Alvaro Austin - 2106752180

Persiapan Malware Analysis

Dari hal yang saya pelajari mengenai REMnux Distro, fungsi perangkat lunak yang terdapat pada distro tersebut adalah:

- *Examining static properties*: Alat ini berguna untuk memeriksa properti statis dari file baik itu berbahaya atau tidak. Proses ini melibatkan analisis struktur dan konten file untuk mengidentifikasi potensi bahaya.
    - Terdapat beberapa pembagian dalam pemeriksaan properti statis dari file. Pada REMnux, pembagian tersebut terdiri atas beberapa tipe file, dengan alat masing-masing yaitu:
        1. General Files: file-file yang umum ditemukan.
            a. TrID: *identify file using signature*
            b. Yara Rules: melakukan pemindaian terhadap file.
        2. Portable Executable (PE) Files: file-file *executable* yang biasa ditemukan pada OS Windows.
            a. Manalyze: melakukan analisa statik terhadap file yang berbahaya
            b. StringSifter: menambahkan peringkat terhadap *Windows executable* yang mencurigakan.
        3. Executable and Linkable Format (ELF) Files: file-file standar umum untuk file yang dapat dieksekusi, kode objek, pustaka bersama, dan dump yang biasa ditemukan pada OS Linux.
            a. pyelftools: library (package) yang digunakan dalam melakukan *parsing* dan analisa ELF file.
        4. .NET Files: file-file biner, rakitan, yang digunakan untuk sepenuhnya terdeskripsi dan berisi program .NET
            a. dnfile: menganalisa properti statik dari .NET file
            b. dotnetfile: sama seperti dnfile, menganalisa properti statik dari .NET file
        5. Deobfuscation Files: file-file yang merupakan cuplikan dari urutan fungsi bertingkat yang dipanggil dalam program hingga saat program tersebut berhenti.
            a. CyberChef: Membantu dalam *decoding* dan analisa data
            b. Malchive: Melakukan analisa statik terhadap beberapa aspek kode yang berbahaya.

- *Statically analyze code*: Alat ini berguna untuk menganalisis kode baik berbahaya atau tidak secara statis. Proses Ini melibatkan pemeriksaan kode file untuk memahami fungsinya dan mengidentifikasi komponen berbahaya.
    - Terdapat beberapa pembagian dalam analisa statis kode. Pada REMnux, pembagian tersebut terdiri atas beberapa tipe file, dengan alat masing-masing yaitu
        1. General Files: file-file yang umum ditemukan.
            a. BinNavi
            b. Ghidra.

2. Portable Executable (PE) Files: file-file *executable* yang biasa ditemukan pada OS Windows.
    a. Malchive
    b. Speakeasy
3. .NET Files: file-file biner, rakitan, yang digunakan untuk sepenuhnya terdeskripsi dan berisi program .NET
    a. de4dot
    b. ILSpy
4. Unpacking
    a. TrID
    b. Bytehist
5. Python
    a. Pyinstaller Extractor
    b. Decompyle++
6. Scripts
    a. ExtractScripts
    b. JS Beautifier
7. Java
    a. cfr
    b. JAD Java Decompiler
8. Flash
    a. Flare
    b. Flasm
9. Android
    a. JADX
    b. apktool

- *Dynamically reverse-engineer code*: Alat ini berguna dalam membuat kembali kode (*reverse engineering*) secara dinamis. Proses ini melibatkan eksekusi kode dalam suatu lingkungan yang terkendali untuk memahami interaksinya dengan komponen yang berkaitan.
    - Terdapat beberapa pembagian dalam *reverse-engineering* yang terdapat pada format file. Pada REMnux, pembagian tersebut terdiri atas beberapa tipe file, dengan alat masing-masing yaitu
        1. General
            a. Frida
            b. Wine
        2. Shellcode
            a. shcode2exe
            b. scdbg
        3. Scripts
            a. SpiderMonkey
            b. JStillery
        4. ELF Files

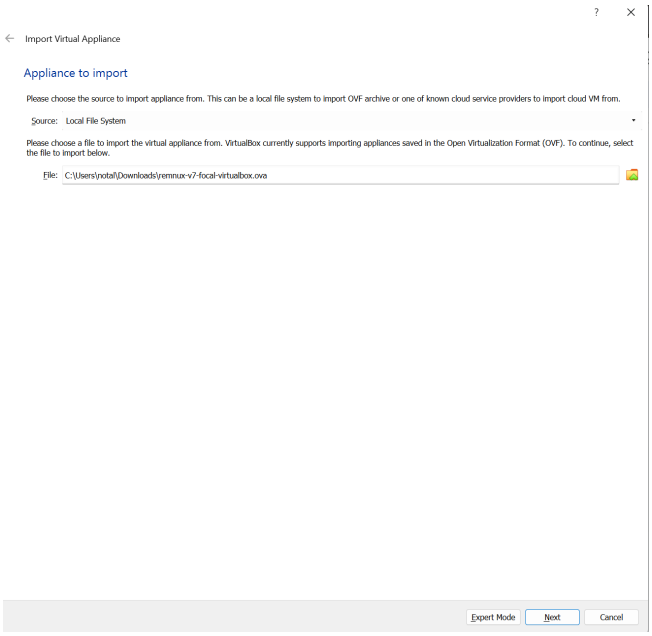
    a. GNU Project Debugger
    b. edb
- *Perform memory forensics*: Alat ini berguna dalam melakukan forensik memori biasanya pada sistem yang terinfeksi. Dimana proses ini melibatkan analisis memori sistem untuk mengidentifikasi aktivitas atau data berbahaya apapun.
    a. Volatility 3: untuk pengecekan memory image (memory dumps)
- *Exploring network interactions for behavioral analysis*: Alat ini berguna dalam menjelajahi interaksi jaringan untuk analisis perilaku. Proses ini melibatkan pemantauan *network traffic* untuk mengidentifikasi pola atau aktivitas tidak biasa yang dapat mengindikasikan infeksi *malware*.
    - Terdapat beberapa pembagian dalam *reverse-engineering* yang terdapat dalam tahap pemantauan jaringan. Pada REMnux, pembagian tersebut terdiri atas beberapa tahap, dengan alat masing-masing pada setiap tahap yaitu
        1. Monitoring
            a. Burp Suite Community Edition
            b. Network Miner Free Edition
        2. Connecting
            a. Unfurl
            b. thug
        3. Services
            a. fakedns
            b. fakemail
- *Investigate system interactions*: Alat ini berguna dalam menyelidiki interaksi malware di tingkat sistem. Proses Ini melibatkan analisis interaksi malware dengan sistem operasi sistem dan komponen lainnya untuk memahami perilaku dan dampaknya.
    a. Sysdig
    b. sandfly-processdecloak
- *Analyzing documents*: Alat ini berguna dalam menganalisis dokumen baik itu berbahaya maupun tidak. Proses Ini melibatkan pemeriksaan dokumen untuk mencari potensi ancaman atau konten berbahaya.
    - Terdapat beberapa . Pada REMnux, analisa dokumen dapat terbagi menjadi berbagai tipe dokumen dengan alat masing-masing pada setiap tipe yaitu
        1. General
            a. base64dump
            b. Tesseract OCR
        2. PDF
            a. peepdf
            b. pdfid.py
        3. Microsoft Office
            a. SSView
            b. EvilClippy
        4. Email Messages
            a. mail-parser
            b. msg-extractor

- *Gathering and analyzing data*: Alat ini berguna dalam mengumpulkan dan menganalisis data. Proses Ini melibatkan pengumpulan dan analisis data tentang potensi ancaman untuk mengidentifikasi pola, tren, dan potensi kerentanan.
    a. malwoverview
    b. VirusTotal API

- *View or edit files*: Alat ini berguna untuk melihat dan juga mengubah file-file. Proses ini melibatkan pemeriksaan file tersebut dengan mencari bagian yang penting pada file.
    a. dos2unix
    b. wxHexEditor

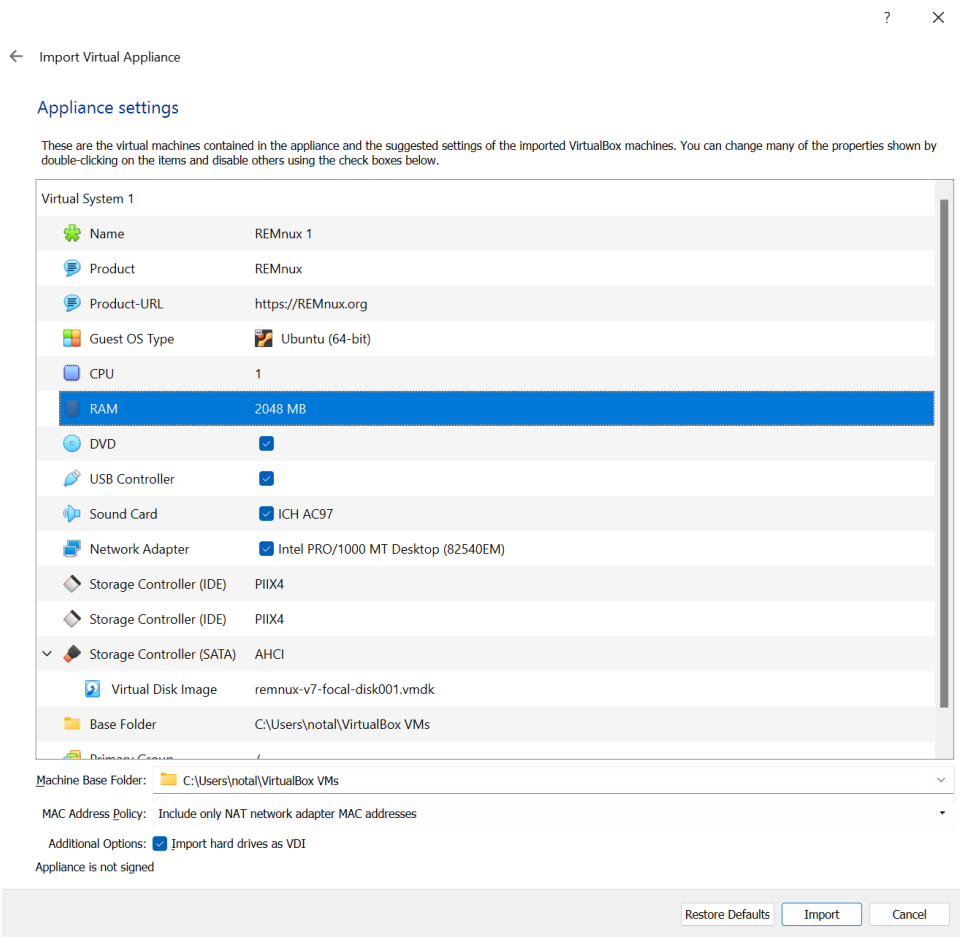
Screenshot Praktik:
Proses

- Installasi
  Pertama-tama, saya mengakses tab “Get the Virtual Appliance” dari dokumentasi remnux yang diberikan di scele (Get the Virtual Appliance - REMnux Documentation). Terdapat 2 pilihan yang dapat kita gunakan, General OVA atau VirtualBox OVA. Saya memilih VirtualBox OVA karena saya akan menggunakan Oracle VM Virtual Box Manager untuk melakukan proses penambahan *virutal appliance* pada aplikasi VM saya.
- Proses yang selanjutnya saya lakukan hanyalah membuka tab “Import Virtual Appliance” pada aplikasi Oracle VM Virtual Box dan referensikan file yang baru saja di download sebelumnya.

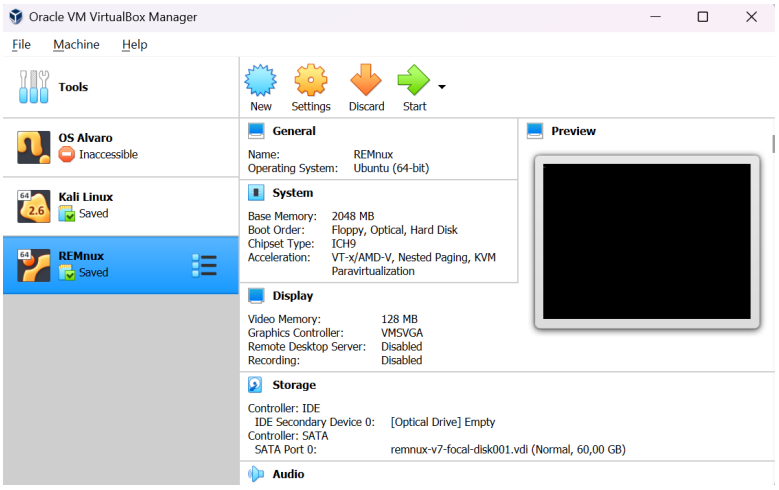


- Ubah RAM menjadi 2048 MB lalu import OVA tersebut.

Alvaro Austin - 2106752180

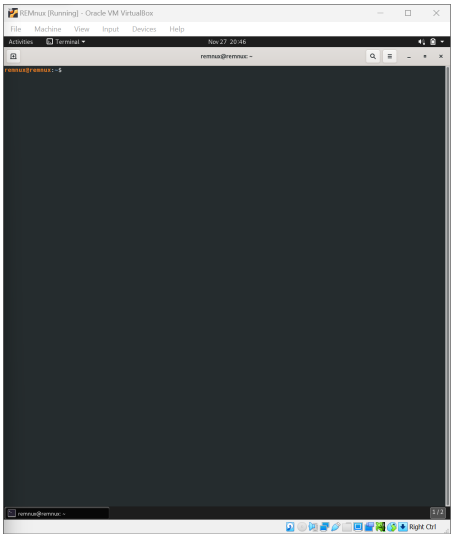


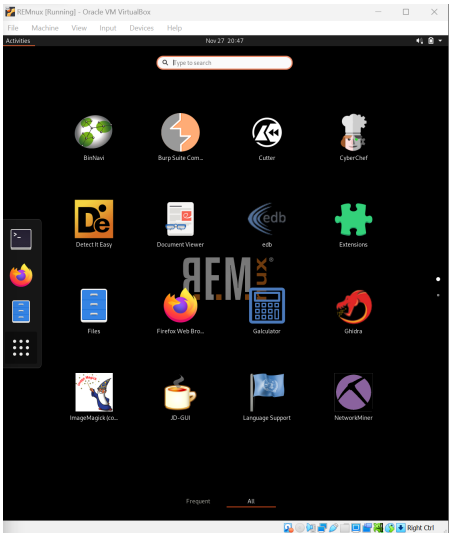
- Selanjutnya VM seharusnya telah ada pada daftar VM kita.



Hasil VM yang berhasil di import:

Alvaro Austin - 2106752180





Mencoba beberapa tools:

- Yara Rules: melakukan pemindaian terhadap file-file dengan menggunakan aturan yang dilakukan.

  Command: `yara <yara_rule_file>.yar /path/to/directory`

  Berikut adalah rule yang saya gunakan (berdasarkan dokumentasi dari Yara Rules (Welcome to YARA's documentation! — yara 4.4.0 documentation))

```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true
    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"
    condition:
        $a or $b or $c
}
```

  Dimana apabila sebuah direktori memiliki string a, b, c maka **silent_baker** akan di report. Pada tahap ini saya membuat yar_rule_files dengan nama test.yar, dan saya gunakan direktori dimana yar_rule_files itu berada dan didapatkan output:

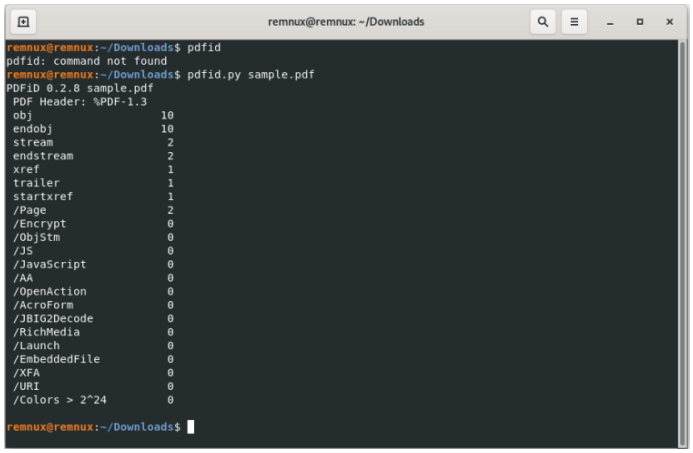
```
remnux@remnux:~/Downloads$ nano test.yar
remnux@remnux:~/Downloads$ ls -l
total 4
-rw-rw-r-- 1 remnux remnux 340 Nov 27 23:09 test.yar
remnux@remnux:~/Downloads$ dir
test.yar
remnux@remnux:~/Downloads$ pwd
/home/remnux/Downloads
remnux@remnux:~/Downloads$ ^C
remnux@remnux:~/Downloads$ ^C
remnux@remnux:~/Downloads$ ^C
remnux@remnux:~/Downloads$ ^C
remnux@remnux:~/Downloads$ yara test.yar /home/remnux/Downloads
silent_banker /home/remnux/Downloads/test.yar
remnux@remnux:~/Downloads$ ^Cdir

ir: command not found
remnux@remnux:~/Downloads$ dir
test.yar
remnux@remnux:~/Downloads$
```

Alvaro Austin - 2106752180

Dapat dilihat bahwa hasil yang ditemukan adalah file test.yar memiliki setidaknya string a, b, c pada namanya. Hal ini dapat dibuktikan karena test.yar memiliki string “a” pada ekstensi formatnya. Sehingga dengan menggunakan tool ini, kita dapat melakukan enforcing rule terhadap file-file yang ada. Apabila ada hal yang terjadi

- Menggunakan pdfid.py
  Alat ini merupakan alat yang melakukan pengecekan terhadap PDF kita (bukan melakukan *parsing* dengan PDF). Saya menggunakan PDF yang saya temukan online dimana terdiri atas 2 halaman *simple* (sample.pdf (africau.edu)).



  Berdasarkan hasil tools diatas, tidak ditemukan element dengan elemen javascript didalamnya. Hal ini karena atribut dari /JS, sampai /XFA menghasilkan nilai 0 sehingga dokumen ini bukanlah dokumen berbahaya.

- wxHexEditor: Tool ini saya gunakan untuk melakukan pengeditan terhadap hex code yang ada dalam suatu file