

제3절 네트워크 사고 분석

1. 사고 유형별 수집 데이터

침입 사고나 네트워크 공격이 발생했을 경우, 네트워크 관리자들이 발생 현황을 파악하고 증거 분석을 위해서 여러 가지 정보를 수집해야 한다. <표 3-7>은 네트워크에서 발생하는 다양한 사고의 종류와 해당 사고가 발생한 경우에 수집해야하는 정보들이다.

<표 3-7> 네트워크 사고 종류 및 수집해야하는 정보

사고의 종류	사 고	수집 정보
불법적인 자원 사용	프로세스 및 저장 장치 불법 사용	호스트: 액세스 로그, 프로세스 상태, CPU 사용률과 파일 및 저장 공간 상태
	네트워크 대역폭 불법 사용	네트워크: 회선 상태, 송수신된 패킷 개수, IP 주소, 프로토콜 사용현황 및 스위치 포트 상태
	메일 및 프록시 서비스의 불법 릴레이 (relay)	호스트: 어플리케이션 로그와 프로세스 상태 네트워크: IP 주소, 프로토콜 사용현황 및 데이터 내용
DoS (denial of service)	서버 자원들을 과도하게 소모하여 서비스 불안정 및 중단	호스트: 프로세스 상태, CPU 사용률 및 비정상적인 패킷 로그 네트워크: 회선 상태, 비정상적인 패킷 개수, IP 주소 및 비정상적인 패킷의 내용
	네트워크 대역폭을 과도하게 점유하여 통신 불안정 및 중단	네트워크: 송수신된 패킷 개수, IP 주소, 프로토콜 사용현황 및 데이터 내용
데이터 손상 및 변조	웹 페이지, 데이터 파일, 프로그램 파일 변조	호스트: 액세스 로그, 파일과 저장 장치 상태, 환경 파일의 내용 등 네트워크: IP 주소, 프로토콜 사용현황, 데이터 내용, 스위치 포트의 상태
정보 누설	비공개 자료의 누설과 통신 가로채기	호스트: 액세스 로그와 파일 및 저장 장치 상태 네트워크: IP 주소, 프로토콜 사용현황, 데이터 내용 및 스위치 포트 상태

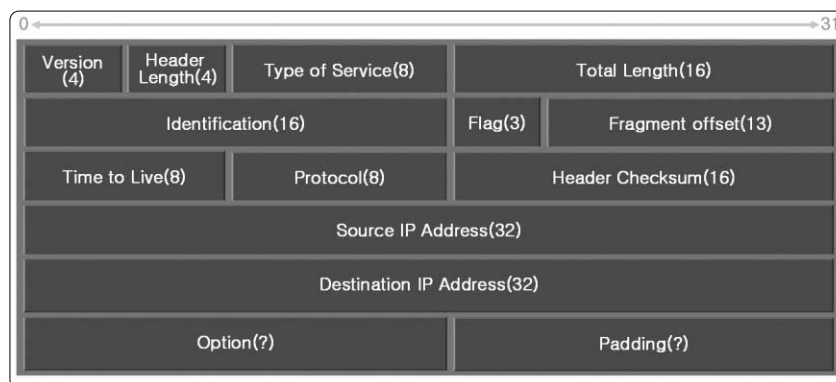


위 <표 3-7>에서와 같이 대부분의 사고나 공격이 발생하는 경우, 그 원인과 발생지를 찾기 위해서 네트워크 트래픽 정보와 패킷은 반드시 수집하여 분석해야 한다. 네트워크 트래픽 정보는 MRTG와 같은 공개 소프트웨어나 네트워크 관리에 사용되는 많은 NMS (Network Management System), 보안을 위하여 사용하는 방화벽, IDS 등에서 수집할 수 있으며, 라우터나 스위치의 간단한 명령어만으로 확인할 수도 있다. 또한 명확한 원인 분석을 위하여 필요한 실제 네트워크 패킷들은 공개 프로토콜 분석기인 Ethereal이나 편리한 사용자 인터페이스를 제공하는 다양한 전문 분석기를 사용하여 간단하게 수집 및 분석할 수 있다.

네트워크에서 수집할 수 있는 주요 정보는 대역폭 사용량, 트래픽을 대량으로 발생시키고 있는 IP 주소, 침입을 위하여 내부의 사용자나 서버의 IP 주소나 TCP/UDP 포트를 스캐닝하고 있는 IP 주소, 프로토콜 (TCP/UDP 포트)별 사용 현황, 라우터나 스위치의 포트별 트래픽 발생 현황 등과 같은 통계 데이터와 실제 데이터를 송수신하고 있는 패킷들이다.

2. 프로토콜 개요

네트워크에서 수집한 패킷들은 많은 IP와 TCP/UDP 헤더 정보를 포함하고 있기 때문에 패킷들을 명확하게 분석하기 위해서는 TCP/IP 헤더에 대한 이해가 필요하다. <그림 3-38>은 IP 헤더에 포함되는 정보이다.



```

Internet Protocol, Src: 192.216.124.35 (192.216.124.35), Dst: 192.216.124.255 (192.216.124.255)
  Version: 4
  Header length: 20 bytes
  [x] Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    0000 00.. = Differentiated Services Codepoint: Default (0x00)
    .... ..0. = ECN-Capable Transport (ECT): 0
    .... ...0 = ECN-CE: 0
  Total Length: 202
  Identification: 0xdec8 (57032)
  [x] Flags: 0x00
    0... = Reserved bit: Not set
    .0.. = Don't fragment: Not set
    ..0. = More fragments: Not set
  Fragment offset: 0
  Time to live: 128
  Protocol: UDP (0x11)
  [x] Header checksum: 0xe086 [correct]
    [Good: True]
    [Bad : False]
  Source: 192.216.124.35 (192.216.124.35)
  Destination: 192.216.124.255 (192.216.124.255)

```

〈그림 3-38〉 IP 헤더

IP 헤더의 일부 필드는 네트워크 공격이나 침입에 사용된다.

- Flag (3 bits)

- Bit “0”: 일반적으로 0으로 설정
- Bit “1”: 0이면 큰 패킷에 대한 조각이며, 1이면 조각을 허용하지 않는 것이다.
- Bit “2”: 0이면 해당 패킷이 마지막 조각이며, 1이면 마지막이 아니다.
- 이 필드는 Ethernet에서 전송할 수 있는 1518bytes의 이하의 패킷을 강제로 작은 조각으로 분리해서 전송하는 경우에 사용한다. 즉, 대상 시스템의 전송 계층이 아닌 IP 계층에서 패킷을 모두 모아야 상위 계층으로 전달하게 된다. 보안 장비에서 전송 계층의 헤더만 검색하는 경우에는 막을 수 없게 하는 방법으로 사용되기도 한다.

- Time to Live (8 bits)

- 일반적으로 TTL이라고 하며, 해당 패킷이 전송과정에서 통과할 수 있는 최대 라우터 개수를 나타낸다.
- 해당 패킷의 전송 수명을 제한하는 것이며, 송신 시스템에서 특정 값으로 설정되어 라우터를 통과할 때마다 하나씩 감소한다.

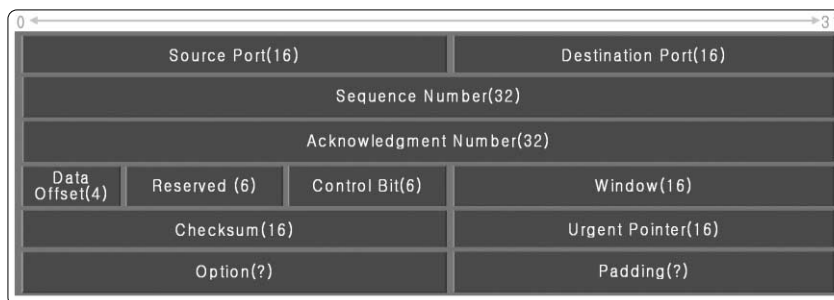


- 이 필드의 값이 “0”에 도달하였을 때에 해당 패킷은 버려지고, 송신 시스템에게 ICMP 프로토콜을 사용하여 전송하는 과정에서 에러가 발생하였음을 통보하게 된다. (무한 라우팅 루프를 방지)
- 내부적으로 시스템 간의 테스트 (local test)를 위하여 “1”로 설정된 패킷을 송신하기도 하지만, 내부의 시스템에서 라우터와 네트워크를 공격하기 위하여 강제로 “1”로 설정하기도 한다.

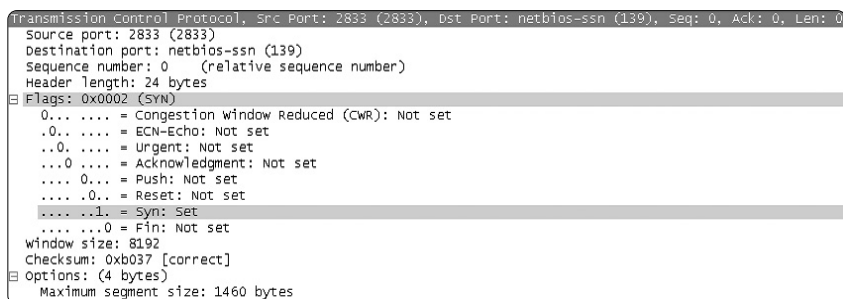
• Protocol (8 bits)

- IP 계층(네트워크)의 상위 계층(전송)에 있는 프로토콜을 표시한다.
- ICMP (1), TCP (6), UDP (17)
- 0부터 255까지의 값을 갖지만, 255는 사용하지 못하게 예약되어 있다. 전송 계층의 프로토콜을 해석하지 못하여 전송 계층의 헤더를 해석하는 장비에서 오류가 발생하도록 공격하는 패킷에서 255로 사용하기도 한다.

〈그림 3-39〉는 전송 계층인 TCP의 헤더이다. TCP 헤더는 시스템과 네트워크를 연결해주는 가장 중요한 계층이기 때문에 가장 복잡하고 공격이나 침입을 파악하기 위하여 주요 필드는 반드시 이해하고 있어야 한다. 대부분의 분석 자료에서 살펴보면, 네트워크 웜(worm), Dos 공격 및 침입 포트 또는 공격 패턴을 설명하기 위하여 많이 인용하는 헤더이다.



제3장 침해사고 분석기술



〈그림 3-39〉 TCP 헤더

침입이나 공격 패킷들을 분석하기 위해서는 다음 TCP 필드들을 살펴보아야 한다.

- Source Port (16 bits)

- 0 ~ 65535까지 정의할 수 있으며, 해당 패킷을 보내는 시스템에서 할당한 논리적인 포트이다. 즉, 해당 패킷에 대한 응답을 받는 시스템의 포트이다.

- Destination Port (16 bits)

- 0 ~ 65535까지 정의할 수 있으며, 해당 패킷을 받는 시스템에서 할당한 논리적인 포트이다.
- 이 두 포트 번호가 상대적으로 서로 일치하면, 한 세션으로 인식된다.
- 네트워크를 통하여 특정 어플리케이션을 공격하는 경우에는, 이 포트가 해당 어플리케이션에서 사용하는 서비스 포트이다. 대부분 한 포트만 공격하지만, 최근의 네트워크 웹들은 여러 개의 포트를 동시에 공격하는 경우도 있다.

- Sequence Number (32 bits)

- 송신자가 전송하는 데이터의 TCP 세그먼트 번호이다.
- 번호는 초기 접속 과정에서 할당되어 전송되는 데이터 세그먼트의 크기만큼 증가한다. 즉, TCP 헤더 다음에 포함되어 있는 데이터 바이트 수만큼 증가한다.



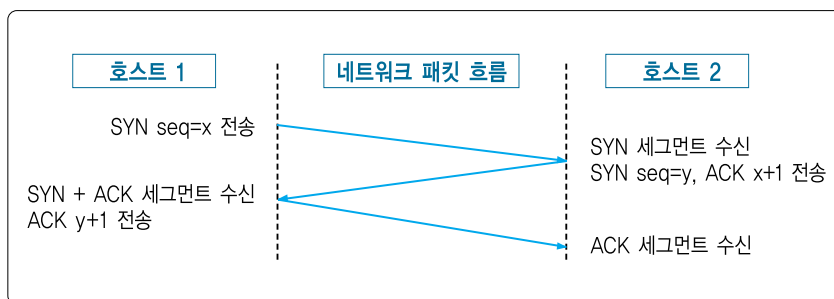
- TCP 헤더 다음에 데이터가 포함되어 있지 않은 경우에는 증가하지 않는다. 즉, Ack 패킷의 경우에는 데이터가 포함되어 있지 않기 때문에 몇 개의 Ack 패킷들에 할당되어 있는 번호가 같게 된다.

- Acknowledgement Number (32 bits)

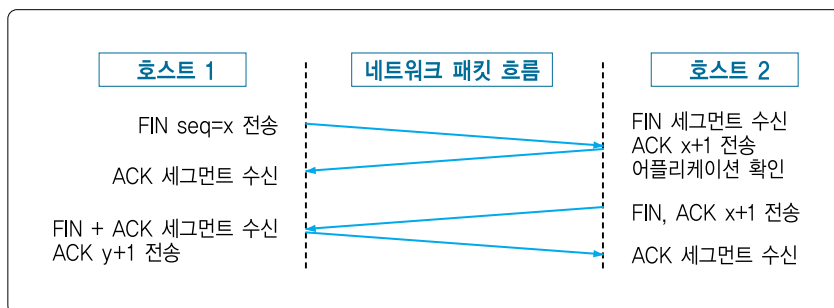
- 송신자가 해당 패킷을 수신하는 호스트로부터 다음에 받아야 하는 패킷의 TCP 세그먼트 번호이다. 즉, 해당 패킷을 수신한 호스트에서는 이 번호가 할당되어 있는 패킷으로 응답한다.

- Control Bits (6 bits)

- TCP 세그먼트의 목적, 즉 해당 패킷의 용도를 표시한다. 각 필드가 “1”로 설정되면, 해당 패킷은 설정된 용도를 위하여 전송되는 것이다.
- URG (Urgent Pointer Field): 긴급을 전달이 필요한 패킷임을 표시한다.
- ACK (Acknowledgement): 이전 패킷에 대한 응답 패킷임을 표시한다.
- PSH (Push): 해당 세그먼트를 메모리에서 재합성하지 말고 어플리케이션에게 바로 전달해야 하는 세그먼트임을 표시한다.
- RST (Reset): 해당 세션을 강제로 종료함을 표시한다.
- SYN (Synchronize): 초기 접속을 시작하는 경우에 사용된다. 이 플래그가 설정된 패킷에는 초기 Sequence Number와 TCP Window 크기가 명시되어 있다. <그림 3-40-A>는 TCP 세션 시작을 위하여 송수신되는 패킷들의 흐름이다.
- FIN (Fin): 세션 종료를 위한 패킷임을 표시한다. <그림 3-40-B>는 TCP 세션 종료 시점에서 송수신되는 패킷의 흐름이다.



〈그림 3-40-A〉 세션 시작을 위한 패킷 흐름



〈그림 3-40-B〉 세션 종료를 위한 패킷 흐름

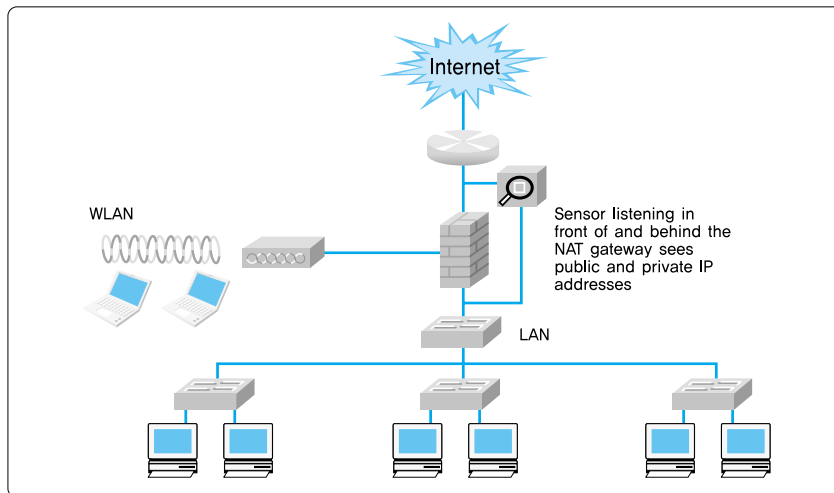
- Window (16 bits)

- TCP에서 데이터의 송수신 과정을 원활하게 하기 위하여 사용되는 버퍼의 크기이다. 초기 접속하는 과정에서 운영체제나 어플리케이션에 따라 다른 크기로 할당된다.
- 데이터 세그먼트를 수신하는 호스트에서는 해당 세그먼트를 버퍼에 누적하는 경우에, Window 크기를 감소시켜서 송신하는 호스트에게 통보해야 하여 데이터 송신을 늦추고, 누적된 세그먼트들을 처리했을 경우에는 다시 Window 크기를 초기화 한다.
- 네트워크 시스템이나 특정 호스트를 공격하기 위하여 많은 패킷들을 송신하는 일부 네트워크 웜의 경우에는 TCP Window가 일정한 크기로 고정되어 있다. 때문에 웜의 특징을 설명할 때에 종종 TCP Window 크기를 명시하기도 한다.

3. 패킷 수집 및 디코드

가. 측정 구간

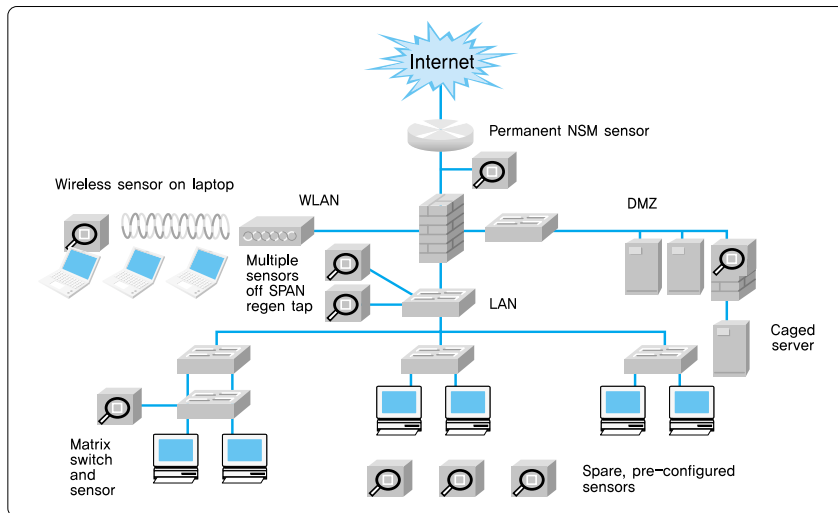
외부의 침입이나 공격을 확인하기 위하여 트래픽과 패킷을 수집해야 하는 위치는 목적에 따라서 다르겠지만, 가장 우선적으로 <그림 3-41>와 같이 라우터와 가까운 위치에서 수집해야 한다. 또한 방화벽을 사용하고 있고, 방화벽에서 IP 주소를 공인에서 사설로 바꾸어 주는 NAT 기능이 동작하고 있다면, 방화벽 앞과 뒤에서 동시에 측정하는 것도 좋은 방법이다. IP 주소는 바뀌지만, 같은 시간대에 생성된 세션을 찾아서 동일한 패킷들로 분석할 수 있다.



<그림 3-41> 트래픽 및 패킷 측정 구간

IP 주소를 변조하여 네트워크를 공격하는 최근의 네트워크 뱀을 찾아서 분석하기 위해서는 내부 네트워크의 전 구간이 측정 대상이 되며, 특히 백본이나 주요 액세스 스위치들의 포트 트래픽과 CPU 상태를 동시에 관찰해야 한다. 대부분의 네트워크 공격들이 외부에서 특정 시스템이 과도한 트래픽을 발생시켜서 라우터나 방화벽 또는 백본 스위치에 영향을 주기 보다는 내부에 있는 호스트가 서로를 공격하도록 하는 DDos (Distributed Dos) 방법을 사

용하고 있다.



〈그림 3-42〉 주요 측정 구간

나. 패킷 수집

대부분의 프로토콜 분석기에는 기본적으로 캡처 옵션들을 설정하여 패킷을 버퍼나 파일로 저장하는 방법을 제공한다. 만약 패킷의 헤더 정보만 분석에 필요하고, 실제 사용자들의 데이터가 필요하지 않은 경우에는 모든 패킷들의 시작부분에서 일정 크기만큼만 캡처하는 기능도 제공한다.

파일로 저장하는 기능에는 일정 개수의 파일을 항상 유지하도록 하는 기능이 있으며, 일정 개수만큼 생성되는 파일에는 항상 최근의 패킷들만 포함되도록 하는 기능도 (Ring Buffer) 유용하게 사용할 수 있다. 즉, 언제 어떤 사고가 발생할지 알 수 없는 상황에서 항상 일정 시간동안의 트래픽 정보와 패킷을 보관하고 있으면, 사고가 발생해도 명확한 원인을 분석할 수 있다.

패킷을 수집하여 파일로 저장할 때, 파일의 크기는 24~32 MBytes로 설정하는 것이 좋다. 파일 크기를 너무 크게 설정하면, 향후에 다시 분석기에서 해당 파일을 읽어 들어서 저



장되어 있는 모든 패킷들을 분석하는데 많은 시간이 필요하게 된다. 저장되는 파일의 개수는 패킷을 수집하는 구간에서 발생하는 트래픽과 패킷 저장이 필요한 시간을 계산하여 설정한다. 즉, 패킷을 캡처하는 동안 파일 하나가 생성되는데 몇 분이 필요한지 파악하면, 원하는 시간동안의 모든 패킷들을 캡처하기 위해서는 몇 개의 파일이 생성되어야 하는지 계산할 수 있다.

다. 트래픽 통계 관찰

트래픽 통계는 NMS나 주요 시스템의 내부 명령어를 사용하여 언제든지 관찰할 수 있으며, 패킷이 캡처된 시간동안의 트래픽 통계도 분석기에서 간단하게 살펴볼 수 있다. 침입에 대한 분석을 하기 위해서는 침입을 탐지하는 시스템의 로그와 침입 대상이 된 시스템의 로그를 함께 관찰하는 것이 좋다. 공격이나 비정상적인 트래픽에 대해서는 시스템에서 제공하는 통계나 분석기에서 제공하는 통계만으로 간단하게 관찰할 수 있다.

- MAC 주소별 트래픽 통계: 사용자들이나 서버들이 연결되어 있는 액세스 스위치 구간에서는 호스트별 MAC 주소를 확인하여 트래픽 발생 상태를 관찰할 수 있다. 하지만, 라우터나 백본 구간에서는 호스트의 MAC을 확인할 수 없기 때문에 중요한 의미를 갖지 못한다.

Endpoints: UDP&TCP Examples.pkt

Ethernet Hosts: 7

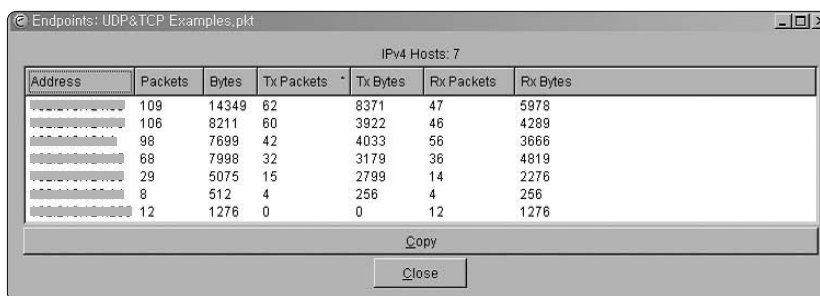
Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
00:0C:29:00:00:00	109	14349	62	8371	47	5978
00:0C:29:00:00:01	106	8211	60	3922	46	4289
00:0C:29:00:00:02	98	7699	42	4033	56	3666
00:0C:29:00:00:03	68	7998	32	3179	36	4819
00:0C:29:00:00:04	29	5075	15	2799	14	2276
00:0C:29:00:00:05	8	512	4	256	4	256
00:0C:29:00:00:06	12	1276	0	0	12	1276

Copy

Close

〈그림 3-43〉 MAC 주소별 트래픽 통계

- IP 주소별 트래픽 통계: 침입이나 공격은 IP를 기반으로 발생하기 때문에 IP 주소별로 모든 트래픽 통계를 분리하여 관찰할 필요가 있다. 만약 네트워크에서 웜이 활동하거나 DDos가 발생하고 있다면, 내부 IP 주소에서 전송하는 패킷 개수가 수신하는 패킷 개수보다 훨씬 많아지게 된다. 또한 패킷 개수에 비하여 송수신되는 바이트 수가 매우 적다는 것을 발견할 수도 있다.



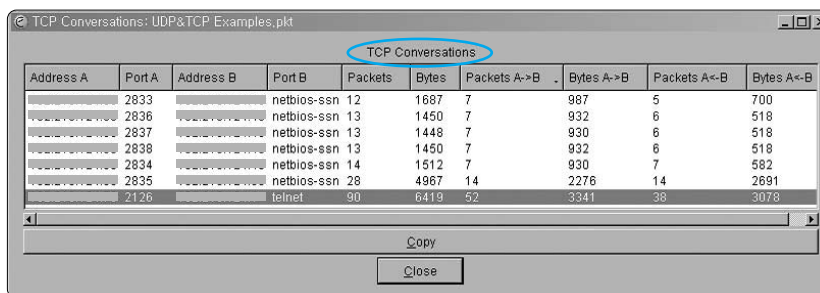
IPv4 Hosts: 7

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
109	14349	62	8371	47	5978	
106	8211	60	3922	46	4289	
98	7699	42	4033	56	3666	
68	7998	32	3179	36	4819	
29	5075	15	2799	14	2276	
8	512	4	256	4	256	
12	1276	0	0	12	1276	

Copy Close

〈그림 3-44〉 IP 주소별 트래픽 통계

- TCP/UDP 세션별 트래픽 통계: 세션별 트래픽 통계에서는 어느 세션이 서버에서 몇 분 동안 작업을 했으며, 어느 정도의 패킷이나 데이터가 송수신 되었는지를 확인할 수 있다. 외부 IP 주소가 서버에 접속하여 무언가 작업을 하거나 데이터를 올리는 경우에는 서버에서 수신한 bytes 수가 더 많을 수도 있다. 대부분의 분석기에는 세션별 트래픽 통계에서 제공하는 IP 주소와 TCP/UDP 포트 번호들을 참조하여 필터를 정의할 수 있으며, 정의된 필터로 해당 세션에 대한 패킷들을 추출할 수 있는 기능을 제공한다.



TCP Conversations

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A->B	Bytes A->B	Packets B->A	Bytes B->A
	2833	netbios-ssn	12	1687	7	987	5	700	
	2836	netbios-ssn	13	1450	7	932	6	518	
	2837	netbios-ssn	13	1448	7	930	6	518	
	2838	netbios-ssn	13	1450	7	932	6	518	
	2834	netbios-ssn	14	1512	7	930	7	582	
	2835	netbios-ssn	28	4967	14	2276	14	2691	
	2126	telnet	90	6419	52	3341	38	3078	

Copy Close

〈그림 3-45〉 TCP 세션별 트래픽 통계



특정 구간에서 일정 시간 동안 모든 패킷을 캡처하면 디코드 화면에 표시되는 패킷들이 무수히 많기 때문에, 필터를 구성하여 관련성이 있는 패킷들만 추출하는 것이 중요하다. 분석기에서는 통계 자료를 기반으로 필터를 정의하는 방법에 대해서 익숙할 수 록 원하는 패킷들을 신속하게 추출할 수 있게 된다.

라. 패킷 디코드

프로토콜 분석기들의 다양한 필터를 사용하여 원하는 패킷들을 추출한 후, 상세하게 분석하기 위해서는 분석기의 디코드 화면을 이해하고 칼럼들을 분석에 적합하도록 구성해야 한다. 패킷 분석에 필요한 디코드 화면과 칼럼 구성은 아래 <그림 3-46>과 같다.

No.	Time	Delta Time	Source	Destination	Length	Protocol	Info	Abs. Time
1	0.000000	0.000000			220	BROWSER	Get Backup List Request	2001-11-29 23:36:54
2	0.000000	0.000000			96	NBNS	Name query NB THE AG GROU	2001-11-29 23:36:54
3	0.060000	0.060000			108	NBNS	Name query response NB 19:	2001-11-29 23:36:54
4	0.060000	0.000000			235	BROWSER	Get Backup List Response	2001-11-29 23:36:54
5	0.060000	0.000000			96	NBNS	Name query NB MIKE-PC<20:	2001-11-29 23:36:54
6	0.060000	0.000000			108	NBNS	Name query response NB 19:	2001-11-29 23:36:54
7	0.060000	0.000000			64	TCP	2833 > netbios-ssn [SYN] :	2001-11-29 23:36:54
8	0.060000	0.000000			64	TCP	netbios-ssn > 2833 [SYN] :	2001-11-29 23:36:54
9	0.060000	0.000000			64	TCP	2833 > netbios-ssn [ACK] :	2001-11-29 23:36:54
10	0.060000	0.000000			130	NBSS	Session request, to MIKE-4	2001-11-29 23:36:54
11	0.060000	0.000000			64	NBSS	Positive session response	2001-11-29 23:36:54
12	0.060000	0.000000			232	SMB	Negotiate Protocol Request	2001-11-29 23:36:54
13	0.060000	0.000000			165	SMB	Negotiate Protocol Respon	2001-11-29 23:36:54
14	0.060000	0.000000			240	SMB	Session Setup Andx Request	2001-11-29 23:36:54
15	0.060000	0.000000			214	SMB	Session Setup Andx Respon	2001-11-29 23:36:54
16	0.060000	0.000000			193	LANMAN	NetserverEnum2 Request, w	2001-11-29 23:36:54
Frame 1 (220 bytes on wire, 220 bytes captured)								
Ethernet II, Src: RPrinter_11:56:de (00:40:95:11:56:de), Dst: Broadcast (ff:ff:ff:ff:ff:ff)								
Internet Protocol, Src: (00:00:00:00:00:00), Dst: (00:00:00:00:00:00)								
User Datagram Protocol, Src Port: netbios-dgm (138), Dst Port: netbios-dgm (138)								
NetBIOS Datagram Service								
SMB (Server Message Block Protocol)								
SMB Mailslot Protocol								
Microsoft Windows Browser Protocol								
<pre> 0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0010 00 c2 de c8 00 00 80 11 e0 86 c0 d8 7c 23 c0 d8 0020 7c ff 00 8a 00 8a 00 b6 bd c3 11 02 8c aa c0 d8 0030 7c 23 00 8a 00 a0 00 00 20 46 44 45 50 45 44 46 #.....FDEPDF 0040 43 43 42 46 45 43 46 46 44 43 41 43 41 43 41 43 CEBPEEFFDCACAC 0050 41 43 41 43 41 43 41 41 41 00 20 46 45 45 49 45 ACACACAA.FEIE 0060 46 43 41 45 42 45 48 43 41 45 48 46 43 45 50 46 FCAEBHC AEHFCEP 0070 46 46 41 43 41 43 41 43 41 42 4e 00 ff 53 4d 42 FFACACAC ABN.SMB 0080 25 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 %..... 0090 00 00 00 00 00 00 00 00 00 00 00 00 11 00 00 06 00a0 00 00 00 00 00 00 00 00 00 e8 03 00 00 00 00 00 00b0 00 00 00 06 00 56 00 03 00 01 00 01 00 02 00 17 V..... 00c0 00 5c 4d 41 49 4c 53 4c 4f 54 5c 42 52 4f 57 53 .MAILSL.OTBROWS 00d0 45 00 09 04 05 00 00 00 04 00 6a 00 00 00 00 00 E.....}. </pre>								

<그림 3-46> 패킷 디코드 화면

대부분 프로토콜 분석기들의 디코드 화면은 유사하다. 먼저 상단 화면에서는 캡처된 패킷들의 목록과 간단한 정보를 표시하며, 중간 화면에서는 상단 화면에서 선택한 한 패킷의

계층별 프로토콜 헤더 정보를 표시한다. 하단 화면에서는 선택한 패킷에 포함되어 있는 실제 내용을 16진수와 ASCII 코드로 표시한다. 다음은 패킷 목록을 표시하는 상단 화면에서 패킷을 관찰하기 위해 필요한 칼럼들이다.

- Time (Relative Time): 기준이 되는 패킷을 0.00초로 하여 다음 패킷들이 캡처되기까지의 시간을 의미한다. 즉, 1번 패킷이 기준 패킷이라고 하면 2, 3, 4번까지 캡처된 시간을 표시한다. 각 패킷들의 Delta Time을 합한 것이다. 이시간은 1초 동안의 몇 개의 패킷이 네트워크에서 발생했는지 또는 특정 패킷까지 몇 초가 걸렸는지를 계산하기 위해서 사용된다.
- Delta Time: 바로 이전 패킷과 해당 패킷 사이의 시간 간격을 의미한다. 위 <그림 3-46>에서 보면, 2번 패킷 다음에 3번 패킷이 캡처된 시간이 0.06초이다. 이 시간은 패킷들이 얼마나 짧은 또는 긴 시간 간격으로 발생했는지를 확인하기 위해서 사용된다.
- Abs. Time (Absolute Time): 각 패킷이 캡처된 실제 시스템의 시간이다. 이시간은 실제 사고가 발생한 시간에 캡처된 패킷을 찾기 위해서 참조하며, 실제 분석에 필요한 것은 아니다.
- Source: 해당 패킷을 송신한 호스트의 주소이다. IP 또는 MAC 주소로 표시할 수 있다.
- Destination: 해당 패킷을 수신하는 호스트의 주소이다. 마찬가지로 IP 또는 MAC 주소로 표시할 수 있다.
- Length: 해당 패킷의 실제 크기이다. MAC 계층의 CRC를 포함하여 표시하기도 하지만, CRC를 제외한 나머지 크기로 표시하기도 한다.



- Protocol: 해당 패킷의 어플리케이션 포트이다. 서버의 TCP/UDP 포트를 근거로 표시된다.
- Information: 해당 패킷에 포함되어 있는 송수신 포트, 패킷의 용도 및 어플리케이션에 대한 간략한 설명이다. 각 패킷의 Seq, Ack 번호 및 TCP Window의 크기도 표시된다.

4. 공격형 트래픽의 특징

공격형 트래픽은 네트워크에 과부하를 발생시키거나 특정 어플리케이션의 서비스를 중단 시키는 형태가 있다. 몇 년 전까지만 해도 어플리케이션의 서비스를 중단시키는 코드를 서버에 삽입하는 (Buffer Overflow) 공격이 많았다. 하지만 최근 몇 년 전부터는 유입되는 네트워크 월들은 서버뿐만 아니라 네트워크 전체에 과부하를 발생시키는 트래픽을 발생시키고 있다.

가. 특정 서버만을 공격하는 패턴

Web, E-mail, DNS 서버와 같은 주요 어플리케이션 서버만을 공격하는 트래픽은 다음과 같은 특징을 가지고 있다.

- 많은 데이터를 전송하는 것처럼 패킷의 크기가 크거나 불규칙하다.
- 해당 서버에서 인식하지 못하는 코드를 전송한다.
- 해당 서버에서 처리할 수 없을 정도의 많은 요청을 보낸다.
- 서버에서 허용되지 않는 문장을 사용하여 요청 패킷을 전송한다.

제3장 침해사고 분석기술

No.	Time	Delta Time	Source	Destination	Length	Protocol	Info
1	0.000000	0.000000	000.000.000.000	000.000.000.000	1514	TCP	[TCP]
2	0.304984	0.304984	000.000.000.000	000.000.000.000	1514	TCP	[TCP]
3	1.280039	0.975054	000.000.000.000	000.000.000.000	1514	TCP	[TCP]

Header length: 20 bytes							
Flags: 0x0010 (ACK)							
window size: 17520							
checksum: 0x8350 [correct]							
TCP segment data (1460 bytes)							

0000	00	05	74	00	50	80	08	00	20	a1	f4	80	08	00	45	00	...	t.P...E.
0010	05	dc	0e	61	40	00	7e	06	62	95	d2	77	a0	5f	3f	f9	...	a@..	b..w..?
0020	d3	55	06	5a	00	50	82	2f	0f	03	2d	f5	a8	a1	50	10	...	U.Z.P./	..-...P.
0030	44	70	83	50	00	00	47	45	54	20	2f	64	65	66	61	75	...	Op.P..GE	T/defau
0040	6c	74	2e	69	64	61	3f	58	58	58	58	58	58	58	58	58	...	lt.ida?x	xxxxxxxx
0050	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	...	xxxxxxxx	xxxxxxxx
0060	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	...	xxxxxxxx	xxxxxxxx
0070	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	...	xxxxxxxx	xxxxxxxx
0080	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	...	xxxxxxxx	xxxxxxxx
0090	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	...	xxxxxxxx	xxxxxxxx
00a0	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	...	xxxxxxxx	xxxxxxxx
00b0	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	...	xxxxxxxx	xxxxxxxx
00c0	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	...	xxxxxxxx	xxxxxxxx
00d0	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	...	xxxxxxxx	xxxxxxxx
00e0	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	...	xxxxxxxx	xxxxxxxx
00f0	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	...	xxxxxxxx	xxxxxxxx
0100	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	...	xxxxxxxx	xxxxxxxx
0110	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	...	xxxxxxxx	xxxxxxxx
0120	58	58	58	58	58	58	58	25	75	39	30	39	30	25	75	36	...	xxxxxxxx	u9090%u6

〈그림 3-47〉 C#Red 패턴

나. 네트워크에 과부하를 발생시키는 공격 패턴

네트워크에 과부하를 발생시켜서 스위치나 방화벽 또는 라우터에 영향을 주는 트래픽은 간단하게 찾을 수 있지만, 가장 어려운 문제는 이러한 트래픽을 발생시키는 호스트를 찾는 것이다. 많은 웜이나 공격이 IP 주소를 변조하기 때문에 발생지를 찾기 어려워지고 있으며, 때문에 이러한 트래픽이 발견되면, 각 스위치의 포트 상태를 점검해야 한다.

- 불특정 IP 주소를 순차적으로 사용하여 한 IP 주소를 대상으로 같은 형태의 패킷을 송신한다.
- 특정 IP 주소에서 불특정 다수의 IP 주소를 대상으로 순차적으로 같은 형태의 패킷이 발생한다.
- 1초에 송신하는 패킷의 개수가 수백 ~ 수 만개까지 발생한다.
- 한개 또는 몇 개의 TCP 포트를 대상으로 SYN 패킷을 전송한다.
- 발생하는 패킷의 크기가 작다.
- 패킷의 개수가 수신보다 송신이 매우 많다. (100 ~ 1000배 이상)



No.	Time	Delta Time	Source	Destination	Length	Protocol	Info
1	0.000000	0.000000	1160.1160	1160.80	60	TCP	1160 > http [SYN] Seq=0 Ack=0 win=16384 Len=0
2	0.000144	0.000144	1160.1160	1160.80	60	TCP	1671 > http [SYN] Seq=0 Ack=0 win=16384 Len=0
3	0.000166	0.000022	1160.1160	1160.80	60	TCP	1450 > http [SYN] Seq=0 Ack=0 win=16384 Len=0
4	0.000307	0.000141	1160.1160	1160.80	60	TCP	1872 > http [SYN] Seq=0 Ack=0 win=16384 Len=0
5	0.000330	0.000022	1160.1160	1160.80	60	TCP	1516 > http [SYN] Seq=0 Ack=0 win=16384 Len=0
6	0.000346	0.000016	1160.1160	1160.80	60	TCP	1729 > http [SYN] Seq=0 Ack=0 win=16384 Len=0
7	0.000470	0.000124	1160.1160	1160.80	60	TCP	1110 > http [SYN] Seq=0 Ack=0 win=16384 Len=0
8	0.000493	0.000023	1160.1160	1160.80	60	TCP	1916 > http [SYN] Seq=0 Ack=0 win=16384 Len=0
9	0.000633	0.000141	1160.1160	1160.80	60	TCP	1741 > http [SYN] Seq=0 Ack=0 win=16384 Len=0
10	0.000657	0.000022	1160.1160	1160.80	60	TCP	1189 > http [SYN] Seq=0 Ack=0 win=16384 Len=0
11	0.000674	0.000017	1160.1160	1160.80	60	TCP	1656 > http [SYN] Seq=0 Ack=0 win=16384 Len=0
12	0.000800	0.000126	1160.1160	1160.80	60	TCP	1888 > http [SYN] Seq=0 Ack=0 win=16384 Len=0
13	0.000838	0.000038	1160.1160	1160.80	60	TCP	1931 > http [SYN] Seq=0 Ack=0 win=16384 Len=0

Transmission Control Protocol, Src Port: 1160 (1160), Dst Port: http (80), Seq: 0, Ack: 0, Len: 0
 Source port: 1160 (1160)
 Destination port: http (80)
 Sequence number: 0 (relative sequence number)
 Header length: 20 bytes
 Flags: 0x0002 (SYN)
 0... .. = Congestion window Reduced (CWR): Not set
 .0... .. = ECN-Echo: Not set
 ..0... .. = Urgent: Not set
 ...0... .. = Acknowledgment: Not set
0... .. = Push: Not set
0... .. = Reset: Not set
1... .. = Syn: Set
0... .. = Fin: Not set
 Window size: 16384
 checksum: 0x7265 [correct]

〈그림 3-48〉 MS Blaster 패턴

〈그림 3-48〉의 패턴을 보면, 여러 IP 주소가 순차적으로 바뀌면서 한 IP 주소에 TCP 80 포트 SYN 패킷을 전송하고 있다. 또한 패킷을 전송하는 시간은 Delta Time을 참조하면, 짧게는 0.00001초에서 길게는 0.00014초 밖에 차이하지 않는다.

No.	Time	Delta Time	Source	Destination	Length	Protocol	Info
1	0.000000	0.000000	100.103.100.103	100.103.100.103	60	TCP	3579 > http [SYN] Seq=0 Ack=0 win=64240 Len=0
2	0.013047	0.013047	100.103.100.103	100.103.100.103	62	TCP	3588 > 2745 [SYN] Seq=0 Ack=0 win=64240 Len=0
3	0.013917	0.000870	100.103.100.103	100.103.100.103	62	TCP	3591 > epmap [SYN] Seq=0 Ack=0 win=64240 Len=0
4	0.036069	0.022152	100.103.100.103	100.103.100.103	62	TCP	3592 > 1025 [SYN] Seq=0 Ack=0 win=64240 Len=0
5	0.036935	0.000866	100.103.100.103	100.103.100.103	62	TCP	3594 > microsoft-ds [SYN] Seq=0 Ack=0 win=64240 Len=0
6	0.045572	0.008637	100.103.100.103	100.103.100.103	62	TCP	3601 > 3127 [SYN] Seq=0 Ack=0 win=64240 Len=0
7	0.052788	0.007216	100.103.100.103	100.103.100.103	62	TCP	3611 > 6129 [SYN] Seq=0 Ack=0 win=64240 Len=0
8	0.060290	0.007502	100.103.100.103	100.103.100.103	62	TCP	3612 > netbios-ssn [SYN] Seq=0 Ack=0 win=64240 Len=0
9	0.061163	0.000873	100.103.100.103	100.103.100.103	62	TCP	3614 > http [SYN] Seq=0 Ack=0 win=64240 Len=0
10	0.073443	0.014279	100.103.100.103	100.103.100.103	62	TCP	3615 > 2745 [SYN] Seq=0 Ack=0 win=64240 Len=0
11	0.076349	0.000906	100.103.100.103	100.103.100.103	62	TCP	3617 > epmap [SYN] Seq=0 Ack=0 win=64240 Len=0
12	0.089203	0.012854	100.103.100.103	100.103.100.103	62	TCP	3625 > 1025 [SYN] Seq=0 Ack=0 win=64240 Len=0
13	0.090109	0.000906	100.103.100.103	100.103.100.103	62	TCP	3626 > microsoft-ds [SYN] Seq=0 Ack=0 win=64240 Len=0
14	0.094909	0.004800	100.103.100.103	100.103.100.103	62	TCP	1378 > microsoft-ds [SYN] Seq=0 Ack=0 win=64240 Len=0
15	0.095597	0.000688	100.103.100.103	100.103.100.103	62	TCP	1377 > 1025 [SYN] Seq=0 Ack=0 win=64240 Len=0
16	0.096291	0.000694	100.103.100.103	100.103.100.103	62	TCP	1373 > 2745 [SYN] Seq=0 Ack=0 win=64240 Len=0

Transmission Control Protocol, Src Port: 3579 (3579), Dst Port: http (80), Seq: 0, Ack: 0, Len: 0
 Source port: 3579 (3579)
 Destination port: http (80)
 Sequence number: 0 (relative sequence number)
 Header length: 28 bytes
 Flags: 0x0002 (SYN)
 0... .. = Congestion window Reduced (CWR): Not set
 .0... .. = ECN-Echo: Not set
 ..0... .. = Urgent: Not set
 ...0... .. = Acknowledgment: Not set
0... .. = Push: Not set
0... .. = Reset: Not set
1... .. = Syn: Set
0... .. = Fin: Not set
 Window size: 64240
 checksum: 0xc2a9 [correct]
 options: (8 bytes)

IPv4 Hosts: 382						
Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
4848	328897	4587	301478	261	27419	
142	18779	65	6427	77	12352	
123	14073	56	5433	67	8640	
83	11630	38	5032	45	6598	
21	5509	11	4263	10	1246	
24	1680	24	1680	0	0	
21	1470	21	1470	0	0	
20	1400	20	1400	0	0	

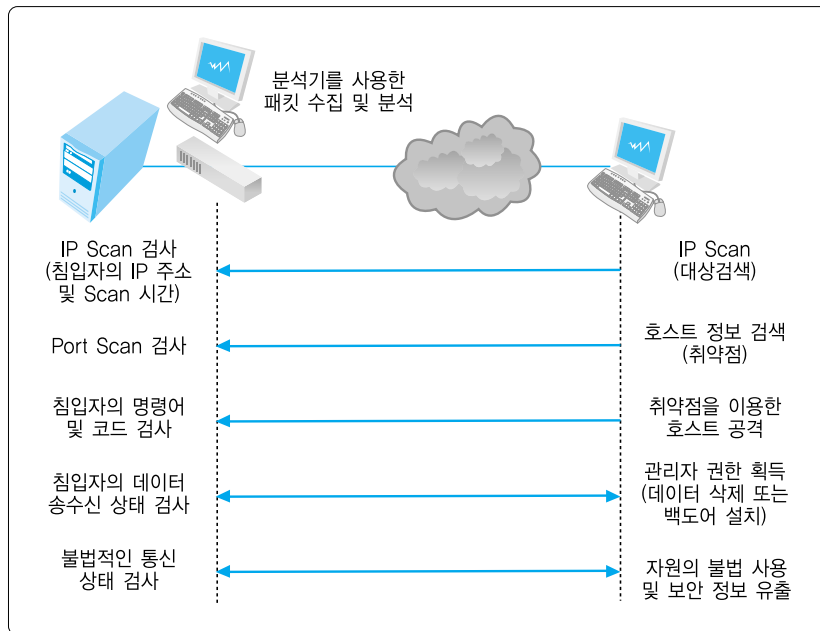
〈그림 3-49〉 Gaobot 패턴

〈그림 3-49〉의 패턴을 보면, 한 IP 주소가 여러 IP 주소들의 다중 TCP 포트에 대해서 SYN 패킷을 전송하고 있다.

5. 침입형 트래픽의 특징

침입하려는 대상을 알고 있는 상태라면, 접속이 가능한 포트를 검색한 후에 접속을 시도하기 위하여 여러 가지 방법이나 도구를 사용한다. 하지만, 대상을 모르고 있다면, 먼저 네트워크를 통하여 접속이 가능한 IP 주소와 포트를 검색한 후에 접속을 시도한다. IP 주소와 포트를 검색하는 과정에서 발생하는 트래픽의 특징은 아래와 같다.

- 해당 IP 주소에서 송신한 패킷 개수와 수신한 패킷 개수가 비슷하다.
- 일반적으로 검색 과정에서는 실제 데이터 송수신이 발생하지 않기 때문에 송수신한 바이트 수도 비슷하다.
- 패킷 가운데 RST이나 login Failure가 포함된 패킷이 많다.
- 검색하려는 IP 주소 또는 포트가 순차적으로 바뀐다.



〈그림 3-50〉 통상적인 침입 과정

다음 〈그림 3-51〉는 네트워크에 연결되어 있는 호스트의 IP 주소를 검색하는 패턴이다. IP 주소를 검색하는 방법으로 가장 많이 사용되는 명령어가 Ping이다. Ping 명령어를 이용하여 Echo request를 전송하면, 네트워크에 연결되어 있지 않는 호스트에서는 응답이 없지만, 연결되어 있는 호스트에서는 Echo Reply로 응답하기 때문에, 가장 간단한 방법으로 사용된다.

제3장 침해사고 분석기술

No.	Time	Delta Time	Source	Destination	Length	Protocol	Info
1	0.000000	0.000000			98	ICMP	Echo (ping) request
2	0.001327	0.001327			98	ICMP	Echo (ping) request
3	0.002624	0.001297			98	ICMP	Echo (ping) request
4	0.003924	0.001300			98	ICMP	Echo (ping) request
5	0.005220	0.001296			98	ICMP	Echo (ping) request
6	0.006510	0.001289			98	ICMP	Echo (ping) request
7	0.007803	0.001293			98	ICMP	Echo (ping) request
8	0.009315	0.001512			98	ICMP	Echo (ping) request
9	0.010647	0.001332			98	ICMP	Echo (ping) request
10	0.013344	0.002696			98	ICMP	Echo (ping) request

Frame 1 (98 bytes on wire, 98 bytes captured)

Ethernet II, Src: Intel_36:44:49 (00:90:27:36:44:49), Dst: Sercomm_67:86:40 (00:c0:02:67:86:40)

Internet Protocol, Src: 192.168.0.16 (192.168.0.16), Dst: ()

Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0

Checksum: 0x1a52 [correct]

Identifier: 0x0200

Sequence number: 0xff00

Data (56 bytes)

〈그림 3-51〉 IP Scan 패턴

〈그림 3-52〉은 TCP 포트를 검색하는 패턴이다. 한 IP 주소의 TCP 포트에 대해서 순차적으로 SYN 패킷을 전송하고 있으며, 대상 호스트에서 해당 포트로 접속을 거부하는 경우에는 RST 패킷이 발생하고 있다. 하지만, 만약 TCP 7(echo), 9(discard)번 포트인 경우에 대해서는 SYN ACK 패킷이 발생하여 접속이 가능함을 알려주고 있다.

No.	Time	Delta Time	Source	Destination	Length	Protocol	Info
1	0.000000	0.000000			60	TCP	2294 > 2294 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
2	0.000290	0.000290			64	TCP	1 > 2294 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
3	0.006753	0.006463			78	TCP	2296 > 2 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
4	0.006890	0.000137			64	TCP	2 > 2296 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
5	0.012702	0.005812			78	TCP	2298 > 3 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
6	0.012809	0.000106			64	TCP	3 > 2298 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
7	0.017518	0.004709			78	TCP	2300 > 4 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
8	0.017627	0.000108			64	TCP	4 > 2300 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
9	0.022309	0.004682			78	TCP	2302 > 5 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
10	0.022443	0.000134			64	TCP	5 > 2302 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
11	0.027259	0.004816			78	TCP	2304 > 6 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
12	0.027386	0.000126			64	TCP	6 > 2304 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
13	0.033003	0.005617			78	TCP	2306 > echo [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
14	0.033260	0.000257			64	TCP	echo > 2306 [SYN, ACK] Seq=0 Ack=1 Win=8191 Len=0 MSS=1460
15	0.033484	0.000224			60	TCP	2306 > echo [ACK] Seq=1 Ack=1 Win=8760 Len=0
16	0.038277	0.004793			78	TCP	2308 > 8 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
17	0.038519	0.000241			64	TCP	8 > 2308 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
18	0.043487	0.004968			78	TCP	2310 > discard [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
19	0.043832	0.000345			64	TCP	discard > 2310 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
20	0.044043	0.000211			60	TCP	2310 > discard [ACK] Seq=1 Ack=1 Win=8760 Len=0
21	0.049285	0.005241			78	TCP	2312 > 10 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
22	0.049539	0.000254			64	TCP	10 > 2312 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
23	0.055083	0.005544			78	TCP	2314 > systat [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
24	0.055193	0.000109			64	TCP	systat > 2314 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
25	0.061076	0.005883			78	TCP	2316 > 12 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
26	0.061192	0.000115			64	TCP	12 > 2316 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0

〈그림 3-52〉 Port Scan Pattern