情 報 保 護 學 會 誌
第18卷 第4號, 2008. 8

봇넷 분류법 및 진화된 봇넷 구조

전 용 희*, 오 진 태**

요 약

인터넷이 직면하고 있는 최대 위협중의 하나는 봇넷이라는 수많은 감염되거나 침해된 좀비 머신의 존재이다. 최근 이러한 봇넷이 인터넷 공격의 근본 원인이 되고 있다. 그동안 봇넷은 IRC(Internet Relay Chat) 기반이 주류를 이루어 왔으나, 중앙 집중 구조로 인하여 쉽게 차단되는 특성이 있기 때문에, 앞으로는 HTTP 봇넷, P2P 봇넷과 같은 더욱 더 탄력성 있는 구조와 여러 가지 회피 기법을 가진 진화된 구조를 가진 봇넷의 출현이 전망된다. 따라서 본 논문에서는 봇넷에 대한 보다 나은 이해를 위하여 봇넷을 분류하기 위한 분류법(taxonomy)을 소개하고, 가까운 미래에 봇마스터들에 의하여 개발될 수 있는 진화된(advanced) 봇넷 구조로 계층구조와 혼합구조에 대하여 분석 기술하고자 한다.

Ⅰ. 서 론

봇이란 용어는 로봇(robot)으로부터 유도되었으며, 미리 정해진 기능들을 수행하기 위하여 자동화된 방법으로 설계된 스크립트(script) 혹은 스크립트 집합을 나타내기 위하여 사용되는 일반적인 용어이다. 이런 봇들의 네트워크가 봇넷(Botnet)이며, 감염된 혹은 침해된 기계들의 네트워크이다.

이런 봇넷 혹은 좀비(zombie) 군대라고 하는 침해된 머신의 대규모 네트워크를 이용한 공격이 최근 인터넷에 대한 가장 큰 위협 중의 하나가 되고 있다. 봇마스터(Botmaster)라고 하는 단일 혹은 작은 그룹의 공격자의 제어 하에, 봇넷은 인터넷 자원을 대상으로 한 분산 서비스 거부(DDoS : Distributed Denial of Service) 공격, 전자메일 스팸 공격, 감염 호스트로부터 개인정보와 같은 유익한 정보를 수집하기 위하여 키 로거(key logger)나 트로이 목마의 내려 받기나 설치, 백도어 설치, 불법 데이터 공유, 새로운 멀웨어를 확산하기 위하여 사용된다[18].

봇넷의 출현 이후 새로운 정교한 소프트웨어 모듈들이 기존의 봇넷에 첨가되어, 컴퓨터를 침해하기 위한 다양한 방법을 제공하고 잠재적으로 훨씬 더 해로운 공격을 제공한다[10]. 기존 봇의 주요 특징은 다음과 같다[4] : ICMP, SYN, UDP 및 HTTP를 이용한 플러딩 공격; SOCKS4, HTTP 및 HTTPS 프록시 기능; TCP 포트 Redirect; 패스워드 스니핑; 쿠키 수집; 스팸 메일 전송; 포트 스캔; 키로거.

[18]에서는 IRC-기반 봇넷을 기반으로 하는 봇넷 기술 개요에 대하여 알아보고 분석하였다. 그러나 IRC-기반 봇넷은 어느 한 지점에서 중앙 명령 및 제어 위치를 가진 중앙 집중구조를 가지기 때문에 쉽게 차단이 될 수 있다[19]. 국내에서도 악성 IRCBot 웜이 감소되는 추세이다[17]. 그러므로 공격자들은 더욱 더 탄력성 있는 구조로 이동하는 경향이다. 이런 종류의 봇넷 구조로는 HTTP 봇넷, P2P 봇넷이 있다[5]. 본 논문에서는 이런 좀 더 진화되고 정교화 된 봇넷 위협 기술에 대하여 분류하고 분석하고자 한다. 이를 위하여 먼저 봇넷 행위와 통상적인 특성을 분석하기 위하여 봇넷을 분류하기 위한 모델에 대하여 기술하고자 한다[10]. 다음으로 진화된 봇넷 구조로 계층 구조와 혼합 구조에 대하여 알아본다.

Ⅱ. 봇넷 분류 기준

현재까지 봇넷의 행위와 특성에 대한 좀 더 나은 이해를 위하여 많은 연구가 수행되어 왔으나, 아직도 봇넷의 구조와 메커니즘에 대한 것은 더 규명될 필요가 있다. [10]에서는 기존의 유용한 분류 기법과 새로운 제안을 결합하여 아래와 같은 봇넷 분류 기준을 제시한다 :

• 공격 행위 : DDoS와 scan과 같은 공격을 위하여

* 대구가톨릭대학교 컴퓨터정보통신공학부 (yhjeon@cu.ac.kr)
** 한국전자통신연구원 정보보호연구본부 보안게이트웨이연구팀 (showme@etri.re.kr)

사용되는 방법

- 명령 및 제어(Command & Control : C&C) 모델 : 집중형, 분산형 및 P2P 등과 같은 명령 및 제어를 위한 구조
- 통신 프로토콜 : IRC, HTTP, P2P[16] 등과 같은 네트워크상의 통신을 위하여 사용하는 프로토콜
- 관측 가능한 봇넷 행위 : 비정상 시스템 호출과 한꺼번에 많은 짧은 패킷 등과 같은 봇넷에 의하여 나타나는 비정상 행위
- 집결(rally) 메커니즘 : hard-coded IP와 동적 DNS, 분산 DNS 등과 같은 새로운 봇을 발견하고 봇마스터 아래 집결시키기 위한 메커니즘
- 회피 기법 : HTTP/VoIP 터널링, IPv6 터널링, P2P 암호 트래픽 등과 같은 탐지를 회피하기 위하여 사용되는 기법

Ⅲ. 분류법[10]

3.1 공격행위

공격자는 공격을 위한 수단으로 여러 가지의 도구들을 사용한다. 이러한 도구들로는 원격 익스플로잇, 피싱과 바이러스 부착물과 같은 정크 메일, 웹사이트 피싱, 스파이웨어, ID 절도 등과 같은 것이 있다. 공격과정에서 봇넷은 통상 대량의 비정상 트래픽을 생성하며, 이것으로 탐지를 용이하게 할 수 있다.

다음과 같은 네 가지 관점에서 공격 행위를 논의한다 :

- 새로운 호스트 감염 : 봇넷은 바이러스와 웜 같은 다른 멀웨어와 유사한 방법을 사용하여 새로운 호스트를 감염시킨다. 흔히 사용되는 방법으로 사회공학 및 악성 메일 분배가 있다. 봇은 원격 익스플로잇을 이용하여 알려진 취약성을 가진 호스트들에 대한 탐색을 시도한다. 만일 취약 호스트가 발견되면 봇은 침해하기 위하여 공격을 개시한다. 봇에 감염된 호스트가 가능한 희생 머신을 탐색할 때, 보통 한꺼번에 많은 짧은 패킷을 생성하게 되고, 이것이 네트워크 모니터링 장치에 의하여 확인될 수 있다.
- DDoS : 가장 오래된 봇넷 공격 메커니즘 중 하나로써, 과거에 발생하였든 Yahoo와 마이크로소프트에 대한 봇넷 공격도 DDoS를 통하여 이루어 졌다. 봇넷은 UDP 플러딩, TCP SYN 플러딩과 HTTP 플러딩과 같은 다양한 공격 도구들을 합성하여 사용하기 때문에, DDoS에 의한 공격이 빈도와 양이 감소하였지만 여전히 발생하고 있다. DDoS 공격을 개시할 수 있는 봇들로 AgoBot, SDBot, PhatBot 등이 있다.
- 피싱 및 스팸 프록시 : 봇넷이 스팸 분배에 넓게 사용되는데, 그 이유는 소스에 대한 추적이 불가능하고, 훨씬 더 많은 양의 스팸을 분배할 수 있기 때문이다. 이를 통하여 익스플로잇(멀웨어)를 분배하거나 취약성을 이용하여 멀웨어를 설치하기도 한다. 더구나 어떤 스팸은 피싱 공격을 사용한 ID 절도를 위하여 사용되기도 하였다.
- 개인 정보 도용 : 최근 봇넷들은 침해된 기계로부터 민감한 정보를 훔치기 위하여, 키로거(keylogger)와 네트워크 트래픽 스니퍼 같은 정교한 도구들을 채택하였다. 민감한 데이터들이 이러한 도구들에 의하여 기록되고 일정한 형태로 컴파일 되어 다양한 통신 채널들을 통하여 봇 마스터에게 전송된다. 주로 사용되는 방법은 봇넷에 의하여 생성된 지정된 IRC 채널이나 지정된 이메일 주소로 이메일을 통하여 데이터를 전송하는 것이다.

3.2 명령 및 제어

봇넷 분류기준으로 C&C가 포함된 이유는 다음과 같다 :

- 봇넷의 C&C는 유일하며 봇과 그 변형 사이에서 변경될 가능성이 별로 없다.
- 봇넷 C&C는 운용적이고 효과적인 봇넷을 지원하기 위하여 필수적이다.

활성화된 C&C를 차단시키거나 단순히 통신 링크를 차단만 하면, 봇마스터가 많은 수의 봇을 접촉하거나 대규모의 협동 공격 개시가 불가하기 때문에, 다른 한편으로 C&C는 또한 봇넷의 운용 관점에서 가장 취약한 링크이기도 하다. 그러므로 봇넷에서 C&C 기능에 대한 이해가 봇넷에 대응하기 위하여 아주 중요하게 여겨진다.

C&C 시스템은 대략 세 개의 다른 모델로 구분될 수 있다 :

- 집중형(centralized) 모델
- 피어-대-피어(P2P) 모델
- 랜덤(random) 모델

3.2.1 집중형 C&C 모델

기존 봇넷들이 사용하는 지배적인 모델로, AgoBot, SDBot, RBot과 같은 잘 알려진 봇들이 이 범주에 속한다. 집중형 모델에서는 봇마스터가 모든 봇의 접촉점(C&C 서버)으로 단일 고대역폭 호스트를 선정한다. C&C 서버는 IRC나 HTTP 같은 네트워크 서비스를 수행하고, 새로운 컴퓨터가 봇에 의하여 감염되면 그 C&C 서버로 연결을 개시하여 봇넷에 가입하게 된다. 일단 적당한 C&C 서버 채널에 가입되면, 그 봇은 봇마스터로부터 명령을 기다린다.

집중형 모델의 장점은 아래와 같이 세 가지로 요약될 수 있다.

(1) 풍부하고 다양한 소프트웨어 도구들로 인하여 구현이 용이하다. 그러한 도구들로는 IRC 서버상의 IRC 봇 스크립트와 IRC 봇이 있다. 이 모델을 사용하면 한 개의 봇 마스터가 수천 개의 봇을 쉽게 제어할 수 있게 된다. 최근 봇들은 금전적인 이득을 노리기 때문에 그들의 이익을 최대화하고 많은 봇들을 제어할 수 있게 하기 위하여 집중형 모델을 선호한다.

(2) 봇넷에 대한 대응책이 사용되고 있지만 완벽한 것은 아니기 때문에, 실제로 집중형 모델은 나름대로의 생존성을 유지하고 있다. 물론 단일 실패점에 대한 결점은 가지고 있지만, 당분간은 C&C 시스템을 변경하기 위하여 봇마스터들이 엄청난 노력을 들일 만큼의 동기는 없어 보인다는 분석이다.

(3) 이 모델에서의 메시징 지연이 작기 때문에 봇마스터들이 봇넷을 조정하고 공격을 개시하는 것이 쉽다.

집중형 모델에서는 서버에서 대부분의 대화가 발생하는 중요한 장소이기 때문에 심각한 결점을 가지고 있다. 그러므로 C&C 서버가 봇넷에서 가장 취약한 링크가 된다. 해당 C&C 서버만 찾아서 차단하면 전체 봇넷이 없어지게 된다.

3.2.2 P2P-기반 C&C 모델

봇넷 제작자들은 네트워크에서 실패에 더욱 탄력성 있는 대안적인 봇넷 통신 시스템을 구축 하였다. 최근에 P2P 통신을 이용한 봇넷이 증가하는 추세이다[5]. 예를 들어, Phatbot의 어떤 변형은 봇넷을 제어하기 위한 수단으로 P2P 통신을 사용한다.

집중형 C&C 모델에 비하여 P2P 기반 C&C 모델은 발견하고 없애는 것이 훨씬 더 어렵다. 통신 시스템이 소수의 선택된 서버에 과도하게 의존하지 않기 때문에, 몇 개의 봇을 없앤다고 하여도 전체 봇넷이 파괴되지는 않는다. 이러한 점 때문에 P2P 기반 C&C 모델이 가까운 미래에 봇넷에서 점점 더 많이 사용될 것이다. 따라서 P2P 기반 C&C 모델을 사용하는 봇넷에 대응하는 것은 더 많은 도전을 제시하고 있다.

한편 기존 P2P 기반 C&C 시스템은 아래와 같은 많은 제한사항을 가지고 있다.

- 지원되는 그룹 크기가 집중형에 비하여 너무 작다. 집중형에서는 천 개 정도가 소규모에 속하는 반면, P2P 모델에서는 통상 10개 내지 50개 정도의 범위에 불과하다.
- 기존 P2P 시스템은 메시지 전달과 전파 지연을 보장하지 못한다. 그러므로 P2P 통신을 사용하는 경우, 집중형 모델을 사용하는 것보다 봇들을 조정하는 것이 더 힘들게 된다.

이런 두 가지 제약사항들이 봇넷에서 P2P 기반 통신의 광범위한 채택을 제한시켜 왔다. 따라서 기존의 P2P-기반 봇넷들은 소수의 목표 호스트들을 공격하기 위하여 사용되었다. 그러나 P2P-기반 봇넷 지식이 축적됨에 따라, 이러한 제약사항을 극복한 새로운 P2P-기반 봇넷이 출현할 것이다. 그리하여 집중형 통신보다 더욱 견고한 P2P-기반 통신으로 더욱 더 많은 봇넷들이 이동할 것이다.

3.2.3 랜덤 C&C 모델

이 모델은 실제 봇넷에서는 사용되지 않았지만 높은 생존성을 원하는 어떤 미래 형태의 봇넷에게 흥미가 있다[3]. 제안된 랜덤 C&C 모델에서는 봇은 다른 봇이나 봇마스터를 능동적으로 접촉하지 않는다. 대신에 봇은 자신의 봇마스터로 부터의 입력 연결을 청취한다. 공격을 개시하기 위하여 봇마스터는 봇을 발견하기 위하여 인터넷을 스캔한다. 봇이 발견되면 봇마스터는 그 봇에게 명령을 발행한다.

이런 모델은 구현이 용이하고, 발견 및 파괴에 아주 탄력적이지만, 본질적으로 확장성 문제가 있고 대규모

협동 공격에 사용되기는 어렵다.

3.3 통신 프로토콜

대부분의 경우에 봇넷은 그들의 통신을 위하여 새로운 네트워크 프로토콜을 생성하는 대신에 이미 이용 가능한 소프트웨어 도구에 의하여 구현된 기존의 통신 프로토콜을 사용한다. 봇넷에 의하여 사용되는 통신 프로토콜의 이해가 중요한 이유는 다음과 같다 :

- 통신 특성이 봇넷 기원과 사용되고 있는 가능한 소프트웨어 도구들에 대한 이해를 제공한다.
- 통신 프로토콜에 대한 이해를 통하여 봇과 마스터 사이에서 발생하는 대화를 해독할 수 있다.

3.3.1 IRC 프로토콜

아직도 가장 많이 지배적으로 사용되는 프로토콜이다. IRC 프로토콜은 "채널"이라는 토의 포럼에서 그룹 통신을 위하여 주로 설계되었으나, 개인 메시지를 통한 일대일 통신을 허용한다. 이런 융통성 있는 통신 모델이 전체적으로 봇넷을 명령할 수 있고 또한 일대일 통신을 이용하여 선택적으로 봇들을 명령할 수 있기 때문에 봇 마스터에게 아주 유용하다. 보다 자세한 IRC-기반 봇넷 구조에 대하여는 [9,14]를 참조할 수 있다.

정상적인 IRC 서버/클라이언트와 IRC 봇넷이 다른 점은 봇은 고객화된(customized) 신택스를 가지고 그들의 채널을 통하여 전송되는 메시지를 파스하는 스크립트를 가지고 있다. 봇넷에 따라서 신택스도 달라지기 때문에, 신택스를 조사함으로써 봇넷과 그 기원에 대한 추가적인 정보를 얻을 수 있다.

3.3.2 HTTP 프로토콜

봇넷 통신을 위하여 또한 인기 있는 방법으로 HTTP 프로토콜이 있으며, 그 이유는 다음과 같다 :

- IRC 프로토콜을 이용한 봇넷은 잘 알려져 있기 때문에, 봇넷을 탐지하기 위하여 IRC 트래픽 모니터링이 많이 수행된다. 반면 HTTP 트래픽은 인터넷 트래픽의 대부분을 차지하기 때문에 탐지가 더 어렵다.
- 대부분의 조직에서는 게이트웨이 방화벽을 구현하고, 통상적으로 IRC 포트를 포함하여 원하지 않는 포트로의 입/출력 트래픽을 차단한다. 그러므로 IRC 프로토콜을 사용하면 봇은 C&C 서버와 통신을 할 수 없기 때문에, 방화벽 보안 정책을 우회할 수 있는 통신 채널로 HTTP를 사용한다. 예를 들어, Bobax[6] 봇은 C&C 서버와 통신하기 위하여 HTTP를 사용한다.

HTTP를 사용하는 봇넷을 탐지하는 것은 봇넷 트래픽이 엄청난 양의 정상적인 HTTP 트래픽 속에 숨겨져 있기 때문에 훨씬 더 어렵다. 그러나 봇넷에 의하여 생성되는 HTTP 트래픽은 정상적인 HTTP 트래픽과는 다르기 때문에 적절한 필터가 있다면 탐지가 가능하다.

3.3.3 기타 프로토콜

다른 진화된 봇넷들은 IM(Instant Messenger) 프로토콜, P2P 프로토콜을 사용한다. AgoBot의 후손인 Phatbot의 최근 변형은 P2P 통신을 사용한 것으로 알려져 있다. Phatbot의 변형에서는 개인적인 메시징과 작은 수의 그룹사이의 파일 전달을 위하여 암호화된 P2P 프로토콜을 구현한 코드를 사용하였다. IRC와 HTTP 프로토콜이 아닌 다른 프로토콜을 사용하는 봇넷의 수는 아직 상대적으로 작지만, 미래에는 더욱 더 많이 사용될 전망이기 때문에, 그에 따른 봇넷 탐지도 더 힘들 것으로 분석된다.

3.4 관측 가능한 봇넷 행위

봇넷의 존재를 탐지하기 위하여, 봇넷이 보여주는 비정상 행위를 발견해야 한다. 관측 가능한 봇넷의 행위는 아래와 같이 세 가지 형태로 분류될 수 있다 :

- 네트워크-기반 행위
- 호스트-기반 행위
- 전역적 상호관련 행위

이런 봇넷 행위를 빨리 잡아내는 것이 대응책을 강구하고 봇넷을 탐지하는데 매우 중요하다.

3.4.1 네트워크-기반 행위

봇마스터는 봇들과 통신하고 공격을 개시하기 위한 기능을 수행하는데, 이 때 관측 가능한 네트워크 트래픽

패턴을 생성한다. 이를 이용하여 개별 봇들과 C&C 서버를 탐지할 수 있다.

봇넷들이 봇들과 통신을 위하여 IRC와 HTTP를 사용하기 때문에, 비정상 패턴을 가진 관측 가능한 IRC와 HTTP 트래픽이 봇의 존재를 나타내기 위하여 사용될 수 있다. 예를 들어, IRC 서비스가 허용되지 않는 내부 기업망으로의 내향/외향 IRC 트래픽, 쉽게 이해되지 않는 어떤 신택스를 따르는 대화 등이 있다.

많은 봇넷들이 그들의 C&C 서버를 찾기 위하여 동적 DNS 도메인 네임을 사용하는데, 이 때 비정상 DNS 질의가 봇넷을 탐지하기 위하여 사용될 수 있다. 예를 들어, 관측 가능한 트래픽 스트림 내의 C&C 서버의 IP 주소를 수집하거나, 주기적으로 변경되는 특정 도메인 네임과 관련된 IP 주소 등을 조사하는 것이다. 봇넷들에 의하여 생성되는 어떤 종류의 통신 트래픽은 정상 트래픽보다 군집적(bursty)이기 때문에, 네트워크 트래픽 플로를 관찰함으로써도 발견될 수 있다.

예를 들어 봇넷들이 DDoS TCP SYN 플러딩 공격을 수행한다고 가정하면, 엉터리 소스 IP 주소를 가진 많은 수의 유효하지 않은 TCP SYN 패킷을 전송할 수 있다. 그러므로 네트워크 감시 장비가 유효하지 않은 소스 IP 주소를 가진 많은 수의 외향 TCP SYN 패킷을 발견한다면, 어떤 내부 호스트가 침해되었고 DDoS 공격에 적극적으로 참여하고 있다는 것을 알아 낼 수 있다.

3.4.2 호스트-기반 행위

봇이 실행될 때, 봇은 시스템 레지스터리와 시스템 파일을 변경하고, 네트워크 연결을 생성하거나 앤티바이러스 프로그램을 불능화 시키는 일련의 시스템/라이브러리 호출을 한다. 봇에 의하여 만들어지는 이런 일련의 시스템/라이브러리 호출은 합법적인 프로그램과 애플리케이션들에 의하여 만들어진 것과는 다를 수 있다. 이런 행위들이 보안 시스템에게 관측될 수 있다. 시스템/라이브러리 호출 시퀀스에 대한 조사를 통하여 봇을 탐지하는 도구에 대한 소개는 [11]을 참조할 수 있다.

3.4.3 전역적 상호관련 행위

탐지 효율성 관점에서 전역적인 행위 관측이 가장 좋다고 여겨진다. 그리하여 전역적인 행위 특성이 봇넷의 기본적인 구조와 메커니즘과 자주 결부된다. 왜냐하면, 봇넷의 구조와 메커니즘이 재설계되고 재구현되지 않는 한, 그 것들이 봇넷마다 거의 변경되지 않기 때문이다. 결과적으로, 이런 전역적으로 관측 가능한 행위가 봇넷류를 탐지하기 위하여 가장 유용하다.

예를 들어, 봇넷이 C&C 서버를 발견하기 위하여 동적 DNS 항목을 사용하는데, 이 DNS 항목에 대하여 전역적으로 DNS 질의 트래픽이 증가될 수 있다. 이를 통하여, 해당 동적 DNS 도메인 네임이 봇넷에 의하여 C&C 서버로 사용되고 있다는 확률이 높게 된다. 이러한 특징은 봇넷이 통신을 위하여 IRC나 HTTP를 사용하는지에 관계없이 거의 변경될 가능성이 없다. 다른 전역적인 행위의 예로 네트워크 플로가 있다.

3.5 집결 메커니즘

집결 메커니즘은 봇넷들이 새로운 봇들을 발견하고 그들의 봇마스터 아래에 집결시키기 위하여 아주 중요하다. 통상적으로 사용되는 집결 메커니즘은 아래와 같다.

3.5.1 하드-코드된 IP 주소

통상적으로 봇은 자신의 바이너리 속에 하드 코드된 C&C 서버 IP 주소를 가지고 있다. 봇이 어떤 컴퓨터를 처음 감염시킬 때, 자신의 바이너리 코드 안에 포함된 하드 코드된 서버 IP 서버 주소를 이용하여 C&C 서버로 다시 연결하게 된다. 예를 들어, WORM_ZOTOB.E은 하드 코드된 IP 주소를 사용하는 IRC 봇이다[10]. 이 주소를 사용하는 문제점은 C&C 서버가 쉽게 탐지되고 통신 채널이 쉽게 차단될 수 있다는 것이다. 이런 방법으로 C&C 서버가 단절되면, 봇넷은 완전히 비활성화 될 수 있다. 이 때문에 하드 코드된 IP 서버 주소는 봇의 최근 변형에서는 많이 사용되지 않는다.

3.5.2 동적 DNS 도메인 네임

현재의 봇넷은 동적 DNS 제공자에 의하여 할당된 하드-코드된 도메인 네임을 포함한다. 이 방법의 장점은, 만약에 C&C 서버가 당국에 의하여 차단된다면, 봇마스터는 다른 곳의 새로운 C&C 서버를 생성하고 해당 동적 DNS 항목 내의 IP 주소를 갱신함으로써 자신의 제어를 다시 쉽게 개시할 수 있다는 것이다. 구 C&C 서버로의 연결이 실패하면 봇은 DNS 질의를 수행하고 새로운

C&C 서버로 재지시(redirect) 된다. 이런 DNS 재지시(redirection : 혹은 경로 재지정)를 "herding"이라 한다. 이와 같이, 동적 DNS를 사용하면 봇마스터는 기존 C&C 서버가 동작하는데 실패할 때에도 자신의 봇넷 상의 제어를 유지할 수 있게 된다. 때로는 봇마스터가 C&C 서버의 위치를 이동하기 위하여 주기적으로 동적 DNS 항목을 갱신할 수 있으며, 이것은 탐지를 더 어렵게 만든다.

3.5.3 분산 DNS 서비스

좀 더 새로운 봇넷류는 보안 당국의 법집행 범위에서 벗어나는 위치에 자신의 분산 DNS 서비스를 운영한다. 봇은 이런 DNS 서버들의 주소를 가지고 있으며 C&C 서버의 IP 주소를 해결하기 위하여 이런 서버들을 접촉한다. 많은 경우 이런 DNS 서비스들은 게이트웨이에서 보안 장치에 의한 탐지를 회피하기 위하여 상위 포트 번호에서 수행되도록 선택된다. 분산 DNS 서비스를 사용하는 봇넷이 위에서 기술된 다른 형태의 봇넷과 비교하여 탐지하고 파괴하기가 가장 어렵다.

3.6 회피 기법

봇넷의 진화는 계속되고 있으며 점점 더 정교하여 지고 있다. 첨단 봇은 항바이러스 엔진과 시그니처-기반 침입탐지시스템(IDS)을 회피할 뿐만 아니라, 비정상-기반 탐지 시스템도 회피할 수 있다. 회피 기술로 다양한 기법들이 사용되고 있으며, 예를 들어 정교한 실행 패킷, 루트킷, 프로토콜 회피 기법 등이 있다.

게다가 봇넷은 통신 흔적을 숨기기 위하여 새로운 메커니즘을 지속적으로 추가하고 있다. 봇넷을 탐지하기 위하여 IRC 트래픽에 대한 감시가 증가하기 때문에, 어떤 봇넷은 IRC를 더 이상 사용하지 않고 대신 수정된 IRC 프로토콜이나, HTTP나 VoIP 같은 다른 프로토콜을 사용하기 시작하였다. 내용물이 노출되는 것을 피하기 위하여 암호 기법도 사용된다. 어떤 봇넷은 TCP와 ICMP 터널링과 하물며 IPv6 터널링[7]과 같은 변환 채널 통신을 사용하기도 한다. SKYPE[15]와 IM[1]의 사용 가능성을 논의하는 분석도 있다. 이런 새로운 봇넷이 출현하는 것은 단지 시간문제라는 분석이다.

새로운 회피 기법의 개발과 더불어 이에 대응할 수 있는 새로운 봇넷 탐지 및 방지 기술에 대한 개발도 이루어 질 것이며, '블랙햇'과 '레드햇' 사이의 전쟁 수위는 상승될 전망이다.

Ⅳ. 봇넷 계층 구조[4]

현재의 봇넷이 가지고 있는 문제점을 극복하기 위하여 미래의 봇넷에 의하여 사용될 수 있는 개선사항으로 P2P 네트워크, 인증(디지털 서명) 및 은닉된(covert) 채널 기법들을 사용한 계층화된(layered) 방법을 제시하고 있으며, 아래와 같다. 제시된 계층은 제어 계층, 통신 계층 및 감염 계층으로 나누어지며, 먼저 제어 계층의 기능에 대하여 제시한다.

4.1 제어 계층

4.1.1 문자기반/XML

현재의 봇넷을 분석한 결과, 봇넷 제작자들은 봇 명령을 위한 가장 좋은 포맷은 순수한 문자라는 것을 이미 인식하였다는 것이 분명하다고 기술한다. 새로운 인터넷 기능을 지원하기 위하여 생성되는 대부분의 신규 프로토콜들은 문자-기반이다. 웹 2.0 집합의 기술에서는 XML과 HTTP가 여러 가지 다른 방법으로 사용될 수 있다는 것을 보여주며, 봇넷도 예외가 아니라는 지적이다.

현재 봇들은 이미 단순한 형태의 문자 통신을 사용하고 있으며, XML을 사용하면 명령 파서의 생성도 쉬워지고, 프로그래머로 하여금 XMLSIG 같은 새로운 표준으로부터의 기능성을 반영하는 것을 허용한다. 다른 시스템들에서 명령이 동일하게 번역될 복수-플랫폼 봇넷도 생성될 것이다.

4.1.2 디지털 서명

XML 관련 표준을 통한 명령들의 서명과 암호화는 봇넷 마스터로 하여금 봇들과의 통신을 보호하여 주며 순수한 문자를 가지고 작업을 하는 장점을 손상함이 없이 통제할 수 있도록 보장하여 준다.

4.1.3 채널 독립성

XML을 사용함으로써 봇넷은 봇에게 명령을 전달하기 위하여 사용되는 통신시스템과는 완전히 독립적이게 된다는 것이다. 계층화된 설계를 통하여, 봇의 제어 계

층은 하위 계층인 통신 계층의 상세한 세부적인 사항을 알 필요가 없게 되며, 봇들은 다른 통신 방법을 사용할 수 있다는 것이다.

봇의 특징을 나타내는 봇의 부분인 페이로드도 별 개의 계층으로 개발될 수 있다고 기술한다. 여러 개의 특징 모듈을 구성하여 필요한 경우 새로운 모듈을 명령 계층으로부터 다운로드 받아 사용하면 봇의 명령 집합을 확장할 필요가 없다는 것이다. 확장 기능을 가진 융통성 있는 봇을 만드는 다른 방법으로, PowerShell이나 IronPython과 같은 스크립트 언어를 사용할 수 있다.

4.2 통신 계층

봇넷 제작자들에 의하여 사용될 수 있는 융통성 있는 대안적인 통신에 대하여 기술하고 있다. 제시된 개념들은 최근 봇넷에서 식별되고 있으며, 봇넷 제작자들이 이런 문제를 해결하기 위하여 매우 노력하고 있다는 것을 보여준다고 지적한다.

봇이 명령을 어디서 얻는지 모르도록 IP 주소를 가지지 않는 한 가지 방법으로, OTP(one time password) 알고리즘의 사용을 기술하고 있다. 봇과 마스터는 씨드(seed)를 전달할 수 있으며, 마스터는 XML에 의하여 전송된 명령에 의하여 추후 갱신이 가능하고, 인터넷상에서 발견될 필요가 있는 스트링을 on the fly로 계산한다. 이 스트링을 발견하고 봇은 자신의 마스터 혹은 명령을 받을 수 있는 주소를 또한 발견할 수 있다.

시간 동기화에 대한 필요성을 없애기 위하여, 봇은 시간 단위로 한 개의 OTP를 계산하여 24개의 다른 값을 생성하도록 한다. 가능한 통신 채널로 아래와 같은 방법들이 사용될 수 있다.

- IM(Instant Messenger) : 봇넷을 위하여 가장 많이 사용되는 방법 중의 하나이다.
- DNS : 원래 소스가 차단되고 제거될 때에도, 이용 가능한 정보를 유지하는데 도움을 주는 캐싱 시스템에 대한 장점이 있다.
- SPAM : 스팸 메일을 통하여 OTP 스트링도 전송될 수 있으며, 명령을 전달 할 수 있게 된다.
- 웹 메일 시스템 : 마스터가 계정이름으로 OTP 스트링을 사용하여 웹메일 시스템에 e-메일 계정을 연다. 디폴트 패스워드를 사용하여 봇은 이 계정에 로긴할 수 있고 명령을 찾을 수 있다. 봇마스터에 의하여 생성된 정보만 처리하도록 보증하기 위하여 디지털 서명에 의한 콘텐츠 인증이 사용된다.
- 탐색 엔진 : OTP 스트링을 포함하고 있는 웹 사이트를 찾기 위하여 웹 탐색을 사용할 수 있고, 봇이 필요한 정보를 게시할 수 있다.
- SBLOG : 블로그 포스트로부터의 코멘트 필드가 OTP 스트링을 출판하기 위한 장소로 이용될 수 있다.
- P2P 네트워크 : P2P 파일 공유 시스템을 이용하여 OTP 스트링을 가진 파일을 탐색할 수 있다. 봇을 위한 여러 개의 명령이나 추가적인 모듈을 포함할 수 있는 대규모 파일도 쉽게 전송할 수 있는 장점이 있다. 봇은 프로브(probe)를 전송하여 피어 사이에서 다른 봇의 존재를 조사할 수 있고, 봇으로 알려진 피어들의 목록을 보유할 수 있다. P2P 네트워크를 이용한 명령의 전달은 더욱 비밀 통신을 전개하도록 하여준다.
- Skype : 인터넷 전화를 위하여 많이 사용되는 Skype가 봇 통신을 위한 수단이 될 수 있다. 봇과 봇마스터는 Skype 내의 IM이나. Profile 데이터 필드에 의하여 정보를 교환할 수 있다. 봇넷 설계자에게 가장 관심 있는 Skype의 특징은, 이 프로토콜이 암호화될 뿐만 아니라 방화벽 정책 실행 규칙을 우회하기 위하여 설계되었다는 것이다.

4.3 감염 계층

봇이 다른 컴퓨터를 효과적으로 감염시키고 확산하기 위하여 새로운 익스플로잇을 채택할 것으로 전망되며, 지속적으로 개선되어 나갈 것이다. 봇넷 제작자들이 익스플로잇 프레임워크를 또한 사용할 수 있으며, 이를 위하여 봇을 프레임워크 페이로드 포맷에 이동하여야 하고, 봇 내에 프레임워크 연결 기능성을 통합해야 한다. 이 작업이 쉽지는 않지만, 봇 제작자들에 의하여 이미 이런 프로그래밍 숙련도는 보여주었다는 분석이다. 익스플로잇 프레임워크와 호환함으로써 봇은 단지 새로운 익스플로잇 모듈을 다운로드만하면 되게 된다.

V. 혼합 P2P 봇넷[12]

5.1 기존 P2P 봇넷의 문제점

Slapper[2], Sinit[13], Phatbot[8] 등과 같은 봇넷들이

P2P 제어 구조를 가지고 구현된 대표적인 것들이다. 예를 들어, 대응 시스템에 의하여 봇넷을 쉽게 차단하는데 이용될 수 있는 부트스트랩(bootstrap) 과정을 제거하기 위하여, Slapper 웜은 전파 동안 모든 감염 컴퓨터에 대한 알려진 봇들의 목록을 구축한다. 비슷하게 Sinit도 부트스트랩 과정이 없으며 갱신 인증을 위하여 공개키 암호를 사용한다.

그럼에도 불구하고 기존의 P2P 봇넷은 많은 약점을 가지고 있다고 분석하였다. Sinit 봇은 통신하기 위한 다른 Sinit 봇들을 찾기 위하여 random probing을 사용한다. 이것이 구축된 봇넷에 대한 열악한 연결성을 초래하고 과도한 probing 트래픽으로 하여금 쉽게 탐지 된다. Phatbot은 자신의 부트스트랩 과정을 위하여 Gnutella 캐시 서버를 이용한다. 이것 또한 봇넷이 쉽게 차단 되도록 한다. 게다가 기반이 되는 WASTE P2P 프로토콜은 대규모 네트워크에 확장성이 없다. Slapper는 암호 및 명령 인증을 구현하지 않음으로써 다른 것에 의하여 쉽게 하이재킹 당할 수 있다. 게다가 알려진 봇의 목록은 봇넷의 (거의) 모든 구성원들을 포함하기 때문에, 한 개의 봇만 잡혀도 전체 봇넷이 노출되게 된다. 또한 복잡한 통신 메커니즘으로 인하여 많은 트래픽을 생성하여 네트워크 플로 분석에 의하여 감시당하기 쉽다.

5.2 진화된 P2P 봇넷의 특징

집중형과 기존의 P2P 봇넷의 문제점들을 고려하여, 진화된 봇넷의 설계는 봇마스터에 의하여 직면하는 아래와 같은 현실적인 문제들을 고려해야 한다고 기술하고 있다 :

(1) 방어시스템에 의하여 상당한 부분의 봇넷이 제거된 후에도 남아있는 봇들을 제어할 수 있는 견고한 봇넷을 생성하는 방법.
(2) 방어시스템에 의하여 어떤 봇들이 포획될 때 네트워크 토폴로지의 심각한 노출을 방지하는 방법.
(3) 봇마스터에 의하여 봇넷의 완전한 정보를 쉽게 감시하고 획득하는 방법
(4) 통신 트래픽 패턴을 통한 봇 탐지를 방지하는 방법
(5) 봇들의 동적 혹은 사설 IP 주소, 매일의 온라인/오프라인 재산과 같은 네트워크 관련 이슈를 고려한 설계

이와 같은 사항들을 고려하여, 아래와 같은 특징을 가진 혼합 P2P 봇넷을 제안하고 있다 :

- 부트스트랩 절차가 필요 없는 봇넷
- 봇넷은 각 봇에 포함된 피어 목록을 통하여 통신한다. 각 봇은 고정된, 제한된 크기의 피어 목록을 가지며 자신의 목록을 다른 봇에게 공개하지 않는다. 이런 방법으로, 한 봇이 잡혀도 피어 목록에 있는 단지 제한된 수의 봇들만 노출된다.
- 봇마스터는 *report* 명령을 발행하여 전체 봇넷을 쉽게 감시할 수 있다. 이 명령은 봇들이 봇마스터에 의하여 제어되는 특정 침해 머신(*센서 호스트*라고 함)에게 보고하도록 지시한다. *report* 명령에 있는 센서 호스트의 IP 주소는 명령이 발행될 때마다 변경된다.
- 위의 *report* 명령을 통하여 봇넷에 대한 정보를 수집한 후, 봇마스터는 필요한 경우 모든 봇들에게 자신들의 피어 목록을 갱신하도록 센서 호스트를 접촉하기 위하여 *update* 명령을 발행할 수 있다. 이렇게 함으로써 균형 있고 견고한 연결성을 보유하고, 부서진 봇넷을 다시 연결하게 된다.
- 인터넷으로부터 접근 가능한 정적 전역 IP 주소를 가진 봇만이 피어 목록에 있는 후보가 된다. 이런 호스트는 동시에 클라이언트와 서버 특징을 가지고 행동하기 때문에 P2P 용어로 *servent* 봇이라 한다. 이렇게 함으로써 각 봇 내의 피어 목록이 긴 수명을 가지도록 한다.
- 모든 *servent* 봇은 다른 봇들로부터의 입력 연결을 위한 자기-결정 서비스 포트를 청취하고 입력 트래픽에 대하여 자기-생성 대칭 암호화 키를 사용한다. 이 개별적인 암호화 및 개별 서비스 설계가 봇넷 통신 트래픽의 네트워크 플로 분석을 통한 봇넷 탐지를 매우 어렵게 만든다.

5.3 혼합 P2P 봇넷 구조

[12]에서 제안된 진화된 혼합 P2P 구조는 아래와 같은 특징을 가진다 :

- 두 종류의 봇 : 서번트 봇과 클라이언트 봇
  서번트 봇은 정적, 비-사설 IP 주소를 가지며 전역 인터넷으로부터 접근이 가능하다. 클라이언트 봇은 아래와 같은 나머지 봇들을 포함 한다 :
  - 동적으로 할당되는 IP 주소를 가진 봇
  - 사설 IP 주소를 가진 봇

- 전역 인터넷으로부터 연결될 수 없는 방화벽 뒤의 봇

서번트 봇은 클라이언트와 서버 역할을 동시에 수행하며, 클라이언트 봇은 입력 연결을 수락하지 않는다.

- 봇넷 명령 및 제어 구조 : 봇 마스터는 봇넷의 어떤 봇을 통하여도 자신의 명령을 주입할 수 있다. 클라이언트와 서번트 봇 모두는 능동적이고 주기적으로 자신들의 피어 목록에 있는 서번트 봇에 연결하여, 봇마스터에 의하여 발행된 명령을 검색한다. 새로운 명령에 대하여는 자신의 피어 목록에 있는 모든 서번트 봇에게 그 명령을 즉시 전달한다.
- 전통적인 C&C 봇넷과의 연관성
제안된 구조는 집중형 봇넷의 사실상의 확장이며, 서번트 봇이 C&C 서버 역할을 하게 된다. 그러나 서번트 봇의 수가 크게 늘어나고 서로 연결되는 구조이다. 많은 수의 서번트 봇이, 제안된 봇넷이 차단되기 어렵게 되는 주된 이유이다.

진화된 혼합 봇넷의 명령 및 제어 통신과 봇넷 구성 및 방어에 대한 사항은 [12]를 참조할 수 있다. 혼합 P2P 봇넷에 대한 방어 전략을 요약하면 아래와 같다 :

- 정적 전역 IP 주소를 가진 컴퓨터들이 침해되지 않도록 예방하거나, 침해된 경우 빨리 제거함으로써 봇넷 통신 네트워크 구축을 하지 못하도록 한다.
- 신속한 탐지 및 대응 시스템을 개발하여, 봇 마스터가 첫 번째 갱신 명령을 발행하기 전에, 새로이 생성된 봇넷의 서번트 봇의 초기 집합을 빨리 차단한다.
- 하니팟 기법을 기반으로 P2P 봇넷의 통신 채널을 poison하도록 노력한다. 감염 하니팟을 봇넷에 가입시키고 정적 전역 IP 주소를 가졌다고 하면, 서번트 봇으로 대우된다. 결과적으로 많은 봇 속의 피어 목록에 위치를 차지하고 유효한 통신 채널의 수를 감소시킨다.

Ⅵ. 맺음말

봇넷이 인터넷 환경에 심각한 위협을 제시하고 있다. 이 봇넷 위협에 효과적으로 대처하기 위하여 기존 봇넷에 대한 탐지, 대응 및 완화 방안이 요구된다. 또한 미래의 봇넷 공격에 대하여도 준비하기 위하여 과거에 출현하였든 봇넷에 대한 탐지 및 방어 전략뿐만 아니라 미래의 진화된 봇넷에 대하여도 연구할 필요가 있다. 따라서 본 논문에서는 이미 알려진 그리고 알려지지 않은 봇넷 행위에 대하여 보다 나은 이해를 하기 위하여 먼저 봇넷 분류법에 대하여 기술하였다. 그리고 미래의 진화된 봇넷 구조에 대하여 소개함으로써 향후의 봇넷 위협에 대하여 정확히 식별하고, 탐지하고, 대응할 수 있는 방안을 모색하고자 하였다.

가까운 미래에 봇은 아래와 같은 방법으로 설계될 수 있다고 전망한다[4] :

- 계층화 설계를 따름으로써 쉽게 확장되고 갱신될 수 있다.
- 봇이 사용하는 통신이나 프로토콜의 종류와는 독립적으로, 복수 형태의 네트워크와 프로토콜을 경유할 수 있다.
- 마스터는 정확한 위치를 알 수 없기 때문에 발견이 힘들게 된다.
- 디지털로 서명된 명령을 채택하여 쉽게 하이재킹되지 않는다.
- 신용장(credential)을 훔치지 않고도 웹 사이트와 온-라인 은행 상의 사용자에 의한 거래를 직접 변경할 수 있다.
- DNS와 HTTP 같은 쉽게 차단될 수 없는 통신 프로토콜을 사용한다.

국내에서도 봇넷의 위협에 대한 효과적인 대책을 강구하기 위하여 산 · 학 · 연이 유기적인 관계를 가지고 지속적이고 체계적으로 연구가 수행되어야 한다고 판단된다.

참고문헌

[1] AOL Instant Messenger, http://en.wikipedia.org/wiki/AOL_Instant_Messenger

[2] I. Arce and E. Levy, "An analysis of the slapper worm", IEEE Security and Privacy Magazine, Jan.-Feb. 2003.

[3] Evan Cooke, Farnam Jahanian, and Danny McPherson, "The Zombie Roundup : Understanding, Detecting, and Disrupting Botnets". Proc. of Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI '05), Boston, 2005.

[4] Andre Fucs, Augusto Paes de Barros, and Victor

Pereira, "New botnets trends and threats", Web document.

[5] Julian B. Grizzard et al., "Peer-to-Peer Botnets : Overview and Case Study", http://www.usenix.org/event/hotbots07/tech/full_papers/grizzard/grizzard_html/

[6] LURHQ Threat Intelligence Group, Bobax Trojan Analysis, http://www.lurhq.com/bobax.html

[7] Malware Tunneling in IPv6, http://www.us-cert.gov/reading_room/ IPv6Malware-Tunneling.pdf

[8] Phatbot trojan analysis, Http://www.lurhq.com/phatbot.html

[9] Ramneek Puri, Bots & Botnet : An Overview, GSEC Practical Assignment Version 1.4b, Aug. 2003, SANS Institute.

[10] Trend Micro, Taxonomy of Bonet Threats, A Trend Micro White Paper, Nov. 2006.

[11] Jau-Hwang Wang et al., "Virus detection using data mining techniques", Proceedings of IEEE 37th Annual 2003 International Carnahan Conference on Security, 2003.

[12] Ping Wang, Sherri Sparks and Cliff C. Zou, "An Advanced Hybrid Peer-to-Peer Botnet", Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, 2007.

[13] Sinit P2P trojan analysis, Http://www.lurhq.com/sinit.html

[14] Jianwei Zhuge et al., "Characterizing the IRC-based Botnet Phenomenon", Reihe Informatik, TR-2007-010, Dec. 2007.

[15] http://www.skype.com/helloagain.html

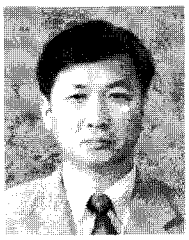
[16] 나재훈, "P2P 지식정보보안 기술 및 표준동향, ITFind 메일진 353호, 2008.7.4.

[17] 장영준, 차민석, 정진성, 조시행, "악성 코드 동향과 그 미래 전망", 한국정보보호학회 정보보호학회지 제 18권 제 3호, pp.1-16, 2008년 6월.

[18] 전용희, "봇넷 개요 및 전망 분석", 한국정보보호학회 정보보호학회지 제 18권 제 3호, pp.101-108, 2008년 6월.

[19] 정현철, Detection and Response Technology for Botnets, KRnet '08 자료, 2008.6.25.

〈著者紹介〉



전 용 희 (Yong-Hee Jeon)
종신회원

1971년 3월~1978년 2월 : 고려대학교 전기전자전파공학부
1985년 8월~1987년 8월 : 미국 플로리다 공대 대학원 컴퓨터공학과
1987년 8월~1992년 12월 : 미국 노스캐롤라이나주립 대학원 Elec. and Comp. Eng. 석사, 박사
1978년 1월~1978년 11월 : 삼성중공업(주)
1978년 11월~1985년 7월 : 한국전력기술(주)
1979년 6월~1980년 6월 : 벨기에 벨가톰사 연수
1989년 1월~1989년 6월 : 미국 노스캐롤라이나주립대 Dept of Elec. and Comp. Eng. TA
1989년 7월~1992년 9월 : 미국 노스캐롤라이나주립대 부설 CCSP (Center For Comm. & Signal Processing) RA
1992년 10월~1994년 2월 : 한국전자통신연구원 광대역통신망연구부 선임연구원
1994년 3월~현재 : 대구가톨릭대학교 컴퓨터 · 정보통신공학부 교수
2001년 3월~2003년 2월 : 대구가톨릭대학교 공과대학장 역임
2004년 2월~2005년 2월 : 한국전자통신연구원 정보보호연구단 초빙연구원
2007년 1월~2007년 12월 : 한국정보보호학회 학회지 편집위원장
2008년 1월~현재 : 한국정보보호학회 부회장
<관심분야> 네트워크 보안, 웜 모델링 및 대응 기술, 통신망 성능분석

오 진 태 (Jintae Oh)
정회원

1990년 2월 : 경북대학교 전자공학과 공학사
1992년 2월 : 경북대학교 전자공학과 석사
1992년 2월~1998년 2월 : 한국전자통신연구원 선임연구원
1998년 3월~1999년 1월 : 미국 MinMax Tech. 연구원
1999년 2월~2001년 10월 : 미국 Engedi Networks. Director
2001년 10월~2003년 1월 : 미국 Winnow Tech. Co-founder, CTO 부사장
2003년 3월~현재 : 한국전자통신연구원 선임연구원, 정보보호연구본부 보안게이트웨이연구팀 팀장
2008년 1월~현재 : 한국정보보호학회 학회지 편집위원(간사)
<관심분야> 네트워크보안, 비정상행위탐지기술, 공격 시그니처 자동생성기술, 보안하드웨어기술