DevSecOps 클라우드 보안 전문가

보안 사고 분석 대응

작성자: 정재호

작성일: 2024.11.14

0. 목차

[1. 서술형 평가 3](#_Toc182498818)

[1.1. /var/log: wtmp, utmp, btmp, lastlog 파일에 대한 설명 3](#_Toc182498819)

[1.2. 악성코드의 정의 4](#_Toc182498820)

[1.3. 침해사고 분석 시 수집해야 될 아티팩트(Artifact)에 대한 설명 5](#_Toc182498821)

[2. 실습 과제 8](#_Toc182498822)

[2.1. Log 서버 구축 8](#_Toc182498823)

[2.2. WAF(Web Application Firewall) 구축 11](#_Toc182498824)

[2.3. SIEM 솔루션 구축 12](#_Toc182498825)

[2.4. Flare VM 구축 16](#_Toc182498826)

[2.5. Lab01-01.exe 분석 17](#_Toc182498827)

[2.6. 악성 코드 2종 분석 21](#_Toc182498828)

[3. 문제 해결 과제 24](#_Toc182498829)

[3.1. 웹 서버 취약점 공략 후 IDS 탐지 24](#_Toc182498830)

[3.2. 침해 사고 증거 수집 및 분석 24](#_Toc182498831)

[3.3. 침해사고 분석 보고서 24](#_Toc182498832)

# 1. 서술형 평가

## 1.1. /var/log: wtmp, utmp, btmp, lastlog 파일에 대한 설명

### 1) wtmp

- 시스템에 로그인/로그아웃한 모든 사용자와 시스템 재부팅 등의 기록을 저장

- last 명령어로 wtmp 파일 내용을 확인할 수 있다.

### 2) utmp

- 현재 시스템에 로그인되어 있는 사용자 세션 정보를 저장

- who 또는 w 명령어를 통해 현재 로그인된 사용자 정보를 확인할 수 있다.

### 3) btmp

- 실패한 로그인 시도를 기록, 비정상적인 로그인 시도나 brute-force 공격 등을 확인할 사용한다.

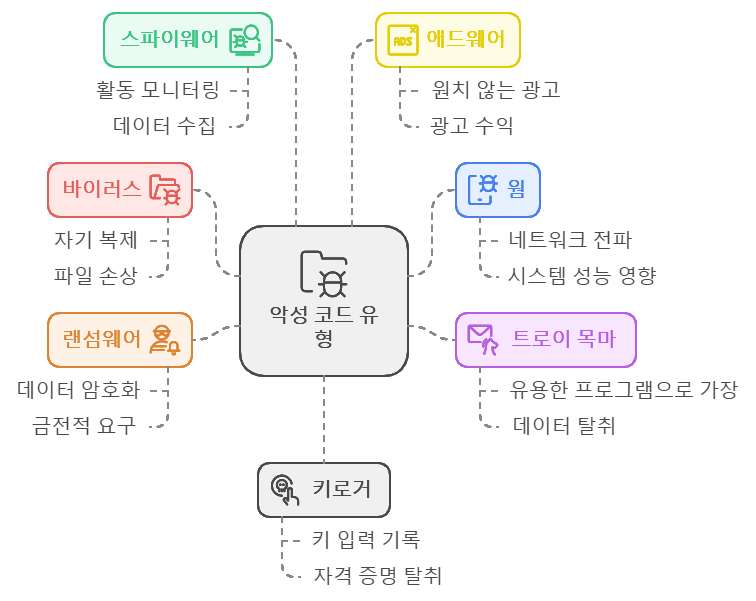
- lastb 명령어로 파일 내용을 확인할 수 있다.

### 4) lastlog

- 모든 시스템 사용자 계정의 마지막 로그인 시간을 기록한다.

- lastlog 명령어로 각 사용자의 마지막 로그인 시간을 조회할 수 있다.

## 1.2. 악성코드의 정의



- 시스템, 네트워크, 프로그램, 데이터 등에 악의적인 피해를 가하기 위해 고의로 제작된 소프트웨어

- 주로 시스템을 손상시키거나 데이터를 탈취하고, 사용자의 권한을 탈취하는 목적으로 제작된다.

## 1.3. 침해사고 분석 시 수집해야 될 아티팩트(Artifact)에 대한 설명

### 1) 가공되지 않은 데이터 자체

**- 로그 파일(Log Files)**

시스템 및 애플리케이션에서 발생한 이벤트를 기록한 파일

운영체제의 보안 로그, 웹 서버의 접속 로그, 방화벽 로그 등 해당

**- 네트워크 트래픽 캡쳐(Network Traffic Capture)**

네트워크 상 모든 데이터를 캡처하여 분석하는 파일

Wireshark와 같은 패킷 캡쳐 도구를 사용하여 수집할 수 있다.

공격자가 외부와 통신한 내역, 비정상적인 트래픽을 식별하는 데 사용

**- 메모리 덤프(Memory Dump)**

RAM에 저장된 데이터를 그대로 ‘덤프’하여 저장한 파일

프로그램의 실행 정보, 로그인 세션, 암호화되지 않은 데이터 등이 포함될 수 있다.

공격자가 메모리에 남긴 흔적을 분석할 수 있다.

**- 시스템 이미지(System Image)**

시스템 전체의 디스크 이미지를 생성한 것

파일 시스템을 포함한 모든 데이터가 그대로 유지

공격자가 남긴 파일, 설정 변경 사하 등을 분석> 침해사고 전후 관계를 확인하는 데 사용

### 2) 주요 아티팩트 목록

**- Process**

시스템에서 실행 중인 프로세스 목록과 그 세부 정보

특정 악성 코드나 비정상적인 프로세스가 실행 중인지 확인하는 데 중요한 단서가 된다.

pslist, tasklist 같은 명령어를 통해 확인

**- Event Log**

운영체제가 발생하는 다양한 이벤트를 기록한 로그

각 이벤트는 시간, 사용자, 이벤트 코드 등을 포함

사용자 로그인 기록, 파일 접근, 시스템 오류 등을 통해 공격자가 수행한 행동을 추적할 수 있다.

**- MFT (Master File Table)**

파일 메타데이터를 관리하는 테이블

파일 생성/수정/삭제 시간, 파일 위치 등의 정보를 포함

삭제된 파일의 흔적을 추적, 공격자가 사용한 파일의 시간 정보를 확인

침해사고 발생 시간을 파악하는데 사용

**- Prefetch**

운영체제에서 자주 사용되는 프로그램 실행 정보를 저장

프로그램 실행 속도를 높이는 데 사용

특정 프로그램이 언제 실행되었는지 알 수 있다.

악성 프로그램이나 비정상적인 실행 기록을 확인하는 데 사용

**- Suspicious File**

악성코드나 이상행동을 유발한 것으로 의심되는 파일을 의미

분석을 통해 악성 여부를 확인, 공격자의 툴 또는 백도어 파일을 식별하는 데 사용

**- Memory Dump**

메모리에 남아 있는 데이터나 프로세스 정보를 그대로 덤프하여 저장한 파일

휘발성 데이터(암호, 세션 정보, 악성코드 실행 흔적 등) 확보할 수 있다.

악성코드의 활동 형태를 추적할 수 있다.

**- Web Browser (Cookie, History, Cache 등)**

사용자가 웹 브라우저에서 방문한 사이트, 저장된 쿠키, 캐시 파일 등의 정보를 수집

공격자가 브라우저를 통해 접근한 경우, 관련 세션 정보나 브라우저 캐시에 남아 있는 데이터를 통해 경격자의 흔적을 찾을 수 있다.

**- Persistence**

시스템 재부팅 후에도 자동 실행되도록 설정된 영구적 유지 방식

공격자는 시작 프로그램, 서비스 등록, 레지스트리 키 등을 수정하여 지속적으로 접근 권한을 유지

공격자가 설정한 유지 방법을 파악하고 제거할 수 있다.

**- Registry**

Windows 운영체제에서 설정 정보가 저장되는 데이터베이스

시스템 설정, 사용자 정보, 프로그램 설치 경로 등을 저장

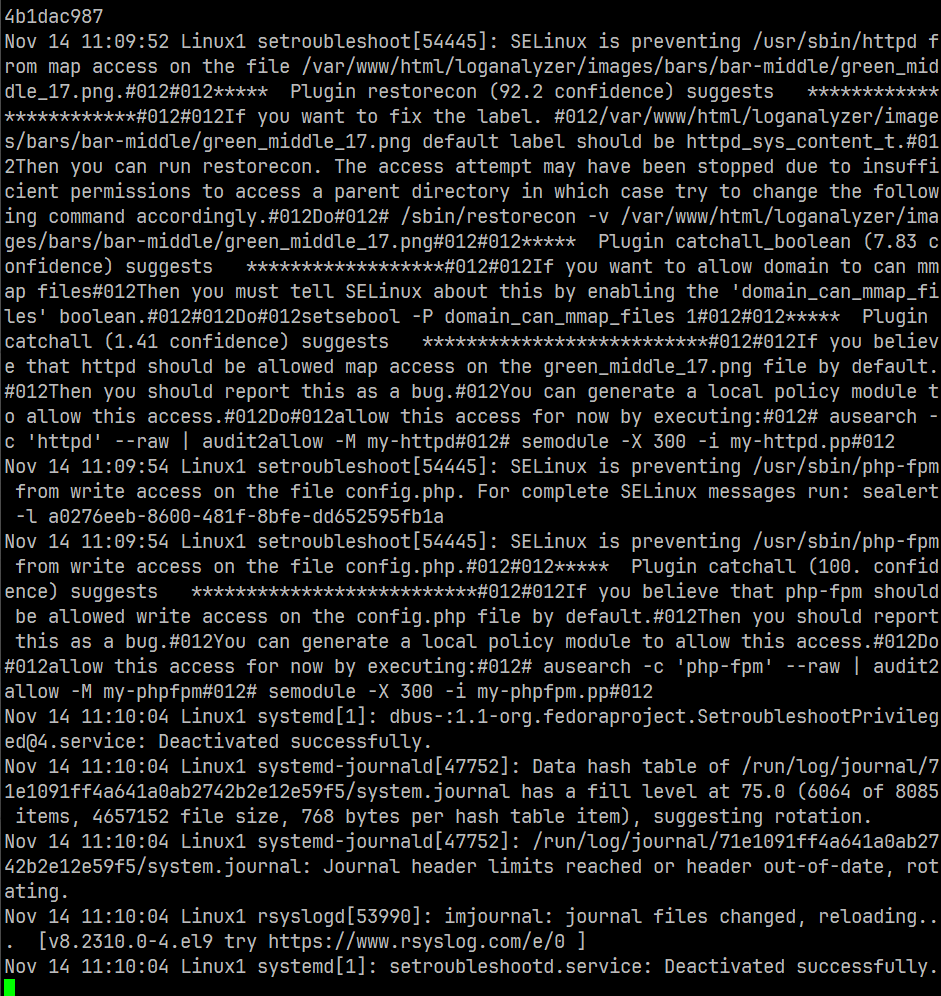
레지스트리에 추가된 의심스러운 키를 통해 공격자가 설정한 백도어나 자동 실행 프로그램 등을 파악할 수 있다.

# 2. 실습 과제

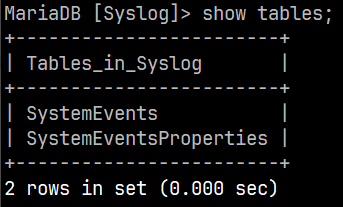
## 2.1. Log 서버 구축

### 1) Log Server/Log Database 구축

**- ‘tail -f /var/log/messages’**

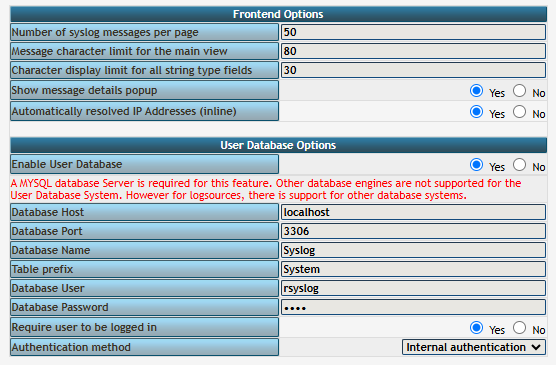


**- Log database/Log Tables**

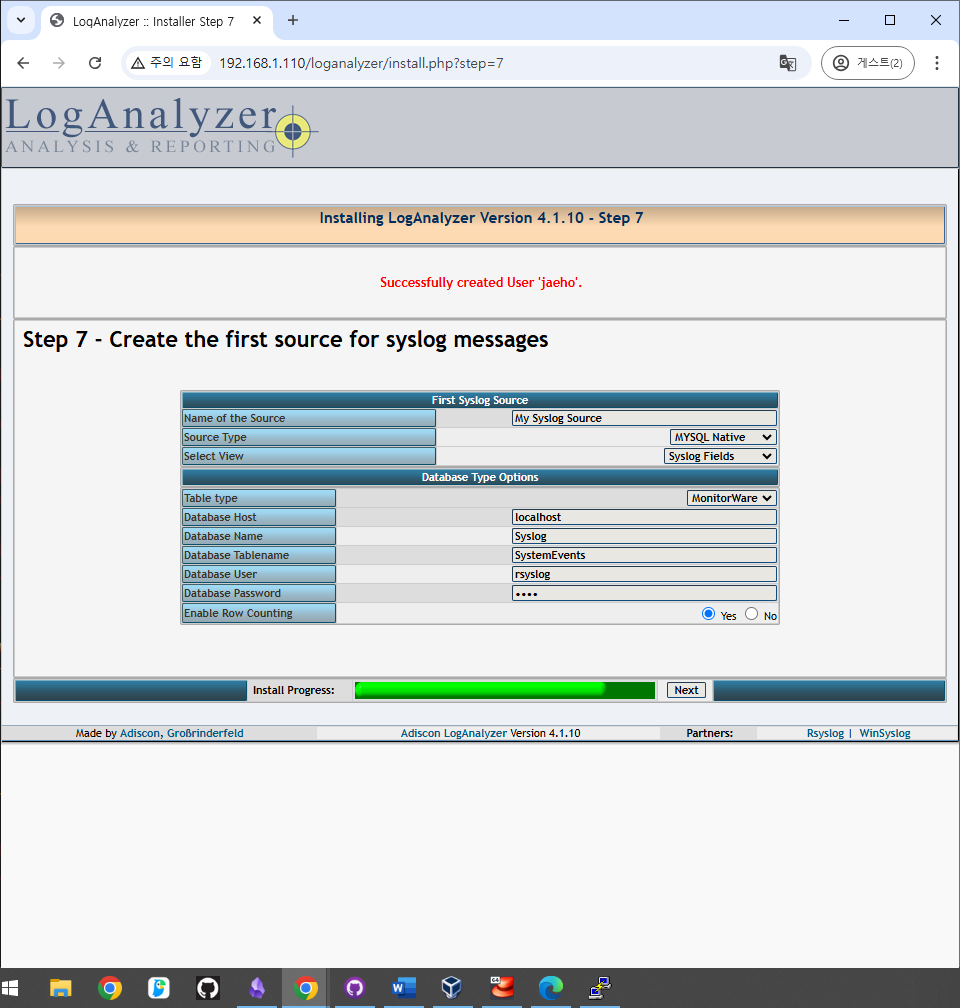


### 2) Loganalyser Web

**- Loganalyser Web Setup**



**- Loganalyser Web**

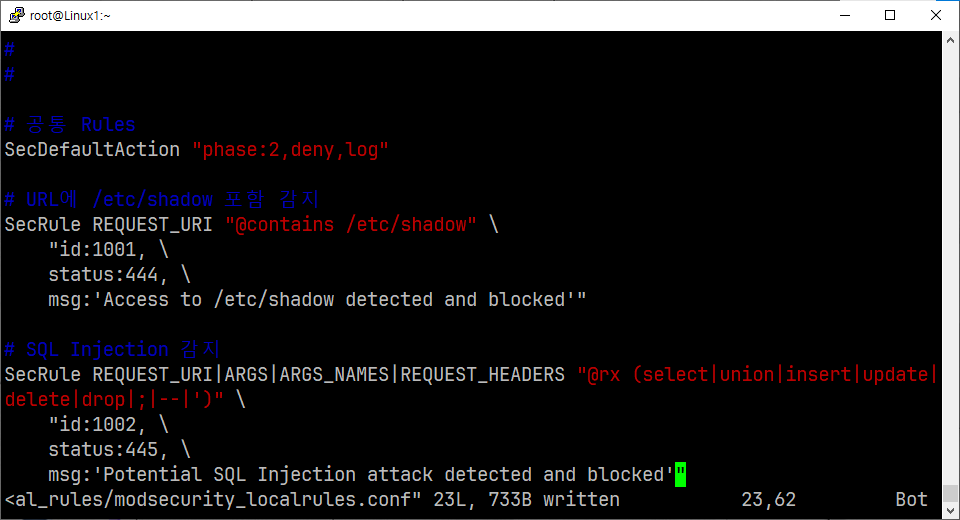


정석 대로라면 set 7 설정 후 index.php 접속이 가능해야 하는데 ‘HTTP ERROR 500’이 발생하며 접속이 이뤄지지 않았다.

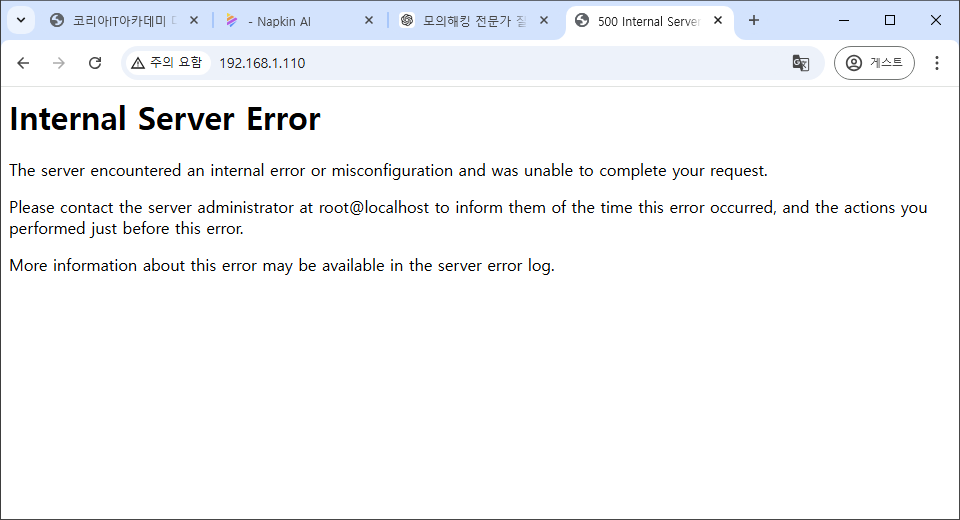
원인을 분석하고 해결하기에 시간이 부족함으로 모든 문제를 해결한 후 다시 진행하는 것으로 결정

## 2.2. WAF(Web Application Firewall) 구축

### 1) WAF 모듈 설치 및 Rule 작성



### 2) 웹 테스트



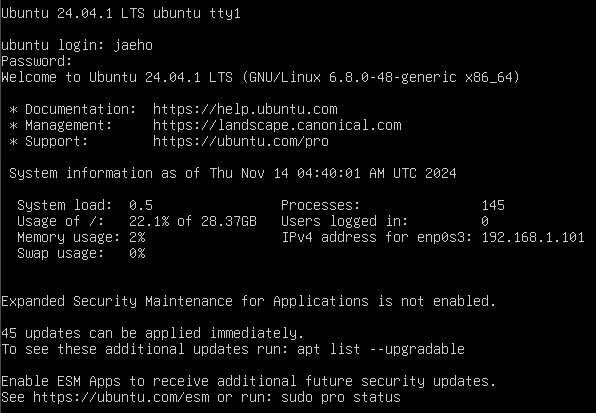
WAF Rule을 설정하고 웹에 접속하면 서버 에러가 발생한다.

예상 가능한 원인(Module 충돌, Rule 문법 오류 등)을 해결하고 웹 서비스가 정상적으로 동작하도록 수정하자

## 2.3. SIEM 솔루션 구축

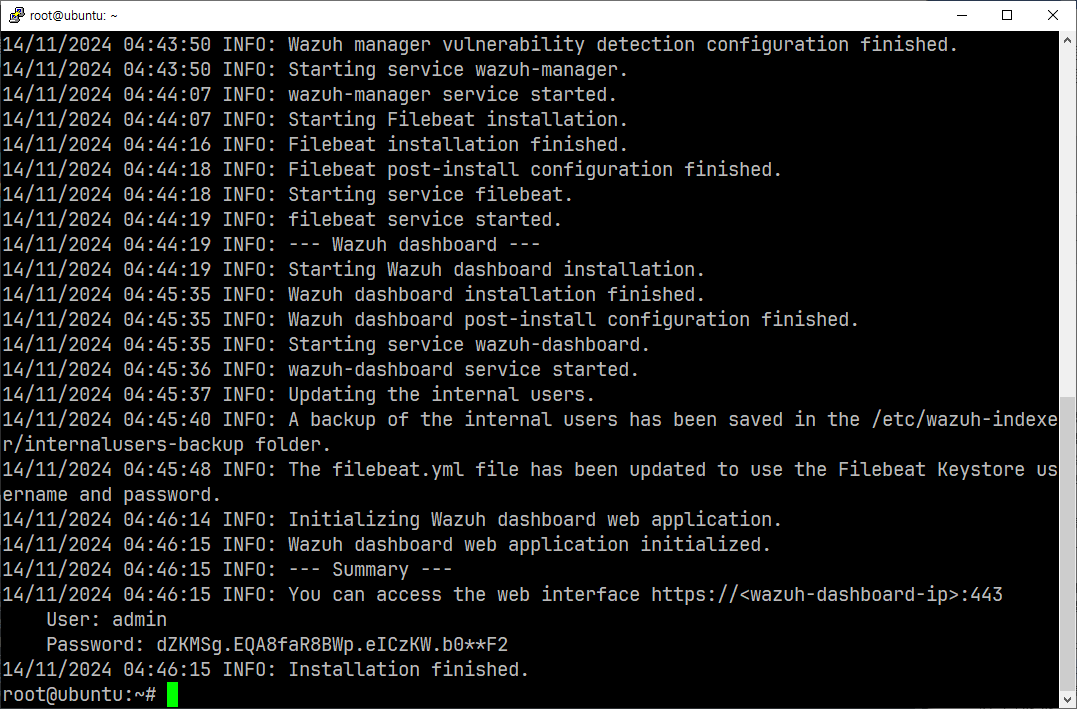
### 1) Ubuntu Wazuh 솔루션 설치

**- Ubuntu 환경 구성**



**- Wazuh 설치**

curl -sO https://packages.wazuh.com/4.9/wazuh-install.sh && sudo bash ./wazuh-install.sh -a



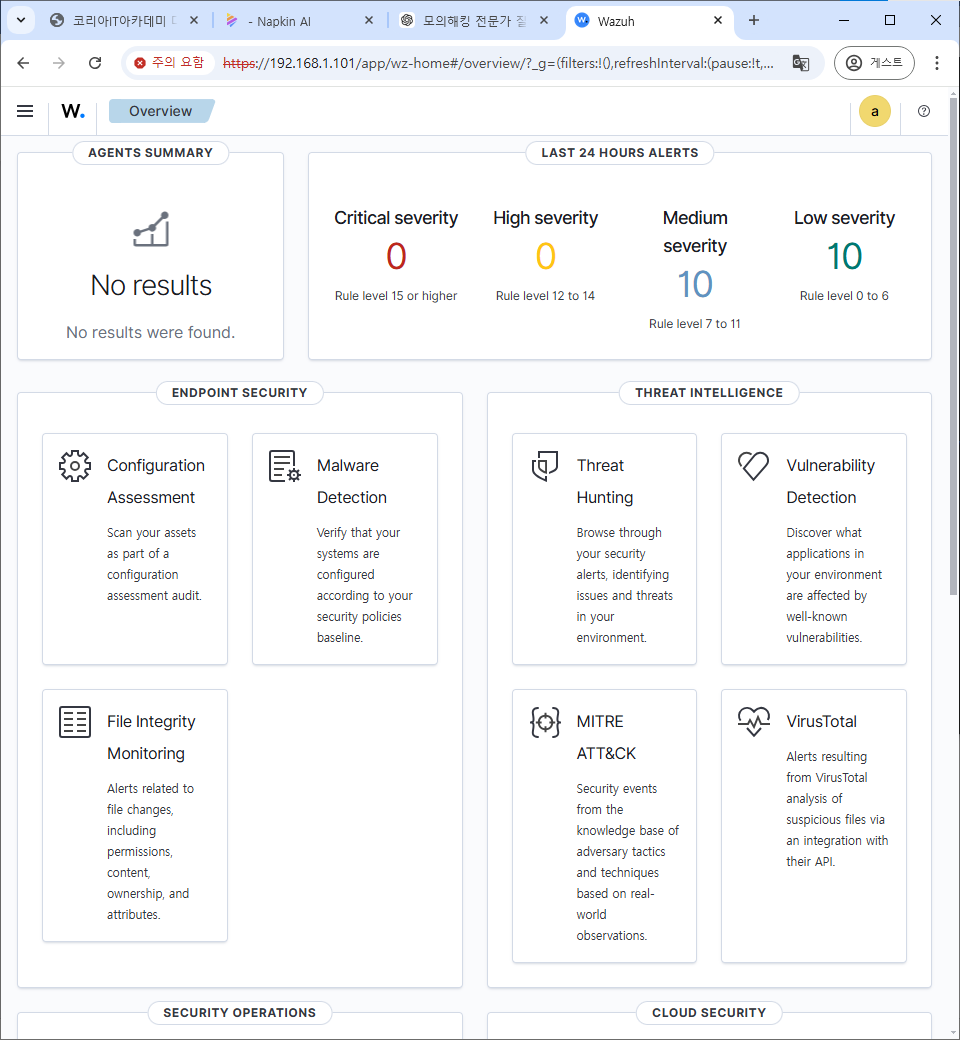
### 2) Wazuh Web

**- Wazuh 계정 정보**

Username: admin

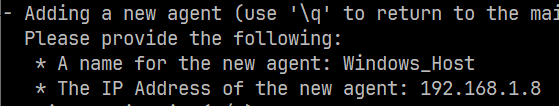
Password: Wn.owwNy.iM+kGOe4K390qDFpBhIqN+T(설치 과정에서 문제가 있어 재설치 하였습니다.)

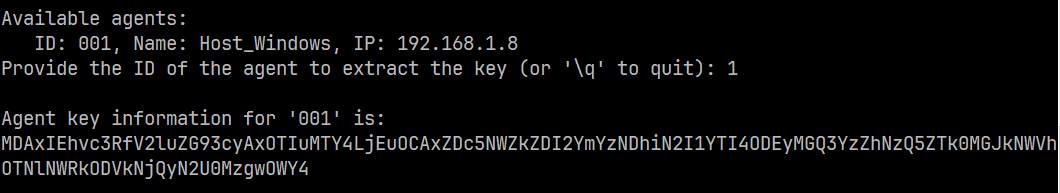
**- Dashboard**

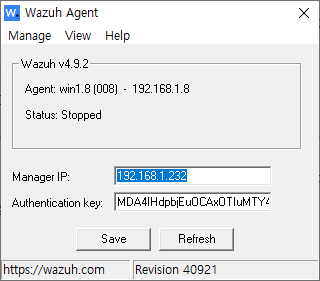
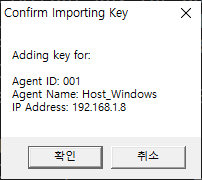
****

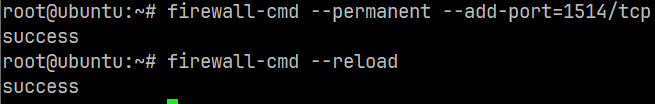
### 3) Wazuh Agent

**- Windows Agent 등록**





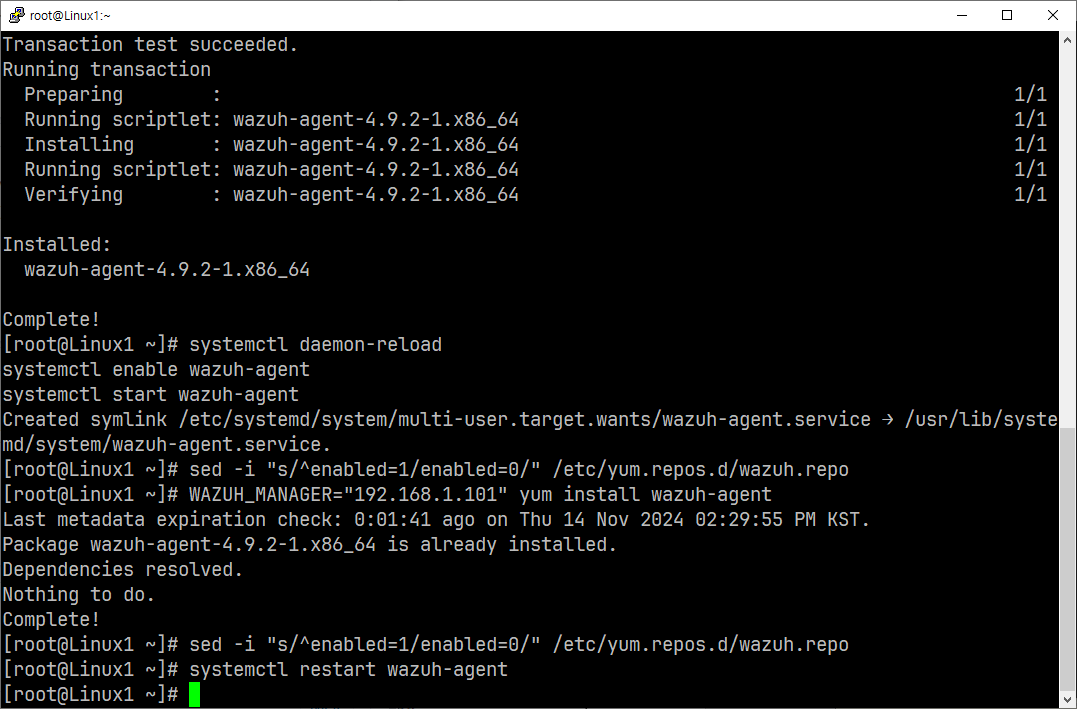
 



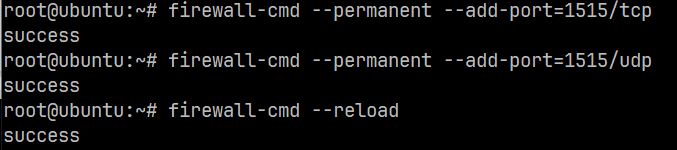


**- Rocky\_Linux**

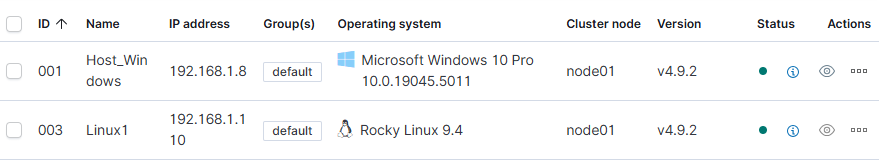
Agent 설치



Manager 시스템 port 개방

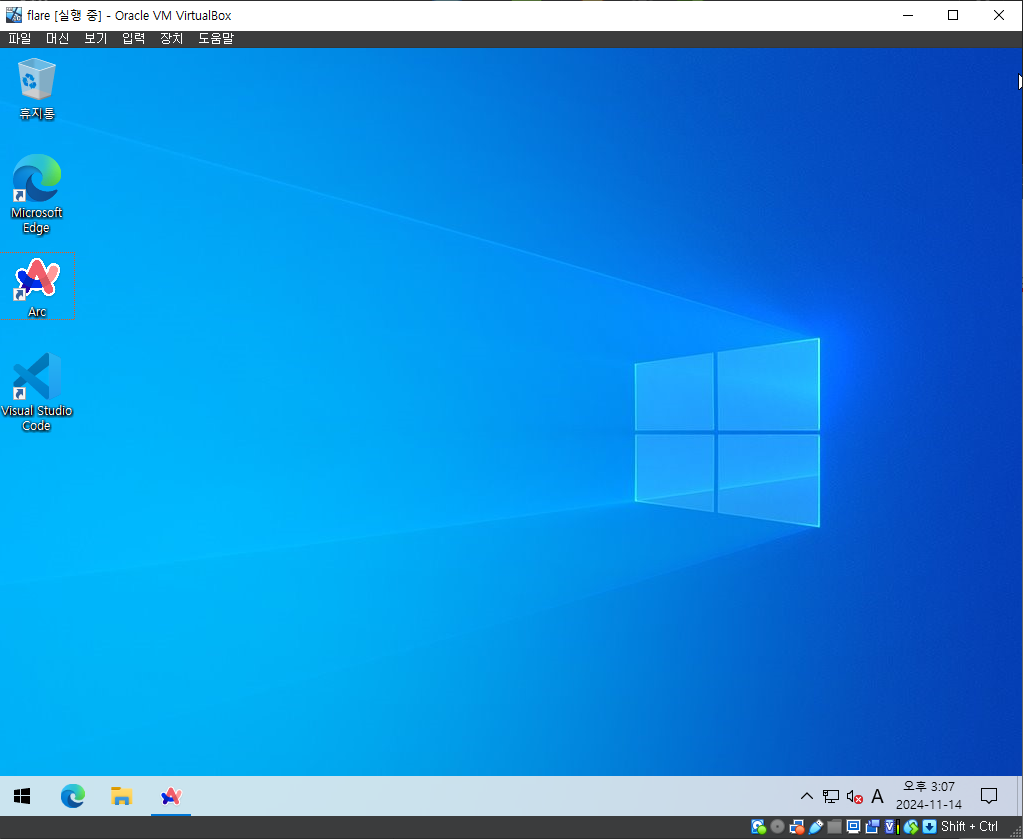


연결 확인

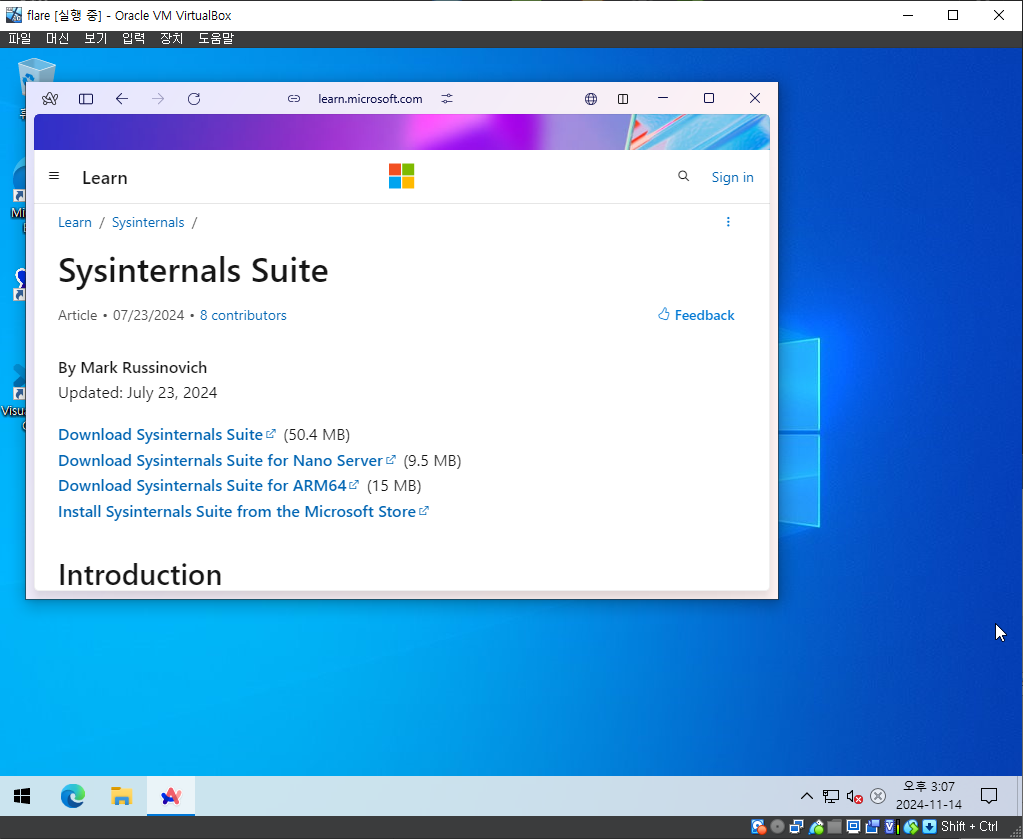


## 2.4. Flare VM 구축

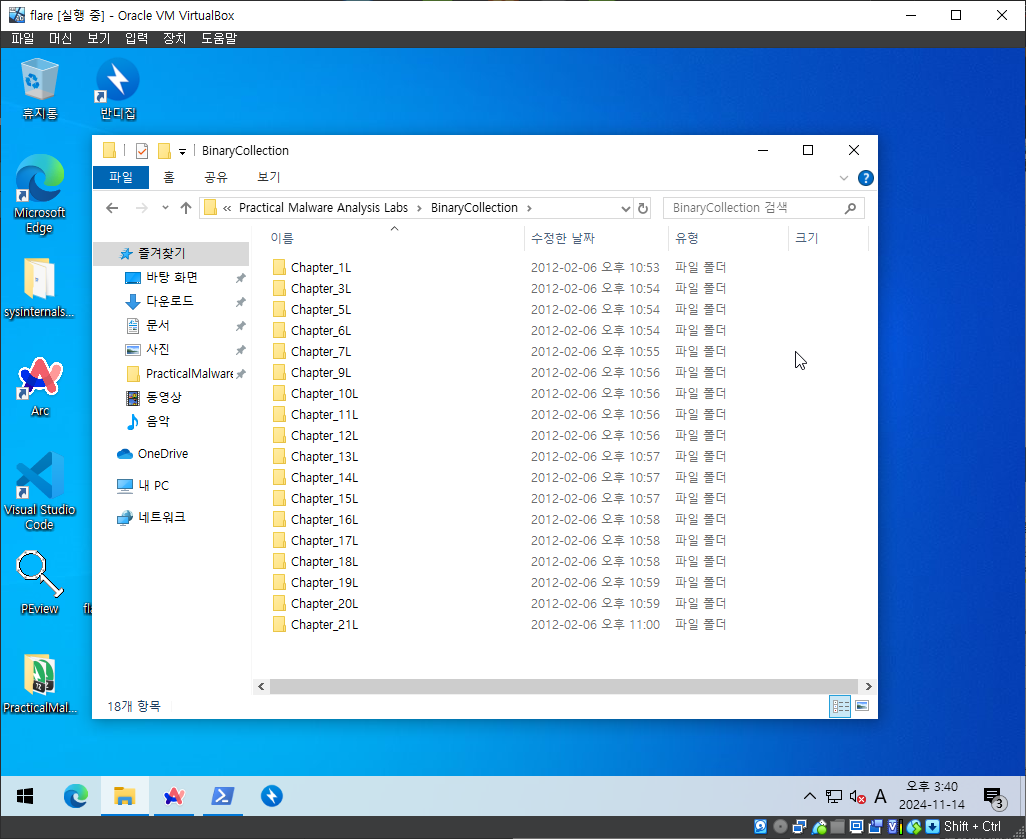
### 1) Windows VM 구축



### 2) sysinternals suite 설치



### 3) Practical Malware 다운



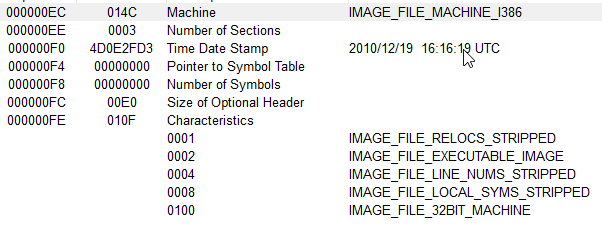
악성 코드 분석에 필요한 툴들은 개별 설치했습니다.

## 2.5. Lab01-01.exe 분석

**- PEView**

해당 파일이 실행 파일인 점과 컴파일 시간을 확인



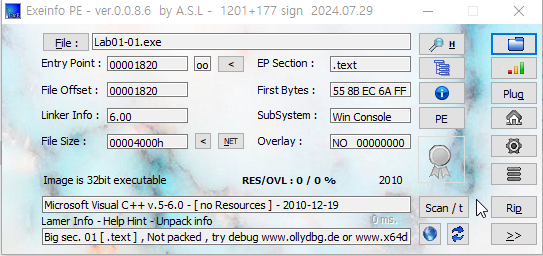


파일 형태: 실행 파일

컴파일 시간: 10년/12/19 16:16:19 UTC

**- Exeinfo PE**

파일의 패킹, 암호화 여부 확인

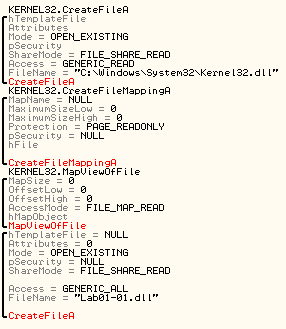


편집 퉅: VS C++ v.5-6.0

패킹 여부: Not packed

**- OllyDBG**

실행 파일의 어셈블리 코드(실행 시 침해 과정) 추측 가능



파일 실행 결과: “C:\Windows\System32\Kerne132.dll”을 사용해 새로운 파일을 생성한다.

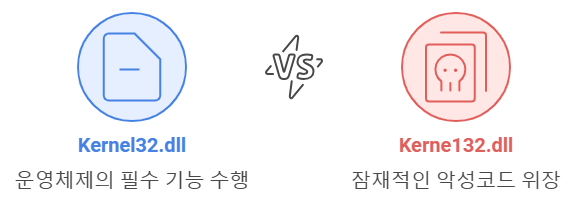
**- BinText**



실행 결과로 만들어지는 파일의 정체가 ‘kerne132.dll’로 예상된다.

Kernel32.dll 파일은 (검증된)다른 실행 파일을 실행할 때 대부분 필요한 파일로 ‘메모리 관리’, ‘입출력 명령’, ‘프로세스와 스레드 생성’ 과 같이 운영체제에서 필요한 동작을 수행한다.

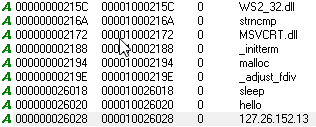
악성코드 유포자는 특정 실행 파일 실행 시 kernel32.dll 파일을 유사한 이름의 kerne132.dll로 바꿔서 실행되도록 유도해 원하는 결과를 얻고자 하는 것으로 예상된다.





Kernel132.dll 생성(실행) 후 Lab01-01.dll 이 실행되는 것을 발견했다.

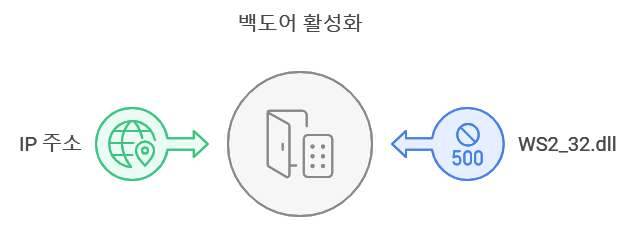
해당 파일을 분석하면 유포자가 의도한 실행 결과를 알 수 있을 것 같다.



WS2\_32.dll 파일이 호출(실행)되는 것을 확인했으며 이어서 IP 주소를 확인했다.

WS2\_32.dll 파일은 Windows Socket 사용을 위한 함수들의 정의되어 있는 파일로 네트워크 통신을 위해 사용된다.

WS2\_32.dll 파일의 호출과 함께 IP 주소가 함께 값으로 저장하는 것으로 유추한 결과 해당 파일은 실행 시 유포자의 PC로 통하는 백도어가 생성(활성화) 된다고 예상된다.



**- 악성코드 분석 결과**

|  |  |  |  |
| --- | --- | --- | --- |
| **패킹 여부** | No | | |
| **코드 내 함수 실행** | Kernel32.dll -> Kerne132.dll (import>파일 변조)  Lab01-01.dll (import) | | |
| **호스트 기반 증거** | Kerne132.dll | **네트워크 기반 증거** | Lab01-01.dll |
| **유포 목적 추론** | 1. Kerne132.dll 생성으로 피해자(사용자)의 PC 시스템 지연  2. 침해 사실 인지 지연/탐지 방지  3. 공격자(유포자)의 시스템에 접속(또는 접속을 허용)하는 백도어 생성 | | |
| **대응 방안** | 불필요한 프로세스 중지  기존 서비스 내용 점검(kerne132.dll 파일은 육안으로 검증해야 함)  백신 설치 및 업데이트 최신화 | | |

## 2.6. 악성 코드 2종 분석

### 1) Lab03-01

**- PEView**

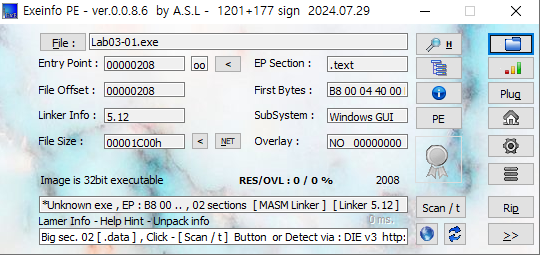


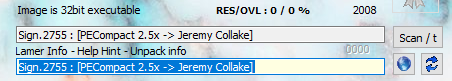
파일 형태: 실행 파일

컴파일 시간: 08년/01/06 14:51:31 UTC

**- Exeinfo PE**

파일의 패킹, 암호화 여부 확인





편집 퉅: 불명

패킹 여부: PECompact 2.5x 툴로 압축됨

**- BinText**



프로세스 제어 및 네트워크 통신을 위한 라이브러리 참조 확인



HTTP 연결을 시도하는 문자열을 확인



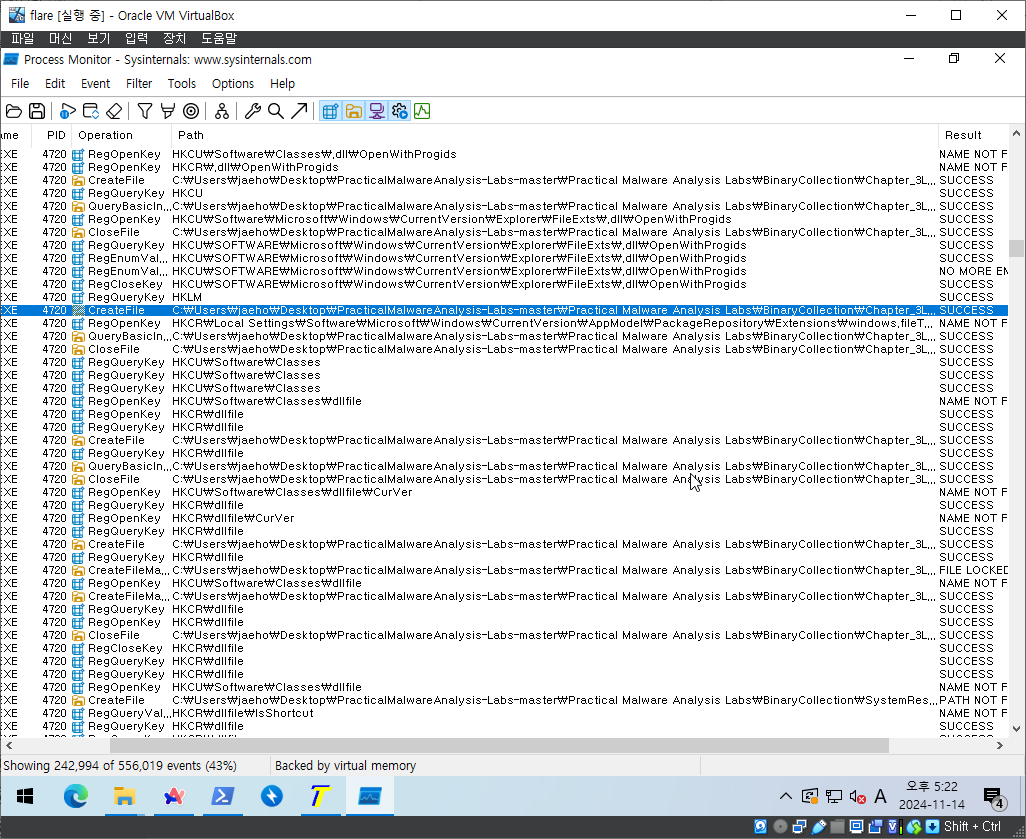
특정 웹 서버에 접속 시도



특정 파일을 윈도우 부팅시마다 실행하도록 설정하는 것을 확인

그 위에 등장하는 vmx32to64.exe 파일이 의심되어 자료 조사

**- Process Monitor(동적 점검)**



모니터링 툴을 활성화 한 상태에서 악성 프로그램 실행 시 모니터링 툴에서 지속적으로 악성 프로그램을 호출하고 실행하는 정황을 포착함

지속해서 자기 자신을 호출(실행)하는 것으로 미루어 짐작하여 해당 악성 프로그램은 웜 종류의 자가 복제 악성 프로그램으로 추측된다.

**- 악성코드 분석 결과**

|  |  |  |  |
| --- | --- | --- | --- |
| **생성 시점** | 08년/01/06 14:51:31 UTC | | |
| **패킹 여부** | PECompact 2.5x | | |
| **코드 내 함수 실행** | Kernel32.dll, ws2\_32 | | |
| **호스트 기반 증거** | Vmx32to64.exe | **네트워크 기반 증거** | www.practicalma lwareanalysis.com |
| **유포 목적 추론** | 1. 피해자의 PC가 네트워크와 연결되어 있는지 여부를 확인  2. 악성 프로그램이 부팅마다 자동 실행되도록 시스템 설정  3. system32/에 자신의 복제를 생성하는 것을 반복  4. 연결된 네트워크를 통해 연결된 다른 PC에도 증식 시도 | | |
| **대응 방안** | 불필요한 프로세스 중지  기존 서비스 내용 점검  시스템 예약(자동 실행) 기능에 불필요한 시스템 제거  백신 설치 및 업데이트 최신화 | | |

### 2) Lab06-01

시간 관계상 분석하지 못했습니다.

# 3. 문제 해결 과제

문제 풀이 시간이 부족하여 진행하지 못했습니다.

## 3.1. 웹 서버 취약점 공략 후 IDS 탐지

## 3.2. 침해 사고 증거 수집 및 분석

## 3.3. 침해사고 분석 보고서