HTTP Get Flooding 기술을 이용한 APT(지능적 지속 위협)공격 도구의 설계와 구현

천우봉[†] · 박원형[††] · 정태명[†††]

요 약

최근 사이버공격을 보면 전세계 해킹공격 트랜드로 APT 공격이 지속 발생하고 있다. 특히, HTTP Get Flooding 공격은 사이버공격기법 중 가장 효과적인 공격 중 하나이다. 기존의 HTTP Get Flooding 공격 기술에 대해 알아보고 ATP 공격 특성을 결합한 새로운 공격 기술을 제안한다. 본 논문은 HTTP Get Flooding 기술을 이용하여 효과적인 APT 공격 도구 제작 관한 내용이다. 이 공격도구를 통해 지속적인 DDoS 공격에 대한 적극적 방어 대책이 필요하다.

주제어 : APT 공격 도구, HTTP Get Flooding 공격, 분산서비스거부공격

Design and Implementation of ATP(Advanced Persistent Threat) Attack Tool Using HTTP Get Flooding Technology

Woo Bong Cheon[†] · Won Hyung Park[††] · Tai Myoung Chung[†††]

ABSTRACT

As we can see from the recent cyber attack, APT(Advanced Persistent Threat) is trend of hacking attack in the World. Thus, HTTP Get Flooding attack is considered to be one of the most successful attacks in cyber attack method. In this paper, designs and implements new technique for the cyber attack using HTTP get flooding technology. also, I need a defence about DDoS attack through APT Tools.

Keywords : Advanced Persistent Threat Attack, HTTP Get Flooding Attack, DDoS

† 정 회 원: 성균관대학교 정보통신대학원 박사과정
†† 정 회 원: 서울과학기술대학교 산업정보시스템공학과 겸임교수
††† 정 회 원: 성균관대학교 정보통신대학원 교수
논문접수: 2011년 08월 29일, 심사완료: 2011년 09월 06일, 게재확정: 2011년 11월 03일

1. 서 론

최근 APT(Advanced Persistent Threat) 공격[1]으로 3.4 디도스가 발생하였으며, 지난 2009년 7월 7일에 발생한 사이버테러[2]를 전후로 HTTP Get Flooding[3] 공격이 DDoS 공격의 주를 이루고 있다. 이는 Get Flooding 공격이 효과적으로 홈페이지를 마비시키고 사고조사와 대응을 힘들게 하기 때문이다. 또한, 해커는 마치 생존을 위해 보호색(protection coloration)으로 위장하는 동물의 생존 본능처럼, 공격자들도 공격에 사용되는 트래픽을 최대한 정상적인 유저처럼 가장하면서도 대량의 집중적인 접속을 통해 공격 대상이 정상적인 서비스를 진행하지 못하도록 함으로써 치명적인 피해를 유발하도록 하는 것과 같은 맥락이라 할 수 있다.

또한, 이전의 양상과 달라진 초기에는 보안 관리자들조차 정상적인 접속과 구별하기 어려운 새로운 형태의 공격에 대해 어떻게 모니터링하고 대응해야 하는지 전혀 갈피를 잡지 못하였고 정상 접속과 공격을 구분할 수 없어, 공격이 당하고 있는 것을 보고도 달리 손을 쓸 수가 없었다. 이후 한 번 두 번씩 유사한 공격들을 반복적으로 경험하면서 점차적으로 공격 트래픽의 양상을 이해할 수 있게 되었고, 공격의 특성을 파악, 대응해 오면서 지금까지 온 것이 사실이다.

해커는 지금까지의 창을 막아왔던 방패를 뚫기 위해, 한 단계 더 진일보한 새로운 형태의 공격을 준비하고 있을 것으로 보이며 조만간 그 공격이 수면위로 가시화되고 있다.

웹을 공격하는 공격자의 목적은 단순하다. 바로 대량의 접속을 통해 해당 사이트의 “웹 서비스를 중지”시키는 것이다. 사실 GET Flooding 공격은 최근에 나온 용어 같지만 실제 오래전부터 사용되었는데, 수년 전 웹에 대한 초기의 DoS 공격 양상은 매우 단순하다고 할 수 있다. 이를테면 단순히 웹 서버에 대량의 세션을 맺거나 특정 이미지나 고용량 파일을 반복적으로 다운로드 받아 웹 서버의 세션이나 대역폭을 가득 채우거나 데몬 프로세스를 가득 채우는 형태의 어찌 보면 매우 단순한 공격이었다고 할 수 있다. 그러다 보니, 이러한 공격은 방화벽에서 IP당 세션을 제한하거나 CDN과 같은 고용량 서비스를 이용하면 서비스 중단 없이 간단하게 대응 가능한 초보적 수준의 공격이라고 할 수 있다. 즉, 솔루션이 있는, 대응할 수 있는 형태의 공격이라는 것이며 요즘은 거의 사용되지 않는 공격 형태이다.

이후 공격도 점차 진보하여 7.7을 전후로 하여 최근까지 게시판(bbs)이나 공지(notice)의 특정 게시물을 요청하거나 특정 단어를 검색하는 소위 dynamic 콘텐츠에 대한 요청이거나 아예 특정 홈페이지(http://www.example.com)와 같이 메인 사이트를 대상으로 여러 좀비들이 동시에 GET 요청을 반복적으로 하면서 웹 서버의 부하를 유발하는 사례가 대부분이라고 할 수 있다. 이는 dynamic 콘텐츠가 DB와 통신하면서 좀 더 많은 부하와 응답시간을 유발할 수 있고, 결과적으로 이전의 방식보다 좀 더 효과적인 공격이기 때문이다. 거기에다 HTTP 표준[4]에 맞는 형식 즉, 정상적인 문법에 맞게 GET Request를 한다면 공격 트래픽으로 차단되는 것도 피할 수 있어 더할 나위 없을 것이다. 이렇듯 일정량 이상의 DDoS 공격이 발생한다면 대부분의 사이트들이 제대로 서비스하지 못하고 DDoS 공격을 받아 장애가 유발될 것이다. 이렇듯 정교하게 만들어진 APT 공격도구를 이용해서 특정 기관에 대한 공격시 비정상적인 DDoS 공격과 같은 대량 접속에 대해서는 쉽게 피해가 발생할 수 있다.

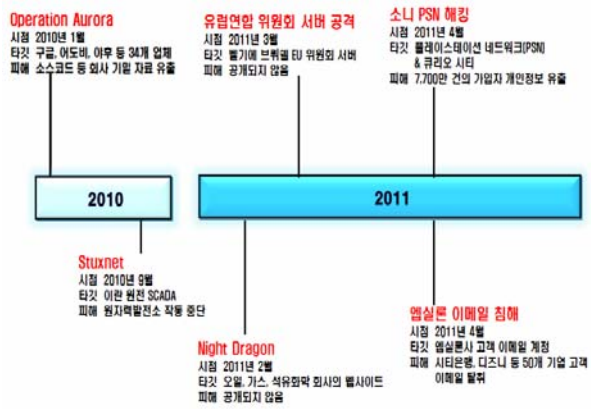
2. 관련 연구

2.1 APT(Advanced Persistent Threat)

APT(Advanced Persistent Threat)는 최초 2006년 미 공군 사령부에서 정부 기관들과의 원활한 커뮤니케이션을 위해 특정 보안 위협의 형태를 지칭[7][8]하며 최근 전세계적인 해킹공격 트랜드로 다양한 IT 기술과 방식들을 이용해 조직적으로 경제적이거나 정치적인 목적을 위해 다양한 보안 위협들을 생산해 지속적으로 특정 대상에게 가하는 일련의 행위를 의미한다.

또한, 지능적 지속 위협의 공격은 정부, 사회기간산업시설, 정보통신, 금융 등 국가중요기반시설을 대상으로 한다. 이에 대한 사례로 2010년 오퍼

레이션 오로라(Operation Aurora)의 회사기밀자료 유출, 2011년 3월 유럽연합위원회서버 공격 등 최근들어 많은 사고가 발생하고 있다[5].



<그림 1> APT 공격 사례

국내의 경우 3.4 디도스 공격 등 특정 대상을 노리는 공격이 증가하는 추세이다. APT 공격자는 기초 정보 수집, 악성코드 침투, 기밀 정보유출 과정을 거친다. 즉, SNS 등 다양한 방법으로 공격 대상조직에 대한 정보를 수집하고 해당 조직 임직원에게 악성코드 이메일을 보내 지속적인 타깃 공격을 한다. 이후 공격 목표인 조직 내부에서 사용하려는 시스템의 취약점을 악용해 정보를 은밀하게 빼내거나 파괴한다.

2.2 동적(Dynamic) 콘텐츠

웹사이트의 구성은 크게 이미지나 플래쉬, CSS, JS등 클라이언트에서 해석하는 static(정적) 콘텐츠와 php나 asp, cgi 등 서버에서 실행하여 실행된 결과를 클라이언트에게 보여주는 dynamic(동적) 콘텐츠로 나눌 수 있다. static 콘텐츠는 서버의 성능만 뛰어나고, 대역폭만 충분하다면 서비스를 하는 데에 크게 문제가 없다. DNS의 Round Robin 기반 또는 L4스위치 등을 이용하여 수평적으로 서버를 늘릴 수도 있고, 간단하게는 CDN과 같은 서비스를 이용해도 된다. 주로 이러한 정적인 콘텐츠에 대해 안정적으로 서비스를 하기 위해서는 아파치나 IIS와 같은 일반 웹 서버가 아닌 Squid와 같은 캐시(cache) 시스템을 활용하면 많은 접속이 몰려도 빠르면서도 안정적으로 서비스가 가능하다. 그러나 이는 개발 프로그램을 수정하고 DB를 마이그레이션 해야 하는 등 단기간에 해결 가능한 수준이 아니라 오랜 시간과 노력이 필요한 작업인 것이다. 또한 static 콘텐츠에 대한 대량 폭주에 대비하여 쉽게 사용이 가능했던 캐시(cache) 시스템을 도입해도 전혀 효과가 없다고 할 수 있다. 왜냐하면 기본적으로 dynamic 콘텐츠는 로그인 정보등과 같이 매번 실시간으로 변경되어야 하므로 데이터의 결과를 캐시할 수 없으며, 캐시시스템에서는 dynamic 콘텐츠에 대해 캐시가 처리하는 것이 아니라 origin 웹 서버로 그대로 바이패스(bypass)만 하게 되기 때문이다. 공격자는 이러한 특성을 이용하여 GET Flooding 공격 수행 시에 dynamic 콘텐츠에 대해서만 집중적으로 공격을 한다.

2.3 HTTP Get Flooding

공격자의 GET Flooding 공격에 사용되는 URL은 아래 <표 1> 같이 3가지가 있다.

<표 1> HTTP GET Flooding 공격 유형[6]

URL 형식	페이지 유형
http://www.example.com/	홈페이지의 첫 페이지
http://www.example.com/bbs/bbs_list.php?search_key=attention&search_value=attack	게시판이나 공지의 특정 게시물
http://www.example.com/bbs/bbs_list.php?search_key=attention&search_value=attack	게시판에서 특정 단어에 대한 검색시도

첫째, 사이트의 첫 페이지를 요청하는 것인데 대부분의 사이트가 첫 페이지에 가급적 실시간적인 많은 정보를 종합적으로 보여주기 위해 많은 dynamic 콘텐츠를 호출하게 된다는 특징을 이용한다. 또한, 사이트에 처음 접속하기 위해서는 대부분이 메인 페이지로 접속한 후 메뉴를 클릭해서 하위 메뉴로 들어가기 때문에 일반적으로 가장 많은 정상 접속이 있다는 점이 APT 공격에 효과적이다. 최근 웹방화벽에서 URL에 대한 특정 문자열로 차단하는 기능을 제공하고 있는데, 만약 관리자가 특정 사이트로 DDoS 공격이 들어와서 이 접속을 차단하게 되면 거의 모든 서비스가 중지될 수 있으므로 현실적으로 차단이 불가능 하다.

둘째, 게시물의 특정 콘텐츠를 공격 대상으로 삼는 것으로 사실 관리자에게 공격이 쉽게 노출될 수 있다는 한계가 있다. 이는 주로 첫 페이지가 static 콘텐츠로 구성된 사이트인 경우 그 대상이 되는데, 방화벽과 같은 보안장비에서 위의 URL로 차단하면 미봉책이나마 어느 정도 대응을 할 수는 있다. 물론 위의 특정 콘텐츠를 접속하는 일부 정상 유저도 차단되며 공격자가 URL을 변경하면 대응에 어려움이 있다는 한계가 있다.

셋째, '자유게시판'에서 특정 검색어로 검색(search)을 요청하는 형태로 자주는 아니더라도 가끔 발견되는 공격의 형태라고 할 수 있다. 단순 게시물 하나를 보는 것에 비해 모든 DB의 내용을 검색하는 것이므로 게시물이 많고, 인덱스가 잘 되어 있지 않은 사이트라면 서버에 많은 부담을 줄 수 있다. 반대로 이는 공격자에게 효율적인 공격방식이 될 수 있다. 하지만 이 방법 역시 일부 오탐이 발생할 수 있으며 방화벽이나 L7 보안장비 등에서 특정 문자열로 차단할 수 있다.

2.4 HTTP GET Flooding 공격 형태

현재, 봇넷의 기술은 마스터가 각 좀비들에게 아래의 조건에 따라 일괄적으로 동시에 접속 요청하도록 명령을 내릴 수 있다.

(1) 특정 URL에 대해 (예:http://www.example.com/)
(2) 특정 비율(속도)로 (예:초당 10회 또는 최대로)
(3) 시간대(예:종료 명령전까지 계속)

GET Flooding의 공격 징후를 보면 다음과 같다.

No.	Time	Source	Destination	Protocol	Info
35	0.228426	192.168.253.135	0.21	HTTP	GET / HTTP/1.1
36	0.228437	192.168.253.135	0.21	HTTP	GET / HTTP/1.1
39	0.228674	192.168.253.135	0.21	HTTP	GET / HTTP/1.1
40	0.228683	192.168.253.135	0.21	HTTP	GET / HTTP/1.1
43	0.229011	192.168.253.135	0.21	HTTP	GET / HTTP/1.1
44	0.229020	192.168.253.135	0.21	HTTP	GET / HTTP/1.1
47	0.229174	192.168.253.135	0.21	HTTP	GET / HTTP/1.1

<그림 2> HTTP Get Flooding 사례

첫째, 공격을 받을 때의 로그를 분석해 보면 짧은 시간에 동시에 같은 URL이 반복적으로 찍히게 된다. 따라서 로그만으로 좀비PC와 공격 대상이 되는 URL을 쉽게 인지하는 것이 가능하다. 그러나 워낙 짧은 시간에 많은 접속요구가 있거나 평상시에 많은 접속이 있는 사이트라면 워낙 많은 로그가 남게 되므로 육안으로 공격과 정상 접속을 구분하는 것은 현실적으로 불가능할 것이다.

두 번째는 특정 IP에서 요청한 정보(URL)에 대한 내용이다. 특정 기간에는 모든 좀비PC가 동일한 URL을 요청시도 하지만 얼마 후 공격자는 URL을 변경하여 재시도를 할 수 있다. 하지만 어느 정도의 기간 동안에는 동일한 URL을 반복적으로 요청하는 특징을 가지고 있는데 특정한 1개의 IP에 대하여 정상 접속과 공격인 경우이다.

```
/common/css/sub.css
/common/js/common.js
/common/css/common.css
/images/common/main_service.gif
/customer/success_story_view.php?biid=733
/common/css/common.css
/images/common/main_service.gif
/images/common/main_industry.gif
```

<그림 3> 정상적인 웹접속의 경우

먼저 로그에서 정상적인 웹 접속의 경우 URI 정보만 뽑은 것이다. 처음에 첫 페이지(/)를 접속하면 index 파일이 클라이언트에게 전달되고, 클라이언트의 브라우저는 html 파일을 파싱(해석)하여 유저에게 보여주는데, 중간에 <img src= abc.gif>와 같이 이미지를 호출하는 링크가 있으면 웹서버에 다시 요청(Hit)을 하게 된다. 그리고 유저가 특정 메뉴나 링크 등 /customer/success_story_view.php?biid=733를 클릭하면 파일의 내용을 파싱하여 페이지에 포함된 static 콘텐츠를 보여주게 된다.

다음은 GET Flooding을 수행하는 좀비PC의 경우이다.

```
/bbs/list/bbs.php
/bbs/list/bbs.php
/sitemap/site.php
/sitemap/site.php
/sitemap/site.php
/notice/notice.php?id=475
/notice/notice.php?id=475
/notice/notice.php?id=475
```

<그림 4> 비정상적인 웹접속의 경우

요청하는 URI는 변경될 수 있지만 중요하게 살펴보아야 할 특징은 바로 'static 콘텐츠에 대한 접속은 하나도 없이 오직 dynamic 콘텐츠만 요청

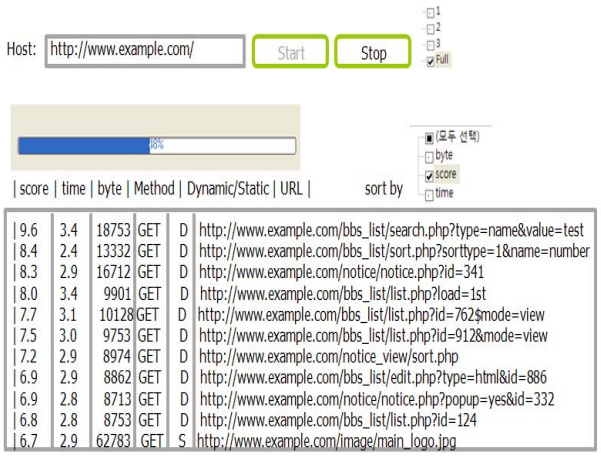
한다'는 것이다. 앞에서 언급한 바와 같이, 일반적으로 정상적인 브라우저의 접속이라면 GET 요청 후 받은 파일에 대한 해석 후 링크되어 있는 이미지 파일에 대한 재요청이 있어야 하지만 로그 파일에는 그러한 것이 없이 지속적으로 dynamic 콘텐츠만 요청하고 있는 것을 확인할 수 있으며 이를 통해 공격임을 알 수 있다.

3. APT(지능형 지속 공격) 도구 설계 및 구현

본 장에서는 지금까지 살펴본 내용을 기반으로 공격자의 입장에서 진보된 공격방법에 대해 설계하고 구현한다.

3.1 APT(지능형 지속 공격) 도구 설계

앞에서 살펴본 바와 같이 서버에 가장 큰 부담을 주는 요청은 응답(response)패킷의 사이즈가 크면서 DB와 통신하는 dynamic 콘텐츠가 된다. . 따라서 공격 대상이 되는 사이트에서 이 요구조건에 맞는 URL을 찾는 것이 필요하다. 이를 위해 웹취약성 스캐너와 같이 대상이 되는 홈페이지 주소만 입력하면 사이트의 모든 링크를 스캔, 응답시간과 패킷 응답 사이즈를 조합하여 가장 최적의 URL을 순서대로 나열하여 보여줄 수 있다.

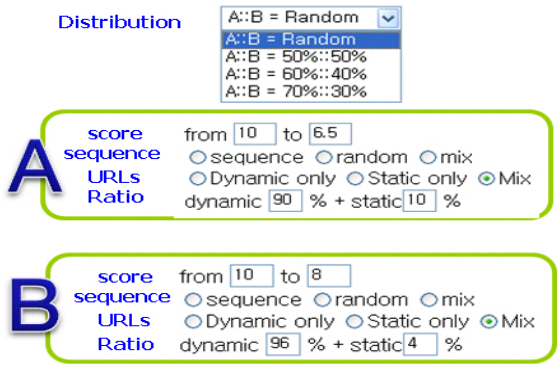


<그림 5> 공격솔루션의 스캔 설정 화면

스캔 화면은 다른 스캐너와 비슷하게 도메인만 입력한 후 Start를 누르면 스캔을 시작하게 된다. 여기에서 스캔 옵션은 모든 링크를 따라 들어가 스캔할 것인지 아니면 1,2,3 depth까지만 할 것인지를 지정할 수 있다. 사이트의 규모에 따라 Full로 스캔하면 좀 더 공격에 효과적인 URL을 얻을 수 있겠지만, 많은 시간이 소요된다는 단점이 있다. 스캔이 진행되면서 스캔된 URL들이 하단에 출력되는데, 이 URL들은 앞에서 살펴본 바에 따라 요청 후 응답까지 실행에 소요된 시간(time), 응답 패킷의 사이즈(byte)를 종합하여 점수(score)를 매기게 된다. 출력되는 순서는 이 두 가지를 조합한 계산식을 이용한 score로 할 것인지 아니면 사이즈나 시간에 따라 각각 할 것인지 선택할 수 있다. 스캔이 종료된 후 또는 중간에 중단한 후에는 공격을 하기 위한 사전 작업을 하게 되는데, 이때 여러 가지 가능한 옵션을 선택할 수 있다. .

3.2 APT(지능형 지속 공격) 도구 구현

아래는 공격 방법을 선택할 수 있는 몇 가지 옵션을 보여주고 있다. 첫 번째는, 모든 좀비PC들을 100% 동일한 룰로 적용하여 이용하는 것이 아니라 크게 두 가지로 나누어 각각 서로 다른 정책으로 공격하도록 할 수 있는데, 이 2개의 그룹은 5대5, 4:6, 3:7 또는 랜덤하게 가중치를 분류할 수도 있다. 예를 들면, 1,000개의 좀비PC가 있다면 이 중 400개는 A그룹의 정책을 선택하고, 600개는 B그룹의 정책을 선택할 수 있도록 한다. 물론 그룹은 좀 더 세밀하게 3개 또는 4개 등으로 나눌 수도 있고, 이 비율도 한번 설정하면 고정되는 것이 아니라 수시로 변화되도록 설정할 수도 있다.



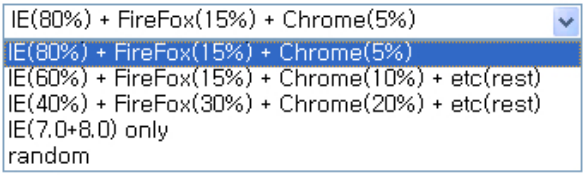
<그림 6> 공격솔루션의 옵션 설정 화면

!

1) HTTP Header의 User-Agent 구현

공격을 탐지하고 대응하는 입장에서는 아주 작

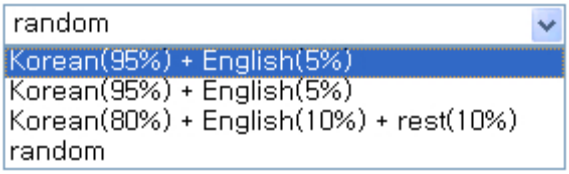
은 단서만으로도 큰 도움이 될 수 있다. 그 중 하나가 바로 User-Agent인데, 만약 특정한 User-Agent만 집중적으로 보이게 된다면 공격이 쉽게 노출될 수 있다. 그래서 최근의 공격들도 분석을 해 보면 대부분 일정 비율로 나누어지거나 random 하게 보이고 있는데, 가급적 모든 공격에는 규칙을 만들지 않는 것이 좀 더 좋은 효과를 낼 수 있으므로 각 그룹에 대해 random하게 설정하는 것이 좋다. 물론, IE에서도 버전에 따라 또는 툴바를 설치하면 세부 문자열이 변경되는데, 이러한 정보들은 인터넷에 연결되어 있다면 자동으로 업데이트 된다.



<그림 7> User-Agent 옵션

2) HTTP Header의 Accept-Language 구현

이는 브라우저의 언어를 알 수 있는 부분이다. 한국어라도 브라우저마다 약간 다른 값이 보이므로 앞에서 선택한 브라우저에 따라 자동으로 변환되어 보인다. 공격 대상이 한국이라면 주로 한국어 버전의 브라우저가 사용된다.

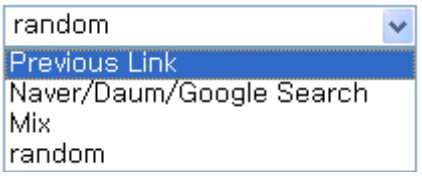


<그림 8> Accept-Language 옵션

3) HTTP Header의 Referer 구현

브라우저의 주소창에 직접 입력하지 않는 한 링크를 클릭해서 사이트에 접속하게 되면 헤더에 Referer이 남게 된다. 따라서 “Previous Link”를 선택하면 스캐닝시에 저장하였던 실제 Referer 정보를 이용하여 보여지게 되며 “Naver/Daum/Google Search”를 선택하면 검색엔진에서 링크된 사이트를 클릭한 것처럼 보이게 된다. 이를테면 daum 검색에서 링크를 클릭한 것처럼 보여지려면 Referer이 아래와 같이 보이게 된다.

Referer:http://search.daum.net/search?w=tot&t_
_nil_searchbox=btn&q=xxxxx.



<그림 9> Referer 옵션

검색엔진을 통해 들어오는 경우도 일부 있기는 하지만 Previous Link가 실제 접속과 동일하므로 이것을 사용하면 Referer 분석을 통한 공격차단은 거의 불가능하다.

4) HTTP Header의 Cookie 구현

웹방화벽이나 DDoS 보안 장비 등에서 일반 브라우저와 공격용 봇을 구분하는 기능 중 하나는 바로 Cookie를 활용하는 것이다. 즉, 3 way handshake 후 클라이언트의 GET 요청에 대한 응답 패킷에 보안장비에서 임의의 값으로 Set-Cookie를 설정하여 보낸 이후 클라이언트가 이 Cookie값을 헤더에 입력해서 접속하면 브라우저이고, 그렇지 않으면 봇 등 자동화된 툴이라고 판단하여 차단 한다. 대부분의 봇들이 Cookie에 대해 잘 응답하지 않기 때문에 보안장비 입장에서 봇을 구분하고 차단하는데 매우 유용한 기능이라고 할 수 있다. 이 옵션은 Set-Cookie에 대해 정상적으로 응답할 것인지에 대한 옵션으로 특별히 성능의 문제가 없다면 Yes로 설정하는 것이 좋다.

<그림 10> Cookie 옵션

이때 서버의 응답 패킷에서는 아래 그림과 같이 Set-Cookie: 로 값을 설정하여 클라이언트에게 보내게 된다.

```
HTTP/1.1 200 OK
Date: Sun, 21 Nov 2010 14:13:18 GMT
Server: Apache
Set-Cookie:
SECURITY=Advanced+GET+Flooding+DDoS+Attacks
Set-Cookie:
SECURITY=Advanced+GET+Flooding+DDoS+Attacks;
expires=Sun, 21-Nov-2010 14:23:18 GMT
Set-Cookie:
SECURITY=Advanced+GET+Flooding+DDoS+Attacks;
expires=Sun, 21-Nov-2010 14:23:18 GMT; path=/;
domain=example.com
Content-Length: 2
Content-Type: text/html
```

<그림 11> Cookie 옵션 설정 공격 사례

4. APT 공격 영향도 평가

4.1 콘텐츠 분포에 따른 공격 영향

본 장에서는 제안한 공격기술을 이용한 공격이 홈페이지 마다 콘텐츠 분포에 따라 공격영향의 변화가 있을 것으로 예상된다. 일반적인 홈페이지의 dynamic 콘텐츠와 static 콘텐츠로 분류하는데 그 상세 정보는 아래 <표 2>와 같다. 물론 이는 사이트의 구성과 목적, 개발 방법에 따라 달라지겠지만 일반적인 사이트의 특징으로서 어느 정도의 객관성을 위해 해당 사이트의 메인 페이지에 대해 1초 동안 접속 요청한 정보만을 분석하였다.

4.2 Dynamic 콘텐츠 공격성능 영향도 분석

Dynamic 콘텐츠 웹 서버의 성능에 어느 정도의 영향을 미치게 될 지에 구체적으로 분석하였다. 다음은 응답의 종류에 따라 초당 최대 처리 가능한 응답횟수에 대한 자료로, 숫자가 크면 클수록 단위 시간에 더 많은 응답을 할 수 있다는 의미이므로 이는 서버 입장에서 부담 없이 처리 가능하다는 것을 의미하며 반대로 이 숫자가 작을수록 서버 입장에서는 더욱 부담을 준다는 의미가 된다. 아래 <표 3>을 해석해 보면 다음과 같은 결과를 확인 하였다.

1) static 파일의 경우 응답 패킷의 사이즈가 크면 클수록 서버에 부담을 준다.
2) DB통신을 하지 않는 dynamic 콘텐츠의 경우 응답 패킷의 사이즈가 크면 클수록 서버에 부담을 준다.
3) DB통신을 하는 dynamic 콘텐츠의 경우 응답 패킷의 사이즈가 크면 클수록 서버에 부담을 준다.
4) DB통신을 하지 않는 dynamic 콘텐츠보다 DB통신을 하는 dynamic 콘텐츠가 서버에 부담을 준다.

<표 2> Dynamic 콘텐츠의 분포

사이트	Static 콘텐츠양	Dynamic 콘텐츠양	비율(dynamic/static)	1 connection당 요청수
A 포탈	106	6	5%	12회
B 신문사	111	2	1.8%	7회
C 쇼핑몰	60	6	10%	3.5회
D 은행	100	1	1%	12.5회
E 커뮤니티	65	3	5%	1.5회
F 이러닝	107	4	4%	7.8회

<표 3> 응답 종류에 따른 성능 비교

Type of Response	Peak Rate (Responses/Second)	Rate (based on 2kB static File)
2kB Static File	4,000	1
64kB Static File	1,400	2.8배
2kB Dynamic (PHP) File without Database Access	1,400	2.8배
64kB Dynamic (PHP) File without Database Access	250	16배
2kB Dynamic (PHP) File with Database Access	850	4.7배
64kB Dynamic (PHP) File with Database Access	250	16배

위의 내용을 종합해 보면 “DB통신을 하는 dynamic 콘텐츠이되 응답 패킷의 사이즈가 크면 클수록 서버에서 부담을 느낀다”는 것을 알 수 있다. 위의 표는 단 한 번의 요청에 대한 응답 패킷의 내용을 분석한 것이라 생각보다 큰 차이가 없는 것처럼 보일 수 있으나 실제 공격이 발생할 경우에는 초당 수천~수만 회 이상 꾸준히 요청이 들어오게 되므로 간단히 이론적으로 계산을 해 보아도 실제로는 매우 큰 차이가 나는 것을 알 수 있다. GET Flooding공격이 초당 1,000회, 1분(60초)동안 집중된다면 2KB의 static 콘텐츠 보다 64KB의 Dynamic 콘텐츠는 16배 x 1,000회 x 60초 = 960,000 가 된다. 즉, 극단적으로 응답 패킷 사이즈가 큰 Dynamic 콘텐츠는 작은 사이즈의 static 콘텐츠에 비해 무려 96만배의 부하 차이가 난다.

5. 결 론

본 논문에서 제안한 공격기술을 이용한 APT 공격이 실제로 개발되어 악용된다면, 현존하는 보안솔루션으로는 탐지 및 차단이 거의 불가능할 것으로 보인다. 그 이유는 공격을 탐지하고 차단을 하려면 공격 트래픽만을 검출해야 하는데, 트래픽이 다양하게 변화하여 공격하기 때문에 이를 차단하기 위한 특정한 규칙을 찾을 수 없다. 또한 기존의 일반적인 DDoS공격 특징을 대부분 우회하는 등 정상 접속과의 구별이 거의 불가능하여 현재 구축되어 있는 보안장비로 대응하는데 한계가 있고 부하를 유발하는 URL만을 수집, 집중적으로 공격하기 때문에 같은 양의 공격을 해도 그 효과가 매우 크다고 할 수 있다. 따라서 지난 7.7과 같이 수만~수십만의 대규모 좀비가 특정 사이트에 집중적으로 공격한다고 가정하면 정부 및 금융기관뿐만 아니라 민간 포털과 언론 등 대규모 사이트공격에 안전하지 않을 것이다.

그러나 이에 대한 대응방안이 전혀 없는 것은 아니다. 본 논문에서 ‘보호색’ 이라는 용어를 사용했듯이 보호색은 보호색 일 뿐 주변과 완전히 동일할 수는 없기 때문이다. 향후 연구과제로 현재의 보안 모델이 “특정 임계치를 넘거나 특정 패턴이 있으면 차단” 인 negative 모델이었다면 반대로 홈페이지에 대한 자동학습을 통해 정상적인 접속 패턴을 DB화하여 positive 모델이 아닌 접속은 차단하도록 하여 이를 구체화하고 오탐이 없는 솔루션이 필요가 있다.

참 고 문 헌

[1] Command Five Pty Ltd. (2011). Advanced Persistent Threat:A Decade in Review.

[2] 강형 등 (2009). 사이버테러에 따른 경제적 피해규모 산정을 위한 모델 연구, 한국사이버테러정보전학회. 9월호.

[3] Debasish Das. et al. (2011). Detection of HTTP flooding attacks in multiple scenarios, **Proceeding ICCC ’11**.

[4] R. Fielding et al(1999). RFC2616 - Hypertext Transfer Protocol. http://faqs.org/rfcs/rfc261.html.

[5] 안철수연구소 (2011). 지능적 타깃 공격 APT 대책 발표. http://blog.ahnlab.com/ahnlab/1252.

[6] 유승엽 등 (2011). 패턴의 특성을 이용한 HTTP get flooding 공격 탐지 알고리즘. 논문지. 한국정보기술학회.

[7] Advanced Persistent Threat(2010), http://en.wikipedia.org/wiki/Advanced_Persistent _Threat

[8] 정윤정 (2011). [핫테크] 지능적 지속 위협(APT). http://www.etnews.com/201108170108. 전자신문.

천 우 봉

2009 성균관대학교
전자전기컴퓨터공학과
(박사과정)
관심분야: 보안관제, 보안정책,
침해사고대응
E-Mail: cwb3242@naver.com

박 원 형

2009 경기대학교
정보보호학과(이학박사)
2010 서울과학기술대학교
산업정보시스템공학과(겸임교수)
관심분야: 보안관제, 융합보안, 윈도우포렌식
E-Mail: infosecure@seoultech.ac.kr

정 태 명

1984 미국 일리노이 주립대학교
전산학과(공학사)
1987 미국 일리노이 주립대학교
컴퓨터공학과(공학석사)
1995 미국 퍼듀대학교 컴퓨터공학과(공학박사)
현재 성균관대학교 정보통신공학부 교수
관심분야: 사이버보안, 소프트웨어공학
E-Mail: tmchung@ece.skku.ac.kr