1:ff00:0
06/11-21:41:32.684038 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {IPV6-ICMP} fe80::1 -> ff02::2
06/11-21:41:32.685894 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {IPV6-ICMP} fe80::1 -> ff02::2
06/11-21:41:32.686644 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {IPV6-ICMP} fe80::1 -> ff02::2
06/11-21:41:34.127355 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.2 -> 192.168.1.7
06/11-21:41:34.127412 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.7 -> 192.168.1.2
06/11-21:41:35.149315 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.2 -> 192.168.1.7
06/11-21:41:35.149371 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.7 -> 192.168.1.2
06/11-21:41:36.130437 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.2 -> 192.168.1.7
06/11-21:41:36.130495 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.7 -> 192.168.1.2
06/11-21:41:37.133471 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.2 -> 192.168.1.7
06/11-21:41:37.133527 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.7 -> 192.168.1.2
06/11-21:41:38.134672 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.2 -> 192.168.1.7
06/11-21:41:38.134727 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.7 -> 192.168.1.2
06/11-21:41:39.157101 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.2 -> 192.168.1.7
06/11-21:41:39.157280 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.7 -> 192.168.1.2
06/11-21:41:40.206774 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.2 -> 192.168.1.7
06/11-21:41:40.206849 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.7 -> 192.168.1.2
06/11-21:41:41.138668 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.2 -> 192.168.1.7
06/11-21:41:41.138744 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.7 -> 192.168.1.2
06/11-21:41:42.191416 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.2 -> 192.168.1.7
06/11-21:41:42.191476 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.7 -> 192.168.1.2
06/11-21:41:43.144887 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.2 -> 192.168.1.7
06/11-21:41:43.144896 [**] [1:100000001:1] Tes ICMP PAKETTT [**] [Priority: 0] {ICMP} 192.168.1.7 -> 192.168.1.2
```

Snort NIDS-Kesimpulan Paket

Summary Packetnya

```
Parrot Terminal Parrot Terminal
Snort ran for 0 days 0 hours 0 minutes 32 seconds
Pkts/sec: 3
=====
Memory usage summary:
  Total non-mapped bytes (arena): 51761152
  Bytes in mapped regions (hblkhd): 22392832
  Total allocated space (uordblks): 45430272
  Total free space (fordblks): 6330880
  Topmost releasable block (keepcost): 133632
=====
Packet I/O Totals:
  Received: 101
  Analyzed: 100 ( 99.010%)
  Dropped: 0 ( 0.000%)
  Filtered: 0 ( 0.000%)
  Outstanding: 1 ( 0.990%)
  Injected: 0
=====
Breakdown by protocol (includes rebuilt packets):
  Eth: 100 (100.000%)
  VLAN: 0 ( 0.000%)
  IP4: 76 ( 76.000%)
  Frag: 0 ( 0.000%)
  ICMP: 47 ( 47.000%)
  UDP: 4 ( 4.000%)
  TCP: 22 ( 22.000%)
  IP6: 22 ( 22.000%)
  IP6 Ext: 44 ( 44.000%)
  IP6 Opts: 22 ( 22.000%)
  Frag6: 0 ( 0.000%)
  ICMP6: 22 ( 22.000%)
  UDP6: 0 ( 0.000%)
  TCP6: 0 ( 0.000%)
  Teredo: 0 ( 0.000%)
  ICMP-IP: 0 ( 0.000%)
  TD1/TD4: 0 ( 0.000%)
  =====
```

CVE

Common Vulnerabilities Exposures adalah metode refensi yang menyediakan kerentanan (vulnerability) dan beserta paparannya (exposure) tentang keamanan yang telah disebar dan diklasifikasikan ke publik. Singkatnya kerentanan yang sudah diketahui dan diklasifikasi berdasarkan nomor CVE. Tingkat kerentanan dapat dinamakan CVSS yang akan kita bahas di nextnya.

CVSS

Common Vulnerabilities Scoring System atau CVSS akan membantu kita dengan tingkat severity pada kerentanan dari 0.0-10.0 (Low-High). Berikut score CVSS dibawah yah biar gampang saya buatkan tabelnya.

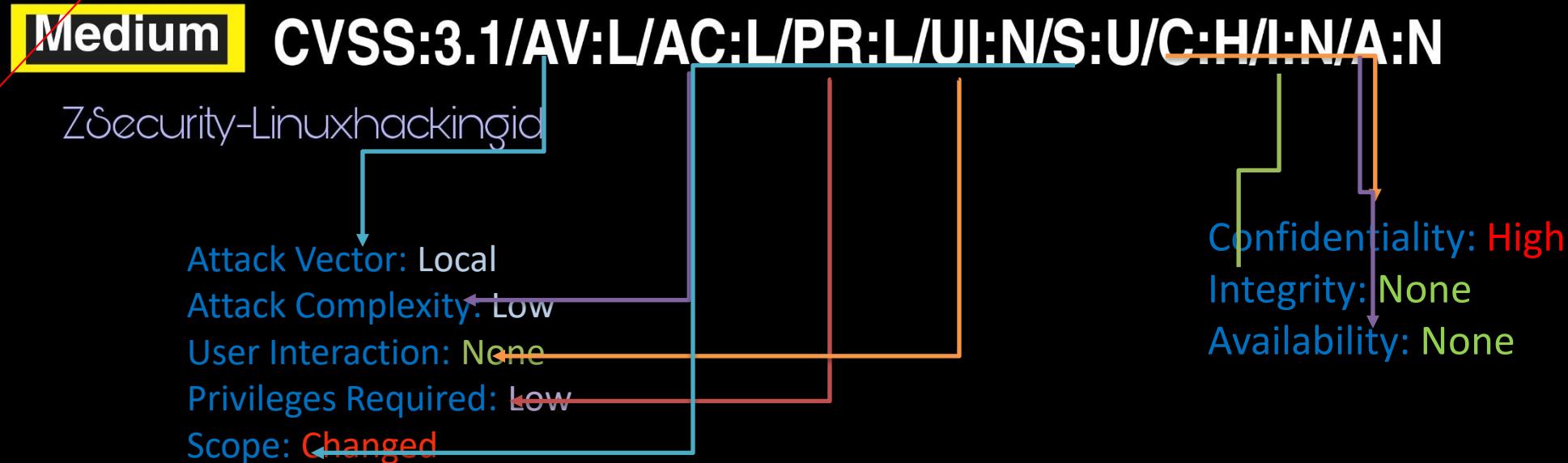
Severity	Score Range	Color pada umumnya
None	0.0	NONE
Low	0.1 - 3.9	
Medium	4.0 - 6.9	
High	7.0 - 8.9	
Critical	9.0 - 10.0	

CVSS-Metrics

- Base Metrics dibagi menjadi 2 grup:
 - Exploitability
 1. Attack Vector: bagaimana vuln agar bisa exploit?
 2. Attack Complexity: susah apa gampang untuk exploit vuln nya?
 3. Authentication: Authenticated?
 4. User Interaction: butuh bantuan user biar exploitnya berhasil?
 5. Privileges Required: butuh privilege tertentu agar exploitnya berhasil?
 - Impact Metrics
 1. Confidentiality: dampak Confidential pada saat data prosesing
 2. Integrity: dampak integrity pada system
 3. Availability: dampak ketersediaan bagi system ketika di exploit

CVSS-Example

CVE-2021-3505



PDF UPDATE

Thanks yang udah baca sampe akhir ini, semoga ilmunya bisa dimanfaatkan juga yahh. Terkait update nya pdf ini, saya ngga bisa jamin untuk V3 nya karena masing-masing penulis disini memiliki kesibukan masing-masing. Tapi bukan berarti ini mentok sampe v2 doang, klo ada waktu luang dan niat juga wkwkwk bisa sii update ke v4 nya.

<https://linuxhackingid.org>

<https://forum.linuxhackingid.org>

<https://t.me/linuxhackingid>

https://t.me/linuxhackingid_channel

Linuxhackingid**.org**

Article Creator



Zsecurity
Founder



Rakmen
Admin



Walfindo
Admin