PLAINBIT

사이버보안전문단 5기 프로젝트

# Sysmon을 이용한 호스트 기반의 사이버 위협 로깅 방안 연구

2023.10

김진국, 장원희, 김서준, 이승형

# 목 차

# 1. 개 요

코로나 19 이후 비대면 원격 근무가 증가함에 따라 사이버 위협은 날이 갈수록 지능화되어가고 있다. 이로 인해, 기업의 피해가 늘면서 사이버 위협의 탐지와 대응은 사이버 보안에서 직면해야 하는 가장 큰 이슈가 되었다. 하지만, 많은 기업에서는 아직까지도 예산 부족으로 인한 보안 제품 및 전문 운영 인력 부족, 사이버 위협에 대한 인식 부족 등의 다양한 이유로 적절한 대응 체계를 갖추기 어려운 환경이다.

최근 수준 높은 공격자들은 자신들이 사용하는 전용 코드나 도구를 흔적이 남지 않는 방식으로 실행하거나, 사용 후 완전 삭제하는 방식으로 철저하게 은닉하고 있다. 공격 과정에서 발생한 흔적들의 대부분이 공격자에 의해 삭제되기 때문에 EDR 과 같이 호스트 이상 징후에 따른 모니터링 방안이 마련되어 있지 않다면 이를 탐지하는 것은 어렵다.

이에 따라 Microsoft 에서 공개한 모니터링 도구인 Sysmon 을 통해 호스트 기반의 사이버 위협을 체계적으로 모니터링 및 탐지할 수 있는 로깅 방안 연구가 필요하다.

본 연구에서는 기존에 공개되어 있는 대표적인 3 개의 Sysmon 로깅 설정 연구와 다년간의 침해사고 분석 경험을 기반으로 국내 기업의 특성과 환경에 맞는 Sysmon 로깅 방안을 제시하고자 하며, 이로 인한 기대 효과는 아래와 같다.

**첫째, 이벤트 로그의 한계점이 보완된 로그로 위협 탐지 상세화**

기본적으로 Windows 에서는 시스템의 성능, 오류, 경고, 운영 정보 등의 중요한 정보를 이벤트 로그로 저장하고 있지만 Sysmon 은 시스템에서 일어나는 이벤트 행위(프로세스 연결, 네트워크 연결, 파일 생성 등)를 더 자세하게 기록한다. 따라서, 서비스 또는 운영체제에 의해 기본적으로 저장되는 로그 외의 시스템 행위에 대해 더 자세히 로깅해 위협을 탐지할 수 있다.

**둘째, 위협 탐지 속도 개선으로 인해 사고 대응 속도 향상**

다년간의 침해사고 분석 경험을 반영해 주요 공격자 행위에 대한 Sysmon 로그 패턴을 분석하고 이를 체계적으로 Sysmon Configure 에 반영했다. 따라서, 최적화된 이벤트로 위협 행위가 로깅 되어 적은 노력으로 대부분의 침해사고 원인과 영향 판별이 가능해 사고 대응 속도를 향상시킬 수 있다.

**셋째, 중소기업 위협 모니터링 강화**

한국인터넷진흥원(KISA)가 공개한 '기업 규모별 사이버 보안 침해사고 신고 현황'에 따르면, 최근 3 년간 전체 신고 수는 지속적으로 증가하고 있다. 이 중 중소기업의 신고가 가장 많은 것으로 확인되며, 실제로 대부분의 중소기업에서 예산 부족으로 인해 위협에 대한 모니터링을 수행하지 못하는 상황이다. 본 연구에 사용된 Sysmon 은 공개된 도구로 누구나 사용할 수 있기 때문에 금전적인 부담이 없이 중소기업에서도 위협에 대한 모니터링을 통해 보안을 강화하는데 기여할 수 있다.

# 2. Sysmon 이란?

Sysmon(System Monitor)은 Microsoft 사의 Sysinternals Suite 에 포함된 시스템 모니터링 도구로, 시스템에서 일어나는 이벤트 행위(프로세스 생성, 이벤트 연결 등)를 더 자세하기 기록한다. 이벤트 로그가 남기지 않는 정보를 추가로 기록할 수 있기 때문에 공격자(또는 악성코드)가 수행한 악성 행위와 시스템 이상 징후 원인 등을 식별하는데 많은 도움이 된다.

## 2.1. Sysmon 주요 기능

Sysmon 에서는 다음과 같은 주요 기능을 지원하고 있다.

[표 1] Sysmon 주요 기능

| 번호 | 주요 기능 |
|---|---|
| 1 | 현재 프로세스와 부모 프로세스의 프로세스 생성 정보 |
| 2 | 여러 해시 알고리즘 (SHA1, SHA256, IMPHASH)를 사용해 프로세스 이미지 파일의 해시 값 기록 |
| 3 | 여러 해시 알고리즘 동시 사용 가능 |
| 4 | 프로세스 생성 이벤트에 프로세스 GUID 를 포함해 Windows 에서 프로세스 ID 를 재사용하는 경우에도 이벤트의 상관관계 분석 가능 |
| 5 | 각 이벤트에 세션 GUID 를 포함해 동일한 로그온 세션의 이벤트 상관 관계 분석 가능 |
| 6 | 드라이버 또는 DLL 로드를 서명 및 해시 정보와 함께 기록 |
| 7 | 디스크 및 볼륨의 Raw Access Read 에 대한 이벤트 존재 |
| 8 | 네트워크 연결을 시도하는 Source 프로세스, IP 주소, Port 번호, 호스트 명, Port 정보 기록 |
| 9 | 공격자 (또는 악성코드)가 변조하는 파일 생성 시간 탐지 |
| 10 | 레지스트리에서 Configure 가 변경된 경우, Configure 를 자동으로 다시 로드 |
| 11 | 필터링 규칙을 이용해 특정 이벤트를 포함하거나 제외 가능 |
| 12 | 부팅 절차부터 Sysmon 이 동작해 시스템 프로세스의 작업 모니터링 수행 |

## 2.2. Sysmon 이벤트 항목

Sysmon 에서는 Microsoft-Windows-Sysmon/Operational.evtx[1]에 아래와 같은 이벤트를 저장하며, 각 이벤트 항목은 아래 표와 같다.

[표 2] Sysmon 이벤트 항목

| 이벤트 ID | 이벤트 명 | 설명 |
|---|---|---|
| 1 | Process Create | 새로 생성된 프로세스에 대한 정보를 기록 |
| 2 | File Creation time changed | 파일 생성 시간이 프로세스에 의해 수정될 때 기록 |
| 3 | Network connection detected | 시스템의 TCP/UDP 연결 기록 |
| 4 | Sysmon service state changed | Sysmon 서비스 상태 (시작 또는 중지) 기록 |
| 5 | Process terminated | 프로세스가 종료될 때 기록 |
| 6 | Driver loaded | 시스템에 로드 되는 드라이브 정보 (드라이브 해시, 서명 정보 등) 기록 |
| 7 | Image loaded | 프로세스에서 모듈이 로드 될 때 기록 |
| 8 | CreateRemoteThread detected | 프로세스가 다른 프로세스에서 스레드를 생성할 때 기록 |
| 9 | RawAccessRead detected | 프로세스가 드라이브에서 읽기 작업을 수행할 때 기록 |
| 10 | Process accessed | 프로세스가 다른 프로세스에 접근할 때 기록 |
| 11 | File created | 프로세스가 파일을 생성하거나 덮어쓸 때 기록 |
| 12 | RegistryEvent, Object added or deleted | 프로세스가 레지스트리 키 또는 값을 생성 및 삭제할 때 기록 |
| 13 | RegistryEvent, Value set | 프로세스가 레지스트리 값을 수정할 때 기록 |
| 14 | RegistryEvent, Object renamed | 프로세스가 레지스트리 이름을 변경할 때 기록 |
| 15 | File stream created | 명명된 파일 스트림이 생성될 때 기록 |
| 16 | Sysmon config state changed | 필터링 규칙이 변경된 경우와 같이 Sysmon 설정 변경 내용이 기록 |
| 17 | PipeEvent, Pipe Created | 명명된 파이프가 생성될 때 기록 |
| 18 | PipeEvent, Pipe Connected | 클라이언트와 서버 간에 명명된 파이프 연결이 이루어질 때 기록 |
| 19 | WmiEvent, WmiEventFilter activity detected | WMI 이벤트 필터가 등록될 때 기록 |
| 20 | WmiEvent, WmiEventConsumer activity detected | WMI Consumers 가 등록될 때 기록 |
| 21 | WmiEvent, WmiEventConsumerToFilder activity detected | WMI Consumer 가 필터에 바인딩 될 때 기록 |
| 22 | DNSEvent, DNS query | 성공 여부와 관계없이 프로세스가 DNS 쿼리를 실행할 때,<br>캐시 되거나 캐시 되지 않을 때 기록 |
| 23 | FileDelete, File Delete archived | 시스템에서 파일이 삭제됐을 때 기록 |
| 24 | Clipboard changed | 시스템 클립보드 내용이 변경될 때 기록 |
| 25 | Process Tampering | 프로세스 변조 기술이 탐지될 때 기록 |
| 26 | File Delete logged | 시스템에서 파일이 삭제됐을 때 기록(ArchivedDirectory 저장 안 함) |
| 27 | File Block Executable | 실행 파일 생성을 탐지하고 차단할 때 기록 |
| 28 | File Block Shredding | 파일 완전 삭제 도구로 완전 삭제 시도를 탐지하고 차단할 때 기록 |
| 29 | File Executable Detected | 실행 파일이 생성될 때 기록 |
| 255 | Error report | Sysmon 내에서 오류가 발생했을 때 기록 |

[1] Sysmon 이벤트로그 저장 경로는 “C:\Windows\System32\winevt\Logs\Microsoft-Windows-Sysmon/Operational.evtx”이다.

## 2.3. Sysmon Configure

Sysmon 을 기본 설정으로 설치하게 되면 로깅 되지 않는 이벤트(네트워크 정보, 파일 생성 및 삭제, 레지스트리 변경 등)가 존재한다. 그러나, Sysmon 설정을 모두 활성화해 로깅하면 너무 많은 데이터가 로깅 되어 위협 탐지를 위한 이벤트가 덮어 씌워질 수 있다. 이로 인해 위협을 탐지하기 위해 필요한 데이터를 확보할 수 없어 전반적인 위협 탐지가 어렵다. 따라서, 운영하고 있는 시스템 용도와 모니터링 목적에 맞는 Configure 를 작성해야 한다.

Configure 파일은 XML 형식으로 작성되어 있으며, 이벤트 유형이 정의된 필드에 논리 연산 (AND, OR)을 적용해 필터링 옵션을 사용할 수 있어 높은 유연성을 가진다.

### 2.3.1. Sysmon Configure 구성

Sysmon Configure 는 아래 항목들로 구성된다.

[표 3] Sysmon Configure 구성 항목

| 태그 | 설명 |
|---|---|
| 〈Sysmon〉 | Configure File 의 최상위 요소로 Schema Version 정의 |
| 〈Configure_Entries〉 | Sysmon Meta Config(Hash Algorithms 등) 정의 |
| 〈EventFiltering〉 | Sysmon 이벤트 필터링을 위한 요소 |
| 〈RuleGroup〉 | 같은 종류의 이벤트에 대한 필터링을 그룹화해 논리 연산 적용 |
| 〈Event_Tag〉 | 필터링할 이벤트 지정 |
| 〈Field〉 | 이벤트에서 필터링할 필드 지정 |

```
<Sysmon schemaversion="0.00">
  <Configure_Entries>...</Configure_Entries>
  <EventFiltering>
    <RuleGroup name="", groupRelation="">
      <Event_Tag onmatch="">
        <!-- Filtering -->
        <Field condition=""> ... </Field>
        <Rule name="", groupRelation="">
          <Field condition=""> ... </Field>
        </Rule>
      </Event_Tag>
    </RuleGroup>
  </EventFiltering>
</Sysmon>
```

[그림 1] Sysmon Configure 구성 예시

### 2.3.2. Sysmon Configure Entries 항목

Sysmon Configure Entries 는 Configure 파일 내에서 Sysmon 설정값을 지정할 수 있으며, Sysmon Configure Entries 항목은 아래 표와 같다.

[표 4] Sysmon Configure Entries 항목

| 항목 구분 | 타입 | 설명 |
|---|---|---|
| ArchiveDirectory | String | 삭제된 파일이 저장되는 Volume roots 디렉터리 이름 지정<br>해당 디렉터리는 시스템 ACL 로 보호 (기본 값: Sysmon) |
| CheckRevocation | Boolean | 폐지된 서명 검사 기능 제어 여부 (기본 값: True) |
| CopyOnDeletePE | Boolean | 삭제된 PE 파일 ArchiveDirectory 복사 여부 (기본 값: False) |
| CopyOnDeleteSIDs | Strings | 파일 삭제 시 ArchiveDirectory 에 복사할 계정 SID 목록 |
| CopyOnDeleteExtensions | Strings | 파일 삭제 시 ArchiveDirectory 에 복사할 확장자명 목록 |
| CopyOnDeleteProcesses | Strings | 파일 삭제 시 ArchiveDirectory 에 복사할 프로세스 목록 |
| DnsLookup | Boolean | Reverse DNS lookup 제어 여부 (기본 값: True) |
| DriverName | String | Sysmon Driver 및 Service 이름을 지정된 이름으로 실행 |
| HashAlgorithms | Strings | Hash 에 적용할 Hash 알고리즘 목록<br>- 지원 알고리즘: MD5, SHA1, SHA256, IMPHASH 및 *(All) |

## 2.4. Sysmon Configure Rules

### 2.4.1. RuleGroup

RuleGroup 은 같은 종류의 이벤트 필터들을 그룹화해 관리할 수 있으며, “groupRelation” 속성을 사용해 필터 간 논리 연산을 수정할 수 있다. RuleGroup 당 하나의 Event Tag 만 가질 수 있으며, 둘 이상의 다른 Event Tag 가 있는 경우 첫 번째 Event Tag 만 적용된다. Schema Version 4.22 이상에서 필터 간 기본 관계는 “AND”로 설정되어 있다.

```
<RuleGroup name="Group Name", groupRelation="and|or">
  ...
</RuleGroup>
```

[그림 2] RuleGroup 사용 예시

### 2.4.2. Event_Tag

Sysmon Configure 에서 필터링하기 위해서는 이벤트 별로 정의된 태그를 사용해야 하며, 정의된 Event_Tag 항목은 아래 표와 같다.

[표 5] Sysmon Configure Event Tag 항목

| 이벤트 ID | 이벤트 명 | 태그 |
|---|---|---|
| 1 | Process Create | Process Create |
| 2 | File creation time changed | FileCreatedTime |
| 3 | Network connection detected | NetworkConnect |
| 4 | Sysmon service state changed | N/A |
| 5 | Process terminated | ProcessTerminate |
| 6 | Driver loaded | DriverLoad |
| 7 | Image loaded | ImageLoad |
| 8 | CreateRemoteThread detected | CreateRemoteThread |
| 9 | RawAccessRead detected | RawAccessRead |
| 10 | Process accessed | ProcessAccess |
| 11 | File created | FileCreate |
| 12 | Registry Object added or deleted | RegistryEvent |
| 13 | Registry Value set | RegistryEvent |
| 14 | Registry Object rename | RegistryEvent |
| 15 | File stream created | FileCreateStreamHash |
| 16 | Sysmon config state changed | N/A |

| 이벤트 ID | 이벤트 명 | 태그 |
|---|---|---|
| 17 | Pipe Created | PipeEvent |
| 18 | Pipe Connected | PipeEvent |
| 19 | WmiEventFilter activity detected | WmiEvent |
| 20 | WmiEventConsumer activity detected | WmiEvent |
| 21 | WmiEventConsumerToFileter activity detected | WmiEvent |
| 22 | DNS query | DNSQuery |
| 23 | File Delete archived | FileDelete |
| 24 | Clipboard changed | ClipboardChange |
| 25 | Process Tampering | ProcessTampering |
| 26 | File Delete logged | FileDeleteDetected |
| 27 | File Block Executable | FileBlockExecutable |
| 28 | File Block Shredding | FileBlockShredding |
| 29 | File Executable Detected | FIleExecutableDetected |

### 2.4.3. onmatch

Event_Tag 에서 반드시 “onmatch” 속성을 지정해야 하며, 이벤트가 일치하게 되면 onmatch 속성의 값인 “include”와 “execlude”에 따라 이벤트를 포함하거나 제외할 수 있다.

```
<ProcessCreate onmatch="include|exclude">
  <Field>...</Field>
</ProcessCreate>
```

[그림 3] “onmatch” 속성 사용 예시

“onmatch” 속성 값에 대한 설명은 아래 표와 같다.

[표 6] “onmatch” 속성 값

| 속성 값 | 설명 | Rule 예시 |
|---|---|---|
| include | 해당 Event Tag 에 대한 이벤트를 기록하지 않음 | 〈Event_Tag onmatch=”include”〉<br>…<br>〈/Event_Tag〉 |
| exclude | 해당 Event Tag 에 대한 모든 이벤트를 기록 | 〈Event_Tag onmatch=”exclude”〉<br>…<br>〈/Event_Tag〉 |

### 2.4.4. Filtering (Field and Condition)

각 이벤트 유형의 Field 를 필터링하기 위해 “condition” 속성을 사용할 수 있으며, 각 Field 에는 0 개 이상의 필터가 포함될 수 있다. 동일한 Field 이름에 대한 필터는 OR 조건으로 동작하고, 다른 Field 이름에 대한 필터는 AND 조건으로 동작한다. Field 값에 대한 condition 은 아래 표와 같이 대소문자 구분 없이 사용 가능하다.

condition 미 설정 시 기본적으로 값은 “is”로 적용되며, name 속성은 이벤트가 기록될 때 이벤트의 RuleName 필드를 채우는데 사용된다.

```
<Image name="Rule Name" condition="is">C:\Windows\system32\audiodg.exe</Image>
```

[그림 4] condition 속성 사용 예시

Sysmon Configure Condition 항목은 아래 표와 같다.

[표 7] Sysmon Configure Condition 항목

| 조건 구분 | 설명 |
|---|---|
| is | 값이 동일 (기본 값) |
| is any | 필드는 세미콜론(;)으로 구분된 값 중 하나 |
| is not | 값이 다름 |
| contains | 필드에 해당 값 포함 |
| contains any | 필드에 세미콜론(;)으로 구분된 값 포함 |
| contain all | 필드에 세미콜론(;)으로 구분된 값 모두 포함 |
| excludes | 필드에 값이 없음 |
| excludes any | 필드에 세미콜론(;)으로 구분된 값이 하나 이상 없음 |
| excludes all | 필드에 세미콜론(;)으로 구분된 값이 모두 없음 |
| begin with | 필드가 해당 값으로 시작 |
| end with | 필드가 해당 값으로 종료 |
| not begin with | 필드가 해당 값으로 시작되지 않음 |
| not end with | 필드가 해당 값으로 종료되지 않음 |
| less than | 비교 시 0 보다 작음 |
| more than | 비교 시 0 보다 큼 |
| image | 이미지 경로가 일치하는 경우 (전체 경로 또는 이미지 명으로 지정 가능) |

# 3. 선행 연구 분석

## 3.1. 선행 연구 선정

Sysmon Configure Rule을 어떻게 설정하는가에 따라 기록되는 이벤트가 달라지며, 공격자 행위에 대한 추가적인 정보를 획득할 수 있는지에 대한 여부가 결정된다. 기존에 사이버 위협 탐지 목적으로 연구되었던 Sysmon Configure들의 Rule을 비교 및 분석해 이벤트 구성 별 관점과 Sysmon Configure 의 적절한 작성 기준을 파악한다. 선행된 Sysmon Configure 연구에서 비교 및 분석할 Configure 연구 선정 기준은 아래와 같다.

[표 8] 선행 연구 선정 기준

| 번호 | 기준 |
|---|---|
| 1 | 오픈소스 형태로 외부에 공개되어 있어 Configure에 접근이 용이해 많은 사용자가 Rule 활용 |
| 2 | 최대한 정밀하게 사이버 위협을 탐지해 이벤트로 로깅 |
| 3 | 사이버 위협을 탐지할 수 있는 이벤트를 정상 이벤트보다 더 많이 로깅할 수 있도록 구성 |
| 4 | 작성된 Rule이 누구나 쉽게 이해하고 활용할 수 있게 구성 |

위에서 정의된 기준에 따라 다음과 같은 3개의 Sysmon Configure 연구를 선정했다.

[표 9] 선정된 선행 연구 목록

| 구분 | 프로젝트 명 | 프로젝트 URL |
|---|---|---|
| SwiftOnSecurity | sysmon-config | https://github.com/SwiftOnSecurity/sysmon-config |
| Neo23x0 | sysmon-config | https://github.com/Neo23x0/sysmon-config |
| olafhartong | sysmon-modular | https://github.com/olafhartong/sysmon-modular |

## 3.2. 선행 연구 소개

### 3.2.1. [SwiftOnSecurity] sysmon-config

SwiftOnSecurity Sysmon Configure 는 Rule 의 거의 모든 코드에 주석이 있고, Event ID 별 설명이 작성되어 있기 때문에 Sysmon에 대한 자습서 및 가이드 역할을 수행하고 있다. 하지만, 2021년 10월 이후에 업데이트되지 않아 최근에 업데이트된 Sysmon 의 추가 이벤트에 대한 Rule은 포함되어 있지 않다.

Sysmon Configure 중에서 가장 높은 접근성을 가지고 있어 Sysmon을 설치한 대다수의 사용자가 해당 Configure를 적용해 사용하고 있으며, Github 의 많은 Sysmon Configure 들이 해당 연구를 포크해 작성되고 있다.

### 3.2.2. [Neo23x0] sysmon-config

Neo23x0 Sysmon Configure는 SwiftOnSecurity의 Sysmon Configure에서 분기 및 수정된 연구 중 가장 큰 연구로, 처음에는 단순히 SwiftOnSecurity 의 Sysmon Configure 를 포크 했지만 이후 30 개 이상의 소규모 연구들이 합쳐져 꾸준히 업데이트되고 있다. 마지막으로 2023 년 6 월에 업데이트되어 SwiftOnSecurity 연구에서 기록되지 않은 이벤트 유형이 포함되어 있다.

### 3.2.3. [olafhartong] sysmon-modular

olafhartong Sysmon Configure는 2023년 9월에 마지막으로 업데이트되었으며, MITRE ATT&CK을 기반으로 발생할 수 있는 행위를 이벤트에 적용시켜 공격자의 행위를 탐지할 수 있도록 연구했다. 또한, Event ID 유형 별로 모듈화되어 있어 사용자가 Configure를 손쉽게 최적화해 수정할 수 있도록 제작했다.

## 3.3. 선행 연구 분석

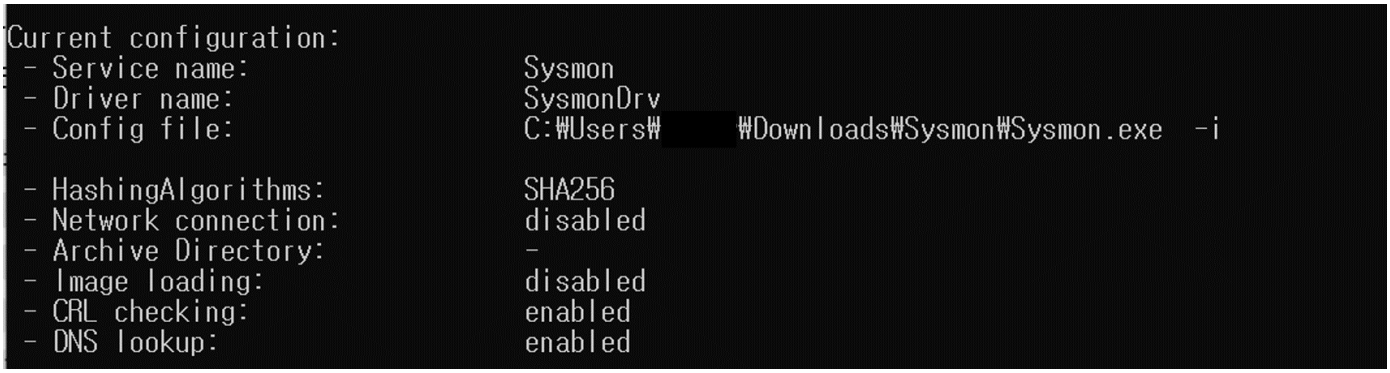
선정된 3 개의 연구와 별도의 기본으로 설치된 Sysmon 을 비교 분석해 선행 연구의 한계점을 파악하고 보다 효과적으로 사이버 위협을 탐지하기 위한 방향을 연구한다. 이를 위해, 각 선행 연구에서 정의된 설정과 Rule 을 파악하고 해당 Configure 를 업무 PC 에 적용해 업무시간에 발생한 이벤트를 비교 분석한다. 분석할 선행 연구의 Configure 파일 목록은 아래와 같다.

[표 10] 분석할 선행 연구의 Configure 파일 목록

| 구분 | 프로젝트 명 | Configure File |
|---|---|---|
| Microsoft | - | Configure 적용 없이 기본 설치 |
| SwiftOnSecurity | sysmon-config | sysmonconfig-export.xml |
| Neo23x0 | sysmon-config | sysmonconfig-trace.xml |
| | | sysmonconfig-export-block.xml |
| olafhartong | sysmon-modular | sysmonconfig-mde-augment.xml |

### 3.3.1. Sysmon 구성 항목

Sysmon 을 설치하고 구성 항목을 확인하면 Sysmon 에 설정된 정보를 확인할 수 있다. 이때, 적용된 설정에서 “Network connection”, “Image loading”, “CRL checking”, “DNS lookup” 옵션에 대한 활성 여부를 확인할 수 있다.




[그림 5] Sysmon 기본 설치 후 구성 설정 화면 (sysmon -c 명령 결과)

각 옵션에 대한 설명은 아래 표와 같다.

[표 11] Sysmon 설정 정보에서 확인 가능한 옵션

| 구성 항목 | 설명 |
|---|---|
| Network connection | 네트워크 연결을 모니터링하고 있으며, 활성화되어 있는 경우 네트워크 활동 추적 가능 |
| Image loading | DLL 로딩을 탐지하며, 활성화되어 있는 경우 새로운 DLL 로딩 식별 가능 |
| CRL checking | 디지털 인증서의 유효성을 확인하고 CRL 을 검사하며, 활성화되어 있는 경우 보안 인증서 상태 확인 |
| DNS lookup | DNS 조회를 탐지하며, 활성화되어 있는 경우 호스트의 이름을 IP 주소로 변환하는 과정 확인 가능 |

Sysmon 을 기본 설치했을 때, Network connection, Image loading 설정이 비활성화되어 네트워크와 이미지에 대한 이벤트를 탐지하지 않는다. SwiftOnSecurity 는 Image Loading 을 제외하고 모두 활성화하고 있고, Neo23x0 는 모든 설정을 활성화해 로깅 되도록 구성했다. 다만, olafhartong 에서는 시스템 성능을 고려해 기본 활성화되어 있는 CRL checking 과 DNS lookup 설정은 비활성화해 구성되었다. 각 연구 별 Configure 적용 설정은 아래 표와 같다.

[표 12] 연구 별로 적용된 구성 항목 비교

| 구성 항목 | Microsoft | SwiftOnSecurity | Neo23x0 | olafhartong |
|---|---|---|---|---|
| Network connection | disabled | enabled | enabled | enabled |
| Image loading | disabled | disabled | enabled | enabled |
| CRL checking | enabled | enabled | enabled | disabled |
| DNS lookup | enabled | enabled | enabled | disabled |

### 3.3.2. Configure 파일에 정의된 Rule 목록

각 선행 연구의 Configure 파일에 정의된 Rule 목록과 Rule 에 대한 “onmatch” 속성 통계는 아래 표와 같다.

[표 13] 각 선행 연구의 Configure 파일에 정의된 Rule 목록 및 “onmatch” 속성 통계

△: 정의되지 않음, ◎: 모두 기록, –: 기록하지 않음

| 이벤트 ID | 이벤트 명 | SwiftOnSecurity | | Neo23x0 – trace | | Neo23x0 – sysmon | | olafhartong | |
|---|---|---|---|---|---|---|---|---|---|
| | | include | exclude | include | exclude | include | exclude | include | exclude |
| 1 | Process Create | △ | 138 | △ | 18 | △ | 147 | 298 | 176 |
| 2 | File Creation time changed | 3 | 9 | ◎ | | 3 | 9 | 6 | 7 |
| 3 | Network connection detected | 84 | 9 | △ | 28 | 100 | 14 | 145 | 29 |
| 4 | Sysmon service state changed | ◎ | | ◎ | | ◎ | | ◎ | |
| 5 | Process terminated | 2 | – | ◎ | | ◎ | | 3 | △ |
| 6 | Driver loaded | △ | 3 | ◎ | | ◎ | | – | |
| 7 | Image loaded | – | | – | 20 | 5 | △ | 67 | 14 |
| 8 | CreateRemoteThread detected | △ | 9 | ◎ | | △ | 9 | 2 | 9 |
| 9 | RawAccessRead detected | – | | △ | 6 | – | | – | |
| 10 | Process accessed | – | | △ | 50 | 4 | 2 | 42 | 62 |
| 11 | File created | 53 | 11 | △ | 15 | 82 | 12 | 103 | 30 |
| 12, 13, 14 | RegistryEvent | 114 | 52 | △ | 43 | 133 | 52 | 246 | 99 |
| 15 | File stream created | 19 | – | ◎ | | ◎ | | ◎ | |
| 16 | Sysmon config state changed | ◎ | | ◎ | | ◎ | | ◎ | |

△: 정의되지 않음, ◎: 모두 기록, -: 기록하지 않음

| 이벤트 ID | 이벤트 명 | SwiftOnSecurity | | Neo23x0 - trace | | Neo23x0 - sysmon | | olafhartong | |
|---|---|---|---|---|---|---|---|---|---|
| | | include | exclude | include | exclude | include | exclude | include | exclude |
| 17, 18 | PipeEvent | 9 | △ | △ | 3 | 49 | 3 | 16 | 65 |
| 19, 20, 21 | WmiEvent | ◎ | | - | | ◎ | | ◎ | |
| 22 | DNSEvent, DNS query | △ | 209 | △ | 1 | △ | 208 | - | |
| 23 | FileDelete, File Delete archived | - | | - | | - | | 98 | 7 |
| 24 | Clipboard changed | - | | ◎ | | - | | - | |
| 25 | Process Tampering | - | | ◎ | | - | | △ | 15 |
| 26 | File Delete logged | - | | △ | 3 | - | | 40 | 7 |
| 27 | File Block Executable | - | | - | | 223 | △ | - | |
| 28 | File Block Shredding | - | | - | | 713 | △ | - | |
| 29 | File Executable Detected | - | | - | | - | | ◎ | |
| 255 | Error report | ◎ | | ◎ | | ◎ | | ◎ | |

### 3.3.3. 선행 연구 적용 결과

선행 연구에서 확인된 Sysmon Configure 파일을 적용해 테스트했으며, 테스트 환경과 결과는 다음과 같다.

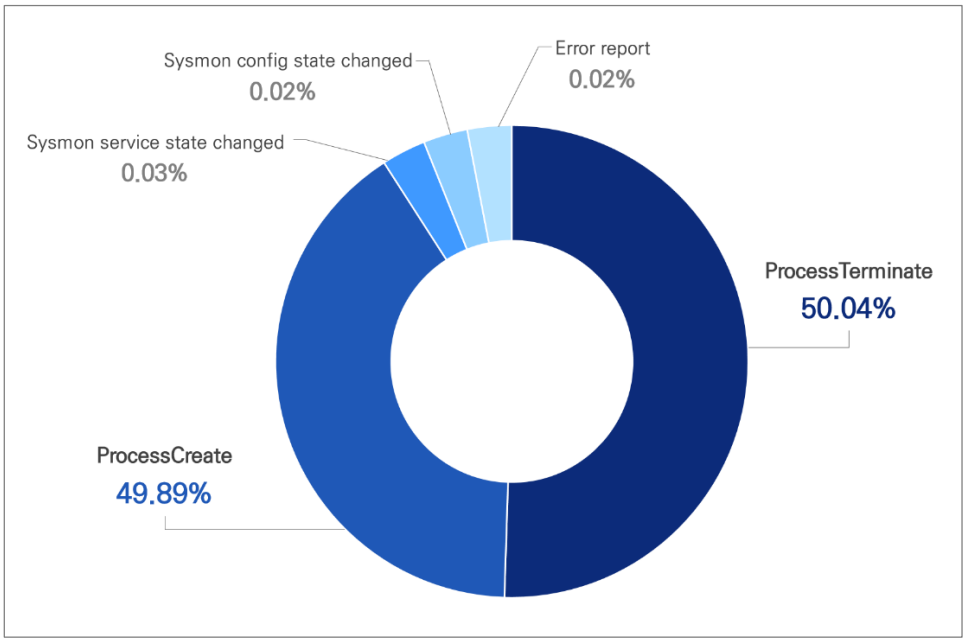
[표 14] 선행 연구 적용 테스트 환경 및 모니터링 기간

| 구분 | 내용 |
|---|---|
| 테스트 환경 | Windows 10, Windows 11 |
| 테스트 용도 | 업무용 PC (문서 작업, 개발, 업무 협업 도구 (메신저 등) 사용, 웹 브라우저를 통한 검색 등) |
| 모니터링 기간 | 2023-08-14 ~ 2023-09-15, 업무 시간에 사용 (오전 9 시 ~ 오후 6 시) |

**추가 설정 없이 Sysmon 만 설치한 상태**[2]에서 로깅을 모니터링한 결과, 1 일 평균 2MB 의 이벤트가 발생했으며, “ProcessCreate”, “Processsterminate” 이벤트만 기록되었다. 따라서, 공격자의 행위를 탐지하기 위해 필요한 네트워크 정보, 파일 생성 및 삭제, 레지스트리 변경 등에 대한 데이터를 확보할 수 없어 다른 이벤트들에 대해 별도로 구성된 Configure 작성이 필요하다.

[표 15] Sysmon만 설치한 환경에서 발생된 상위 5개 이벤트 목록

| 발생된 이벤트 비율 | 이벤트 ID | 이벤트 명 |
|---|---|---|
| 50.04% | 5 | ProcessTerminate |
| 49.89% | 1 | ProcessCreate |
| 0.03% | 4 | Sysmon service state changed |
| 0.02% | 16 | Sysmon config state changed |
| 0.02% | 255 | Error report |



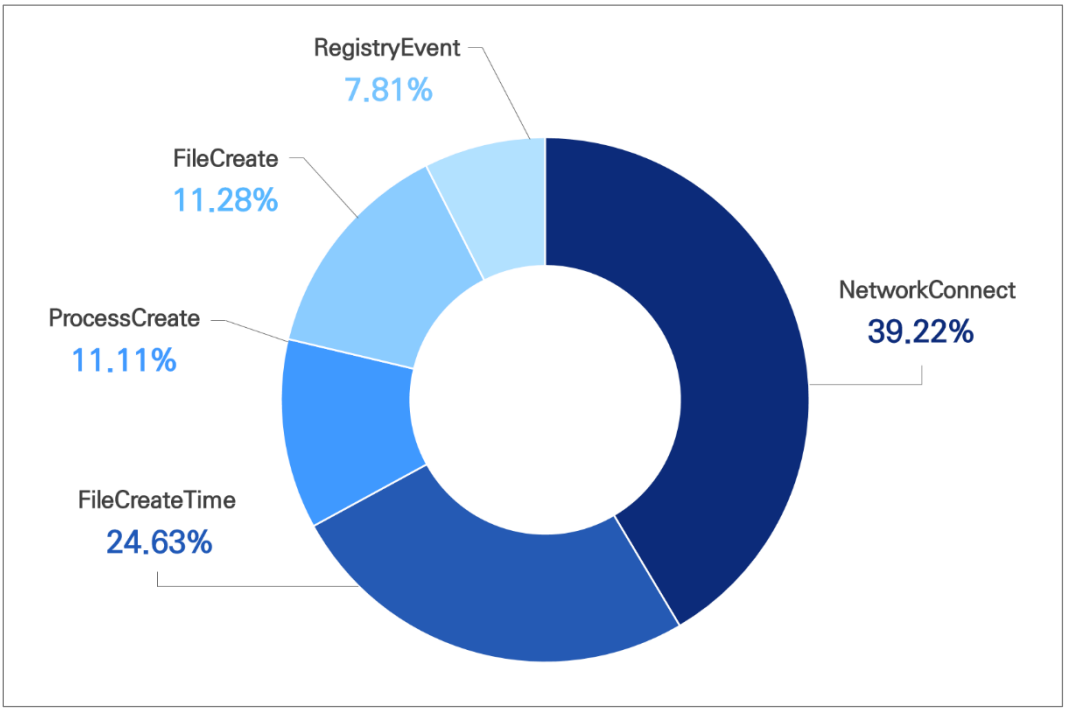

[그림 6] Sysmon만 설치한 환경에서 발생된 상위 5개 이벤트

[2] Sysmon을 별도 Configure 파일을 적용하지 않고 기본 “sysmon –i” 명령으로 설치했을 때를 의미

**SwiftOnSecurity의 sysmonconfig-export.xml** 파일을 적용해 로깅을 모니터링한 결과, 1일 평균 6~7MB의 이벤트가 발생했으며, 이벤트가 많이 발생할 수 있는 "ProcessCreate", "NetworkConnect", "RegistryEvent", "DNSQuery" 이벤트를 중점적으로 정의해 노이즈를 발생시킬 수 있는 많은 이벤트가 로깅 되지 않도록 구성했다. 다만, 2021 년 10 월 이후 Configure 파일이 업데이트되지 않아 최신 버전의 Sysmon 에서 지원하는 이벤트(EventID 23 이상)는 로깅 되지 않았다. 또한, 전체 로그에서 "NetworkConnect"와 "FileCreateTime" 이벤트가 가장 많이 로깅 되어 Rule 에 대한 최적화가 필요하다.

[표 16] SwiftOnSecurity의 sysmonconfig-export.xml 파일 적용 후 발생된 상위 5개 이벤트 목록

| 발생된 이벤트 비율 | 이벤트 ID | 이벤트 명 |
|---|---|---|
| 39.22% | 3 | NetworkConnect |
| 24.63% | 2 | FileCreateTime |
| 11.28% | 11 | FileCreate |
| 11.11% | 1 | ProcessCreate |
| 7.81% | 13 | RegistryEvent, Value set |

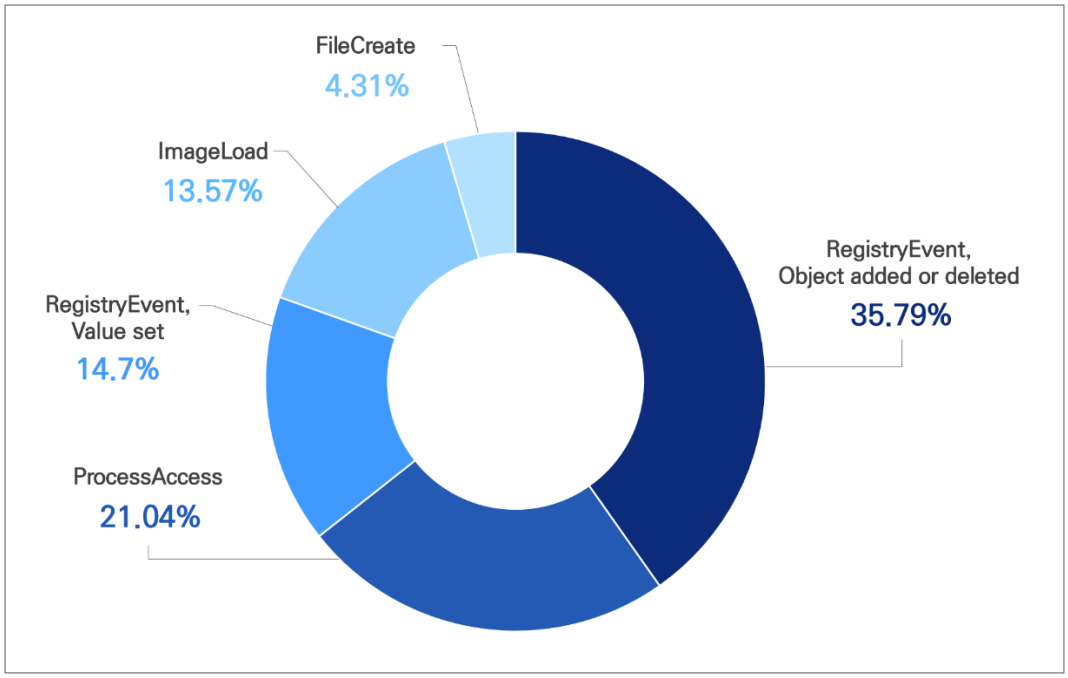



[그림 7] SwiftOnSecurity의 sysmonconfig-export.xml 파일 적용 후 발생된 상위 5개 이벤트

**Neo23x0 의 sysmonconfig-trace.xml** 파일을 적용해 로깅을 모니터링한 결과, 1 일 평균 600MB 의 이벤트가 발생했으며, Sysmon 에서 지원하는 대부분의 이벤트가 모두 로깅 되도록 설정되어 있어 다양한 이벤트가 로깅 된다. 이로 인해 다른 선행 연구의 Configure 를 적용했을 때보다 로깅 주기가 빠르기 때문에 로그 용량을 더 빨리 차지해 정상 이벤트가 위협을 탐지할 수 있는 중요 이벤트를 덮어씌울 수 있다는 한계가 있다. 적용 결과, 시스템 사용 중 레지스트리 변경 작업이 수시로 발생되어 "RegistryEvent" 이벤트가 가장 많이 로깅 되었다.

[표 17] Neo23x0의 sysmonconfig-trace.xml 파일 적용 후 발생된 상위 5개 이벤트 목록

| 발생된 이벤트 비율 | 이벤트 ID | 이벤트 명 |
|---|---|---|
| 35.79% | 12 | RegistryEvent, Object added or deleted |
| 21.04% | 10 | ProcessAccess |
| 14.7% | 13 | RegistryEvent, Value set |
| 13.57% | 7 | ImageLoad |
| 4.31% | 11 | FileCreate |

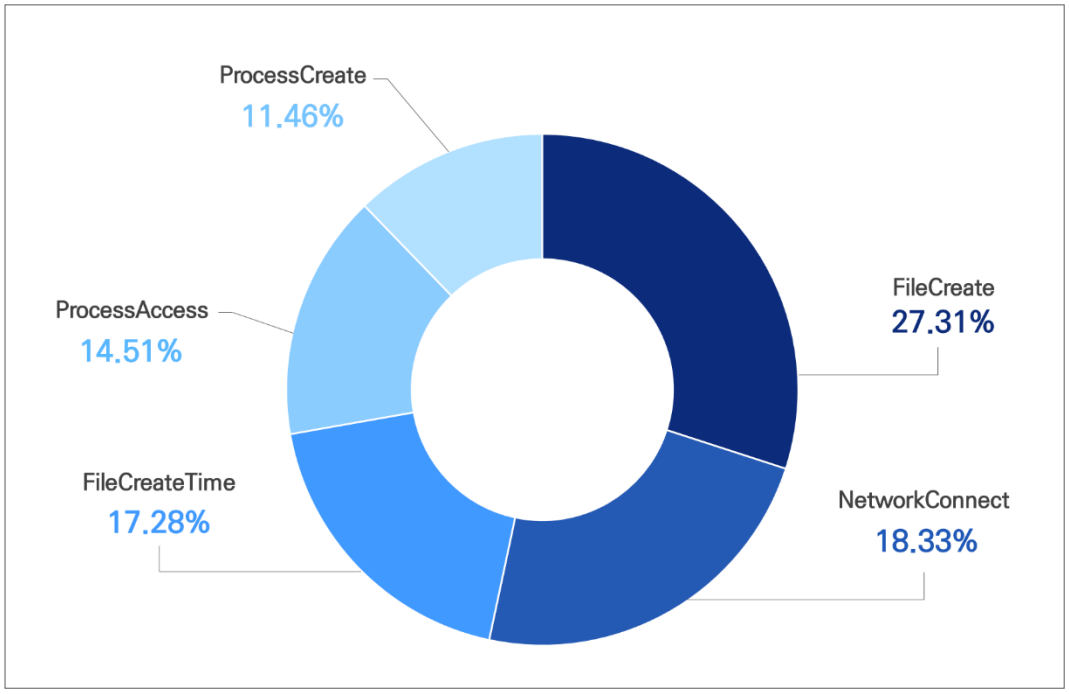



[그림 8] Neo23x0의 sysmonconfig-trace.xml 파일 적용 후 발생된 상위 5개 이벤트

**Neo23x0 의 sysmonconfig-export-block.xml** 파일을 적용해 로깅을 모니터링한 결과, 1 일 평균 8MB 의 이벤트가 발생했으며, 악성 파일과 드라이버 탐지의 정확성을 높이기 위해 이름이 아닌 해시 기반으로 Rule 을 반영했다. 또한, 전체 로그에서 “FileCreate”, “NetworkConnect”, “FileCreateTime” 이벤트가 가장 많이 로깅 되어 Rule 에 대한 최적화가 필요하다.

[표 18] Neo23x0의 sysmonconfig-export-block.xml 파일 적용 후 발생된 상위 5개 이벤트 목록

| 발생된 이벤트 비율 | 이벤트 ID | 이벤트 명 |
|---|---|---|
| 27.31% | 11 | FileCreate |
| 18.33% | 3 | NetworkConnect |
| 17.28% | 2 | FileCreateTime |
| 14.51% | 10 | ProcessAccess |
| 11.46% | 1 | ProcessCreate |

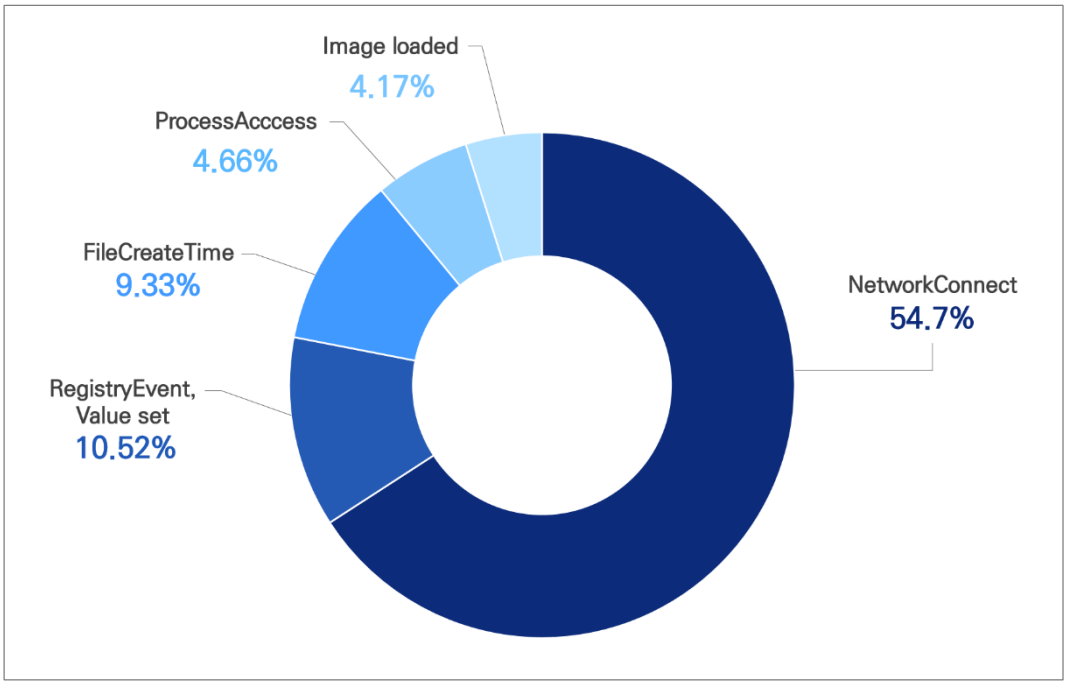



[그림 9] Neo23x0의 sysmonconfig-export-block.xml 파일 적용 후 발생된 상위 5개 이벤트

**olafhartong 의 sysmonconfig-mde-augment.xml** 파일을 적용해 로깅을 모니터링한 결과, 1 일 평균 15MB 의 이벤트가 발생했으며, Rule에 MITRE ATT&CK Techniques ID를 매칭 시켜 어떤 위협 행위를 탐지하기 위한 Rule 인지를 알 수 있도록 정의했다. 또한, 로깅 되도록 정의된 Rule 이 많아 정상적인 이벤트도 다수 로깅 되어 Rule 에 대한 추가적인 필터링이 필요하다.

[표 19] olafhartong의 sysmon-mde-augment.xml 파일 적용 후 발생된 상위 5개 이벤트 목록

| 발생된 이벤트 비율 | 이벤트 ID | 이벤트 명 |
|---|---|---|
| 54.7% | 3 | NetworkConnect |
| 10.52% | 13 | RegistryEvent, Value set |
| 9.33% | 2 | FileCreateTime |
| 4.66% | 10 | ProcessAccess |
| 4.17% | 7 | Image loaded |




[그림 10] olafhartong의 sysmon-mde-augment.xml 파일 적용 후 발생된 상위 5개 이벤트

# 4. 연구 내용

## 4.1. Sysmon Configure Rule 개요

실제 사이버 공격 사례를 분석해 공격자의 행위에 대한 전략을 표준화한 MITRE ATT&CK 프레임워크는 공격자의 행위를 식별할 수 있는 프레임워크로 발전해 국내에서도 사이버 위협을 탐지하고 대응하기 위해 많이 활용되고 있다. 다만, 대부분의 전술이 APT 공격에 조금 더 초점이 맞춰져 있어 국내를 타겟으로 한 공격을 탐지하고 대응하는 데 한계가 있다.

최근 수준 높은 공격자들은 공격 탐지를 회피하기 위해 자신이 수행한 공격을 파일 리스 공격이나 공격 도구 등을 완전 삭제해 철저하게 은닉하고 있다. 랜섬웨어와 같이 표면적으로 드러나는 공격이면 기업에서도 사고를 빠르게 인지할 수 있지만 로그 설정이 미흡한 인프라를 공격자가 침투해 해킹 경유지 등으로 활용하는 경우 로그 부재로 사고에 대한 원인과 영향을 규명하기에 한계가 있다.

본 연구에서는 다년간 침해사고 분석과 대응 경험에 대한 노하우를 살려 국내에서 발생하는 다양한 사이버 위협을 탐지하고 선행된 연구의 한계점을 보완하기 위한 방안을 제안하고자 한다.

### 4.1.1. 선행 연구와의 차별점

선행 연구된 Sysmon Configure 들은 Sysmon 을 EDR[3] 시스템으로 활용해 관제 목적의 모니터링에 중점을 두고 작성되었다. 그러나, BIT-sysmon-config 는 DFIR [4]관점으로 접근해 사이버 위협을 탐지 및 분석할 때 분석가가 유용한 정보를 확인할 수 있도록 Configure 를 작성했다. 선행 연구된 Rule 과의 차별점은 다음과 같다.

[표 20] 선행 연구된 Sysmon Configure Rule과의 차별점

| 번호 | 차별점 |
|---|---|
| 1 | 다양한 사이버 위협 시나리오를 포함해 더 넓은 이벤트 범위 탐지 가능 |
| 2 | 국내에 맞는 환경을 접목시켜 더 명확하고 상세하게 위협 탐지 가능 |
| 3 | DFIR 을 지원하기 위해 설계되어, 분석가가 사고를 신속하게 분석하고 대응할 수 있도록 유용한 정보 제공 |
| 4 | 다년간 경험한 침해사고에서 파악된 주요 공격자 행위를 반영해 현실적인 사이버 위협에 대응할 수 있도록 개선 |
| 5 | 설정을 최적화해 네트워크와 동작되는 시스템 성능에 부담을 주지 않으면서 효과적으로 사이버 위협 탐지 |

[3] Endpoint Detection and Response의 약어로, 엔드포인트에서 발생하는 위협 행위를 실시간으로 탐지하고 이를 분석 및 대응하는 솔루션

[4] Digital Forensics and Incident Response의 약어로, 컴퓨터 및 네트워크 시스템에서 발생한 침해사고를 대응하고 이에 대한 조사를 수행하는 분야

### 4.1.2. 이벤트 구성 별 주요 관점

Sysmon 을 통해 로깅할 수 있는 이벤트에서 공격자의 행위를 구체적으로 식별할 수 있도록 중점적으로 Rule 작성이 필요한 이벤트 구성 요소는 다음과 같다.

[표 21] 중점적으로 Rule 작성이 필요한 이벤트 구성 요소

| 이벤트 ID | 이벤트 명 | 구성 요소 구분 | 설명 |
|---|---|---|---|
| 1 | Process Create | Image | 악성 프로세스가 실행되는 주요 경로 |
| | | OriginalFileName | 공격자가 주로 활용하는 악성 프로세스 명 |
| | | CommandLine | 명령어를 통한 행위 |
| | | Hashes | 공격자가 주로 활용하는 악성 프로세스 해시 |
| | | ParentImage | CMD, PowerShell 을 통해 실행되는 프로세스 |
| 2 | File creation time changed | Image | 공격자가 주로 활용하는 경로 |
| | | TargetFilename | 생성 일시가 수정된 실행 파일(.exe) 명 탐지 |
| 3 | Network connection detected | Image | 공격자가 네트워크를 통해 시스템 탐색 시 주로 사용하는 프로세스 |
| | | DestinationPort | 공격자가 주로 활용하는 알려진 포트 (SSH, Telnet, RDP 등) |
| 5 | Process terminated | Image | 공격자가 주로 활용하는 경로에서 종료된 프로세스 |
| 6 | Driver loaded | Hashes | 공격자가 주로 활용하는 악성 드라이버 해시 |
| 7 | Image loaded | ImageLoaded | 공격자가 주로 활용하는 DLL 명 |
| | | Hashes | 공격자가 주로 활용하는 악성 DLL 해시 |
| 8 | CreateRemoteThread detected | SourceImage | 시스템 경로에서 발생하는 원격 스레드 생성 탐지 |
| 10 | Process accessed | SourceImage | 공격자가 주로 활용하는 경로 |
| | | TargetImage | Process Injection, Credential Dumping 에 주로 활용되는 파일 |
| | | GrantedAccess | Process Injection, Credential Dumping 공격 등을 특정할 수 있는 Access Flags |
| 11 | File created | TargetFilename | 공격자가 주로 활용하는 경로와 확장자 |
| 12, 13 | Registry Event | TargetObject | 공격자가 주로 활용하는 레지스트리 경로 |
| 15 | File stream create | TargetFilename | 공격자가 주로 활용하는 확장자 |
| | | Image | 웹 브라우저를 통한 다운로드 행위 탐지 |
| 17, 18 | PipeEvent | PipeName | 악성 행위 시 주로 사용되는 Pipe 명 |
| 22 | DNS query | QueryName | 네트워크 노이즈를 제거하기 위한 DNS Query 명 |
| 23 | File Delete archived | TargetFilename | 공격자가 주로 활용하는 확장자 |
| 25 | Process Tampering | - | 프로세스 변조 기술을 탐지하기 위해 모든 이벤트 기록 |
| 29 | File Executable Detected | Image | PowerShell 을 통한 실행 파일 다운로드 |

## 4.2. 행위에 따른 Sysmon 로깅 테스트

다년간 침해사고 분석 경험을 반영해 공격자가 주로 수행하는 행위를 시뮬레이션하고, 이에 따른 Sysmon 로그를 수집해 해당 행위가 Sysmon 로그에 어떻게 기록되는지를 분석해 공격자의 행위를 탐지하기 위한 이벤트를 식별했다. 테스트를 수행한 행위 목록은 다음과 같다.

[표 22] 주요 공격 행위에 따른 Sysmon 로깅 테스트

| 대구분 | 소구분 | 행위 |
|---|---|---|
| 파일 | 생성 | 웹 서버를 통한 악성 파일 생성 (IIS, ASP) |
| | | 웹 서버를 통한 악성 파일 생성 (Apache, PHP) |
| | | Windows Explorer 를 통한 파일 생성 |
| | 수정 | Windows Explorer 를 통한 파일 수정 |
| | | 텍스트 편집 프로그램을 통한 파일 수정 |
| | 다운로드 | Chrome 브라우저를 통한 다운로드 |
| | | Microsoft Edge 브라우저를 통한 다운로드 |
| | | Naver Whale 브라우저를 통한 다운로드 |
| | | PowerShell 명령을 통한 다운로드 |
| | 삭제 | 일반 삭제 (Delete) |
| | | 완전 삭제 (Shift + Delete) |
| | 열람 | - |
| | 실행 | 공격 도구 실행 |
| | 복사 | Ctrl + c / Ctrl + v |
| | | 우 클릭을 통한 복사/붙여넣기 |
| | 압축 해제 (Explorer) | Windows Explorer 를 통한 압축 해제 |
| | 압축 해제 (Tools) | 압축 유틸리티를 활용해 압축 해제 |
| 폴더 | 생성 | Windows Explorer 를 통한 폴더 생성 |
| | 수정 | Windows Explorer 를 통한 폴더 수정 |
| | 삭제 | 일반 삭제 (Delete) |
| | | 완전 삭제 (Shift + Delete) |
| | 열람 | - |
| | 복사 | Ctrl + c / Ctrl + v |
| | | 우 클릭을 통한 복사/붙여넣기 |
| | 공유 폴더 | 연결 |

| 대구분 | 소구분 | 행위 |
|---|---|---|
| 폴더 | 공유 폴더 | 연결 해제 |
| | | 접근 |
| | | 폴더 열람 |
| 계정 | 생성 | CMD 를 통한 생성 |
| | | Windows 설정을 통한 생성 |
| | 수정 | 패스워드 변경 |
| | | 그룹 변경 |
| | | 계정 명 변경 |
| | 삭제 | CMD 를 통한 삭제 |
| | | Windows 설정을 통한 삭제 |
| | 권한 상승 | 도구를 통한 상승 (BadPotato) |
| | | 도구를 통한 상승 (JuicyPotato) |
| 로그온 | Inbound | 잠금 해제 |
| | | RDP |
| | | Reverse Shell |
| | Outbound | RDP |
| 로그온 시도 | Inbound | 무차별 대입 공격 |
| 로그오프 | - | - |
| 프로그램 | 설치 | - |
| | 삭제 | - |
| 서비스 | 설치 | - |
| | 수정 | - |
| | 삭제 | - |
| 작업스케줄 | 생성 | CMD 를 통한 생성 |
| | | 작업 스케줄러(taskschd.msc)를 통한 생성 |
| | 수정 | CMD 를 통한 수정 |
| | | 작업 스케줄러(taskschd.msc)를 통한 수정 |
| | 삭제 | CMD 를 통한 삭제 |
| | | 작업 스케줄러(taskschd.msc)를 통한 삭제 |
| 감사정책 | 수정 | - |

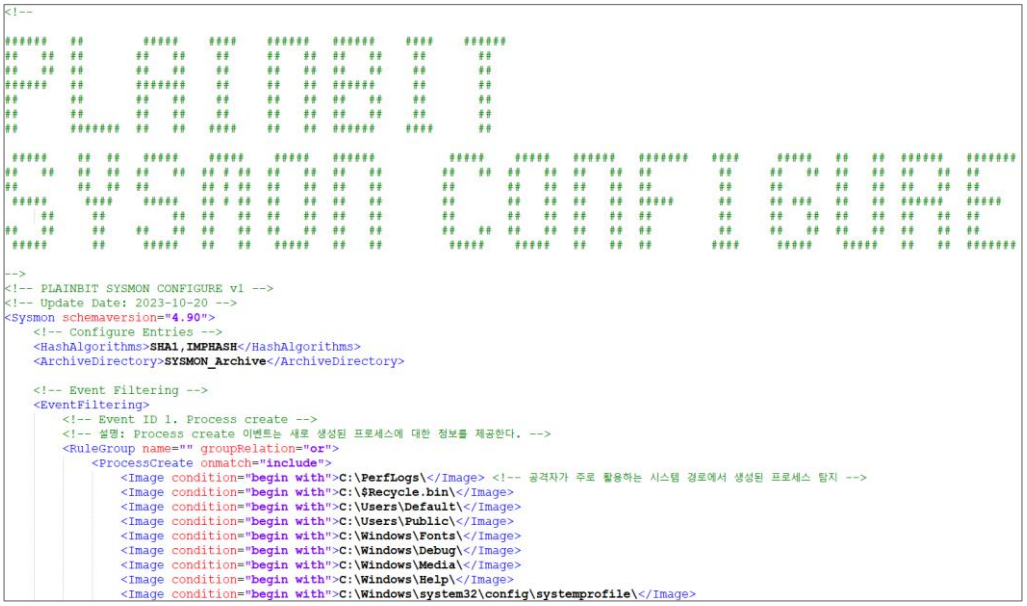
| 대구분 | 소구분 | 행위 |
|---|---|---|
| 레지스트리 | 생성 | CMD 를 통해 생성 |
| | | 레지스트리 편집기 (regedit)를 통해 생성 |
| | 수정 | CMD 를 통해 Name, Value 수정 |
| | | 레지스트리 편집기 (regedit)를 통해 Name, Value 수정 |
| | 삭제 | CMD 를 통해 Name, Value 수정 |
| | | 레지스트리 편집기 (regedit)를 통해 Name, Value 수정 |
| 네트워크 | 스캐닝 | - |
| | RRAS 연결 | - |
| | VPN 연결 | - |
| 명령 실행 | PowerShell | PowerShell 을 통해 명령 실행 |
| | | .ps1 스크립트 실행 |
| | CMD | CMD 를 통해 명령 실행 |
| | | .bat 스크립트 실행 |
| | xp_cmdshell | - |
| | psexec | PowerShell 실행 |
| | | CMD 실행 |
| 안티포렌식 | 백신 | 실시간 검사 비활성화 |
| | 백신 | 탐지 제외 (파일, 폴더) |
| | 백신 | 악성 파일 격리 |
| | 이벤트로그 삭제 | wevtutil.exe 명령 실행 |
| | 이벤트로그 삭제 | 이벤트 뷰어(eventvwr.msc) 활용 |
| | 시간 조작 | 파일 생성 일시 변경 |
| | | 파일 수정 일시 변경 |
| 랜섬웨어 | 비트라커 | - |
| 외장 저장매체 | 연결 | - |
| | 연결 해제 | - |
| | 파일 생성 | 우클릭을 통한 생성 |
| | | CMD 를 통한 생성 |
| | 파일 수정 | 파일 명 변경 |
| 외장 저장매체 | 파일 수정 | 파일 내용 변경 |

| 대구분 | 소구분 | 행위 |
|---|---|---|
| | 파일 삭제 | 우클릭을 통한 삭제 |
| | | 완전 삭제 (Shift + Delete) |
| | 파일 복사 | Host to USB (Ctrl + c / Ctrl + v) |
| | | Host to USB (드래그 앤 드롭) |
| 네트워크 | 원격 데스크톱 연결 | Reverse RDP |

| 소구분 | 구분 | 비고 | | 이벤트 ID | 이벤트 구분 | 이벤트 설명 |
|---|---|---|---|---|---|---|
| 파일 | | PowerShell 명령을 통한 다운로드(3) | | 22 | Dns query (rule: DnsQuery) | Dns query:<br>RuleName: -<br>UtcTime: 2023-07-31 06:10:47.646<br>ProcessGuid: {fa5c402e-4fa1-64c7-1f05-000000000900}<br>ProcessId: 8768<br>QueryName: **download.anydesk.com**<br>QueryStatus: 0<br>QueryResults: ::ffff:188.40.104.135;192.5.6.30;192.33.14.30;192.26.92.30;192.31.80.30;192.1<br>Image: **C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe**<br>User: DESKTOP-Q6DACOA\user |
| | | PowerShell 명령을 통한 다운로드(4)<br>> 다운로드한 파일이 실행 파일이면 발생 | | 29 | File Executable Detected (rule: FileExecutableDetected) | File Executable Detected:<br>RuleName: -<br>UtcTime: 2023-07-31 06:10:50.074<br>ProcessGuid: {fa5c402e-4fa1-64c7-1f05-000000000900}<br>ProcessId: 8768<br>User: DESKTOP-Q6DACOA\user<br>Image: **C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe**<br>TargetFilename: **C:\Users\user\Desktop\AnyDesk.exe**<br>Hashes: SHA1=D579D188488859614E219AFA6332D410E0CA71FC3,MD5=30C9C57AA5700 |
| | 삭제 | 파일 / 일반 삭제(Delete) | | 11 | File created (rule: FileCreate) | File created:<br>RuleName: -<br>UtcTime: 2023-07-28 00:53:55.166<br>ProcessGuid: {fa5c402e-d081-64c1-8000-000000000600}<br>ProcessId: 5644<br>Image: **C:\Windows\Explorer.EXE**<br>TargetFilename: C:**$Recycle.Bin**\S-1-5-21-2740812108-3388594264-3268411955-1001\$I2K<br>CreationUtcTime: 2023-07-28 00:53:55.166<br>User: DESKTOP-Q6DACOA\user |
| | | 파일 / 완전 삭제(Shift delete) | | 23 | File Delete archived (rule: FileDelete) | File Delete archived:<br>RuleName: -<br>UtcTime: 2023-07-28 00:50:23.072<br>ProcessGuid: {fa5c402e-d081-64c1-8000-000000000600}<br>ProcessId: 5644<br>User: DESKTOP-Q6DACOA\user<br>Image: **C:\Windows\Explorer.EXE**<br>TargetFilename: **C:\Users\user\Desktop\test.txt**<br>Hashes: SHA1=A94A8FE5CCB19BA61C4C0873D391E987982FBBD3,MD5=098F6BCD4621<br>IsExecutable: false<br>Archived: false - shredded file with pattern 0x74736574 |
| | 열람 | 열람에 사용되는 프로그램의 프로세스가 생성되면서, 열람 대상을 알 수 있 | | 1 | Process Create (rule: ProcessCreate) | Process Create:<br>RuleName: -<br>UtcTime: 2023-07-31 02:49:40.994<br>ProcessGuid: {fa5c402e-2144-64c7-3a03-000000000900}<br>ProcessId: 10752<br>Image: **C:\Windows\System32\notepad.exe**<br>FileVersion: 10.0.19041.1865 (WinBuild.160101.0800)<br>Description: Notepad<br>Product: Microsoft® Windows® Operating System<br>Company: Microsoft Corporation<br>OriginalFileName: NOTEPAD.EXE<br>CommandLine: "C:\Windows\system32\NOTEPAD.EXE" **C:\Users\user\Desktop\b.txt**<br>CurrentDirectory: C:\Users\user\Desktop\<br>User: DESKTOP-Q6DACOA\user<br>LogonGuid: {fa5c402e-6d0f-64c3-319e-010000000000}<br>LogonId: 0x19E31<br>TerminalSessionId: 1<br>IntegrityLevel: Medium<br>Hashes: SHA1=D05DEFE2C8EFEF10ED5F1361760FA0AE41FA79F5,MD5=27F71B12CB58<br>ParentProcessGuid: {fa5c402e-6d13-64c3-7b00-000000000900} |

[그림 11] 행위에 따른 Sysmon 로깅 테스트 결과 정리 시트 일부

## 4.3. BIT-sysmon-config[5]



```
<!--
######  ##       #####   ####   ######  ######   ####   ######
##   ##  ##      ##   ##   ##    ##   ##  ##   ##      ##
##   ##  ##      ##   ##   ##    ##   ##  ##   ##      ##
######   ##      #######   ##    ##   ######    ##      ##
##       ##      ##   ##   ##    ##   ##  ##   ##      ##
##       ##      ##   ##   ##    ##   ##  ##   ##      ##
##       #######  ##   ##  ####   ##   ##  ##  ######  ####     ##

 #####    ##  ##   #####    #####    #####   ######         #####    #####   ######   #######  ####     #####   ##   ##  ######   #######
##   ##   ##  ##  ##   ##  ## # ##  ##   ##  ##   ##        ##   ##  ##   ##  ##   ##  ##        ##     ##   ##  ##   ##  ##   ##  ##   ##
##        ## ##   ##   ##  ## # ##  ##   ##  ##   ##        ##       ##   ##  ##   ##  ##        ##     ##   ##  ##   ##  ##   ##  ##   ##
 #####    ####    #####   ## # ##  ##   ##  ##   ##        ##       ##   ##  ##   ##  #####     ##     ## ###  ##   ##  ######   #####
     ##    ##         ##  ##   ##  ##   ##  ##   ##        ##       ##   ##  ##   ##  ##        ##     ##   ##  ##   ##  ##   ##  ##
##   ##    ##    ##   ##  ##   ##  ##   ##  ##   ##        ##   ##  ##   ##  ##   ##  ##        ##     ##   ##  ##   ##  ##   ##  ##
 #####     ##     #####   ##   ##  #####   ##   ##         #####    #####   ##   ##  ##       ####    #####    #####   ##   ##  #######

-->
<!-- PLAINBIT SYSMON CONFIGURE v1 -->
<!-- Update Date: 2023-10-20 -->
<Sysmon schemaversion="4.90">
    <!-- Configure Entries -->
    <HashAlgorithms>SHA1,IMPHASH</HashAlgorithms>
    <ArchiveDirectory>SYSMON_Archive</ArchiveDirectory>

    <!-- Event Filtering -->
    <EventFiltering>
        <!-- Event ID 1. Process create -->
        <!-- 설명: Process create 이벤트는 새로 생성된 프로세스에 대한 정보를 제공한다. -->
        <RuleGroup name="" groupRelation="or">
            <ProcessCreate onmatch="include">
                <Image condition="begin with">C:\PerfLogs\</Image> <!-- 공격자가 주로 활용하는 시스템 경로에서 생성된 프로세스 탐지 -->
                <Image condition="begin with">C:\$Recycle.bin\</Image>
                <Image condition="begin with">C:\Users\Default\</Image>
                <Image condition="begin with">C:\Users\Public\</Image>
                <Image condition="begin with">C:\Windows\Fonts\</Image>
                <Image condition="begin with">C:\Windows\Debug\</Image>
                <Image condition="begin with">C:\Windows\Media\</Image>
                <Image condition="begin with">C:\Windows\Help\</Image>
                <Image condition="begin with">C:\Windows\system32\config\systemprofile\</Image>
```

[그림 12] 작성된 BIT-sysmon-config 화면 일부

BIT-sysmon-config Rule에 “Network connection”, “Image Loading”, “CRL Checking”, “DNS lookup” 설정 모두 활성화했으며, 정의된 Rule 목록은 다음과 같다.

[표 23] BIT-sysmon-config Rule에 정의된 Rule 목록

| 이벤트 ID | 이벤트 명 | include | exclude | 비고 |
|---|---|---|---|---|
| 1 | ProcessCreate | 461 | 148 | - |
| 2 | FileCreateTime | 4 | 10 | - |
| 3 | NetworkConnect | 37 | 13 | - |
| 5 | ProcessTerminate | 8 | 5 | - |
| 6 | DriverLoad | 72 | 3 | - |
| 7 | ImageLoad | 70 | 15 | - |
| 8 | CreateRemoteThread | 2 | 9 | - |
| 9 | RawAccessRead | - | - | 기록하지 않음 |
| 10 | ProcessAccess | 27 | 15 | - |
| 11 | FileCreate | 48 | 22 | - |
| 12, 13, 14 | RegistryEvent | 43 | 3 | - |
| 15 | FileCreateStreamHash | 13 | - | 모두 기록 |
| 17, 18 | PipeEvent | 12 | 3 | - |
| 19, 20, 21 | WmiEvent | 0 | - | 모두 기록 |

[5] https://github.com/Plainbit/BIT-sysmon-config

| 이벤트 ID | 이벤트 명 | include | exclude | 비고 |
|---|---|---|---|---|
| 22 | DnsQuery | 0 | 92 | - |
| 23 | FileDelete | 40 | 4 | - |
| 24 | ClipboardChange | - | - | 기록하지 않음 |
| 25 | ProcessTampering | 0 | - | 모두 기록 |
| 26 | FileDeleteDetected | - | - | 기록하지 않음 |
| 27 | FileBlockExecutable | - | - | 기록하지 않음 |
| 28 | FileBlockShredding | - | - | 기록하지 않음 |
| 29 | FileExecutableDetected | 1 | 0 | - |

이벤트 ID 별 Sysmon Configure Rule 작성 기준은 다음과 같다.

[표 24] 이벤트 ID 별 Sysmon Configure Rule 작성 기준

| 이벤트 ID | 이벤트 명 | 작성 기준 |
|---|---|---|
| 1 | ProcessCreate | • 생성되는 프로세스에서 공격자의 흔적 확인<br>• 정상 프로세스에 의한 이벤트 로깅 최적화 |
| 2 | FileCreateTime | • 공격자에 의한 공격 도구 시간 조작 행위 (생성 일시) 탐지<br>• 정상 프로세스에 의한 생성 일시 수정 이벤트 로깅 최적화 |
| 3 | NetworkConnect | • 네트워크 연결을 통해 수행하는 행위 탐지<br>• 정상적인 네트워크 연결의 이벤트 로깅 최적화 |
| 5 | ProcessTerminate | • 종료되는 프로세스에서 공격자 흔적 확인 |
| 6 | DriverLoad | • 시스템에 로드되는 드라이버에서 공격자의 흔적 확인<br>• 정상적인 드라이버가 시스템에 로드 되면서 로깅 되는 이벤트 최적화 |
| 7 | ImageLoad | • 프로세스에 의해 로드되는 모듈에서 공격자 흔적 확인<br>• 정상 DLL 이 시스템에 로드 되면서 로깅 되는 이벤트 최적화 |
| 8 | CreateRemoteThread | • 탐지 회피를 위해 정상 프로세스에 악성 프로세스 스레드 생성 흔적 확인<br>• 정상 시스템 프로세스가 다른 프로세스에서 스레드 생성 시 로깅 되는 이벤트 최적화 |
| 9 | RawAccessRead | • 프로세스가 디스크에 대한 Raw Sector 수준의 읽기 작업 수행하는 것을 탐지할 수 있으나 시스템 부하를 유발할 수 있어 비활성화 |
| 10 | ProcessAccess | • 자격증명 획득, 탐지 회피 등을 위해 정상 프로세스 접근 흔적 확인<br>• 정상 프로세스가 다른 프로세스에 접근 시 로깅 되는 이벤트 최적화 |
| 11 | FileCreate | • 시스템 공격에 활용되는 파일 생성 흔적 확인<br>• 정상 프로세스가 파일을 생성해 로깅 되는 이벤트 최적화 |
| 12, 13, 14 | RegistryEvent | • 레지스트리 키 또는 값 생성/삭제/수정 흔적 확인<br>• 정상 프로세스가 레지스트리 키 또는 값을 생성/수정/삭제 시 로깅 되는 이벤트 최적화 |

| 이벤트 ID | 이벤트 명 | 작성 기준 |
|---|---|---|
| 15 | FileCreateStreamHash | • 웹 브라우저를 통해 공격에 활용되는 파일 생성 흔적 확인 |
| 17, 18 | PipeEvent | • 악성 프로그램을 통해 생성, 연결되면서 명명된 Pipe 흔적 확인<br>• 시스템 프로세스에서 생성, 연결되면서 명명된 Pipe 가 로깅 되는 이벤트 최적화 |
| 19, 20, 21 | WmiEvent | • 공격자가 WMI 를 사용해 시스템 정보 수집, 자동 실행, 명령제어 채널로 활용하는 흔적을 확인하기 위해 모든 WMI 이벤트 로깅 |
| 22 | DnsQuery | • 공격 과정에서 실행되는 DNS 쿼리를 탐지하기 위해 모든 DNS 쿼리 이벤트를 로깅하지만 정상 DNS 쿼리의 빈도가 높아 사용자가 업무를 수행하면서 로깅 되는 정상 DNS 쿼리 이벤트 최적화 |
| 23 | FileDelete | • 공격 과정에서 파일 삭제 흔적 확인<br>• 정상 프로세스에서 주기적으로 파일 삭제 시 로깅 되는 이벤트 최적화 |
| 24 | ClipboardChange | • 시스템 클립보드 내용이 변경될 때 이벤트로 로깅 되어 공격자의 흔적을 추가로 식별할 수 있지만 변경된 데이터가 ArchiveDirectory 에 무제한 저장되어 시스템 성능에 영향을 끼치는 것을 방지 위해 비활성화 |
| 25 | ProcessTampering | • 프로세스 기반 탐지 회피 목적으로 악성 프로세스를 정상 프로세스에 변조하는 모든 흔적을 확인하기 위해 모든 프로세스 변조 이벤트 탐지 |
| 26 | FileDeleteDetected | • 공격자의 파일 삭제 흔적을 이벤트로 로깅하기 때문에 비활성화 |
| 27 | FileBlockExecutable | • 해당 이벤트로 시스템의 특정 경로에 실행 파일이 생성되는 것을 탐지 및 차단이 가능하나, 임의로 설정 시 시스템에 영향을 끼칠 수 있어 비활성화 |
| 28 | FileBlockShredding | • 해당 이벤트로 시스템의 특정 경로에서 파일이 완전 삭제되는 행위를 탐지 및 차단이 가능하나, 임의 설정 시 시스템에 영향을 끼칠 수 있기 때문에 비활성화 |
| 29 | FileExecutableDetected | • 공격에 활용되는 실행 파일을 생성하는 흔적을 확인할 수 있으나, FileCreate 이벤트를 통해 공격자가 시스템에 파일을 생성하는 흔적 로깅<br>• 따라서, PowerShell 에서 생성한 실행 파일만 로깅 |

### 4.3.1. 주요 Rule 설명

이벤트 ID 별로 반영된 주요 Rule 은 다음과 같다.

[표 25] 이벤트 ID 별 반영된 주요 Rule

| 이벤트 ID | 이벤트 명 | 주요 Rule 목록 |
|---|---|---|
| 1 | ProcessCreate | • 공격자가 주로 활용하는 시스템 경로에서 생성된 프로세스 탐지<br>• 공격자가 시스템 정찰, 자격 증명 탈취, 도구 전송, UAC 우회, 간접 명령 실행, 탐지 회피를 위해 주로 사용하는 프로세스 탐지<br>• xp_cmdshell, psexec 등을 이용해 원격에서 실행되는 명령어 탐지<br>• 공격자가 자신의 흔적을 지우기 위해 이벤트 로그 삭제 시 탐지<br>• 공격자가 공격에 활용하기 위한 계정 생성, 비밀번호 변경, 그룹 변경, 삭제하는 행위 탐지<br>• 지속성 확보를 위해 서비스와 작업 스케줄러 생성/수정/삭제 탐지<br>• Windows Defender 우회하기 위해 설정 변경 시 탐지<br>• 악성 파일 이름을 변조해 프로세스 명 기반 Rule 우회를 탐지하기 위해 악성 파일 Hash 탐지<br>• Windows 시스템에서 백그라운드로 상시 동작되는 프로세스 예외 처리<br>• Microsoft Office, .Net 에서 백그라운드로 상시 동작되는 프로세스 예외 처리 |
| 2 | FileCreateTime | • “C:\Temp” 하위 경로에서 발생되는 생성 일시 변조 탐지<br>• “C:\Windows\Temp” 하위에 경로에서 발생하는 생성 일시 변조 탐지<br>• “C:\Users” 하위 경로에서 발생하는 생성 일시 변조 탐지<br>• “.exe” 확장자에 대한 생성 일시 변조 탐지<br>• Setup, Install, Update 관련 프로세스가 생성 일시 수정 이벤트 예외 처리<br>• 한국에서 주로 사용되는 업무 협업 프로그램에서 생성 시간을 수정하는 이벤트 예외 처리 |
| 3 | NetworkConnect | • 공격에 활용되는 포트 연결 탐지<br>• 공격자가 네트워크를 통한 시스템 탐색에 주로 사용하는 프로세스 탐지<br>• PowerShell 로 네트워크에서 파일 다운로드 행위 탐지 |
| 5 | ProcessTerminate | • 공격자가 주로 활용하는 시스템 경로에서 종료된 프로세스 탐지<br>• 한국에서 주로 사용되는 업무 협업 프로그램 프로세스 종료 이벤트 예외 처리 |
| 6 | DriverLoad | • 시스템에 로드되는 악성 드라이버 해시 탐지<br>• Microsoft, Intel 의 시그니처를 가지는 드라이버 로드 이벤트 예외 처리 |
| 7 | ImageLoad | • 공격자가 주로 활용하는 시스템 경로에서 로드되는 DLL 탐지<br>• 프로세스 인젝션에 활용되는 DLL 로드 탐지<br>• 시스템에 로드되는 악성 DLL 해시 탐지<br>• OneDrive 프로세스에서 정상적인 DLL 이 로드되는 이벤트 예외 처리<br>• 시스템 프로세스에서 정상적인 DLL 이 로드되는 이벤트 예외 처리 |
| 8 | CreateRemoteThread | • 시스템 및 네트워크 드라이브에서 발생되는 모든 원격 스레드 생성 탐지<br>• 기본적으로 원격 스레드를 생성하는 시스템 프로세스 예외 처리 |

| 이벤트 ID | 이벤트 명 | 주요 Rule 목록 |
|---|---|---|
| 10 | ProcessAccess | • 공격자가 주로 활용하는 시스템 경로에서 프로세스 접근 탐지<br>• 공격자가 자격 증명 획득에 주로 사용하는 시스템 프로세스 패턴 탐지<br>• CobaltStrike 도구 사용 시 발생되는 패턴 탐지<br>• Windows Defender 프로세스가 다른 프로세스에 접근하는 이벤트 예외 처리<br>• VMWare 프로세스가 다른 프로세스에 접근하는 이벤트 예외 처리 |
| 11 | FileCreate | • 공격자가 주로 사용하는 시스템 경로에서 생성된 파일 탐지<br>(특히, “C:\Windows” 하위에 생성되는 파일 모두 탐지해 시스템 파일로 위장하려는 행위 탐지)<br>• 악성 파일에서 주로 사용되는 확장자를 가진 파일 생성 탐지<br>• 지속성을 위해 작업 스케줄러 생성 탐지<br>• 시스템 시작 시 자동으로 시작되는 폴더에 생성되는 파일 탐지<br>• IIS 웹 서버 프로세스를 통해 생성된 웹셸 탐지<br>• Windows Defender 에서 악성 파일이 탐지되어 저장된 격리 및 탐지 정보 탐지<br>• 정상적인 프로세스가 임시 파일을 수시로 생성하는 이벤트 예외 처리 |
| 12, 13, 14 | RegistryEvent | • 레지스트리 수정을 통해 Windows Defender 비활성화 탐지<br>• 시스템 시작 시 자동으로 실행되는 항목 수정 탐지<br>• 원격 데스크톱 연결 시 레지스트리가 수정되는 것을 탐지<br>• 원격 데스크톱에서 사용하는 포트 변경 탐지<br>• LSA 보호에 대한 설정 변경 탐지<br>• 감사 정책 변경 탐지<br>• 시스템에 USB 가 연결 및 연결 해제 탐지<br>• OneDrive 프로세스에서 발생되는 레지스트리 이벤트 예외 처리<br>• 정상적인 시스템 프로세스에서 발생되는 레지스트리 이벤트 예외 처리 |
| 15 | FileCreateStreamHash | • 악성 파일에서 주로 사용되는 확장자를 가진 파일 생성 탐지<br>(.bat, .cmd, .exe, .lnk, .vbs, .dll, reg, .ps1, hta)<br>• 주요 웹 브라우저를 통해 파일 다운로드 탐지<br>(Chrome, Edge, Whale)<br>• 그 외 명명된 모든 파일 스트림 생성 탐지 |
| 17, 18 | PipeEvent | • 자격 증명 획득 시 주로 사용되는 명명된 Pipe 탐지<br>• 악성 프로세스에서 주로 사용되는 명명된 Pipe 탐지<br>• CobaltStrike 도구에서 주로 사용되는 명명된 Pipe 탐지<br>• SMB 에서 주로 사용되는 명명된 Pipe 탐지<br>• 정상적인 시스템 프로세스가 생성, 연결하는 명명된 Pipe 이벤트 예외 처리 |
| 22 | DnsQuery | • Microsoft 프로그램 사용 중 발생하는 DNS 쿼리 이벤트 예외 처리<br>• 웹 브라우저 사용 중 발생하는 DNS 쿼리 이벤트 예외 처리<br>• 한국에서 주로 사용되는 프로그램에서 발생하는 DNS 쿼리 이벤트 예외 처리 |
| 23 | FileDelete | • Office 문서 확장자를 가진 파일 삭제 탐지<br>• 스크립트 및 페이로드 확장자를 가진 파일 삭제 탐지 |

```
<Rule name="Account Created" groupRelation="and"> <!-- 계정이 생성되는 것을 탐지 -->
    <OriginalFileName condition="is">net.exe</OriginalFileName>
    <CommandLine condition="contains all">user;/add</CommandLine>
</Rule>
<Rule name="Account Password Changed" groupRelation="and"> <!-- 계정 비밀번호가 변경되는 것을 탐지 -->
    <OriginalFileName condition="is">net.exe</OriginalFileName>
    <CommandLine condition="contains all">user;*</CommandLine>
</Rule>
<Rule name="Group of Accounts Changed" groupRelation="and"> <!-- 계정의 그룹이 변경되는 것을 탐지 -->
    <OriginalFileName condition="is">net.exe</OriginalFileName>
    <CommandLine condition="contains all">localgroup;/add</CommandLine>
</Rule>
<Rule name="Account Deleted" groupRelation="and"> <!-- 계정이 삭제되는 것을 탐지 -->
    <OriginalFileName condition="is">net.exe</OriginalFileName>
    <CommandLine condition="contains all">user;/delete</CommandLine>
</Rule>

<Rule name="Service Created" groupRelation="and"> <!-- 서비스가 생성되는 것을 탐지 -->
    <OriginalFileName condition="is">sc.exe</OriginalFileName>
    <CommandLine condition="contains">create</CommandLine>
</Rule>
<Rule name="Service Modified" groupRelation="and"> <!-- 서비스가 수정되는 것을 탐지 -->
    <OriginalFileName condition="is">sc.exe</OriginalFileName>
    <CommandLine condition="contains">config</CommandLine>
</Rule>
<Rule name="Service Deleted" groupRelation="and"> <!-- 서비스가 삭제되는 것을 탐지 -->
    <OriginalFileName condition="is">sc.exe</OriginalFileName>
    <CommandLine condition="contains">delete</CommandLine>
</Rule>

<Rule name="Task Scheduler Created" groupRelation="and"> <!-- 작업 스케줄러가 생성되는 것을 탐지 -->
    <OriginalFileName condition="is">schtasks.exe</OriginalFileName>
    <CommandLine condition="contains">/Create</CommandLine>
</Rule>
<Rule name="Task Scheduler Modified" groupRelation="and"> <!-- 작업 스케줄러가 수정되는 것을 탐지 -->
    <OriginalFileName condition="is">schtasks.exe</OriginalFileName>
    <CommandLine condition="contains">/Change</CommandLine>
</Rule>
<Rule name="Task Scheduler Deleted" groupRelation="and"> <!-- 작업 스케줄러가 삭제되는 것을 탐지 -->
    <OriginalFileName condition="is">schtasks.exe</OriginalFileName>
    <CommandLine condition="contains">/Delete</CommandLine>
</Rule>
```

[그림 13] BIT-sysmon-config 내 공격자 행위에 따른 이벤트 탐지 Rule 일부

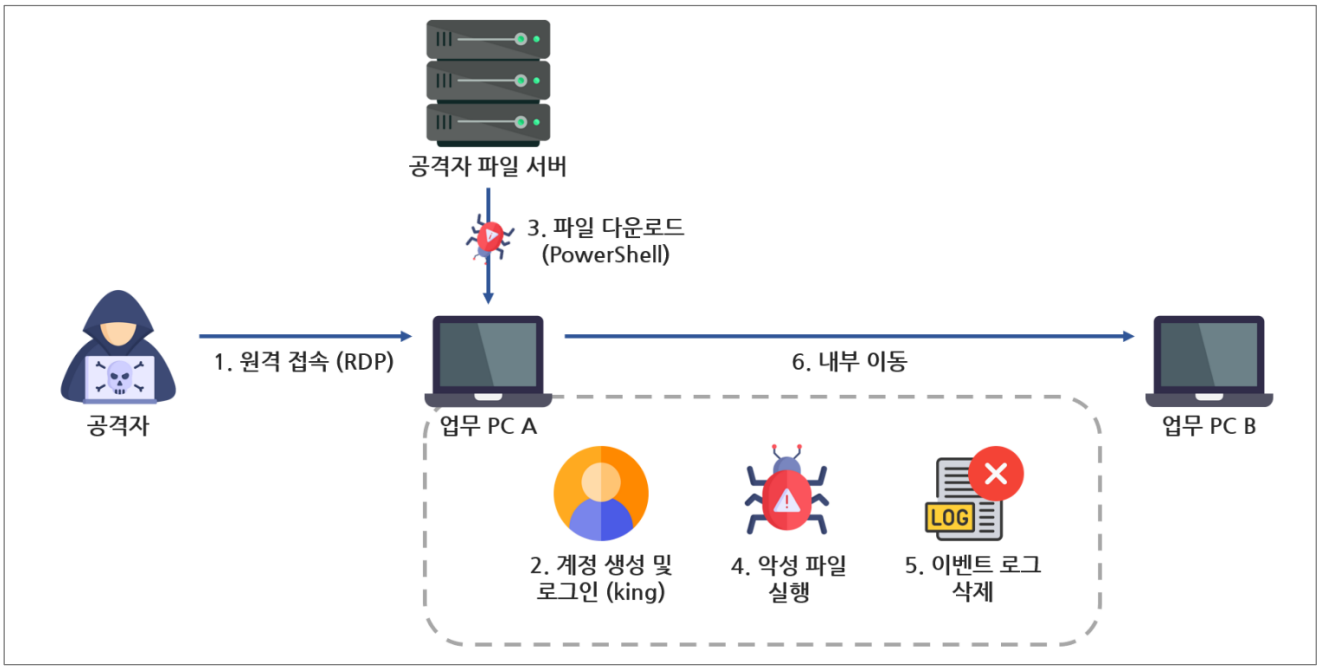
```
<Rule name="Reconnaissance" groupRelation="or"> <!-- 공격자가 시스템 정찰 시 주로 사용하는 프로세스 탐지 -->
    <OriginalFileName condition="is">whoami.exe</OriginalFileName>
    <OriginalFileName condition="is">ipconfig.exe</OriginalFileName>
    <OriginalFileName condition="is">tasklist.exe</OriginalFileName>
    <OriginalFileName condition="is">systeminfo.exe</OriginalFileName>
    <OriginalFileName condition="is">netstat.exe</OriginalFileName>
    <OriginalFileName condition="is">nslookup.exe</OriginalFileName>
    <OriginalFileName condition="is">tracert.exe</OriginalFileName>
    <OriginalFileName condition="is">route.exe</OriginalFileName>
    <OriginalFileName condition="is">klist.exe</OriginalFileName>
    <OriginalFileName condition="is">cmdkey.exe</OriginalFileName>
    <CommandLine condition="contains any">dir C:\users;ls C:\users;dir C:\Users;ls C:\Users</CommandLine>
    <OriginalFileName condition="is">quser.exe</OriginalFileName>
    <OriginalFileName condition="contains any">nltest.exe;nltestk.exe</OriginalFileName>
    <OriginalFileName condition="contains any">nbtstat.exe;nbtinfo.exe</OriginalFileName>
    <OriginalFileName condition="is">netsh.exe</OriginalFileName>
    <CommandLine condition="contains">netsh advfirewall</CommandLine>
    <OriginalFileName condition="contains any">ping.exe</OriginalFileName>
    <OriginalFileName condition="contains any">dsquery.exe</OriginalFileName>
    <CommandLine condition="contains any">net view;net group</CommandLine>
    <OriginalFileName condition="image">qprocess.exe</OriginalFileName>
    <OriginalFileName condition="image">query.exe</OriginalFileName>
    <OriginalFileName condition="image">qwinsta.exe</OriginalFileName>
    <OriginalFileName condition="image">rwinsta.exe</OriginalFileName>
    <OriginalFileName condition="contains any">tree.com;findstr.exe;where.exe</OriginalFileName>
    <OriginalFileName condition="is">pktmon.exe</OriginalFileName>
    <OriginalFileName condition="is">net64.exe</OriginalFileName>
    <OriginalFileName condition="is">ADfind.exe</OriginalFileName>
    <OriginalFileName condition="is">netscan.exe</OriginalFileName>
    <OriginalFileName condition="is">NetworkShare_pre2.exe</OriginalFileName>
</Rule>

<Rule name="Credential Access" groupRelation="or"> <!-- 공격자가 자격증명 탈취에 주로 사용하는 프로세스 탐지 -->
    <OriginalFileName condition="is">mimikatz.exe</OriginalFileName>
    <OriginalFileName condition="is">rdrleakdiag.exe</OriginalFileName>
    <OriginalFileName condition="is">dump64.exe</OriginalFileName>
    <OriginalFileName condition="is">DumpMinitool.exe</OriginalFileName>
    <OriginalFileName condition="is">sqldumper.exe</OriginalFileName>
    <OriginalFileName condition="is">procdump.exe</OriginalFileName>
    <OriginalFileName condition="is">rpcping.exe</OriginalFileName>
    <OriginalFileName condition="is">createdump.exe</OriginalFileName>
</Rule>
```

[그림 14] BIT-sysmon-config 내 공격자가 주로 사용하는 프로세스 탐지 Rule 일부

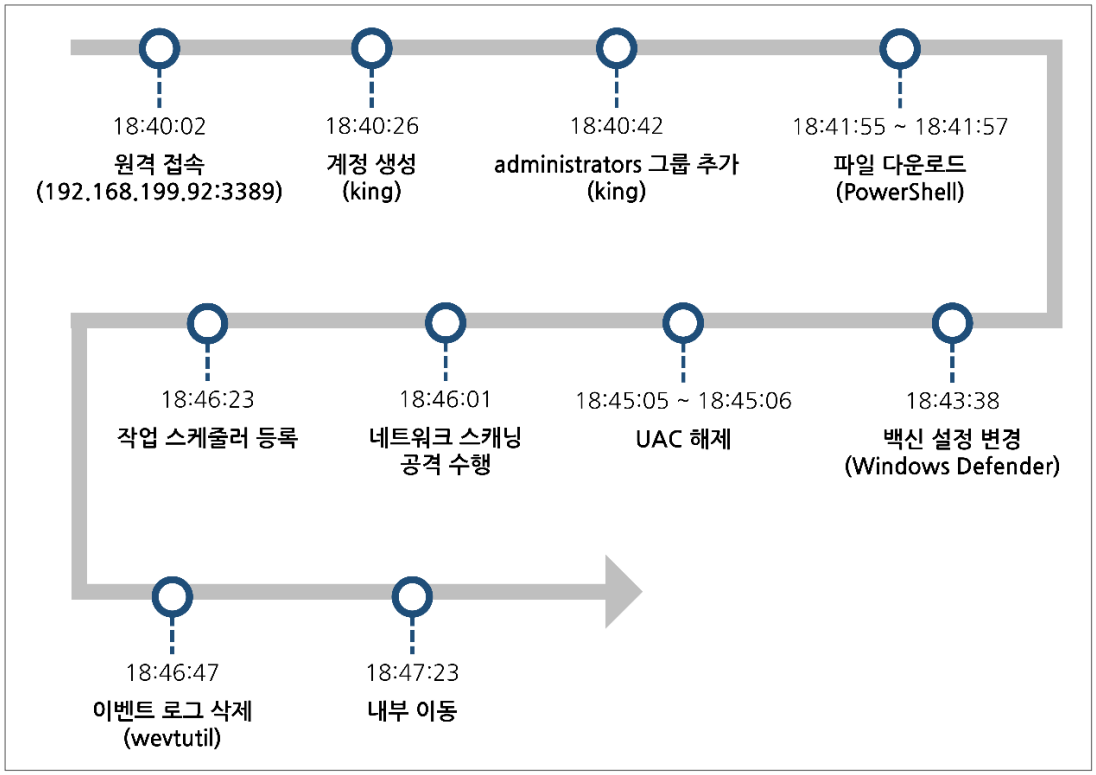
## 4.4. Configure 시나리오 테스트

작성한 BIT-sysmon-config 파일을 국내 환경에서 흔히 발생할 수 있는 가상의 시나리오를 제작한 후 테스트를 수행했으며, 그 결과는 다음과 같다.

### 4.4.1. 시나리오 개요도 및 타임라인




[그림 15] 시나리오 개요도




[그림 16] 시나리오 타임라인 (2023-10-20)

### 4.4.2. 시나리오 테스트 결과

테스트 시나리오 테스트를 통해 확인한 Sysmon 로그 구성 요소는 다음과 같다.

[표 26] 시나리오 테스트 분석 결과에서 확인한 Sysmon 로그 구성 요소

| 일시 | 이벤트 유형 | 상세 내용 | 출처 |
|---|---|---|---|
| 2023-10-20 18:40:02 | 원격 접속 | • **SourceIp**: 192.168.199.92<br>• **DestinationPort**: 3389 | Network Connect |
| 2023-10-20 18:40:26 | 계정 생성 | • **CommandLine**:<br>add user king kong1234!@ /add | Process Create |
| 2023-10-20 18:40:42 | 그룹 추가 (administrators) | • **CommandLine**:<br>net localgroup administrators king /add | Process Create |
| 2023-10-20 18:41:55 | 파일 다운로드 (PowerShell) | • **CommandLine**:<br>powershell.exe Invoke-WebRequest -Uri "http://{IP}/kkkk5555/1234.zip" -OutFile "C:\Users\king\1234.zip" | Process Create |
| 2023-10-20 18:41:57 | 파일 다운로드 (PowerShell) | • **Image**: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe<br>• **DestinationIp**: 3.34.4.133<br>• **DestinationPort**: 80 | Network Connect |
| 2023-10-20 18:43:38 | 백신 비활성화 | • **CommandLine**: "C:\Windows\System32\cmd.exe" /C "C:\Users\king\DisableWindowsDefender.bat" | Process Create |
| 2023-10-20 18:43:38 | 백신 비활성화 | • **ParentCommandLine**:<br>"C:\Windows\System32\cmd.exe" /C "C:\Users\king\DisableWindowsDefender.bat"<br>• **TargetObject**:<br>HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\DisableAntiSpyware Details: DWORD (0x00000001) | Registry Event |
| 2023-10-20 18:45:05 | UAC 해제 | • **CommandLine**:<br>wscript DisableUAC.vbs | Process Create |
| 2023-10-20 18:45:06 | UAC 해제 | • **ParentCommandLine**: wscript DisableUAC.vbs<br>• **CommandLine**:<br>"C:\Windows\System32\cmd.exe" /c reg ADD HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v EnableLUA /t REG_DWORD /d 0 /f | Process Create |
| 2023-10-20 18:46:01 | 네트워크 스캐닝 | • **Image**:<br>C:\Users\king\goon3_win_amd64.exe<br>※ 해당 이벤트 다수 발생 | Network Connect |
| 2023-10-20 18:46:23 | 작업 스케줄러 등록 | • **CommandLine**:<br>"C:\Windows\System32\cmd.exe" /C "C:\Users\king\Scheduler.bat" | Process Create |

| 일시 | 이벤트 유형 | 상세 내용 | 출처 |
|---|---|---|---|
| 2023-10-20 18:46:23 | 작업 스케줄러 등록 | • **ParentCommandLine**: "C:\Windows\System32\cmd.exe" /C "C:\Users\king\Scheduler.bat" • **CommandLine**: schtasks /Create /SC ONCE /TN "MS Office" /TR "C:\Users\king\king.exe" /ST 23:59 /F | Process Create |
| 2023-10-20 18:46:23 | 작업 스케줄러 등록 | • **TargetFilename**: C:\Windows\System32\Tasks\MS Office | File Create |
| 2023-10-20 18:46:47 | 이벤트 로그 삭제 | • **CommandLine**: wevtutil cl Security | Process Create |
| 2023-10-20 18:47:23 | 내부 이동 | • **TargetObject**: HKU\S-1-5-21-3728364944-3941103186-3578969269-1001\Software\Microsoft\Terminal Server Client\Servers\192.168.199.123\UsernameHint • Details: plainbit | Registry Event |

## 4.5. 활용 방안

### 4.5.1. 로그 설정 방안

Sysmon 설치 시 기본적으로 64MB 의 최대 용량을 가지며, 최대 용량이 초과되면 이벤트 로그가 덮어씌워진다. Configure Rule 설정에 따라 프로세스 및 네트워크 관련 이벤트가 다수 발생하기 때문에 Sysmon 기본 로그 용량인 64MB 로는 4~5 일에 해당되는 로그만 로깅 된다. 연구된 BIT-sysmon-config 는 일상적인 업무 환경에서 1 일 평균 2MB 의 이벤트가 발생하지만 공격자가 시스템에 침투해 공격 행위를 수행하면 이벤트 로그가 다수 발생하게 된다. 이로 인해, 침해사고 발생 시 위협을 탐지 및 대응하기 위해서는 최소 1GB 의 로그로 설정하고 주기적으로 백업해 관리해야 한다.

[표 27] 1일 평균 Sysmon 로그 크기 비교

| Microsoft | SwiftOnSecurity | Neo23x0 (trace) | Neo23x0 (sysmon) | olafhartong | PLAINBIT |
|---|---|---|---|---|---|
| 2MB | 6MB | 600MB | 8MB | 15MB | 2MB |

### 4.5.2. Sysmon 을 활용한 공격 위협 차단

Sysmon 에서는 Configure Rule 을 통해 특정 행위를 차단시키는 File Block Executable, File Block Shredding 이벤트가 존재한다. File Block Executable 이벤트는 Configure Rule 에 실행 파일 생성을 차단할 경로를 지정하면 해당 경로에 실행 파일 생성이 차단된다.

```
<RuleGroup name="" groupRelation="or">
  <FileBlockExecutable onmatch="include">
    <TargetFilename condition="contains all">C:\Users;Downloads</TargetFilename>
  </FileBlockExecutable>
</RuleGroup>
```

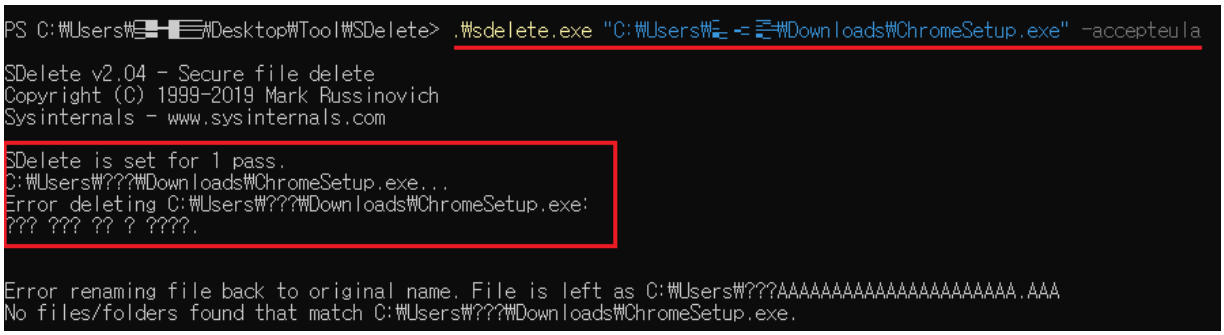
[그림 17] File Block Executable 이벤트 Rule 예시 (C:\Users*\Downloads에 다운로드 되는 실행 파일 차단)


[그림 18] 실행 파일 다운로드 차단 화면 예시

File Block Shredding 이벤트는 Configure Rule 에 SDelete 와 같은 파일 완전 삭제 도구로 삭제되는 것을 차단할 경로를 지정하면 해당 경로에 완전 삭제가 시도될 때 차단된다. 하지만, 일반적으로 삭제하는 행위는 차단되지 않는다.

```
<RuleGroup name="" groupRelation="or">
  <FileBlockShredding onmatch="include">
    <TargetFilename condition="contains all">C:\Users;Downloads</TargetFilename>
  </FileBlockShredding>
</RuleGroup>
```

[그림 19] File Block Shredding 이벤트 Rule 예시 (C:\Users*\Download 경로에서 완전 삭제 차단)




[그림 20] "C:\Users*\Download"에 위치한 파일 완전 삭제 차단 화면

그러나, File Block Executable, File Block Shredding 이벤트는 잘못 정의하게 되면 시스템 운용에 영향을 줄 수 있어 기본적으로 Sysmon Configure 에서 활성화하지 않는다. 운용하는 시스템에서 실행 파일이 생성되지 않는 경로나 공격자가 주로 활용하는 경로를 해당 이벤트를 통해 차단해 공격자에 의한 공격 도구 유입을 사전에 차단할 수 있다.

### 4.5.3. ArchiveDirectory Data 활용

오늘날 공격자가 시스템에 침투한 뒤 공격에 활용한 도구를 삭제하는 행위는 당연시되고 있다. Sysmon 의 파일 삭제 관련 이벤트는 삭제된 파일의 해시 값을 기록하기 때문에 OSINT Tool 로 분석해 알려진 공격 도구의 행위를 식별할 수 있다. 하지만, OSINT Tool 에 존재하지 않는 신규 악성 파일의 행위를 파악하기 위해서는 파일을 확보하는 것이 중요하다.

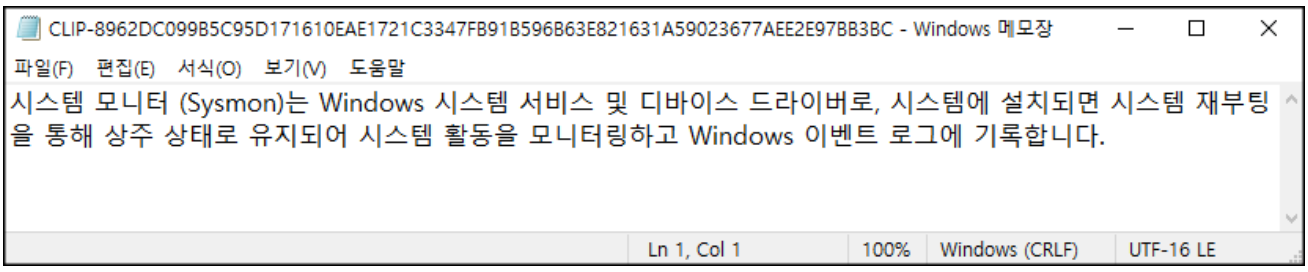
Sysmon 에는 File Delete archived, Clipboard changed 이벤트로 삭제된 파일과 클립보드 내용을 저장하는 ArchiveDirectory 가 존재한다.

File Delete archived 이벤트는 삭제된 파일을 ArchiveDirectory 에 저장해 공격자가 활용한 공격 도구를 확보할 수 있다. 또한, 공격 공격자가 시스템에서 복사한 텍스트를 Clipboard changed 이벤트로 확인할 수 있어 추가적인 공격자의 행위를 식별하는 데 도움이 된다. 삭제된 파일은 사용자가 지정한 ArchiveDirectory 에 "[설정된 해시 값].[확장자]" 형식으로 저장한다. 만약, 삭제된 파일이 0x00 바이트만 포함되어 있다면, ArchiveDirectory 에 저장되지 않는다.

만약, 침해사고 조사 대상 시스템에서 Sysmon 이 동작했다면 ArchiveDirectory 에 저장된 데이터를 확인해 공격자 행위를 식별하는데 활용할 수 있다.


[그림 21] ArchiveDirectory 내 저장된 삭제된 파일


[그림 22] ArchiveDirectory 내 저장된 클립보드 파일




[그림 23] ArchiveDirectory에 저장된 클립보드 파일 내용

PLAINBIT Co., Ltd.

**연구개발센터** 경기도 성남시 분당구 판교역로 18번길 14, 3층
**침해사고대응센터** 서울시 송파구 법원로 11길 12, 2층
**디지털포렌식센터** 서울시 서초구 서초대로 56길 22, 5층
T. 031-8016-0912 F. 031-8016-0913
plainbit.co.kr