**Report**

|  |  |
| --- | --- |
| **시원한 열기** | |
| **10170548** | **하가영(조장)** |
| 10170546 | 김유진 |
| 10171151 | 이채림 |
| 10170890 | 정선영 |

2019. 05. 17 ~ 2019. 06. 02

SYN Flooding

&

TCP Connect Scanning

|  |
| --- |
| Attack |
| * SYN Flooding   TCP 세션이 연결될 때의 취약성을 이용한 서버 공격   * **3-way-handshake**   TCP는 세 차례의 패킷들을 주고받는 과정을 거쳐야만 TCP 연결이  성립된다. 이러한 과정을 ‘3-way-handshake‘라고 한다  텍스트이(가) 표시된 사진  자동 생성된 설명  [3-way–handshake] [SYN Flooding]  : 공격자가 SYN+ACK를 받은 후 ACK를 보내지 않으면 Server가 Client와 연결을 하기 위해 RAM 공간을 점점 더 많이 확보해둔  상태에서 대기하게 됨으로 인해 Server의 RAM을 꽉 차게 만들어서 **Server가 서비스를 계속할 수 없게 만드는 공격임**. |

|  |
| --- |
| * CODE   (교수님이 강의 시간에 만든 패킷(pcap 함수 사용) 소스 참고함)스크린샷이(가) 표시된 사진  자동 생성된 설명  **[공격 판별]**  **→ for문을 이용하여 port 배열 개수 만큼 반복함.**  **• 1번째 if문을 이용해 들어오는 포트를 복사한 배열을 들어오는 port와**  **비교하여, 같으면 a = 1로 설정함.**  **• 2번째 if문은 control flags가 SYN\_ACK이면 count를 1증가함. (여기서, 증가한 공격자가 같은 port로 얼마나 들어오는지 확인 하기 위함)**  **• 3번째 if문은 2번째 if문과 같은 방법, ACK이면 count를 0으로 설정함. (ACK이면 연결이 연결된 것이기 때문임)**  **“SYN Flooding공격 가능성이 있습니다\n”**  **[공격 감지 후 메시지 출력]**  **→** for**문을 빠져 나와 a의 값을 이용해 if문을 실행함.**  **• 첫번째 if문 조건에서 a == 0, 즉, ‘port\_cpy’ 배열에 저장되어 있지 않**  **은 포트 번호를 배열에 저장함.**  **• 두번째 if문은 *count* 즉, 들어오는 패킷의 port 번호가 같아 증가한 *co***  ***unt*가 10이상일 경우 “SYN Flooding 공격 가능성이 있습니다.” 라는**  **메시지를 출력함.**  **∴ 결과**  **: 코드를 작성한 후, 가상 머신으로 공격을 시도하였지만, 공격이 되지**  **않았음. 전에는 성공했지만, 최종적으로 공격을 시도하였지만, 되지 안**  **않아, 구글에 검색하여 이유를 알아보니 사용 환경 “Windows Server**  **2012 R2” 자체에서 막아 실패했을 거라 생각됨.** |
| * 당시 성공한 Wireshark Packet 캡쳐   텍스트이(가) 표시된 사진  매우 높은 신뢰도로 생성된 설명  **→ 공격 환경인 칼리 리눅스에서 ‘msfconsole’ 이라는 공격 툴을 이용함.**  **공격 대상 환경 Windows 2012 R2 으로 공격하여 SYN FLOODING을 성공.**    **• SYN packet 가 무한으로 들어오면서 인터넷 연결이 되지 않음.**  **• cmd로 netstat -an 명령으로**  **“활성 연결: 프로토콜 | 로컬 주소 | 외부 주소 | 상태”에서 “상태”에**  **‘SYN\_RECV’라고 뜨면서 성공함.**    **→ SYN packet을 보내고 나면 서버 측에서 ‘RST(긴급 정지)’ 를 보냄.**  **그 후, 다시 SYN packet을 다시 캡쳐 됨.** |

|  |
| --- |
| * Port Scanning   대상의 어떤 포트가 열려 있는지 확인하는 작업  침입 전 취약점을 분석하기 위한 사전 작업   * **TCP Connect Scanning**   **connect() 함수**를 사용해 각 포트별로 접속하여 스캔하는 방식  완전한 TCP 연결을 맺어 신뢰성 있는 결과를 얻을 수 있으나,  속도가 느리고 로그가 남는다.   * 포트가 열려 있는 경우   SYN+ACK 패킷을 수신하면 ACK 패킷을 전송함으로써 연결  (3-way-handshake 방식이라고도 함)   * 포트가 닫혀 있는 경우   텍스트, 지도이(가) 표시된 사진  자동 생성된 설명SYN+ACK 패킷이 아니라 RST+ACK 응답이 온다.  [TCP Connect Scanning]   * CODE   스크린샷이(가) 표시된 사진  자동 생성된 설명  **[본인 IP 주소 가져오는 함수]**  **→ gethostname은 자신(호스트)의 이름을 얻어 오는 함수,**  **Gethostbyname은 호스트 엔트리를 얻어오는 함수임. If문에서 ipv4 주소**  **타입일때, 메모리를 복사 후, ‘addr’ 배열을 return함.**  스크린샷이(가) 표시된 사진  자동 생성된 설명  **[본인 IP 주소 가져오기]**  **→ ip는 정수로 되어 있는데, 이를 문자열로 바꾸어 배열에 저장하기 위해,**  **‘sprintf’ 또는 ‘sprintf\_s’를 이용하여 바꿈.**  **• 위에 선언한 GetDefaultMyIP()를 불러와, default ipv4주소를 얻어온 후,**  **내 컴퓨터 ipv4 주소를 문자열로 저장함.**  **: ‘ main’에 함수를 불러내, 공격자가 보내는 패킷을 걸러 낼 때, 자신의 ip에 서도 보내기 때문에 자신의 ip를 알아내 걸러 냄..**  **스크린샷이(가) 표시된 사진  자동 생성된 설명**  **[판별 및 검출 & 메시지 출력]**  **→ for문으로 ip배열 개수 만큼 반복함.**    • **첫번째 if문에는 들어온 ip와 서로 다른 ip주소를 저장해 놓은 배열을 이용** **해, 같은 ip인지 비교와 들어오는 flags가 SYN일 때 조건을 줘, 같은 ip가 들어오는지 확인함.**  **• 두번째 for문은 들어오는 port를 배열에 저장함. 2- 첫번째 if문과 두번째 if문은 같은 경우 a =1 바꾸고, 아닐 경우, port\_cpy에 저장함.**  **• for문을 나와 if문은 첫번째 for문에 만족하지 않으면 c = 0일 경우와 들어오는 ip와 배열에 있는 ip가 다를 경우, 다른 배열에 strcpy을 이용해 저장함.**  **• 두번째 if문은 다른 port가 20개를 초과할 경우, 공격이 의심된다는 메시지를 출력함.**  **∴ 결과**    **→** SYN Flooding**과 마찬가지로 같은 환경(공격 : 칼리, 공격 대상 : windows 2012 R2)에서 공격을 시도함.**    **•** ‘nmap**(공격 대상의 포트 오픈 여부)’툴**,  Nmap -sT Ip **(공격 대상 주소)명령어를 사용하여 성공함.**  **만든 감지 툴을 돌려 결과 아래와 같은 얻어냄.**  **[초반에 들어온 패킷]** |
| **[중반에 들어온 패킷]**  **텍스트, 배터리이(가) 표시된 사진  매우 높은 신뢰도로 생성된 설명[다른 port가 20개 초과하여 메시지 출력함]** |
| * 느낀 점 * **하가영** : 처음으로 해킹을 탐지하는 툴을 만들어 봤는데 생각보다 과정이 너무 힘들었다. 해킹을 탐지하는 코드를 작성하는 것도 힘들었지만 공격을 제대로 성공하는 게 더 힘들었다. syn flooding 공격을 성공하지 못해 아쉬웠지만 다 같이 열심히 만든 코드로 탐지에 성공할 수 있어서 뿌듯했다. * **김유진** : 몇번의 팀 프로젝트(및 활동)을 했지만, 감지 툴을 만든다는 것이 힘들다는 것을 알았다. 특히, SYN FLOODING이 전에는 성공했지만, 감지를 하기 위해 가상머신을 돌렸지만, 수 많은 시도를 해도 되지 않았다. 여기서, 해킹이라는 것이 어려움이 가득하다는 것을 알았고, 다시 한번 나의 진로에 대해서 생각을 해보기로 했다. 물론, 어려운 과정 속에서도 팀원 들과의 호흡은 잘 맞았고, 물론 다 완성하지는 못했지만, 큰 아쉬움과 충격을 겪는 거와 동시에 의미 있는 프로젝트이지 않을까 싶다. * **이채림** : 처음으로 해킹 공격을 시연해보고 탐지를 해 보았는데 그 과정이 꽤나 복잡했었고, 많은 시간과 공을 들였지만 Syn flooding공격을 시연하는데 실패해서 너무 속상했다. 공격을 계속해서 시도하다가 한번 성공한 적이 있었는데 얄밉게도 공격 시연 영상을 찍으려 하니 또 공격이 안 됐었다. 또한 공격 시도 도중 cmd창에서 netstat -an 명령어를 이용해 SYN\_RECEIVED 상태가 뜨는 것을 정말 보고싶었지만 딱 한번 성공한 것을 제외하고 보지못해서 많이 아쉬웠다. 그렇지만 port scanning 공격이라도 성공해서 뿌듯함은 있었다. 그리고 wireshark에서 탐지하는 일들을 내가 직접 작성한 코드로 탐지가 되어 신기했었다. * **정선영** : 처음에는 이것도 해보고 저것도 해보자, 라는 의욕이 앞서서 시작했지만 결국 하나라도 완성해보자, 라는 생각으로 끝을 맺었다. 공격의 개념을 이해하는 것부터 시작해서 직접 코드를 짜고, 공격하고, 그 공격을 감지해내고 하는 모든 과정이 어려웠고, 힘들었다. 하지만 나중에는 오기가 생겨서 결국 하나를 완성해냈다. 지금까지 했던 프로젝트 중에 가장 성취감이 높았다. 중간에 실패했던 공격은 언젠가 꼭 다시 도전해서 성공시켜 보고 싶다. |

**● *Site***

<https://skibis.tistory.com/68> (SYN Flooding -1 )

<https://sata.kr/entry/DOSDDOS-SYN-Flooding-%EA%B3%B5%EA%B2%A9%EC%97%90-%EB%8C%80%ED%95%B4%EC%84%9C-%EC%95%8C%EC%95%84%EB%B3%B4%EC%9E%90> (SYN Flooding -2 )

<https://medium.com/@dieswaytoofast/direct-server-return-and-syn-floods-fe6a56889b75> (SYN Flooding -3 )

<https://ko.wikipedia.org/wiki/%ED%8F%AC%ED%8A%B8_%EC%8A%A4%EC%BA%94> (port scanning -1 )

<https://itragdoll.tistory.com/59U> (port scanning -2)

<http://ehpub.co.kr/tag/%EC%9E%90%EA%B8%B0ip-%EC%96%BB%EC%96%B4%EC%98%A4%EA%B8%B0/> **(자신의 ip 찾기)**