VOLATILITY
SHYLOCK

ANGGOTA KELOMPOK:

2501963822 - JOSUA ABRAHAM

2501991536 - VAUSTIN

2502001441 - AYUBI SHOLAHUDIN

2502009412 - RAMADHANA KHALAF SANDHYAKALA

2502018480 - CARLSON KING

MEMORY ACQUISITION

Untuk mendapatkan memory dari malware, harus dilakukan memory dump dengan mengambil memory file dari malware atau menggunakan tools seperti win32dd atau memoryze.

IMAGE ANALYSIS

Setelah mendapatkan memory dump, kita bisa menganalisis memory dump dengan script python:

python vol.py imageinfo -f infected.dmp

imageinfo digunakan untuk mengidentifikasi profile OS dari hasil memory dump malware dan -f digunakan untuk menentukan file dump yang ingin dilihat.

Dari hasil analisis, diketahui bahwa file digunakan pada OS Windows XP dalam environment X86.

PROCESS ANALYSIS

Process enumeration dilakukan menggunakan pslist dengan cara:

python vol.py pslist -f infected.dmp

pslist digunakan untuk memberikan list beberapa proses sebuah sistem.

Setelah menjalankan script tersebut, tidak ditemukan process artifacts dari hasil dump malware.

NETWORK ANALYSIS

Network activity dapat dilihat dengan plugin conscan seperti:

python vol.py connscan -f infected.dmp

connscan digunakan karena merupakan plugin yang bergantung pada OS Windows XP.

Diketahui bahwa proses explorer.exe mengeluarkan network traffic dengan IP 209.190.4.82, dimana IP tersebut ditemukan sebagai IP yang berbahaya. Selain itu, ada IP 184.173.252.203 yang juga ditemukan sebagai IP berbahaya dengan PID 1336, yang menandakan bahwa Firefox terpengaruh dengan aktivitas berbahaya.

TIMELINE ANALYSIS

Timeline analysis dapat dilakukan dengan menggunakan plugin sockets seperti:

python vol.py sockets -f infected.dmp

sockets digunakan untuk menampilkan daftar socket apa saja yang terbuka.

Dari hasil command tersebut, ditemukan bahwa socket explorer.exe tampil pada waktu 11:58:04.

Pada proses-proses berikutnya, exe berbahaya melakukan injeksi code ke dalam proses browser.

DLL INSPECTION

Melakukan enumeration pada DLL yang sudah diload oleh sebuah proses dapat dilihat dengan:

oython vol.py dlllist -f infected.dmp -p 1440

dlllist digunakan untuk menampilkan DLL yang sudah diload oleh sebuah proses.

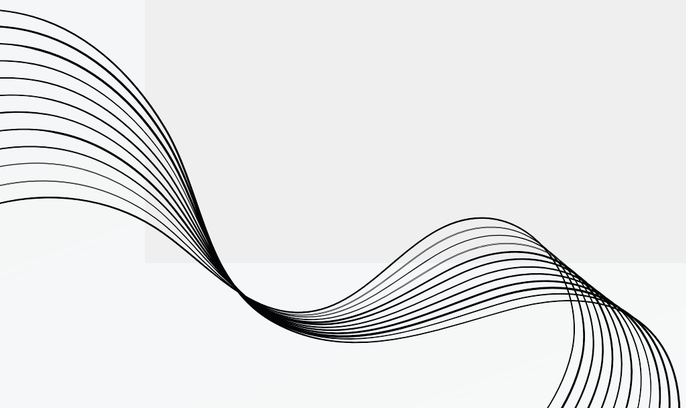
Dalam daftar tersebut, ada beberapa dll yang mencurigakan seperti WINHTTP.dll, DNSAPI.dll, cryptnet.dll, hnetcfg.dll, dan wshtcpip.dll, karena explorer.exe biasanya tidak membutuhkan module networking.

FINDING INJECTED CODE

Setelah mendapatkan list .dll yang berada di dalam explorer sekarang kita bisa mencari injected code dengan cara memakai salah satu plugin Volatility yaitu, malfind.

python vol.py malfind -f infected.dmp -p 1440 -D /home/.../injected_code/

Untuk menganalisis memory dump file, fokus pada suatu ID proses tertentu (PID 1440), dan mengekstrak code yang mungkin berbahaya.



FINDING INJECTED CODE

Lalu kita mendapatkan 2 executables.

file explorer.exe.9d8bda0.01820000-018abfff.dmp
file explorer.exe.9d8bda0.01d30000-01dbbfff.dmp

Kemudian kita coba md5sum untuk calculate dan verify si MD5 hash dari file tersebut.

explorer.exe.9d8bda0.01820000-018abfff.dmp
2eb2380efe9c3a5db32a9adba55834b9

explorer.exe.9d8bda0.01d30000-01dbbfff.dmp
b29d99b940b8a62464032ddbf395f0d8

VAD OVERVIEW

vadinfo plugin digunakan untuk mendapatkan informasi VAD dari suatu memory. Pengunaan plugin tersebut adalah:

python vol.py vadinfo -f infected.dmp -p 1440

Hasil dari command tersebut menampilkan dua memory dump berbeda yang merupakan dua sampel berbeda dari Shylock.

REVEALING HOOKS

Plugin apihooks digunakan untuk mendeteksi hook yang telah dilakukan.

Pengunaan apihooks pada beberapa sampel adalah:
Sampel pertama:
python vol.py apihooks -f infected.dmp -p 1440

Sampel kedua:
python vol.py apihooks -f 2_infected.dmp -p 1480

Firefox hooks:
python vol.py apihooks -f 2_infected.dmp -p 1156

Setelah mengekseskusi ketiga command tersebut, ditemukan ada dll dengan nama hijackdll.dll pada memory Firefox dan diketahui bahwa code berbahaya diinjeksi saat user sedang mengakses bank.

REGISTRY ANALYSIS

Analisis registry dilakukan pada dua sample malware shylock dengan command:

Sample pertama: python vol.py printkey -f infected.dmp -K “Software\Microsoft\Windows\CurrentVersion\Run”

Sample kedua: python vol.py printkey -f 2_infected.dmp -K “Software\Microsoft\Windows\CurrentVersion\Run”

command -K digunakan untuk menunjukkan process pada virtual memory pada directory tertentu.

Setelah memasukkan kedua command tersebut, ada element yaitu “{A3C43CDE-3A97-74B5-88A7-522DC42C016E}” yang merupakan element yang umum dan ditentukan sebagai Shylock.

ANALYSIS MENGGUNAKAN TIMELINER

Timeliner digunakan untuk membuat timeline mengenai artifacts yang ditemukan dalam memory, dan dapat digunakan dengan command:

python vol.py timeliner -f infected.dmp --output-file= shylock_timeline.xls

--output-file digunakan untuk menentukan file yang akan menampung output dari command tersebut.

Dari hasil command tersebut, terlihat ada relasi direct time dengan aktivitas berbahaya yang sebelumnya beserta informasi yang mungkin berhubungan dengan TID.

VOLATILITY 2
VS
VOLATILITY 3

PERBEDAAN

- Volatility 2 dibuat dengan Python 2, sedangkan Volatility 3 dibuat dengan Python 3.
- Volatility 3 memiliki arsitektur plugin yang modular dan dapat diperpanjang dibanding Volatility 2 yang menggunakan arsitektur plugin monolithic yang lebih kompleks.
- Volatility 2 memiliki kompaktibilitas yang dapat digunakan pada hampir semua sistem, sedangkan Volatility 3 tidak dapat digunakan pada beberapa sistem karena versinya yang baru.

TERIMA KASIH