Snort 3 이해: 상태 기반 서명 평가 Byte_Jump

목차

[소개](#)

[배경 정보](#)

[새로운 기능](#)

[지원되는 플랫폼](#)

- [최소 소프트웨어 및 하드웨어 플랫폼](#)

[기능 세부사항](#)

- [기능 기능 설명](#)
  - [어떻게 진행됩니까?](#)
  - [일반 규칙 평가](#)

[데이터 스트림 및 IPS 버퍼](#)

- [규칙 계속](#)
- [사용자 구성](#)

[문제 해결](#)

[샘플 문제](#)

- [문제: 설명](#)
- [문제: 솔루션](#)

[제한 사항 세부 정보 및 일반적인 문제](#)

- [제한 사항 및 기타 고려 사항](#)

소개

이 문서에서는 Snort 3에 추가된 7.4 이후의 새로운 기술에 대해 설명합니다.

배경 정보

- Snort 3 탐지 모듈은 블록 모드에서 작동합니다. 이러한 접근 방식은 성능 우위와 구현 간소화를 제공하지만(상대적으로) 여러 데이터 블록에 걸쳐 있는 시그니처를 탐지하는 데에는 몇 가지 한계가 있습니다.
- 사용자 경험을 용이하게 하기 위해 Snort에서는 다음과 같은 몇 가지 개선 사항이 이미 구현되었습니다.
  1. Flowbits를 사용하면 규칙 작성기가 네트워크 흐름을 사용자 정의 속성으로 표시할 수 있습니다. 이 속성은 흐름에서 모든 패킷에 대해 설정, 삭제 및 테스트될 수 있습니다(패킷을 통한 더 큰 시그니처에 대해 결론을 내리는 방법 제시).
- 스트림 모듈은 유선 패킷을 다시 작성된 패킷에 누적합니다. 이는 원시 패킷보다 더 크고 의미 있는 블록입니다. 다시 작성된 패킷에 대해 IPS 규칙을 평가하면 전체 그림을 보고 더 큰 패턴(시그니처)과 일치할 수 있는 기회가 늘어납니다.
- 어떤 경우에는 재작성된 패킷이 새로운 데이터뿐만 아니라 탐지에 의해 이미 처리된 이전 데이터의 일부를 포함합니다. 다시 말해, 누적된 데이터 블록으로 인해 흐름에서 역방향으로(어느 정도) 스팬되는 시그니처가 탐지될 수 있습니다.

- 스트림 분할기는 흐름을 블록으로 분할하지만, 절단된 지점은 공격자가 패턴 탐지를 피하는 데 사용할 수 있는 잠재적인 약점입니다. 따라서 Snort는 분할을 더 예측할 수 없도록 지터링 메커니즘을 구현했습니다. 이로 인해 공격자에 대한 분석이 더욱 복잡해집니다.

새로운 기능

상태 저장 시그니처 평가는 목록에 추가할 수 있는 새로운 기술입니다. 여러 블록에 대한 IPS 규칙 평가를 활성화하여 탐지 기능을 확장합니다. 따라서, 현재 블록에 데이터가 부족하면 규칙은 즉시 불일치하지 않지만, 대신 더 많은 데이터가 도착하기를 기다린다.

지원되는 플랫폼

최소 소프트웨어 및 하드웨어 플랫폼

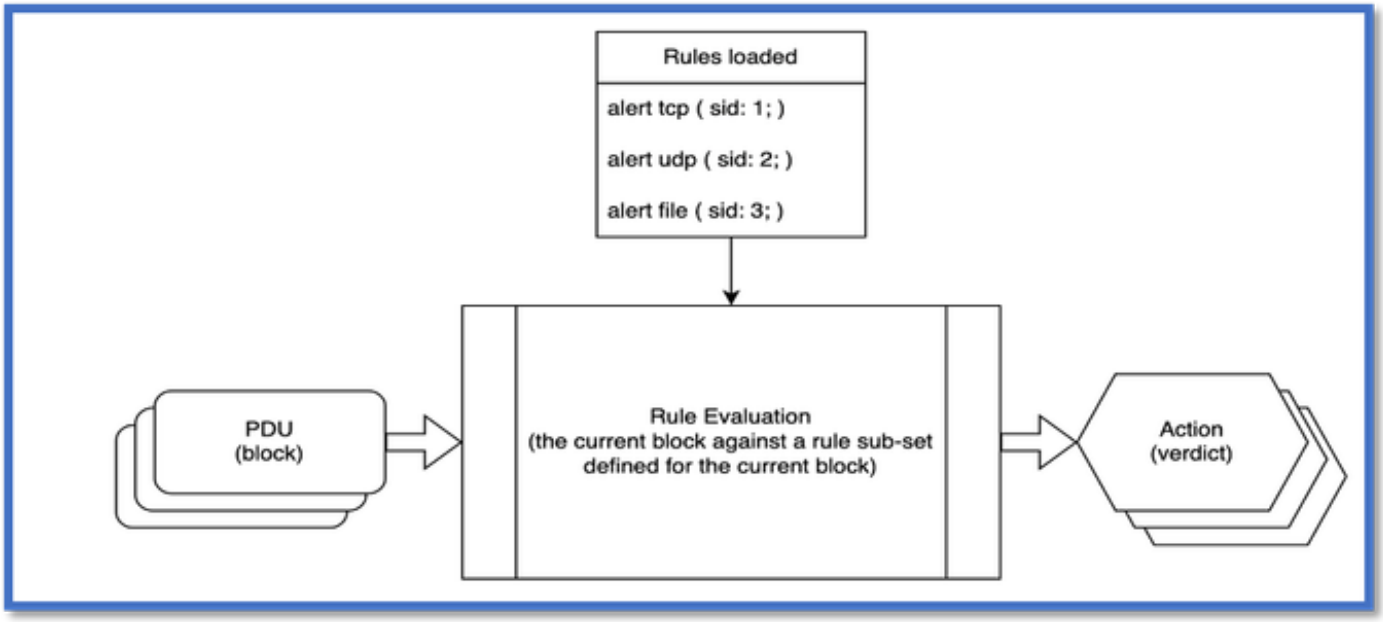
지원되는 최소 관리자 버전	관리되는 디바이스	최소 지원 관리되는 디바이스 버전 필요	참고
Management Center 7.4.0	FTD	7.4.0	Snort 3만
Device Manager 7.4.0	FDM 관리를 지원하는 모든 FTD	7.4.0	Snort 3만

기능 세부사항

기능 기능 설명

어떻게 진행됩니까?

다이어그램에는 탐지 모듈 워크플로가 나와 있습니다. 트래픽 처리 단계에서 모듈에는 이미 모든 규칙이 로드되어 있으며, 모듈에서는 데이터 블록을 하나씩 받아들이고, 규칙을 평가하고, 프로세스 상태 저장 시그니처 평가 블록에 대해 수행할 작업을 정의합니다.



구성표에 대한 참고 사항:

1. 규칙 하위 집합이 현재 데이터 블록에 대해 정의되면, 해당 하위 집합에서 오는 각 규칙은 다른 규칙과 독립적으로 평가됩니다.
2. 각 데이터 블록은 다른 블록과는 독립적으로 평가된다.
3. 데이터 블록은 현재 패킷에 대해 평가되는 IPS 버퍼 집합에 대한 추상화입니다.
4. Action은 현재 패킷에 대해 평가된 작업 목록입니다. 최종 판정은 나중에 결정됩니다.

스테이트풀 시그니처 평가의 작동 방식을 이해하려면 일반적인 IPS 규칙을 평가하는 방법 및 데이터 블록이 스트림을 구성하는 방법을 살펴보십시오.

일반 규칙 평가

IPS 규칙은 다음 형식으로 나타낼 수 있습니다.

```
action protocol source → destination ( option_1: parameters; option_2: parameters;
option_3: parameters; gid: 1; sid: 1; meta_option_1; meta_option_2; meta_option_3; )
```

여기서 각 항목은 다음을 나타냅니다.

action - 규칙이 실행되는 경우 패킷에 대한 IPS 작업

protocol - 일치시킬 프로토콜

소스, 대상 - IP 주소 및 포트

option_1, option_2, option_3 - 규칙 평가의 일부인 IPS 옵션

gid, sid - 규칙을 식별하는 고유한 쌍(메타데이터 옵션과 같음)

meta_option_1, meta_option_2, meta_option3 - 메시지, 클래스 유형 또는 참조와 같은 규칙 메타데이터이며 이러한 옵션은 규칙 평가에 참여하지 않습니다.

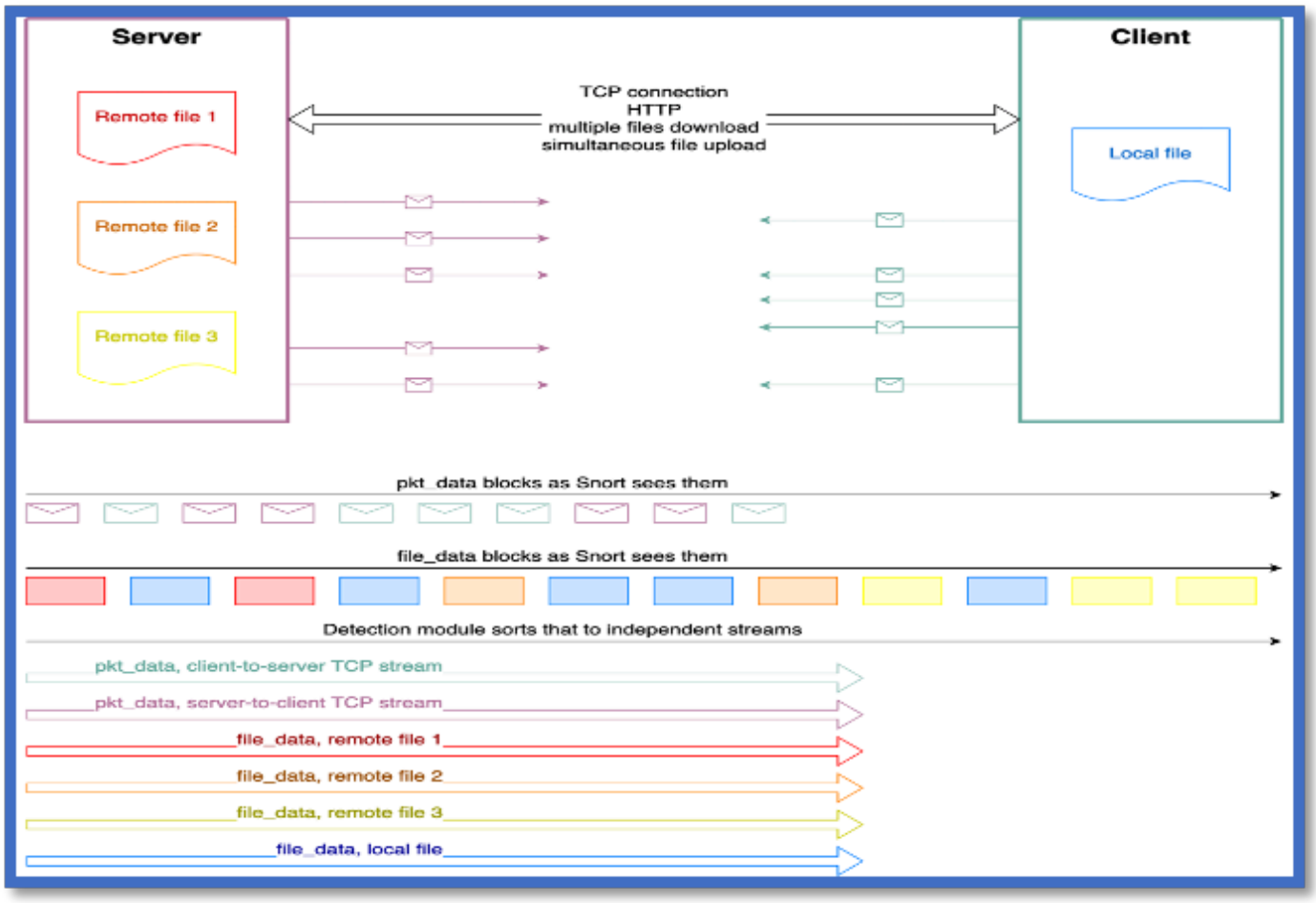
- 프로토콜, 소스 및 대상은 규칙 헤더를 형성합니다. 네트워크 흐름(평가를 위해 수락할 흐름)에 대한 필터 역할을 합니다. 괄호 안의 모든 항목은 규칙 본문입니다. 규칙 본문의 IPS 옵션(규칙 메타데이터 제외)은 데이터 블록에 대해 평가되는 옵션입니다. 이들은 다음과 같은 문구를 준수합니다.
- 옵션은 왼쪽에서 오른쪽으로 엄격하게 평가됩니다.
    1. 두 가지 주요 유형 중 하나일 수 있습니다.
    2. buffer setter, 옵션은 현재 패킷에 대한 IPS 버퍼를 선택합니다.
- 기타(패턴 검색, 수학 연산, 커서 조작, flowbit 연산)
- 커서는 선택한 IPS 버퍼의 위치를 추적하는 데 사용됩니다.

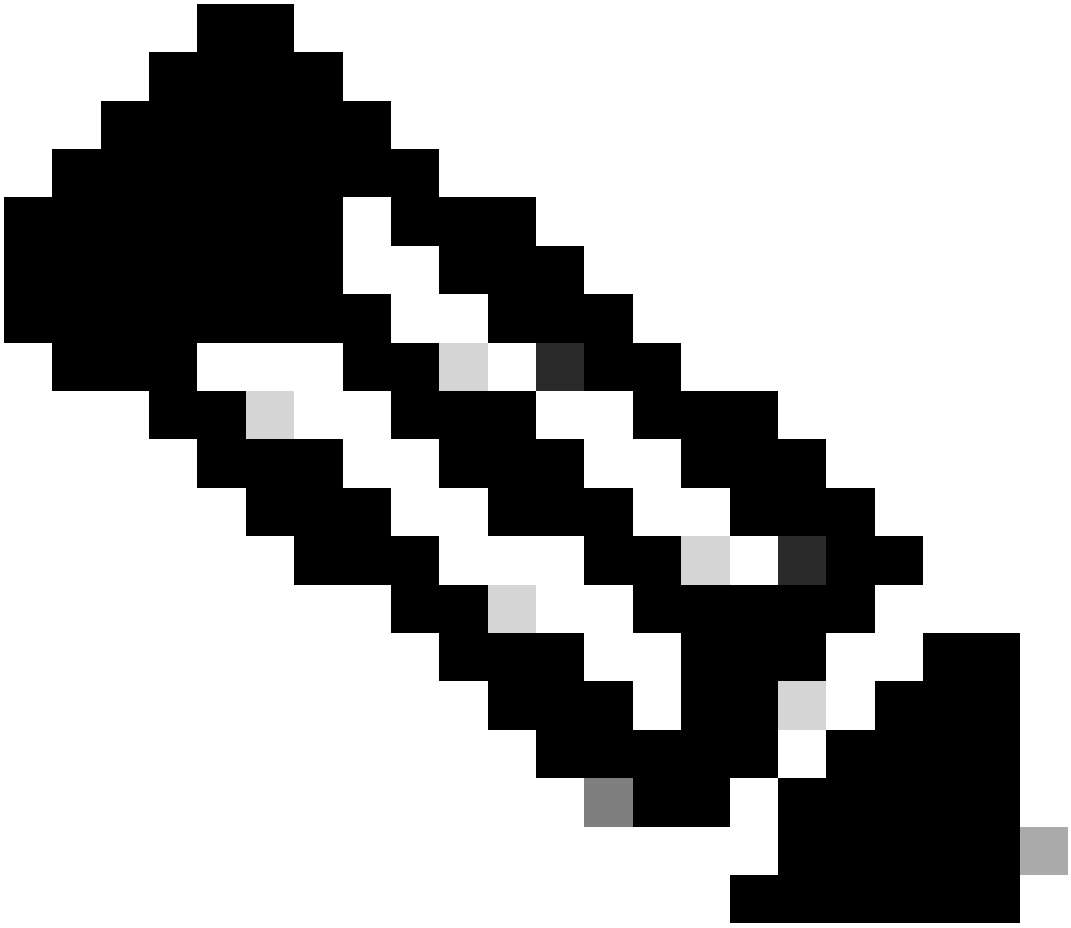
- 옵션은 다음 중 하나일 수 있습니다.
    1. 'absolute', 커서 위치에 따라 달라지지 않습니다.
    2. 'relative', 즉 커서 위치에서 평가를 시작합니다.
- 옵션이 선택된 IPS 버퍼에서 커서를 설정하려고 시도하면 오류가 발생하고 데이터 부족으로 인해 전체 규칙이 일치하지 않습니다
- 마지막으로, 탐지 모듈의 제한이 있습니다. Snort는 리소스가 무제한일 수 있는 경우, 데이터가 사용 가능해지면(더 많은 유선 패킷이 도착함) 표시된 모든 데이터를 캐시하여 규칙을 다시 평가합니다.

데이터 스트림 및 IPS 버퍼

- 데이터 스트림은 동일한 소스에서 연속된 형태의 바이트 스트림입니다. 상태 기반 평가를 지원하기 위해 제시된 새로운 개념입니다. 블록 간의 규칙 평가는 동일한 논리적 데이터(파일, 순수 TCP 스트림 또는 JavaScript 텍스트)에서 수행해야 합니다.

- 일반적으로, 탐지 모듈에 의해 수신된 데이터 블록은 다음을 수행할 수 있다.
    - 다른 IPS 버퍼에서 가져와야 합니다(예: pkt_data와 file_data는 동일하지 않음).
    - 다른 스트림에 속함
    - 스트림을 형성하지 않음(원시 패킷에서 생성된 버퍼)
    - 연속 스트림(ICMP, UDP)을 형성하지 않음
    - 순서가 아님(HTTP Partial Response)
    - 반복 데이터 포함(http_inspect.script_detection 또는 HTTP 청크 응답과 같은 누적 블록)
- 탐지 모듈은 동일한 스트림에서 블록을 연결하기 위해 여러 요소를 정렬할 수 있습니다. 그렇지 않으면 블록 인터리빙에서 원치 않는 간섭이 발생할 수 있습니다.


Server
Remote file 1
Remote file 2
Remote file 3
TCP connection
HTTP
multiple files download
simultaneous file upload
Client
Local file
pkt_data blocks as Snort sees them
file_data blocks as Snort sees them
Detection module sorts that to independent streams
pkt_data, client-to-server TCP stream
pkt_data, server-to-client TCP stream
file_data, remote file 1
file_data, remote file 2
file_data, remote file 3
file_data, local file



참고: 이 예에서는 HTTP 클라이언트가 여러 파일을 동시에 업로드하고 다운로드하는 경우를 보여 줍니다.

- 현재 2개의 IPS 버퍼만 스트림을 나타낼 수 있습니다. pkt_data 및 file_data입니다. 여기서,
  1. pkt_data는 TCP 프로토콜(클라이언트-서버 및 서버-클라이언트 방향)을 위한 두 개의 스트림을 형성합니다.
  2. file_data는 파일, MIME 첨부 파일 및 기타 프로토콜 데이터(예: HTTP HTML 페이지 및 /또는 기타 Content-Type)에 대한 스트림을 형성해야 합니다.
- 상태 기반 평가는 데이터 스트림 내에서 엄격하게 수행됩니다.

규칙 계속

- 이전 절은 커서가 현재 IPS 버퍼에서 벗어나게 설정할 경우 IPS 옵션이 일치하지 않는다는 설명으로 끝납니다. 그러나 IPS 버퍼가 데이터 스트림을 형성하면 상태 저장 시그니처 평가 기능이 Snort 흐름 객체에 규칙 평가 컨텍스트를 단계적으로 저장합니다. 저장된 평가 컨텍스트(상태)를 규칙 연속이라고 합니다. 상태 저장 시그니처 평가는 더 많은 데이터를 사용할 수 있을 때까지 규칙의 최종 판정을 연기합니다.

- 규칙 연속에는 IPS 버퍼 이름, 버퍼 소스 및 대상 커서 위치라는 세 가지 주요 부분이 있습니다(버퍼 소스는 데이터 스트림의 고유 식별자).
- 데이터 블록이 탐지 모듈에 의해 처리되면 다음 작업이 수행됩니다.
    - 상태 저장 시그니처 평가에서는 다음과 같은 경우 규칙 지속을 생성하여 흐름에 연결합니다.
        - IPS 옵션(byte_jump, content, pcre 또는 커서 위치를 업데이트하는 기타 옵션)은 현재 IPS 버퍼 뒤에 커서를 설정합니다
        - 현재 IPS 버퍼는 데이터 스트림을 지원합니다.
        - 현재 IPS 버퍼는 데이터 스트림을 형성합니다.
- 다음과 같은 경우 상태 저장 시그니처 평가에서는 방금 생성한 규칙 지속을 취소하고 흐름에서 제거합니다.
    - IPS 규칙이 현재 데이터 블록에 대해 실행되었습니다(규칙은 블록의 다른 위치에 일치함).
- 상태 저장 시그니처 평가에서는 보류 중인 규칙 연속성을 거부하고, 다음과 같은 경우 흐름에서 제거합니다.
    - IPS 버퍼는 연속 스트림을 형성하지 않습니다(예: 블록에 반복된 데이터가 있거나 간격이 있습니다(데이터 일부가 누락되었거나 블록이 순서가 잘못됨).
- 상태 저장 시그니처 평가에서는 다음과 같은 경우에 사용할 수 있는 새 데이터로 대상 커서 위치를 업데이트합니다.
    - 규칙 연속의 버퍼 소스가 선택한 버퍼 소스와 같습니다
    - IPS 버퍼가 연속 스트림을 형성함
- 상태 저장 시그니처 평가는 다음과 같은 경우 규칙 지속을 IPS 규칙 엔진으로 다시 전송합니다.
    1. 선택한 IPS 버퍼 내부의 대상 커서 위치 지점(규칙 평가를 완료하는 데 필요한 모든 데이터를 최종적으로 수신함).

사용자 구성

- 규칙 연속은 메모리를 사용하므로 Snort는 무제한으로 메모리를 저장할 수 없습니다. 제한을 제어하는 구성 옵션이 있습니다.
    1. Detection.max_continuations_per_flow = 1024: 흐름에 동시에 저장되는 최대 연속 수 { 0:65535 }
- 상태 저장 시그니처 평가가 제한에 도달하면 가장 오래된 규칙 지속을 새 규칙 지속으로 교체합니다.
- 플로우에 존재하는 가장 오래된 규칙 연속은 너무 오래 존재하므로 규칙 평가를 재개할 조건을 충족하지 못합니다.
- 또한 IPS 규칙을 미세 조정하는 데 사용할 수 있는 페그 수는 많으며(주요 중점 사항이어야 함), 제한 사항(필요한 경우)은 다음과 같습니다.
    1. detection.cont_creations: 생성된 총 연속 수(합계)
    2. detection.cont_recalls: 회수된 총 연속 수(합계)
    3. detection.cont_flows: 연속(sum)을 사용한 총 플로우 수
    4. detection.cont_evals: 조건이 충족된 총 연속 수(sum)
    5. detection.cont_matches: 일치하는 총 연속 수(sum)
    6. detection.cont_mismatches: 일치하지 않는 총 연속 수(sum)
    7. detection.cont_max_num: 플로우당 최대 동시 연속 수(최대)
    8. detection.cont_match_distance: 일치하는 연속으로 건너뛴 총 바이트 수(sum)

9. detection.cont_mismatch_distance: 일치하지 않는 연속으로 건너뛴 총 바이트 수(sum)

문제 해결

이 기능은 기존 탐지 프로세스를 개선한 것이므로 명시적으로 문제를 해결할 수 없습니다. 탐지에 장애가 발생하는 경우 규칙, 컨피그레이션 또는 트래픽을 검사해야 합니다.

샘플 문제

문제: 설명

- 시그니처가 파일의 시작과 끝을 동시에 확인해야 한다고 가정해 보겠습니다.
- 예를 들어, 이 구조의 대상 파일(헤더, 본문, 메타데이터)에서 해당 메타데이터의 값이 0인지 확인해야 합니다.

- 파일 바이트: e1 f3 22 03 7f ff xx ... xx 01 00 02 00 여기서
    - e1 f3 22 03 - 파일 유형을 식별하는 매직 번호의 경우 4바이트
    - 7f ff - 본문 크기에 2바이트
    - xx xx ... xx - 일부 데이터의 32kb
    - 01 00 02 00 - 태그 값 형식의 메타 데이터 4바이트(각각 1바이트)
- IPS 규칙은 다음과 같습니다. 경고 파일(file_data; content:"|e1f32203|",fast_pattern; byte_jump:2,0,relative; content:"00",within:4, relative; sid: 1;)
    - 위치
        - 파일 프로토콜은 규칙이 재작성된 패킷만 수락하는지 확인합니다(원시 패킷은 스테이트풀 서명 평가에 참여하지 않음).
        - 'file_data 옵션은 스트림을 형성할 수 있는 파일 데이터 버퍼를 선택한다
        - 첫 번째 content 옵션은 빠른 패턴이며 magic number(원하는 파일 형식인 경우)를 확인합니다.
        - byte_jump 옵션은 파일 본문 크기를 읽고 파일 본문을 이동합니다.
        - 두 번째 content 옵션은 매개변수 제한 검색 깊이 내에서 메타데이터 값에 대한 최종 검사를 수행하고 해당 옵션을 relative로 설정합니다.

문제: 솔루션

규칙은 다음과 같이 평가됩니다.

파일 헤더와 본문 일부를 전달하는 첫 번째 패킷(8kB 크기)에서 다음을 수행합니다.

1. IPS buffer file_data가 선택됩니다. 커서는 0번째 바이트 e1을 가리킵니다.
2. 빠른 패턴 옵션은 매직 번호 바로 뒤의 커서 위치를 바이트 7f를 가리키는 것과 일치하고 설정합니다.
3. byte_jump 옵션은 파일 본문 크기의 2바이트를 읽습니다. 커서는 이 2바이트로 업데이트됩니다. 그런 다음 byte_jump는 32768바이트 이상의 점프를 계산합니다.
4. 상태 저장 시그니처 평가에서는 규칙 연속을 생성하며, 여기서 24578바이트 더 많은 값이 필

요합니다(32768 - (8kB - 헤더 4바이트 - 본문 크기 2바이트).
5. byte_jump 옵션에서 커서 위치를 해당 범위까지 설정하지 못하므로 전체 규칙이 일치하지 않습니다.

파일 본문 부분을 전달하는 두 번째 패킷(16kB 크기)에서 다음을 수행합니다.

1. 상태 저장 시그니처 평가에 보류 중인 규칙 지속이 표시됩니다.
2. 버퍼 이름으로 버퍼를 선택하고 file_data가 사용 가능하고 새 데이터 크기가 16384임을 확인합니다.
3. 업데이트된 커서에 8194바이트가 여전히 필요한 것으로 표시됩니다(24578 - 16384)
4. 규칙이 다시 시작되지 않습니다.

파일 본문 부분 및 메타데이터를 전달하는 세 번째 패킷(8198개 크기):

1. 상태 저장 시그니처 평가에 보류 중인 규칙 지속이 표시됩니다.
2. 이름으로 버퍼를 선택하고 file_data가 사용 가능하고 새 데이터 크기가 8198인 것을 확인합니다.
3. 업데이트된 커서는 버퍼에 충분한 데이터가 있음을 나타냅니다. 커서 위치는 8194입니다.
4. 상태 저장 시그니처 평가에서 규칙 계속을 삭제합니다.
5. 상태 저장 시그니처 평가에서는 커서가 바이트 01을 가리키는 두 번째 content 옵션에서 규칙 평가를 다시 시작합니다.
6. content 옵션은 검색된 두 번째 바이트에서 일치 항목을 찾습니다.
7. 모든 규칙이 마침내 실행되었다.

제한 사항 세부 정보 및 일반적인 문제

제한 사항 및 기타 고려 사항

- 스테이트풀 시그니처 평가 구현으로 인해 Snort는 컨피그레이션을 다시 로드할 때 보류 중인 모든 규칙 연속을 삭제합니다. 삭제되더라도 다음 데이터 블록이 탐지 모듈로 전송될 때까지 규칙 연속은 여전히 Snort 메모리를 차지합니다.
- 상태 저장 평가에서 IPS 규칙에 대한 규칙 레이턴시 기능은 일반 규칙 평가와 동일하게 작동합니다. 서로 다른 데이터 블록의 규칙 파트에 대한 평가 시간이 요약됩니다. 시간이 제한을 초과하면 규칙 평가에서 단락되어 더 일찍 종료됩니다.
- Flowbits 연산은 '정적' 옵션과 마찬가지로 계속 수행되지만 의미를 유지합니다.
  현재 알려진 컨텍스트 내에서 flowbit set/clear/test 작업이 수행됩니다. 따라서 규칙 연속에서 flowbit 옵션이 평가되는 경우 규칙이 평가를 시작할 때가 아니라 현재 환경(flowbits 집합)을 고려합니다.

또한 룰 라이터는 빠른 패턴의 위치에 주목해야 한다.

규칙의 어떤 부분에도 포함될 수 있더라도 fast-pattern 옵션은 전체 규칙 이전에 평가됩니다. 규칙 평가를 트리거합니다. 상태 저장 시그니처 평가 기반 규칙의 경우 규칙 연속 지점이 fast-pattern 옵션 뒤에 있어야 합니다.
또한 IPS 규칙의 평가에는 여러 규칙 연속이 포함될 수 있습니다(동시에 연속되는 것이 아니라 하나씩). 규칙 본문의 모든 옵션은 연속성을 가질 수 있으므로, 규칙 작성기는 동일한 IPS 규칙을 사용하여 데이터 스트림의 다른 위치에서 추가 검사를 수행할 수 있습니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.