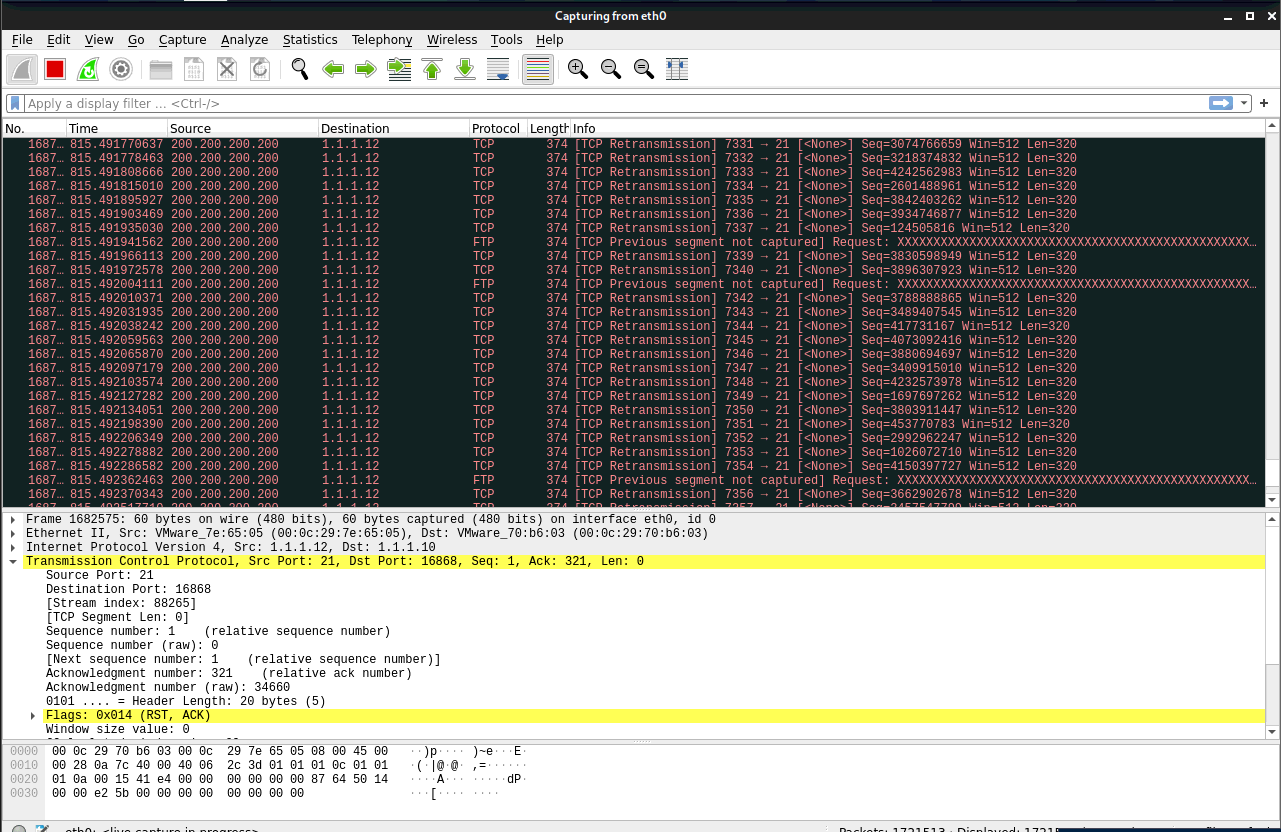
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | KITRI 모의해킹 28기 | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | 작성: 신세은 | | |  | |
|  | DOS | | | | | | |  |
|  | | -Teardrop Attack- | | | |  | | |
|  | |  | | | |  | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |

|  |
| --- |
| 1. Teardrop Attack |
|  |
| 1. Teardrop Attack 이란? |
| * 네트워크상에서 IP가 정상적으로 패킷을 전송할 때 단편화(fragmentation)가 발생하게 되고 수신자는 재조립을 통해 단편화된 데이터를 복구한다. * 이때, 재조립 시 정확한 조립을 위해 offset 이라는 값을 더하게 되는 이 offset값을 단편화 간에 중복되도록 고의적으로 수정하거나 더 큰 값을 더해 그 범위를 넘어서는 오버플로우를 일으켜 시스템의 기능을 마비시키는 방법이다. |
| 1. 원리 |
| C:\Users\USER\Desktop\1.png   |  |  |  | | --- | --- | --- | | **패킷 번호** | **정상 시퀀스 넘버`** | **Teardrop 시퀀스 넘버** | | 1 | 1~101 | 1~101 | | 2 | 101~201 | 81~181 | | 3 | 201~301 | 221~321 | | 4 | 301~401 | 251~351 | |
| * 정상적인 fragmentation 인 경우 1~100, 101~200, 201~300와 같이 fragment간 offset이 차례대로 나열됨을 알 수 있다. * Teardrop 공격 후 fragment 1 과 fragment 2는 offset 이 겹치게 되고, fragment 3와 fragment 4은 사이의 간격이 생기게 되는 것을 확인 할 수 있다. * 이때, offset이 겹치거나 간격이 있으면 재조합을 할 수 없다. |
|  |
|  |
|  |
|  |

|  |
| --- |
|  |
| 2. 공격과정 (필요시 가정이나 예상 시나리오 포함) |
| 실습 환경  공격자 PC(kali linux): 1.1.1.10/25  타겟 PC(cent os7) : 1.1.1.12 /25 |
| 1. 공격 수행 |
| * **# hping3 –a 200.200.200.200 <타겟 IP> --id 3200 –seqnum –p 21 –d 320 –flood**     Hping3 –a 200.200.200.200 1.1.1.12 –id 3200 –seqnum –p 21 –d 320 –flood   |  |  | | --- | --- | | --id 3200 | TCP 패킷의 ID 값, 같은 ID 값이면 동일한 세션의 TCP 패킷으로 간주 | | --seqnum | TCP 패킷의 시퀀스 넘버를 임의로 설정 | | -d 320 | 패킷의 길이를 320 byte로 설정 | |
|  |
| 1. 공격자 PC 패킷 분석 |

* 출발지 주소 200.200.200.200 목적지 주소(타겟PC) 1.1.1.12 인 21번 포트로 시퀀스 넘버가 임의로 생성되어 공격되어지는 것을 확인할 수 있다.

. 

|  |
| --- |
|  |
| 3. 공격 결과 |
|  |
| 1. 패킷 분석(wireshark) |
| * 출발지 주소 200.200.200.200 로 시퀀스 넘버가 정상적이지 않은 임의의 시퀀스 넘버로 공격되고 있는 것을 확인 할 수 있으며, 이러한 신뢰성이 확인되지 않은 데이터 전송에 대하여 반복적인 재요구와 수정을 하게 되는 과정에서 시스템 자원이 고갈되어 과부화가 오는 것을 확인 할 수 있다. |
| 1. 트래픽 과부화 |
| * 공격 직후 타겟 PC 에서 과도한 트래팩이 발생된 것을 확인 할 수 있었고, 속도 또한 저하된 것을 확인 할 수 있다. |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| 4. 보안진단 및 대책 |
| * 현재는 대부분의 OS에서 해당 취약점이 해결되어 이론적으로만 존재하는 공격이다. |
| 1. 침입차단 시스템 (IPS), 방화벽으로 우회할 수 있다. |
|  |
| 1. OS 패치 |

|  |
| --- |
|  |
| 5. 보안대책 적용시 공격결과 (최종 매뉴얼에 포함) |
|  |
|  |
|  |
|  |
|  |
|  |