제26회 한국정보처리학회 추계학술발표대회 논문집 제13권 제2호 (2006. 11)

DNS기반의 봇넷 탐지 시스템

이한우*, 최현상, 이희조
*고려대학교 컴퓨터 정보통신대학원 대학원 정보통신학과
고려대학교 컴퓨터학과
e-mail:hanwoo, hyunsang, heejo@korea.ac.kr

DNS-based Botnet Detection and Monitoring System

Hanwoo Lee*, Hyunsang Choi, Heejo Lee
Dept of Computer Science, Korea University

요 약

인터넷의 비약적 성장과 더불어 인터넷에서의 악성행위 기술도 함께 빠르게 고도화 되고 있으며 이에 따른 피해의 규모 또한 점점 커지고 있다. 종래의 대표적 악성코드인 바이러스, 웜과는 달리 근래에는 해킹을 이용해 경제적 이득을 취하기 위해 전문적이고 조직적인 네트워크를 형성하고 이를 통해 악성행위를 수행하고 있다. 최근 봇넷이라고 하는 네트워크가 이러한 악성행위를 수행하는데 주로 이용되고 있다. 즉, 봇넷이란 악성 프로그램인 봇에 감염된 다수의 컴퓨터들이 네트워크로 연결되어져 있는 형태를 말하며, 이렇게 구축된 거대한 봇넷을 이용하여 개인정보의 탈취, DDoS공격과 피싱 및 스팸 메일의 발송 등과 같은 여러 악성행위를 수행한다. 이처럼 최근 사이버 보안의 최대의 위협 요소로 대두되고 있는 봇넷에 의한 피해를 최소화하기 위해서 그 대응책이 절실히 필요하다. 이에 본 연구에서는 봇넷의 C&C(Command & Control)과정에서 발생되는 DNS 쿼리를 분석하여 정상적인 DNS 쿼리와 구분되는 봇넷 DNS 쿼리 특성을 이용한 봇넷 탐지 시스템을 제안한다.

Keywords: 봇넷, Bot, Zombie, DNS, Enterprise Network

1. 서론

인터넷의 비약적 성장과 더불어 인터넷을 구성하는 많은 기반요소의 취약점이 노출되고 이를 악용하여 불특정 다수를 공격하는 사이버 테러가 빈번히 발생하고 있다. 사이버 공격은 갈수록 지능화 조직화 되고 있으며 해킹을 이용해 경제적 이득을 취득하기 위해 전문적이고 조직적인 네트워크를 형성하고 있다. 이처럼 근래 사이버 공격자들은 금전적 이득을 얻고 불법행위를 대행하기 위해서 자신들의 존재와 위치 등의 추적이 쉽지 않도록 하고, 자신들의 통제 하에 자유자재로 움직일 수 있는 수많은 컴퓨터를 제어하는 되는 기술을 고안하게 되었으며 이러한 목적으로 봇넷이 등장하게 되었다. 공격자는 수천에서 수만대의 컴퓨터에 봇을 설치하고 이들을 네트워크를 통해 동시에 제어를 함으로써 악성행위를 수행한다. 악성 봇에 감염된 PC가 늘어남에 따라 봇넷은 규모가 커지고 있으며 이러한 봇을 이용해 다양한 종류의 악성공격과 그에 따른 피해 또한 점점 늘어가고 있다. 게다가 봇넷의 공격에 의해 개인정보 유출 뿐 만아니라 인터넷 기반구조에 심각한 피해를 입히는 DDoS 공격을 수행하는 주요 원인으로 알려져 있는 등 여러 가지의 측면에서 그 심각성이 대두되고 있다. 최근 봇넷에 의한 피해규모의 확산을 막기 위해 봇을 탐지하는 여러 기술들이 제안이 되었으나 아직까지 대부분의 제안된 기술들이 그 한계점을 갖고 있다. 이에 본 연구에서는 봇넷의 C&C(Command & Control)과정에서 이용되는 DNS 쿼리를 분석하여 정상적인 DNS 쿼리와 구분되는 특성을 이용해 봇넷 탐지 시스템을 개발하였다.

2. 관련연구

Symantec[1]과 SANS[2] 같은 보안 관련기관에서는 실제 인터넷의 트래픽 중 봇에 관련된 트래픽을 분석하고, 또는 Honynet등을 운영하면서 이를 이용해 다양한 봇들의 특징 및 행위, 전파방법을 분석과 봇의 탐지 및 대응기법에 대한 연구가 진행되고 있다. 이외에도 Jim Jones는 봇넷의 구체적인 특징과 번식과 활동의 방법을 확인하고 실제 봇넷의 위험성을 완화하기 방안을 제안하였다

[3]. 그리고 E. Cooke은 IRC의 특성을 연구하여 봇마스터와 봇 사이의 C&C모델을 분석하여 이를 탐지하는 기법에 대한 제시를 하였고 Centralized, P2P, Random의 향후 나타날 수 있는 세 가지 봇넷의 토폴로지를 제시하고 그에 대한 대응책을 제안하였다[4].

봇 마스터의 통제 하에 있는 봇들은 봇 마스터와 C&C과정을 통해 여러 가지 공격을 수행한다. 이 과정에서 C&C에 사용되는 채널 서버의 도메인 네임을 파악하고 도메인 네임서버(DNS) 싱크홀을 적용, 봇 감염 agent가 C&C채널 서버로 연결되는 것을 차단 및 탐지하는 방법을 통해 봇넷의 규모의 파악과 봇에 감염된 PC의 탐지에 성과를 거두고 있다[5]. 그러나 채널서버에서 사용하는 도메인네임을 자동으로 탐지하는 기법에 대해서 연구된 경우가 거의 없으며 실제 이용되고 있는 방법 또한 쉽게 회피가 가능하다는 점 등 한계점을 지닌다.

3. 봇넷 (Botnet)

3.1. 봇넷의 정의

공격자가 원격지에서 조정통제 가능한 봇(Bot)에 감염된 컴퓨터들 간에 논리적인 네트워크를 봇넷이라 한다. 즉, 봇들을 통제하는 권한을 가진 봇 마스터(Bot master)에 의해 원격 조종되는 수천에서 수십만 대의 봇들이 네트워크로 연결되어 명령에 따라 악성행위들을 자동으로 수행한다. 봇 마스터와 봇넷 사이의 명령전달 및 제어를 하기 위하여 대부분의 경우 IRC(Internet Relay Chat) 채널을 이용한다. IRC의 여러 장점을 이용하여 봇 마스터는 자신만의 채널을 만들어 봇넷을 안전하고 편하게 제어할 수 있다. 또한 봇넷이 C&C를 위해 주고받는 트래픽은 정상적인 IRC 트래픽과 차이가 없고 SSH를 이용하는 경우 IDS나 IPS에서 그 탐지의 어려움이 있다.

3.2. 봇넷의 현황

여러 통계자료를 보면 세계적으로 2백만대 정도의 PC가 악성 Bot에 감염된 것으로 추청하고 있으며, 이중 국내에는 약 40만여대 정도가 Bot에 감염되어 있는 것으로 추정된다.

구분	2005년								2005 총계
	1	2	3	4	5	6	7	8	
봇(bot)	24%	26%	25%	24%	20%	18%	19%	19%	22%

표 1 (KISA) 세계 봇 PC중 국내 봇 감염 PC 비율

봇넷에서 C&C를 위하여 이용하는 포트는 TCP 6665 ~ 6669를 사용하며, 공격의 유형도 DDoS나 Syn flooding 공격 뿐 만아니라 피싱과 파밍 등의 공격으로 불특정 다수의 계좌번호나 카드번호등을 갈취하여 범죄적 목적으로 이용을 하기도 하며 불법 와레즈 사이트를 운영하며 스팸을 대행해주고 금전적 이득을 취하기도 한다.

3.3. 봇넷 탐지기술

현재 봇넷을 탐지 연구는 크게 클라이언트 기반의 기법과 네트워크 기반의 기법으로 나눌 수 있으며, 클라이언트 기반의 연구는 다시 대부분의 시그너쳐 기반의 탐지 기술과 이상행위 탐지 기반의 연구로 구분 할 수 있다.

대다수의 안티바이러스 솔루션에서 제안하는 봇의 탐지 기법이 대표적인 클라이언트 기반의 시그너쳐 탐지기법으로 볼 수 있다. 이 기법의 특성상 변종 봇이나 신종 봇에 대한 탐지가 힘들고 빠른 대응을 하기가 힘들 뿐만 아니라 패킹기법을 이용해 탐지를 힘들게 할 수 있다는 한계점들을 갖는다. 수많은 봇들이 그 소스코드가 공개되어있어 쉽게 변종의 생산이 가능하여 실제 사용되는 대 부분의 봇들은 패킹기법을 이용해서 탐지를 힘들게 한다.

클라이언트 기반의 이상탐지 기법은 봇의 행위의 특성이 인간의 정상적인 동작과 구분하기 힘들기 때문에 개발이 어려울 뿐만 아니라 현재 개발된 기법들이 한계점을 갖는다. 네트워크 기반의 시그너쳐 탐지 기법은 봇넷이 발생시키는 트래픽을 패턴매칭을 하여 탐지하는 기법으로 클라이언트와 같은 한계점을 갖는다. 최근 대부분의 봇넷 탐지 연구가 네트워크 기반의 이상행위 탐지 쪽에서 진행이 되고 있다. 네트워크 기반에서 봇넷의 제어서버인 C&C(Command and Control) 서버를 찾아내서 제거하는 방법이나 같은 맥락으로 DNS 싱크홀을 이용해 봇을 탐지하고 봇넷을 제거하는 방법 등의 연구가 있었다. DNS 서버에 전달되는 쿼리(Query log) 조사, Netflow를 이용한 통계 및 휴리스틱 분석, 봇넷 트래픽에 대한 패킷 캡쳐 및 분석, 악성프로그램 분석, 악의적인 봇 시그네처를 근간으로 한 침입탐지 시스템의 이용과 같은 방법이 존재한다[5]. 그리고 봇넷의 다양한 진화 및 변형으로 인해 특별한 패턴을 규정할 수 없고 위 방법들은 사고 발생 이후의 탐지 및 조치만 될 뿐, 예방적인 탐지 및 대응능력에는 여전히 한계를 가진다. 더불어 대부분의 C&C 서버가 Fast-Flux(fast-changing-IP-address)의 특징을 기지고 있어 그 탐지가 힘들다[5]. 이전의 봇들은 봇의 소스에 고정된 IP주소를 입력하여 감염코드를 만들고 이를 이용해 봇넷을 구성하였기 때문에 역어셈블 또는 봇의 트래픽을 분석하여 고정된 IP주소를 찾아내서 싱크홀 기법이나 기타 다른 방법을 이용해 효과적인 대응을 수행을 할 수 있었다. 그러나 이러한 대응기법을 회피하기 위하여, 봇 제작자들은 IP 대신 DNS를 이용하여 위의 탐지 기법을 회피하였다. 즉, 고정된 IP주소가 아니라 유동적인 도메인 네임과 자유롭게 변경이 가능한 IP등을 연결해주는 Dynamic DNS 기법을 이용해서 탐지를 힘들게 하였다. 이러한 변화에 따라 최근 봇넷의 C&C서버의 도메인네임을 찾아내는 기법에 대한 연구가 진행되고 있으나 알려진 기법들이 대부분 쉽게 회피가 가능하다는 한계점을 갖는다.

4. DNS 기반 봇넷 탐지 시스템

4.1. 봇넷 DNS의 특징

봇 마스터는 봇넷을 이용하기 위해서 IRC서버 채널를 이용해 C&C를 하여 봇넷을 이용한다. 봇에 감염된 PC는 스스로 이 IRC서버의 채널로 접속을 하게 되며 서버의 IP주

소를 미리 알고 있는 것이 아니라 도메인네임만을 기억하고 있다가 DNS 쿼리를 보내 채널서버의 IP주소를 응답으로 받아 이를 통해 접속을 한다. 처음 감염 후에 IRC서버에 접속할 때 이외에도 IRC서버의 IP주소가 변경된 경우와 특정 명령을 수행하는 경우 봇들이 IRC서버에 접속하기 위해서 스스로 DNS쿼리를 전송한다. 실제로 Dynamic DNS를 이용하는 IRC서버의 경우에 빈번하게 그 IP주소가 변경이 되며 DNS레코드의 TTL값이 작게 설정이 되어 있기 때문에 봇들에서 DNS쿼리가 자주 발생하게 된다. 이외에도 스팸메일을 보내거나 DNS를 이용해 DDoS공격을 수행하는 경우 봇이 자신을 업데이트 하기위해 특정 도메인에 접속하는 경우 등 많은 실제 악성행위를 수행하는 과정에서도 대량의 DNS 쿼리가 발생하게 된다.
이러한 봇넷에서 발생하는 DNS쿼리들을 보면 표2와 같은 고유의 특징을 지니고 있다는 것을 알 수 있다.

	Sources	Timing	(D)DNS
Legitimate DNS queries	- Anonymous	- Continuous - Regular	- Usually DNS
Botnet DNS queries	- Fixed Group (little variation)	- Temporary - Irregular	- Usually DDNS

표 2 봇넷 DNS와 정상 DNS의 비교

먼저 DNS 쿼리를 보내는 소스가 정상의 DNS 쿼리의 경우에는 불특정 유저이지만 봇넷의 쿼리의 경우에는 항상 정해진 일정한 사이즈의 그룹이라는 차이가 있다. 그리고 DNS 쿼리를 전송하는 시간의 경우에도 정상의 경우에는 지속적인데 반해 봇넷의 경우는 일시적으로 나타나고 규칙적이지 않다는 특징을 갖는다. 그리고 정상적인 DNS 쿼리의 경우 DDNS를 사용하는 경우가 상대적으로 적은데 반해 봇넷의 경우 최근에는 거의 대부분 DDNS를 사용하기 때문에 이부분에서 차이점을 갖는다.

4.2. 봇넷 탐지 알고리즘

앞에서 살펴본 봇넷의 C&C과정에서 발생하는 DNS쿼리들의 특징을 이용해서 정상적인 DNS쿼리들과 봇넷의 DNS 쿼리를 구분할 수 있는 봇넷 탐지 알고리즘을 개발하였다. 알고리즘은 DNS 쿼리를 일정 시간단위로 저장을 하여 각각 시간에 따라 준비된 자료구조에 입력과 삭제 과정을 하는 부분과 봇넷 DNS쿼리 탐지를 하는 부분으로 나누어져 있다. 그림 4에서 기본 입력과 삭제 알고리즘이 나타나 있다. 우선 특정 시간 t−1에서 t사이에 DNS쿼리를 모아서 준비된 Array A에 저장을 한다. 이때 이미 존재하는 도메인네임에 대해서는 그 도메인에 대한 IP 리스트 안에 그 쿼리를 보낸 IP주소가 존재하는지 아닌지를 확인한다. 이는 모든 쿼리를 도메인네임을 기준으로 해서 IP주소를 그룹 짓는 것으로 볼 수 있다. 삭제부분에서는 도메인네임이 화이트리스트에 있는지의 여부와 그룹 지어진 IP주소 리스트의 사이즈가 미리 정해진 임계치 T 값을 넘는가의 여부 두 가지를 각각 확인하여 하나라도 참인 경우 A에서 도메인네임과 그 IP리스트 모두를 삭제하게 된다.

Insert-Query (Q_t) $Q_t \leftarrow$ DNS Queries in t-1 and t
1 $A \leftarrow$ Array for Queries
2 FOR k = 1 to n
3 $DN_k \leftarrow$ Domain name
4 IF DN_k is not exist in A
5 insert(DN_k, A)
6 IP_k, $IPList \leftarrow$ IP, IP List
7 insert(IP_k, $DN_k \rightarrow IPList$)
8 ELSE IF IP_k is not exist in ($DN_k \rightarrow IPList$)
9 $cnt \leftarrow$ size of ($DN_k \rightarrow IPList$)
10 ($DN_k \rightarrow cnt$)++
11 insert(IP_k, $DN_k \rightarrow IPList$)
12 ENDIF
13 ENDFOR
End of Insert-Query

delete-Query (Q_t)
1 FOR k = 1 to n
2 W, $T \leftarrow$ Whitelist, Threshold
3 IF (DN_k is not in W) AND ($DN_k \rightarrow cnt > T$)
4 blacklisting(DN_k, BL) $BL \leftarrow$ Blacklist
5 ENDIF
6 ENDFOR
End of delete-Query

그림 1 DNS 쿼리의 입력과 삭제 알고리즘

이와 같은 과정을 필요한 이유는 DNS쿼리의 대부분을 차지하고 있는 널리 알려진 도메인에 대한 쿼리와 아주 적은 수의 IP그룹을 갖는 도메인 쿼리의 불필요한 처리과정을 없애기 위해서 이다. 앞의 두 과정을 거친 후 실제 봇넷의 DNS 쿼리를 탐지하는 과정을 수행하게 된다.
앞의 DNS쿼리의 입력과 삭제 부분에서 임계값을 넘는 그룹을 갖는 도메인네임에 대해서 블랙리스트에 추가를 하는 부분이 있는데 이 블랙리스트를 기준은 이미 존재하는 도메인네임인 경우 봇 DNS 탐지 알고리즘을 수행한다.

Detect-Bot-Query (Q_t)
1 IF ($DN_t \rightarrow cnt$) > T AND DN_t already exist in BL (DN_{t2})
2 compare($DN_t \rightarrow IPList$, $DN_{t2} \rightarrow IPList$)
3 N $\leftarrow$ number of same IPs between DN_t and DN_{t2}
4 IF N/($DN_t \rightarrow cnt$) > α, $\alpha \leftarrow$ Threshold
5 DN_t is Bot domain name
6 ENDIF
End of Detect-Bot-Query

그림 2 봇 DNS 탐지 알고리즘

서로 다른 시간대에 블랙리스트에 오른 같은 도메인네임을 갖는 도메인에 대해 같은 도메인네임을 갖는 그 도메인네임이 각각 갖고 있는 IP그룹(IP List)을 서로 비교하게 된다. 비교 과정을 통해서 얼마나 많은 수가 일치하는가(Correlation)를 계산을 하여 이 Correlation값이 임계값

α를 넘는가를 확인하는 과정을 통해 봇의 DNS 탐지를 수행한다.

4.3. 시스템 구조

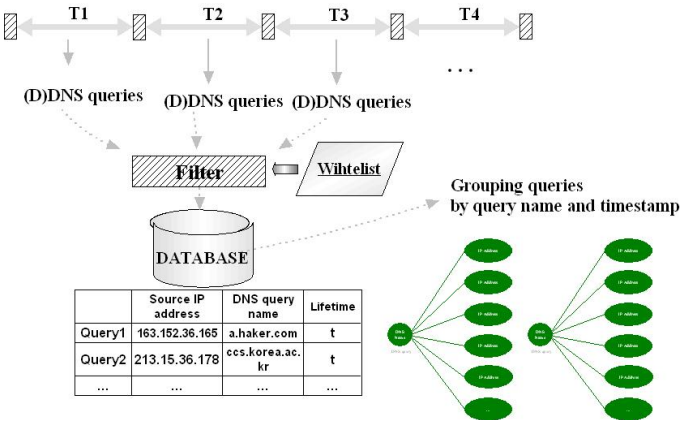

그림 3 봇넷 DNS 탐지 시스템 구조

본 시스템은 오프라인의 DNS 트래픽 데이터를 분석하는 시스템으로 이용을 하는 것 뿐 만아니라 실시간으로 네트워크의 DNS트래픽을 모니터링 함으로써 봇넷의 존재 여부와 그 감염 PC들을 탐지하는 시스템으로 동작할 수 있다. 시스템의 위치는 이상적인 상황을 가정하면 Root DNS서버에서 보내주는 DNS트래픽 데이터를 이용해서 오프라인/온라인으로 동작하는 경우이고 DNS트래픽을 제공하는 DNS서버의 규모가 작으면 작을수록 임계값들의 조정이 필요하고 트래픽 수집의 타이밍t의 조정이 필요하다.

4.4. 탐지 결과

시스템이 봇넷의 탐지를 수행하는 가를 검증하기 위해서 우리는 실제 허니넷(honeynet)망에 봇넷을 설치하고 C&C과정을 통해 여러 악성행위를 수행하는 과정을 거쳐서 네트워크 트래픽을 수집하여 테스트를 하였다. 검증에 이용된 봇들은 Agobot, SDBot, Rbot, Evilbot 등이며 이 중 대표적인 봇인 Agobot에 대한 테스트 결과가 표3과 같다. 허니넷 망은 1Gbps 규모이고 봇넷으로 이용된 감염 PC는 총 50대, 테스트 수행 시간은 약 10시간 이었다.

여기서 쿼리 데이터의 입력과 삭제 및 탐지 알고리즘에서 사용한 시간 간격은 1시간 단위 이다. 실제 테스트를 한 결과 대부분의 쿼리들은 IP리스트 그룹크기에 대한 임계값 20을 넘지 못했다. 네트워크의 규모와 시간간격의 설정에 따른 차이는 있겠지만 대부분의 정상적인 상황에서는 비슷한 시간에 서로 다른 PC에서 같은 사이트에 접속을 하는 경우가 많지 않다는 것을 알 수 있다. 테스트에서 그룹크기 임계값 20을 넘은 경우도 대부분 다음 그림 4에서 보이는 것과 같은 세 가지 패턴을 보이고 있었다. 그림 4는 각 시간에 따라 한 도메인에 대한 쿼리의 IP리스트 사이즈에 대해 이전 시간에 같은 도메인에 대한 쿼리의 IP리스트가 얼마나 일치하는 가를 나타낸 것이다. 즉, Correlation = (Duplicated IP number of T and T−1) / (IPList size of T)

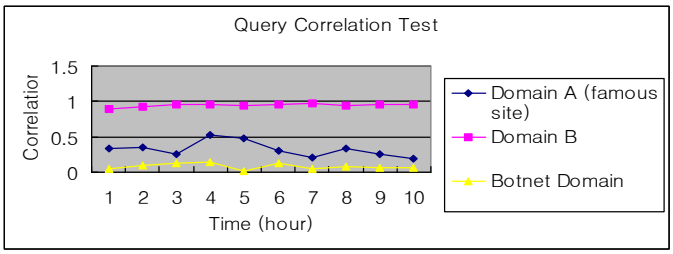

그림 4 DNS 쿼리 Correlation 테스트

첫 번째 Domain A의 경우 널리 알려진 사이트에 대해 Correlation의 변화를 살펴보았는데 대부분 0.3~0.5의 값 근처를 갖고 있었다. 즉, 어떤 시간에 도메인 A에 접속한 사람이 30~50%의 확률로 다음 시간 간격사이에 다시 접속을 하는 것으로 생각할 수 있다. 널리 알려진 사이트가 아닌 Domain B의 경우 Correlation값이 훨씬 낮은 것으로 측정 되었다. 이에 반해 봇넷의 도메인의 경우는 거의 항상 1에 가까운 Correlation 값을 갖게 되는데 실험에서 임계값 α = 0.8 로 설정한 경우 항상 탐지가 되었다. 실험에서 flash crowd와 같은 몇 가지 오탐이 발생한 경우가 있었으나 시간간격의 설정값과 임계값의 조정 화이트리스트를 이용하는 방법들을 이용하면 오탐을 낮출 수 있다. 더 큰 규모의 DNS 서버의 쿼리를 이용한 시험을 예정 중에 있으며 오탐을 줄이는 방안을 또한 연구 중이다. 그리고 실제 망에서 얼마나 봇을 정확하게 탐지하는 가에 대한 탐지율 측정 실험도 예정 중에 있다.

5. 결론

공격자가 원격지에서 조정통제 가능한 봇에 감염된 컴퓨터들간에 논리적인 네트워크를 봇넷이라한다. 봇넷에 의한 피해를 줄이기 위해 봇넷을 탐지하는 기술의 개발이 필요하다. 지금까지 봇넷의 C&C서버의 도메인네임을 찾아내는 기법에 대한 연구를 비롯해 봇넷의 탐지를 위한 다양한 연구가 진행되고 있으나 알려진 기법들이 대부분 쉽게 회피가 가능하다는 점 등의 한계점을 갖고 있었다.

본 연구에서는 DNS를 이용한 봇넷을 탐지하는 알고리즘을 제안하였으며 실제로 봇넷의 탐지가 가능하다는 것을 실험을 통해 확인 하였다.

참고문헌

[1] Symantec Inc, http://www.symantecom/.

[2] SNAS institure, http://www.sans.org/.

[3] Jim Jones, "Botnet : Detecting and Mitigation February", FedCIRC Operations, 2003

[4] E. Cooke, F. Jahanian, D. McPherson, "The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets", USENIX SRUTI 2005

[5] 국가사이버안전센터, “DNS를 이용한 봇넷(Bot-Net) 탐지”, 사이버시큐리티, June 2006