DOI: 10.3745/KIPSTC.2011.18C.3.127

ALADDIN의 어플리케이션 계층 공격 탐지 블록 ALAB 알고리즘의 최적 임계값 도출 및 알고리즘 확장

유 승 엽[†] · 박 동 규[††] · 오 진 태[†††] · 전 인 오[††††]

요 약

악성 봇넷은 DDoS(Distributed Denial of Service) 공격이나 각종 스팸 메시지 발송, 개인 정보 탈취, 클릭 사기 등 많은 악성 행위에 이용되고 있다. 이를 방지하기 위해 많은 연구가 선행되었지만 악성 봇넷 또한 진화하여 탐지 시스템을 회피하고 있다. 특히 최근에는 어플리케이션 계층의 취약성을 공략한 HTTP GET 공격이 주로 사용되고 있다. 한국전자통신연구원에서 개발한 ALADDIN 시스템의 ALAB(Application Layer Attack detection Block)는 서비스 거부 공격 HTTP GET, Incomplete GET Request flooding 공격을 탐지하는 알고리즘이 적용된 탐지 시스템이다. 본 논문에서는 ALAB 탐지 알고리즘의 Incomplete GET 탐지 알고리즘을 확장하고 장기간 조사한 정상적인 패킷 및 공격 패킷들의 분석을 통해 최적 threshold를 도출하여 ALAB 알고리즘의 유효성을 검증한다.

키워드 : ALADDIN, ALAB, DDOS, HTTP GET Request, Incomplete GET

Optimal thresholds of algorithm and expansion of Application-layer attack detection block ALAB in ALADDIN

Seungyeop Yoo[†] · Donggue Park[††] · Jintae Oh[†††] · Inho Jeon[††††]

ABSTRACT

Malicious botnet has been used for more malicious activities, such as DDoS attacks, sending spam messages, steal personal information, etc. To prevent this, many studies have been preceded. But malicious botnets have evolved and evaded detection systems. In particular, HTTP GET Request attack that exploits the vulnerability of the application layer is used. ALAB of ALADDIN proposed by ETRI is DDoS attack detection system that HTTP GET, Incomplete GET request flooding attack detection algorithm is applied. In this paper, we extend Incomplete GET detection algorithm of ALAB and derive the optimal configuration parameters to verify the validity of the algorithm ALAB by the study of the normal and attack packets.

Keywords : ALADDIN, ALAB, DDOS, HTTP GET Request, Incomplete GET

1. 서 론

오늘날 고도의 정보화 사회에서 컴퓨터 통신의 급속적인 발전과 인터넷 사용이 폭발적으로 증가함에 따라 네트워크를 통한 사이버 공격 또한 발달하고 있다. 공격 형태가 다양해지고 전파속도가 단축되면서 더욱 치명적인 공격 형태로 진화하여 최근 인터넷 서비스 제공 업체 및 공공 기관의 업무를 방해하여 개인의 이득을 취하려는 다양한 방식의 분산 서비스 거부(DDoS) 공격의 증가에 따른 DDoS 공격 탐지 및 방어 기술 개발의 필요성이 요구되고 있다.

DDoS 공격은 전 세계적으로 악성 바이러스(악성 봇)를 유포시키고 악성 봇넷을 이루어 대상 서버에 대역폭, 프로세스 처리 능력 및 시스템 자원을 고갈시켜 정상적인 서비스를 제공하지 못하게 만드는 악성 행위이다. 이를 방지하기 위해 많은 연구가 선행되었지만 악성 봇넷 또한 진화하여 탐지 시스템을 회피하고 있다. 특히 기존 IRC(Internet Relay Chat) 프로토콜을 이용하던 악성 봇넷은 IRC 프로토콜을 차단함으로써 HTTP(Hypertext Transfer Protocol) 프로토콜로 이주가 많이 발생하게 되었다. 이는 HTTP를 이용하는 서비스가 많아지고, 네트워크 트래픽의 대부분을 HTTP가 차지함으로써 보안 제품에서 HTTP 트래픽에 대

† 준 회 원 : (주)지엔텔
†† 정 회 원 : 순천향대학교 정보통신공학과 교수(교신저자)
††† 종신회원 : 한국전자통신연구원 책임연구원
†††† 종신회원 : 호서대학교 글로벌창업대학원 교수
논문접수 : 2011년 1월 16일
수 정 일 : 1차 2011년 3월 28일
심사완료 : 2011년 3월 28일

한 검사가 잘 이루어지지 않았었기 때문이다. 즉 보안 제품에 친화적인 HTTP 프로토콜을 이용함으로써 탐지 시스템을 회피하고자 하는 노력인 것이다.[1] 그 외에도 감시를 피하기 위한 많은 노력을 하고 있다.

ALADDIN의 ALAB은 서비스 거부 공격 중 HTTP 프로토콜을 사용하는 HTTP GET Request, Incomplete GET flooding 공격을 방어하는 알고리즘이 적용된 탐지 시스템이다. 기존의 HTTP GET Request flooding 공격을 탐지하는 방법으로는 단순한 rate limit부터 정상적인 웹 접근 패턴 평균에서 벗어나는 패턴의 수를 측정하는 방법, 사용된 URI(Uniform Resource Identifier)를 이용하는 방법까지 다양한 방법을 사용하고 있으나 탐지 알고리즘의 복잡성과 낮은 탐지율이 문제가 되어왔다. ALAB의 알고리즘은 매우 간단하면서 일반적인 통신 패킷에서는 전혀 볼 수 없는 특성을 공격 패킷의 특성을 이용한다는 장점을 가지고 있다.

본 논문에서는 ALAB에서 사용하는 알고리즘의 특성에 맞게 기본 설정 변수 값 interval과 threshold가 적절한지 검증하고 최적의 설정 변수를 도출할 것이다. 탐지 효율의 문제는 곧 탐지 기준을 얼마나 적합하게 잡느냐에 따라 크게 달라질 수 있는 문제이며 ALAB 뿐만 아니라 대부분의 탐지 알고리즘에 설정 변수 최적화 연구는 반드시 필요하다. 또한 ALAB는 세션 성립 이후 Incomplete GET Request 공격을 탐지하기 위해서 GET만을 고려하고 있기 때문에 GET 방식과 유사한 POST 요청 방식에 의한 Incomplete Request flood 공격에서 false negative를 유발할 가능성이 있다. 따라서 ALAB에서 Incomplete GET 탐지 알고리즘을 확장하여 다양한 HTTP GET flooding 공격 툴과 일반적인 HTTP GET 행위들의 데이터를 수집하여 최적의 설정 변수들을 도출하고 HTTP GET flooding 공격을 가하였을 때 기존 알고리즘과 확장된 알고리즘의 정확성을 검증할 것이다.

본 논문의 구성은 다음과 같다. 2장에서 HTTP GET flooding 공격의 특징과 기존의 공격 분석 방법과 HTTP GET flooding 공격을 방어하기 위한 연구에 대해 논하고, 3장에서는 ALAB알고리즘의 오탐의 예와 확장된 알고리즘을 제안한다. 4장에서 최적의 설정 변수 값 도출하고 확장된 알고리즘이 적용된 프로그램을 구현하여 실험 결과 탐지율과 탐지 시간 정도를 알아본다. 5장에서 결론을 내린다.

2. 관련 연구

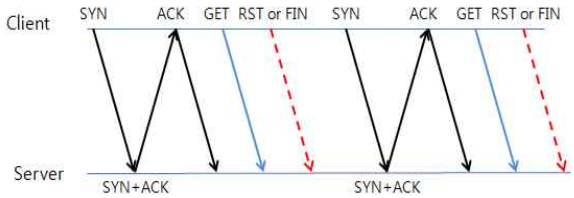
2.1 HTTP GET Request 공격

HTTP GET Request 공격은 보통 TCP session을 반드시 열게 된다. session이 이루어지고 HTTP GET 요청을 빨리 반복 되었을 때 그것이 HTTP Sever의 과도한 응답을 요구하게 되는 현상으로 웹 서비스를 제공하는 서버에 장애를 일으키는 분산 서비스 거부 공격의 진화된 형태로 어플리케이션 계층 공격 중 하나이다.

HTTP GET 공격의 종류는 다음 두 가지로 나뉠 수 있다.

2.1.1 반복적인 단일 TCP connection 이후 GET Request

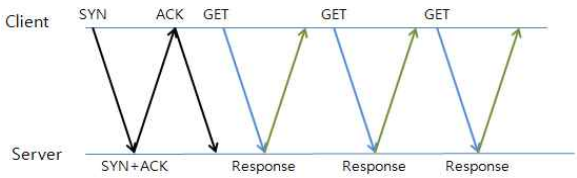
단일 TCP connection 이후 GET Request는 3 ways hand shake로 세션을 열고 GET Request를 한번 요청하여 바로 세션을 닫는 세션의 생성과 종료를 지속적으로 만들어 내는 형태로 공격대상 서버들을 끊임없이 응답하게 만들어 서버 장비에 부하를 주는 공격이다. 이 공격은 더 적은 공격 대역폭으로 공격대상에게 더욱 큰 손해를 입힌다. 매번 HTTP GET이 생성되기 전에 TCP SYN이 먼저 생성되고 srcIP와 dstIP 간에 세션을 형성하기 때문에 간혹 TCP SYN flood 공격으로 보일 수 있다. (그림 1)은 HTTP GET Request 공격 패킷 흐름을 보여주고 있다.



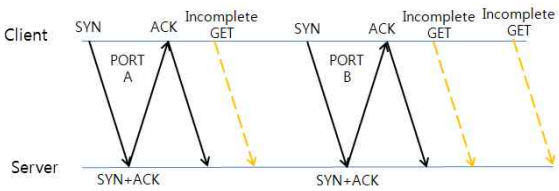
(그림 1) HTTP GET Request 공격 패킷 흐름

2.1.2 단일 TCP connection 안에 다중 GET Request

단일 TCP connection 안에 다중 GET Request는 HTTP 공격의 새로운 형태로써 HTTP 1.1의 특성을 이용하여 (그림 2)와 같이 단 하나의 TCP connection에서 다중 GET Request를 발생 시킨다. 따라서 이 공격은 SYN rate limit 탐지 방식으로는 탐지 할 수 없는 방식이며 하나의 connection 안에서 멀티 GET 을 만들어 동일한 페이지 또는 DB에 부담을 주기위해 동일한 페이지를 지속적으로 요청하거나 탐지를 피해 페이지 수를 변경하면서 지속적으로 요청하는 공격 형태가 될 수 있다.



(그림 2) 단일 TCP connection 안에 다중 GET Request



(그림 3) 단일 TCP connection 안에 다중 Incomplete GET Request

(그림 3)은 특정 서버의 timeout 내에서 끊임없이 Incomplete GET을 서버에 전송하여 Complete GET이 들어올 때까지 기다리게 만들어 다른 정상적인 사용자가 서버에

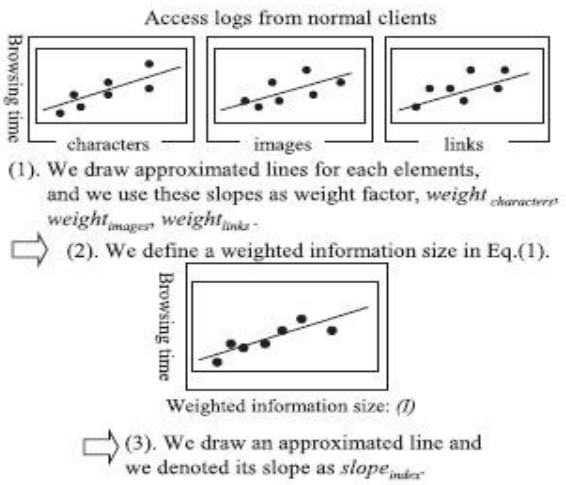
접근할 수 없도록 허용된 접근 기회를 모두 고갈시켜 정상적인 서비스 제공을 방해하는 공격 형태이다.

2.2 기존 HTTP GET flooding 공격 탐지 연구

2.2.1 브라우저 행동기반 탐지기법[2]

브라우저 행동기반 탐지기법은 2가지 알고리즘을 제안하고 있다. 첫 번째, 웹 브라우징 패턴 순서가 같은 호스트를 공격자라고 판단한다. 동일한 바이러스 또는 봇에 감염된 pc가 특정 서버에 지속적으로 동일한 페이지 브라우징을 한다. 동일한 브라우징 방식으로 지속적으로 웹 페이지에 접근하는 방식으로 4개의 레코드를 만들어 비교하여 공격자 IP를 탐지하게 된다. 대규모 네트워크에서 수백만대의 좀비 pc 또는 일반 pc가 동시에 공격 또는 악의 없이 접근하더라고 웹 요청이 많아 순서 비교를 위한 연산이 증가한다.

두 번째, 각 소스 IP별로 접근 웹 페이지와 브라우징시간을 기록하여 브라우징 시간과 information 길이 correlation을 계산하여 correlation이 낮으면 공격으로 판단한다. 정상적인 클라이언트가 웹 페이지에 접근할 때 정보의 크기에 따라 웹 브라우징 시간이 증가하게 된다. 이런 특성을 이용하여 (그림 4)와 같이 정상적인 slope를 계산하여 소스 IP의 지난 접근 기록을 계산하여 slope_index를 얻어낸다. 현재 slope_client값이 slope_index 이상이면 정상 트래픽이고 미만이면 공격 트래픽으로 판단한다.



(그림 4) slope_index 계산 알고리즘

2.2.2 per URL counting[3]

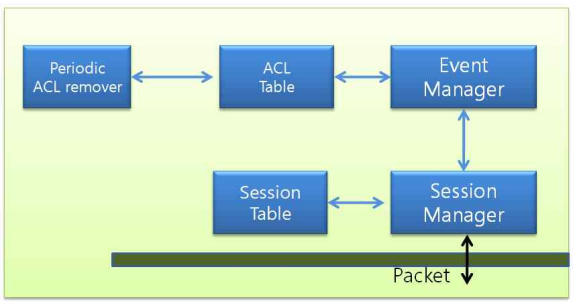
per URL counting은 합법적인 사용자와 봇의 웹 서버 접근 행동을 구별하기 위해서 한 srcIP에서 웹 서버에 요청하는 URL 정보가 정상적인 경우 매우 반복적이지 않지만 봇의 웹 서버 접근의 경우는 동일한 URL 정보가 반복적으로 요청되는 특징을 이용하여 HTTP GET 공격을 탐지하는 방식이다. per-URL 카운팅 메커니즘은 위 그림과 같이 3단계 (1)~(3)로 나뉜다. 첫 번째 단계는 어떤 URL이 공격을 받고 있는지 사용된 모든 URL에 대한 카운팅을 한다. $N_{th}^1 = \alpha \times I_t \times C_s (\alpha = 15)$ 에서 α는 상수, I_t는 측정 인터벌 시간, C_s는 해당 URL에 연결된 IP 주소의 개수를 의미한다. 카운팅 한 결과 $C_{r_i} > N_{th}^1$ 이면 해당 URL은 HTTP GET flooding 공격을 받고 있는 것으로 판단하여 PAL(Potential Attacker List)에 전달한다. 두 번째 단계, PAL 테이블을 만들어 공격자를 찾아내기 위해 srcIP들이 특정 URL을 사용한 횟수를 카운팅한다. 주어진 시간 I_t 동안 각 srcIP에서 victim URL에 요구한 HTTP GET의 회수 S_i를 저장하고 PAL 테이블에서 threshold는 다음과 같이 계산된다. $N_{th}^2 = \alpha \times I_t$. $S_i > N_{th}^2$ 일 경우 대응되는 IP주소는 공격자의 IP로 탐지된다.

이 탐지 방법은 false positive와 false negative가 0을 갖는 매우 낮은 오탐률을 기록하며 공격을 막는 좋은 성능 결과가 있었으나 알고리즘이 다른 사용자의 패킷의 영향을 받아 예상과 다른 결과를 얻을 수 있으며 DB 공격에 사용되는 페이지 넘버링과 같은 유사한 URL을 연속적으로 보내서 시스템에 과부하를 주는 공격에는 적용되지 않는다.

3. ALAB의 탐지 알고리즘 및 확장 알고리즘

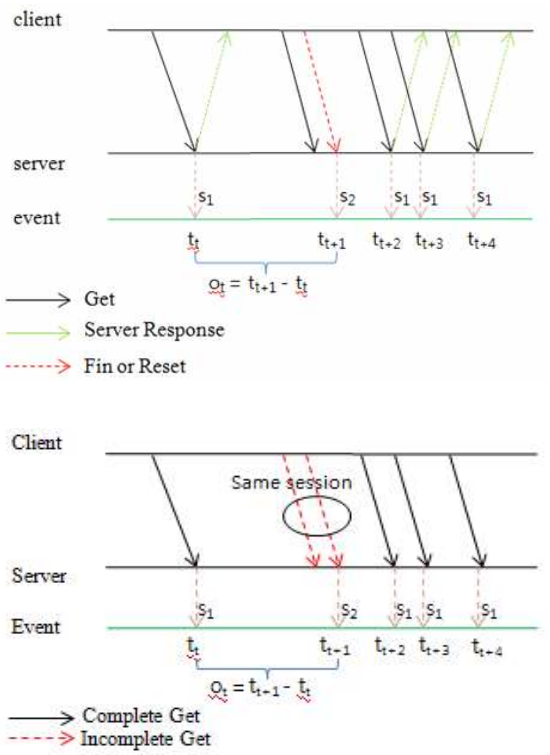
3.1 ALAB의 탐지 알고리즘

ALAB 의 시스템의 구성은 (그림 5)와 같다. 각 구성 모듈 및 저장소간에 흐름은 다음과 같다. 패킷을 Session Manager가 받아 세션이 이루어졌는지 확인한다. 확인되는 과정은 모두 Session Table에 저장되고 세션이 이루어지고 HTTP GET Request와 정상적인 통신과 구별 할 수 있는 두 가지 상태로 세션을 정의한다. 정상적인 상태를 S1, 이상 상태를 S2로 정의하여 상태 정보를 Event Manager에게 전송한다.



(그림 5) ALAB 탐지 시스템 구성도

TCP connection에서 GET Request를 하고 정상적으로 응답이 오면 해당 connection을 상태 S1으로 정의하고 GET Request를 전달하고 바로 (그림 6)과 같이 Response를 받기도 전에 바로 FIN 또는 RST를 서버에 전송하여 connection을 끊는 connection을 상태 S2로 정의한다. 또한 Incomplete GET이 들어온 상태에서 Incomplete GET이 들어왔을 경우도 S2로 정의한다.



(그림 6) ALAB S1, S2 상태 정의

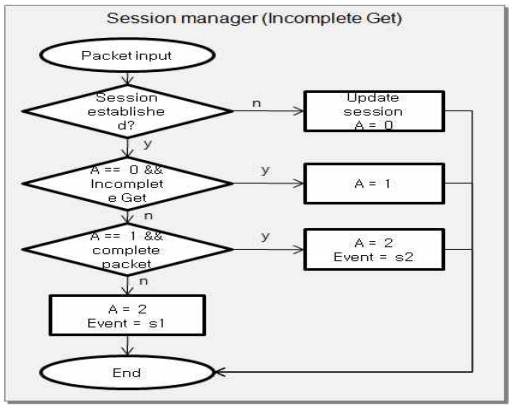
상태 S2를 받은 Event Manager는 S2를 만들어낸 세션 정보를 ACL 리스트에 있는지 확인하고 t 시간 동안 발생한 횟수를 누적시키면서 threshold 변수로 설정된 l(max)와 동일해지는 순간 알람을 울리게 된다. Periodic ACL remover는 정해진 t 시간이 지나면 자동으로 ACL에 있는 모든 정보를 제거한다. ALAB는 이와 같이 각 세션마다 정의된 상태 S2의 시간대 비율을 측정하는 방식으로 공격 패킷을 탐지한다.

3.2 SESSION MANAGER 확장

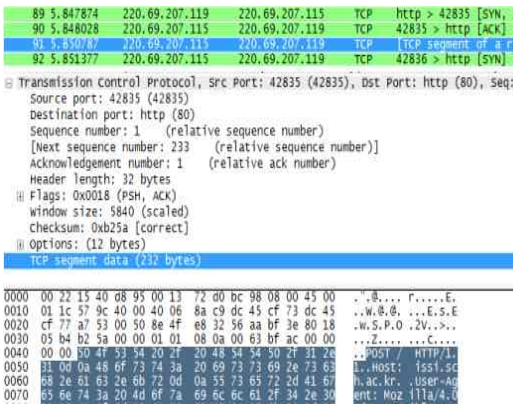
ALAB session manager는 세션을 유지하고 GET의 이상 상태를 탐지하는 주요한 탐지 모듈이다. 적용된 두 가지 탐지 알고리즘 중에 Incomplete GET을 탐지하는 알고리즘은 다음 (그림 7)과 같다.

패킷을 입력으로 받아 세션이 성립되면 A=0, 세tus이 성립 되었으니 Incomplete GET이면 A=1, A=1인 상태에서 세션을 종료하는 패킷이 들어오면 A=2, 그리고 이상 상태(S1)을 발생시키고 정상적으로 종료시키는 complete packet이 들어오면 A=2 그리고 정상 상태(S2) 이벤트를 발생시킨다.

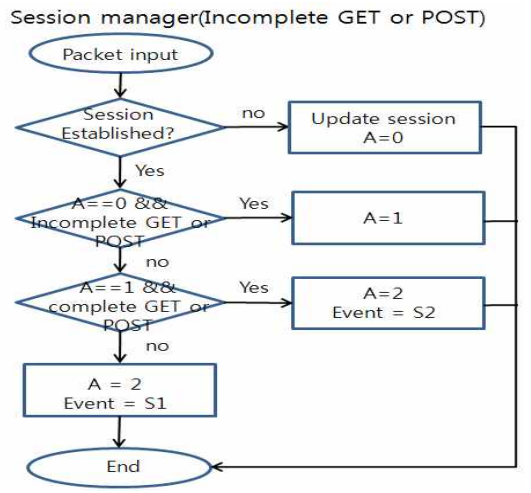
하지만 Slowloris를 통해 공격 패킷을 분석한 결과 (그림 8)과 같이 GET 패킷이 아닌 POST 패킷을 사용하여 공격을 행하는 모습을 보게 되었고 공격은 성공적으로 1초 만에 Apache server 2.2 유효 connection 450개를 모두 점유하였다. POST를 이용한 공격은 ALAB 알고리즘에 탐지 되지 않는다는 것을 의미한다. POST 방식과 GET 방식의 차이는 GET이 서버에 어떤 데이터를 가져와서 보여주는 행위



(그림 7) Session manager incomplete GET 탐지 알고리즘



(그림 8) Slowloris의 POST 패킷을 이용한 공격



(그림 9) 확장된 Session manager incomplete GET 탐지 알고리즘

및 검색 행위와 같으며 POST는 서버의 값이나 상태를 바꾸기 위해서 사용한다는 점에 있다. Incomplete GET과 Complete GET의 차이는 Incomplete GET 패킷의 마지막에 CRLF(\r\n)가 존재하지 않는다는 것이다. 물론 POST를 이용한 패킷에도 CRLF(\r\n)가 존재하지 않았다.

따라서 기존 ALAB의 Incomplete GET 탐지 알고리즘을 POST를 이용한 Incomplete POST도 탐지 할 수 있어야 할 것이다. (그림 9)는 기존 ALAB의 알고리즘을 확장 한 알고리즘이며 패킷 검사 시 GET 뿐만 아니라 POST 패킷도 검사 할 수 있도록 하였다.

4. ALAB의 기존 및 확장 알고리즘 성능 검증

네트워크 트래픽 이상을 탐지하는 기법 중에 가장 기본적인 방법으로 미리 이전의 정상적인 데이터로 threshold를 정의하고 네트워크에서 들어오는 현재 데이터를 비교하여 thresholds를 초과하는 순간 탐지 알람을 울리는 방법을 사용한다[4].

기존의 탐지 알고리즘에서는 최적의 threshold를 얻기 위해서 수분의 시간을 기다려야하는 지연시간이 있어 실시간 탐지에 어려움이 있는 반면 본 탐지 알고리즘은 정상적인 패킷 연속 패턴과 확실히 구별되는 간단한 알고리즘을 사용하여 지연시간이 거의 없다고 볼 수 있다. 본 탐지 알고리즘에서 가장 중요한 변수는 이상행위의 발생 횟수와 interval이다. 본 장에서는 정상적인 통신 패턴을 공격으로 잘못 탐지하지 않는 범위에서 모든 공격을 탐지 할 수 있는 최적의 threshold과 interval을 도출하고 도출된 threshold 및 interval을 적용하여 기존 탐지 기법들과 확장된 ALAB 알고리즘의 탐지율을 비교한다.

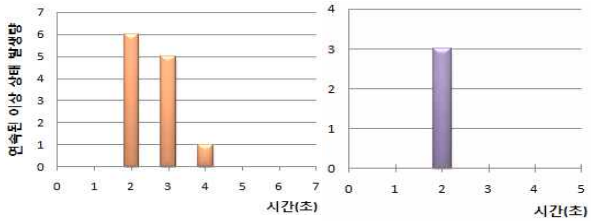
4.1 최적의 threshold, interval 도출

탐지 알고리즘에서 threshold 설정에서 가장 중요한 것은 최대한 낮은 오탐율을 얻는데 있다. 그리고 기존 ALAB 알고리즘에서 S2는 2회, interval은 1초를 기본 설정 값으로 정하였으나 본 논문에서는 <표 1>과 같이 탐지 방법에 따라 threshold를 GET+FIN or RST 연속 패턴에 의한 S2는 C1, Incomplete GET에 의한 S2는 C2 그리고 interval은 It1, It2로 따로 정의하고 최적 변수를 도출한다.

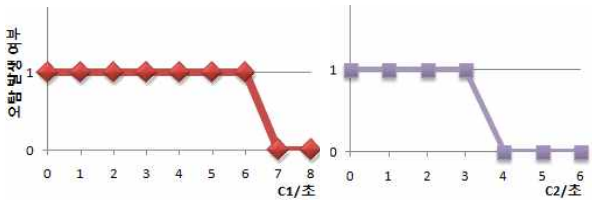
도출 실험을 하기 위해서 1주일 동안 공격이 없는 정상 TCP 패킷 5G byte 대략 1 천만 개의 패킷을 수집하였다. 또한 의도적으로 S2를 발생시킬 가능성이 있는 행위로 웹서핑, 파일 다운로드, 업로드, 웹에서 동영상 및 음악 플레이, 음성 통화 그리고 기타 복합적인 행위를 수행하였고 분석 프로그램을 통해 분석한 결과 C1과 C2를 증가 시키는 이상행위가 발생 할 수 있다는 것을 발견하게 되었고 (그림 10)과 같은 이상 상태 발생 샘플을 추출하였다.

Incomplete GET의 C2값의 최적 값을 간단히 얻어낼 수 있다. 브라우저의 종류에 따라 URI의 최대 길이가 다르며 Microsoft internet Explorer의 경우 2,048 byte를 허용한다. 추가되는 쿠키 크기 4Kbyte와 합치면 60,048 byte이며 패킷 허용 크기는 1,460 byte 로 최대 5개 패킷으로 나뉘며 S2가 (그림 10(b))와 같이 최대 3회 나올 수 있다. C2 값을 3회 이하로 정하게 되면 Incomplete GET이 자주 발생하지는 않지만 정상적인 통신 흐름에서 탐지될 수 있는 소지가 될 수 있다. 따라서 Incomplete GET의 경우 C2는 4회로 정하였다. 4회 이상 Incomplete GET이 한 세션에서 연속 발생한다면 정상적이지 않은 이상 패턴으로 보아야 한다.

정상적인 패킷은 추출한 공격 특성의 연속된 패턴과 쉽게 구별될 수 있기 때문에 본 탐지 알고리즘은 실시간으로도 탐지가 가능하다. 따라서 interval은 최대한 짧게 1초 로 잡아도 문제가 없을 것이라 생각된다. interval이 증가함에 따라 C1, C2 값이 선형적으로 증가하는 비례 관계이며 1초안에 사용자의 의도치 않은 행위로 인한 이상 패킷 패턴의 연속이 나오기는 어려워 보인다.



(a) C1 관련 S2 발생량 (b) C2 관련 S2 발생량
(그림 10) 정상 패킷에서 발생한 S2 발생 샘플



(a) C1 변화에 따른 오탐지 (b) C2 변화에 따른 오탐지
(그림 11) 초당 C1, C2 값에 따른 오탐지 발생 가능성

<표 1>과 같이 ACL for incomplete GET의 threshold는 4회 interval은 1초로 정하였다. ACL for GET 또한 1초의 interval을 주었고 1주일 동안 수집한 공격이 없는 패킷에서 발생한 C1에 해당하는 이상행위(그림 10(a))를 분석한 결과 (그림 12)와 같이 Internet Explorer에서 새로 고침에 의한 것으로 밝혀졌고 기존 ALAB의 탐지 알고리즘으로 새로 고침 버튼이 눌리는 경우 잘못 탐지할 수 있다는 가능성을 발견하였다.

Time	Source	Destination	Protoco	Info
10.355306	220.69.207.111	74.125.155.106	TCP	53280 > http [SYN] Seq=
10.486375	74.125.155.106	220.69.207.111	TCP	http > 53280 [SYN, ACK]
10.486449	220.69.207.111	74.125.155.106	TCP	53280 > http [ACK] Seq=
10.486572	220.69.207.111	74.125.155.106	HTTP	GET / HTTP/1.1
10.544372	220.69.207.111	74.125.155.106	TCP	53280 > http [RST, ACK]
10.547442	220.69.207.111	74.125.155.106	TCP	53281 > http [SYN] Seq=
10.674544	74.125.155.106	220.69.207.111	TCP	http > 53281 [SYN, ACK]
10.674595	220.69.207.111	74.125.155.106	TCP	53281 > http [ACK] Seq=
10.674675	220.69.207.111	74.125.155.106	HTTP	GET / HTTP/1.1
10.718757	220.69.207.111	74.125.155.106	TCP	53281 > http [RST, ACK]
10.720924	220.69.207.111	74.125.155.106	TCP	53282 > http [SYN] Seq=
10.848068	74.125.155.106	220.69.207.111	TCP	http > 53282 [SYN, ACK]
10.848142	220.69.207.111	74.125.155.106	TCP	53282 > http [ACK] Seq=
10.848267	220.69.207.111	74.125.155.106	HTTP	GET / HTTP/1.1
10.889083	220.69.207.111	74.125.155.106	TCP	53282 > http [RST, ACK]

(그림 12) Internet Explorer에서 새로 고침 버튼이 연속 눌렸을 경우 발생 패킷

새로 고침 버튼이 평상시 눌리는 횟수는 많지 않다. 하지만 의도적으로 연속 누르는 경우 사람이 누를 수 있는 회수는 한계점이 있으며 그 한계점 바로 위 값을 threshold로 정하여 의도적인 공격과 구별할 수 있다. 또한 그 값이 매우 작기 때문에 공격 탐지에도 문제가 없을 것이라 생각된다. (그림 10(a))가 평상시 새로 고침 버튼이 연속으로 눌렸을 때 발생한 연속 이상상태를 잘 표현하고 있다. 또한 그 한계점은 초당 6회 이하이다.

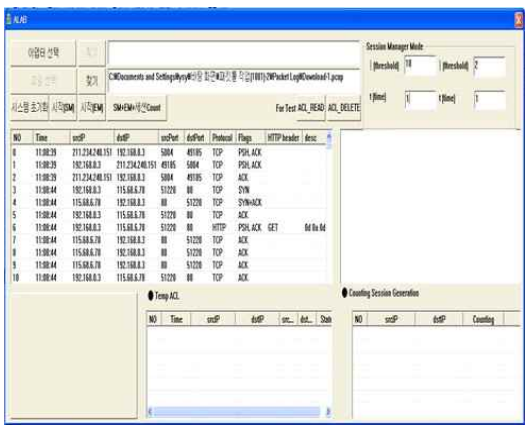
(그림 10, 11)을 바탕으로 다음 <표 1>과 같이 interval과 threshold를 도출하였고 어떠한 정상적인 웹 서버 접근 행위에도 잘못 탐지하는 일은 발생하지 않을 것이라 생각된다.

〈표 1〉 설정 변수 interval과 threshold 별도 정의

detection method	ALAB interval/threshold	expanded ALAB interval/threshold
GET+FIN or RST	interval=1s threshold=2	interval(It1) = 1s threshold(C1) = 7
Incomplete GET		interval(It2) = 1s threshold(C2) = 4

4.2 ALAB 탐지율 검증 소프트웨어 구현

ALAB 프로그램에서 패킷을 수집하기 위해 패킷 캡쳐링 툴로 널리 알려진 Ethereal과 Wireshark를 개발한 CACE technologies에서 제공하는 윈도우 패킷 캡쳐 라이브러리 WinPcap[5]을 사용하였다.



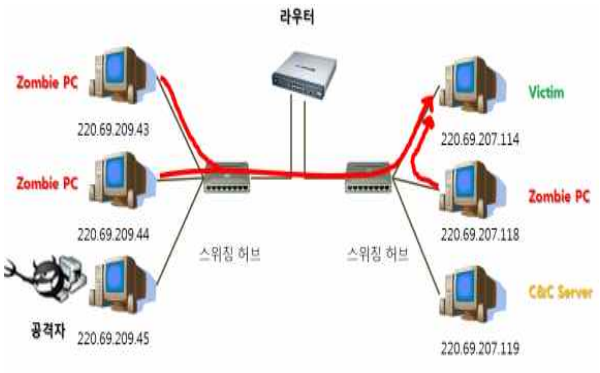
(그림 13) 프로그램 실행 예

ALAB 프로그램에서 패킷을 수집하는 방법으로 사용 가능한 네트워크 인터페이스 목록을 획득하여 실시간으로 캡쳐하는 기능과 패킷 모니터링 툴로 수집하여 저장된 pcap 파일을 읽어오는 정적인 방법이 가능하나 본 논문에서는 정적으로 얻어진 상황별 표준 패킷 로그들을 입력 데이터 값에 알고리즘을 적용하였다. (그림 13)은 구현된 프로그램을 실행한 예를 보여주고 있다.

프로그램의 전체적인 실행 흐름은 패킷 로그 파일을 불러와서 분석 여부를 선택한다. 시작(SM) 버튼을 눌러 S2 상태를 만든 세션의 정보만 temp_acl Table에 저장한다. t(time) 시간 단위로 EM 알고리즘을 적용하여 ACL 테이블에 저장하면서 C(발생회수) 값을 누적시킨다. C값이 threshold와 동일해지는 순간 탐지 알람을 출력한다.

4.3 실험 환경 구성

구현한 프로그램은 ALAB에서 제안하는 알고리즘의 유효성을 검증하는 것이므로 알고리즘의 특성상 하나의 세션 특성만으로 탐지가 가능하기 때문에 (그림 14)와 같이HTTP GET flooding 공격을 가하기 위한 소규모 네트워크를 구성하였다. 1대의 공격자 PC, 3대의 좀비 PC, 1대의 C&C Server, 1대의 타깃 서버로 구성된다.



(그림 14) 공격 패킷 로그 수집 네트워크 환경

패킷 수집은 Victim PC에서 수행하였고 Victim PC는 Window 7 32 bit운영체제와 응용프로그램으로 Wireshark-win32가 유일하게 설치되어 있다. 또한 위 네트워크는 인터넷이 연결되어있지 않은 독립된 네트워크로 구성된다.

4.4 성능 평가 비교

확장된 알고리즘이 적용된 ALAB 프로그램에 도출된 최적 설정 변수 값을 적용하여 HTTP GET flooding 공격 하는 공격 툴 BlackEnergy, NetBot, Slowloris와 웹서핑, 파일 다운로드, 업로드, 웹에서 동영상 및 음악 플레이, 음성 통화 그리고 기타 복합적인 행위들의 패킷으로 실험한 결과 탐지율 및 공격을 탐지하는데 걸린 소요시간을 측정하였다.

〈표 2〉 설정 변수 도출 방식에 따른 탐지율

Detection rate \ Method		GET, FIN or RST	Incomplete GET, POST
ALAB	true positive	66.6%	0%
	false positive	≈0.1%	33.3% (false negative)
Expanded ALAB	true positive	66.6%	33.3%
	false positive	0%	0%

〈표 3〉 확장 ALAB의 설정 변수 도출 방식에 따른 공격 탐지 소요 시간

탐지 소요 시간 / 공격 툴	GET, FIN or RST	Incomplete GET, POST
Blackenergy	0.011 second	X
NetBot	0.05 second	X
Slowloris	X	0.005 second

탐지 결과 <표 2>와 같이 기존의 ALAB 탐지 알고리즘과 확장된 ALAB 알고리즘의 탐지율을 비교하였다. 기존의 ALAB 탐지 알고리즘에서 웹서핑 도중 눌리는 새로 고침 버튼에 의한 잘못된 탐지가 발생하여 0.1% false positive 유발하였고 post 패킷을 이용한 incomplete post 공격을 탐지 하지 못한 false negative를 유발하였다. 확장된 알고리즘에서는 공격 툴의 공격 모두 탐지 되었으며 정상 패킷에서는 전혀 이상 패킷 특성을 보이지 않아 오탐이 전혀 발생하지 않았고 <표 3>과 같이 0.01초 내외에서 탐지가 가능했다.

<표 4>는 다양한 탐지 방법들의 성능 차이결과를 비교하여 보여준다. 모든 기존 연구들은 공격자의 트래픽에 구별될 수 있는 유일한 특성을 가지고 있지 않는 반면 ALAB와 본 논문의 확장된 ALAB에서만 고려를 하고 있다.

확장된 ALAB는 기존의 사용하는 설정 변수와 다르게 GET+RST 또는 FIN의 연속된 사용에서 탐지하는 기준 threshold와 Incomplete GET을 탐지하는 threshold를 다르게 주었다. 또한 Incomplete GET 외에 Incomplete POST도 발생할 수 있다는 가능성을 고려하여 더 확장된 탐지 알고리즘으로서 높은 탐지율을 보일 것으로 생각된다.

5. 결 론

HTTP GET 공격과 정상적인 통신 패킷 특성 차이를 이용하여 HTTP GET flooding 공격을 탐지하는 ALAB와 동일하게 프로그램으로 구현하여 ALAB 알고리즘의 성능을 검증하고 최적의 설정 변수를 도출하였다.

ALAB 알고리즘 실험 프로그램 구성한 네트워크 환경에서 공격 툴 공격 행동과 일반 사용자가 행할 수 있는 웹 서버 접근 행동의 패킷 로그를 수집하여 실험하였으나 Slowloris 공격의 일부가 요청 방식 중 POST 방식을 사용하는 사례가 발견되어 ALAB session manager 알고리즘을 일부 확장하여 POST 방식 까지 탐지할 수 있도록 하였다.

정상적인 패킷에서 S2가 발생할 수 있는 최대 수치를 예상하고 오탐이 발생하지 않는 범위에서 고정된 최적의 설정 변수를 도출하였다.

이는 기존에 설정된 기본 설정변수 t=1s, threshold=2와 다르게 이상 상태 종류에 따라 다르게 나타났다. 도출된 설정 변수를 적용하여 Blackenergy, slowloris, Netbot 그리고 수집한 1 천만 개의 정상 HTTP 패킷로그를 실험 대상으로 탐지한 결과 100%탐지율과 0% 오탐율을 얻어냈다.

ALAB에 오탐의 요인이 될 수 있는 문제를 발견하여 알고리즘을 확장하였고 3개의 공격 툴을 사용하여 알고리즘 유효성 검증에 부족한 면이 있지만 HTTP GET 공격이 성공하기 위해선 공격 도구의 소스 코드는 3개의 공격 툴과 유사하게 만들어질 것이라고 본다. 곧, 단시간에 많은 GET을 발생시키기 위해 GET과 FIN, RST 조합이 나오기 마련이다.

이와 같은 조합을 가지는 공격 도구는 모두 탐지 할 수 있으며 실험 결과 정상 패킷들에서는 오탐이 전혀 발생하지 않았다는 점이 매우 주목할만하다. 더욱이 0.01초 내외의 아주 짧은 시간에 공격을 탐지 할 수 있으며 실시간 탐지에 전혀 무리가 없다고 생각된다.

참 고 문 헌

[1] Markus J, Zulfikar R., "Crimeware: Understanding New Attacks and Defenses", Addison Wesley Professional, ISBN 0-321-50195-0, April, 2008.

[2] Takeshi Yatagai, et al., “A HTTP Flooding Detection Method

〈표 4〉 탐지 기법 성능 비교

Detection methods	Model	False-positive	True-positive
Detection of HTTP GET flood attack[2]	Correlation with browsing time	1% 10%	22% 100%
Web based traffic anomally[6]	WGLR	0.69%	86%
	EPD	0.35%	86%
HTTP flooding detection method[7]	HSMM classify sessions by inline request	6.6%	98.43%
ALAB t = 70, l = 2	HSMM classify by event generation	≈0.1%	≈100%
Proposed algorithm considering POST It1=1s, C1=7, It2=1s, C2=4	HSMM classify by event generation	≈0%	≈100%

Based on Browser Behavior", 2007.
[3] Jinghe Jin, Nazarov Nodir, Chaetae Im, Seung Yeob Nam, "Mitigating HTTP GET Flooding Attacks through Modified NetFPGA Reference Router", 1st Asia NetFPGA Developers Workshop, Daejeon, Korea, June 14, 2010.
[4] Maxion Roy A, "Anomaly detection for diagnosis", in Proceedings of the 20 th Internationl symposium Fault-Tolerant computing(FTCS-20), 1990. 20-27.
[5] http://www.winpcap.org/
[6] Jun LV, Tong Li, Xing Li, "Network Traffic Prediction Algorithm and its Practical Application in real network", Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference, 18-21 Sept., 2007.
[7] Wei-Zhou Lu, Shun-Zheng Yu, "An HTTP Flooding Detection Method Based on Browser Behavior", Computational Intelligence and Security, 2006 International Conference, 1151 - 1154, 3-6 Nov., 2006.

유 승 엽
e-mail : yoosy35@nate.com
2008년 순천향대학교 정보통신공학과 (학사)
2011년 순천향대학원 정보통신공학과 (석사)
관심분야 : 네트워크 보안, 시스템 보안

박 동 규
e-mail : dgpark@sch.ac.kr
1992년 한양대학교 전자공학과(공학박사)
1999년~2003년 순천향대학교 정보기술 공학부 부교수
2004년~현 재 순천향대학교 정보통신 공학과 교수
관심분야 : 네트워크 보안, 유비쿼터스 컴퓨팅 보안

오 진 태
e-mail : showme@etri.re.kr
1990년 경북대학교 전자공학과
1992년 경북대학교 전자공학과(석사)
2011년 충남대학교 컴퓨터공학과(박사)
1992년~1998년 한국전자통신연구원 선임 연구원
1998년~2001년 MINMAX tech. and Engedy Networks 부장
2001년~2003년 Winnow Tech. 공동창업, CTO, 부사장
2005년~2009년 한국전자통신연구원 보안게이트웨이 팀장
2003년~현 재 한국전자통신연구원 책임연구원
관심분야 : 정보보호, 고속 하드웨어 설계사

전 인 오
e-mail : eric@hoseo.edu
2000년 중앙대학교 경영학과(석사)
2005년 호서대학교 컴퓨터응용기술학과 소프트웨어공학(공학박사)
2004년~2005년 호서대학교 디지털비즈 니스학부 겸임교수
2005년~현 재 호서대학교 벤처전문대학원 교수
2005년~현 재 호서대학교 글로벌창업대학원 교수
2006년~2007년 서울벤처정보대학원대학교 전시컨벤션학과장
관심분야 : 소프트웨어공학(S/W 품질보증과 평가 및 품질감리), 전시/컨벤션산업 ,중소기업창업