KODE DAN AKTIVITAS MENCURIGAKAN

9.1. DASAR TEORI

Malware merupakan singkatan dari *Malicious Software*, secara umum program ini sengaja dibuat dengan tujuan merusak, mengganggu, atau mencuri informasi dari sistem komputer atau perangkat lainnya.

Jenis-jenis malware:

- Virus: Merupakan jenis malware yang menggandakan dirinya sendiri dan menyebar ke file atau program lainnya. Virus dapat merusak sistem atau file yang terinfeksi.
- Worm: Mirip dengan virus, worm juga dapat menggandakan dirinya sendiri. Namun, worm tidak memerlukan file induk dan dapat menyebar melalui jaringan komputer.
- Trojan: Trojan horse adalah perangkat lunak yang menyamar sebagai program yang berguna atau sah tetapi sebenarnya memiliki fungsi jahat. Biasanya, trojan digunakan untuk mencuri informasi sensitif atau memberikan akses ke sistem komputer kepada penyerang.
- Ransomware: Ransomware adalah jenis malware yang mengenkripsi data pengguna dan menuntut pembayaran tebusan untuk mendapatkan kunci dekripsi. Ini berarti pengguna tidak dapat mengakses data mereka kecuali mereka membayar tebusan kepada penyerang.
- Spyware: Spyware dirancang untuk mengumpulkan informasi tentang pengguna tanpa sepengetahuan mereka. Biasanya, spyware mengumpulkan data pribadi, seperti riwayat penjelajahan, kata sandi, atau informasi keuangan, dan mengirimkannya ke pihak yang tidak berwenang.
-

Metode Penyebaran yang sering digunakan, antara lain:

- Email: Malware dapat menyebar melalui lampiran berbahaya dalam email yang dikirim ke pengguna. Lampiran tersebut dapat meminta pengguna untuk mengklik atau membuka file untuk mengaktifkan malware.
- Situs web berbahaya: Penyebaran malware juga dapat terjadi melalui situs web yang terinfeksi atau situs web palsu yang dirancang untuk menipu pengguna agar mengunduh dan menginstal malware.
- Perangkat bergerak: Malware juga dapat menyebar melalui perangkat bergerak, seperti ponsel cerdas atau tablet, melalui aplikasi atau tautan yang meragukan.
- Jaringan: Beberapa jenis malware, seperti worm, dapat menyebar melalui jaringan komputer dan memanfaatkan kerentanan yang tidak terpatch pada sistem yang terhubung.

Reverse Engineering (RE) atau biasa dikenal sebagai back engineering, adalah proses dimana objek buatan manusia di dekonstruksi ulang untuk menampilkan desain, arsitektur, kode, atau mendapatkan informasi dari objek tersebut. Jika ingin dibahas secara spesifik, Reverse Engineering dalam dunia security adalah proses mengembalikan suatu program yang telah dicompile (machine code ataupun bytecode) dan mengembalikannya menjadi format yang lebih bisa dibaca oleh manusia dengan tujuan tertentu. Reverse Engineering bisa dikatakan sebagai seni membongkar software maupun hardware.

Proses reverse engineering perlu menggunakan bantuan alat/tools. Tools ini bervariasi dan dapat kita kategorikan dari debugger, disassembler, hingga decompiler yang dijelaskan lebih lanjut sebagai berikut:

➢ Debugger
Debugger merupakan tools yang membantu untuk menganalisa workflow dari program yang kita reverse. Perlu diingat bahwa debugger pada reverse engineering memiliki kemampuan membaca stack frame dan heap frame, mapping library, dan linking sehingga lebih efektif dibanding debug dengan IDE.

➢ Disassembler
Disassembler adalah tools yang membantu mengembalikan bahasa mesin ke bahasa assembly — kebalikan dari assembler.

➢ Decompiling
Decompiling adalah tools yang membantu mengembalikan bahasa mesin ke bahasa semula (tidak sempurna).

Sayangnya, tidak semua program dan bahasa pemrograman dapat di-decompile. Proses decompile yang mengembalikan bahasa tingkat mesin menjadi bahasa tingkat tinggi menyebabkan adanya kekeliruan yang sering terjadi. Hal ini dapat berujung pada kesalahan algoritma dan nilai memori saat dianalisis. Debugger digunakan untuk analisis dinamis yang akan dijelaskan lebih lanjut. Oleh sebab itu, para Reverse Engineer — sebutan bagi para profesional dibidang ini — mengkombinasikan ketiga tools tersebut.
Reverse Engineering dibagi menjadi dua, yaitu Static reverse engineering dan Dynamic reverse engineering. Static reverse engineering adalah kondisi dimana seorang Reverse Engineer membaca hasil decompile atau disassembly secara langsung, memahami algoritmanya, dan membuat pseudocode dari hasil reverse engineering tersebut. Dynamic reverse engineering adalah proses reverse engineering yang menganalisis cara kerja program dengan menjalankan program aplikasi dan menggunakan debugger.
Aplikasi aplikasi yang membutuhkan virtual machine seperti java virtual machine meng-compile kode menjadi intermediate code. Oleh sebab itulah aplikasi seperti java atau android lebih mudah untuk dianalisis saat proses RE

9.2. TUJUAN PERCOBAAN

1. Memahami cara kerja suatu malware sederhana
2. Memahami konsep Reverse Engineering
3. Menerapkan Implementasi Reverse Engineering dalam analisa Malware

9.3. ALAT DAN BAHAN

Dalam memenuhi pelaksanaan langkah-langkah praktikum yang terdapat pada bab ini, peserta praktikum perlu menyiapkan beberapa perangkat berikut:

- Perangkat komputer Desktop/Laptop dengan kapasitas RAM minimum 8 GB
- Perangkat lunak virtualisasi (Oracle VM VirtualBox/VMWare Workstation)
- Koneksi internet
- Beberapa tools yang dibutuhkan dalam praktikum

9.4. PROSEDUR PERCOBAAN

Disclaimer

Resources atau sumber yang digunakan pada praktikum ini, mohon digunakan secara bijaksana dan hati-hati, Penyalahgunaan maupun kesalahan yang terjadi akibat resource tersebut menjadi **tanggung jawab pribadi**.

Eksekusi Malware

1. Jalankan sistem operasi Linux (Desktop/Server) pada aplikasi Virtual Machine (VM),

2. Jalankan command berikut pada VM Terminal Anda:

```
sudo apt update && sudo apt install jd-gui # Debian
paru -Sy --aur --noconfirm jd-gui-bin # ArchLinux
```

Note: jika tidak memiliki perintah `paru` didalam sistem, install menggunakan langkah berikut: https://github.com/Morganamilo/paru#installation

3. Kemudian buatlah folder baru:

```
mkdir ~/victim_NIManda
```

4. Berikutnya, unduh file yang kita butuhkan, dan masukkan ke dalam folder yang telah kita buat sebelumnya, tautan *resource*, (**unduh file MyApp.jar**): https://drive.google.com/drive/u/4/folders/1fcDWz_HVJpQvRnTWRGsDgF-egtlrOx4C

5. Masuk ke laman virustotal.com, kemudian upload file MyApp.jar ke dalam laman tersebut. **Jelaskan hasil yang diberikan dari virustotal tersebut**.

Berikan screenshot dan Penjelasan

6. Kemudian, buka terminal anda dan masuk ke dalam path direktori tempat file MyApp.jar

```
cd ~/victim_NIManda
```

7. Tambahkan beberapa file (bebas) ke dalam folder tersebut.
8. Jalankan file MyApp.jar dengan perintah berikut:

```
java -jar MyApp.jar
```

Jelaskan apa yang terjadi pada file lainnya setelah program tersebut dijalankan.

Berikan screenshot dan Penjelasan

Analisa Malware

9. Berikutnya, kita jalankan decompiler tools yang telah kita install (jd-gui) melalui terminal dengan perintah:

```
jd-gui
```

10. Setelah jd-gui berhasil dijalankan masuk ke bagian file -> open file, kemudian pilih file MyApp.jar

11. Kemudian, kita lakukan analisa melalui jd-gui tersebut.
File .class apa saja yang terdapat dalam file malware tersebut ?

Berikan screenshot dan Penjelasan

12. Apa yang dilakukan FIleUtility.class dalam file malware tersebut?

Berikan Penjelasan

13. Apa yang dilakukan AES.class dalam file malware tersebut, algoritma kriptografi apa saja yang digunakan dalam malware tersebut?

Berikan Penjelasan

14. Apa yang dilakukan RansomwareKI.class dalam file malware tersebut?

Berikan Penjelasan

Mitigasi dan Pemulihan dari Malware

15. Setelah kita melakukan analisa pada malware tersebut, kita dapat melakukan pemulihan kembali pada file kita yang terenkripsi.

16. **Unduh file 'Very Important Document.pdf.dokb'** dan masukkan ke dalam folder ~/victim_NIManda dari tautan Drive sebelumnya pada nomor 4.

17. Buatlah folder dengan nama 'result' di dalam folder ~/victim_NIManda.

18. Jalankan script code yang dapat membantu kita memulihkan salah satu file penting berjudul "Very Important Document.pdf.dokb" kembali menjadi file .pdf

script.py

```python
from hashlib import sha1 # Import library kriptografi sha1
from Crypto.Cipher import AES # Import library kriptografi AES
import string # Bantuan lib string untuk import lowercase text

# Mengakses/membuka file yang terenkripsi
encryptedfile = open('Very Important Document.pdf.dokb', 'rb').read()

for i in string.ascii_lowercase:
  # Generate kunci dari tiap karakter a-z
  key = i * 16 # Pada tiap karakter akan digandakan sebanyak 16 misal 'aaaaaaaaaaaaaaaa'
  key = sha1(key.encode()).digest()[:16] # Mengambil 16 bytes pertama dari hasil SHA-1 digest bytes 0 sampai 15

  aes = AES.new(key, AES.MODE_ECB) # Membuat AES cipher dari key yang didapat dari SHA-1 digest sebelumnya
menggunakan mode ECB
  result = aes.decrypt(encryptedfile) # Melakukan dekripsi file dengan algoritma kriptografi AES yang telah
didefinisikan sebelumnya

  # Pastikan terdapat direktori result
  # Write file baru hasil proses dekripsi, seharusnya ada 26 file baru dan hanya ada 1 file yang dapat diakses.
  open(f'result/Very Important Document_Char_{i}.pdf', 'wb').write(result)
```

19. Jalankan kode yang telah anda buat, dan bukalah dokumen yang berhasil dipulihkan.

Berikan Screenshot
Flag:

Dalam file tersebut, anda akan menemukan sebuah password juga. Password ini dapat kalian gunakan untuk menjalankan *decryptor* yang tersedia pada drive google yang ada pada nomor 5, apabila terjadi hal yang tidak diinginkan akibat malware tersebut.

9.5. EVALUASI

1. Malware jenis apa yang kita jalankan pada praktikum ini, dan jelaskan secara singkat bagaimana proses malware tersebut bekerja.

2. Jelaskan bagaimana cara kerja kode script yang ada pada nomor 18, dalam men-dekripsi dan memulihkan file dari malware tersebut.

3. Hal-hal apa saja yang kita perlu lakukan agar terhindar dari serangan malware