CPN을 이용한 Honeypot 모델 설계

정회원 현 병 기*, 구 경 옥**, 조 도 은***, 조 용 환***

Honeypot Model Analysis using CPN

Byoung-Ki Hyun*, Kyoung-Ok Koo**, Do-Eun Cho***,
Yong-Hwan Cho*** *Regular Member*

요 약

본 논문은 CPN(Colored Petri Nets)을 이용한 Honypot 모델의 설계 및 구현에 관한 것이다. 제안된 Honeypot 모델은 해커의 침입을 능동적으로 유도하고 침입을 탐지 및 행동패턴의 파악을 위해 보안커널 모듈과 유도된 해커의 활동을 위한 가상 모듈로 구성되어 있으며, CPN을 이용한 모델과 기존의 Denning 모델 및 Shieh 모델과 비교 분석하였다. 본 논문에서 제안된 CPN을 이용한 Honeypot 모델은 침입패턴의 특성에 대한 분류가 가능하고, 침입패턴의 모델링과 패턴매칭 과정의 모델링이 가능하며, 다중 호스트를 통한 DDoS 공격의 탐지가 가능하고, 마지막으로 침입패턴의 분석을 위한 학습모델의 기반 제공이 가능하다.

key words : CPN, Honeypot, Intrusion Detection Model

ABSTRACT

This paper is a study about Honey-pot Model using CPN(Colored Petri Nets) that is a method of intrusion detection. Suggested Honey-pot model consists of two parts : ① security kernel module for active induction of hacker's intrusion, intrusion detection and behavior pattern analysis. ② virtual module for activity of induced hackers. However, suggested model was compared and analysed with conventional Denning model and Shieh nodel. The Honey-pot model using CPN can classify the characteristic of intrusion pattern, modeling intrusion pattern and pattern matching procedure, detect DDoS attack through multi hosts, and provide basis of study model for analysing intrusion pattern, finally.

Ⅰ. 서 론

정보화와 그 역기능의 관점에서 인터넷의 발전은 정보화를 촉진 시켰으며, 이는 해커의 활동 무대의 확산을 가져오는 그 역기능을 초래하였다. 해커의 공격을 방어하기 위한 침입차단 시스템의 개발 이후에 좀더 능동적인 방어를 위해서 침입탐지시스템의 개발이 진행되었다. 현재는 침입자 역추적 시스템에 대한 연구가 국내외적으로 활발하게 진행되고 있다.

이러한 보안기술과 더불어 Honeypot 기술은 악의적인 해커의 침입을 탐지 및 유인하고, 그 행동양식을 파악할 뿐만 아니라 포렌식 도구로도 활용할 수 있는 유용한 기술로 발전하고 있다. 즉, 취약한 시스템을 가장해 해커를 유인하여 그들이 입력한 모든 내용과 그들이 사용한 도구, 활동내역, 심지어는 온라인 채팅내용까지 조사하는 등의 방법으로 해커들이 어떤 식으로 공격대상을 정하는지, 어느 정도의 실력을 갖추었는지, 어떤 종류의 공격방법을 즐겨 사용하는지, 그리고 어떻게 감쪽같이

* 대검찰청 사무관, ** 영동대학 사무자동화과, *** 충북대학교 BK21 우수조교
**** 충북대학교 컴퓨터정보통신연구소(yhcho@chungbuk.ac.kr)
논문번호 : #020304-0716, 접수일자 : 2002년 10월 5일

흔적을 남기지 않고 컴퓨터 망에 자리를 잡는지를 파악하여 이를 추적하는 기법이다.[1,2,3,4]

본 논문에서는 침입탐지방법에서 사용된 CPN을 이용한 Honeypot 모델을 설계하고 이에 대한 성능을 평가하였다[5,6,7]. 기존의 모델로 사용된 것은 Denning 모델과 Shieh 모델이다.[6]

제안된 CPN을 이용한 Honeypot 모델은 침입패턴의 특성에 대한 분류 가능, 침입패턴의 모델링, 매칭과정의 모델링이 가능, 다중 호스트를 통한 DoS 공격의 탐지 가능, 침입패턴의 정형화 형태의 기반을 제공, 그리고 마지막으로 침입패턴의 분석을 위한 학습 모델의 기반을 제공하는 것이 가능하다.

제 2장에서는 기존의 침입탐지모델에 대한 기술 분석을 하였으며, 제 3장에서는 본 논문에서 제안한 CPN을 이용한 Honeypot 모델에 관한 상세 기술을 기술하였다. 제 4장에서는 실험 및 결과를 통해 성능평가를 하였으며, 제 5장의 결론으로 구성되어 있다.

Ⅱ. 기존의 침입탐지모델과 제안된 Honypot 모델과의 비교 분석

침입 탐지 모델은 침입 탐지 시스템을 추상화하여 설명하기 위해 사용되며, 침입 탐지 모델의 종류는 Denning, Shieh, Kumar 등의 모델이 있다. 대부분의 침입 탐지 시스템은 침입 탐지 모델을 통하여 입력의 타입이나 시스템의 상태, 특정 침입 패턴의 탐지 등을 나타낸다.[6]

Denning의 모델은 가장 처음 제안된 모델로 거의 모든 침입 탐지 시스템을 나타낼 수 있고, 호스트 레벨의 감사 데이터만 표현하는 것이 가능하다.

Shieh의 모델은 호스트 레벨에서 주체와 객체간의 오퍼레이션에 중점을 둔 침입 탐지 모델로 간단하게 read, write, drive, control의 오퍼레이션으로만 시스템을 모델화한다.

Kumar의 모델은 침입 탐지 모델이기보다는 침입을 표현하는 모델이라는 말이 더 적합하다. 단순히 침입 패턴을 표현하는 모델만을 제공하고 그 이외에 이 패턴을 수정하거나 새로운 패턴을 추가하는 면에서는 연구된 적이 없다.

표 2.1은 제안한 Honeypot 모델인 다중 침입 탐지 모델과 기존의 Denning, Shieh, Kumar의 모델을 각각의 모델이 지원하는 입력 이벤트의 종류와 탐지 가능한 패턴의 종류, 그리고 침입 탐지 시스템에서 모델링 가능한 부분을 선정하여 비교하였다.

Honeypot 모델과 Kumar 모델이 audit 이벤트와 네트워크 패킷을 입력 이벤트로 표현이 가능하기 때문에 호스트나 네트워크 레벨의 침입 탐지 시스템의 모델링이 가능하고, Denning과 Shieh 모델은 audit 이벤트만 표현이 가능하므로 호스트 레벨의 침입 탐지 시스템만 모델링을 한다는 것을 나타낸다. Honeypot 모델은 순차 침입 이외에도 다중 침입의 탐지가 가능한 데 비해 다른 모델들은 순차 침입의 탐지만 가능하기 때문에 다중 침입을 탐지하는 네트워크 레벨의 침입 탐지 시스템이 존재하는 경우 이 시스템을 모델링 하기 어렵다. Honeypot 모델과 Denning 모델은 침입 탐지 시스템의 모든 부분을 모델링할 수 있지만 Shieh 모델은 보호 집합과 흐름 정책을 통하여 모델링하기 때문에 행위 프로파일이나 규칙 집합과 같은 부분을 계속 수정하거나 추가하는 부분이 없고, Kumar 모델은 단순히 침입 패턴만을 모델링하기 때문에 다른 부분을 표현하기 어렵다.

표 2-1 제안된 모델과 기존의 모델과의 비교

모델명 / 비교항목	Honey Pot 모델	Denning 모델	Shieh 모델	Kumar 모델
입력 이벤트	audit 이벤트,	audit 이벤트	audit 이벤트	audit 이벤트, 네트워크 패킷
탐지 패턴	순차 침입, 다중 침입	순차 침입	순차 침입	순차 침입
모델 부분	침입탐지 증거확보, 역추적가능	침입탐지	침입탐지	침입 탐지

Ⅲ. CPN을 이용한 Honeypot 모델 설계

3.1 Honeypot 모델 구성도

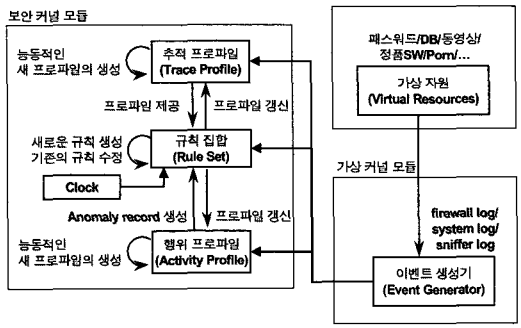

그림 3.1 Honeypot 모델

Honeypot은 크게 보안이 유지되는 보안커널 모듈(주 시스템)과 그렇지 않은 가상시스템(부속시스템) 모듈로 분류된다. 주 시스템은 콘솔과 극히 제한된 몇몇 관리자 호스트(adminhost)에서만 원격접속(remote login)을 할 수 있고, 여기에서는 시스템의 모든 자원을 사용할 수 있는 반면 부속시스템은 일반 원격 호스트에 대한 서비스를 담당한다.

일반적으로 알려진 부속 시스템은 극히 제한된 서비스만을 수행한다. 익명 ftp 서비스는 부속시스템의 가장 좋은 예이다. 이것은 사용자명 ftp의 홈디렉토리 및 하부 디렉토리만 접근하도록 권한을 제한하며 서비스에 필요한 모든 명령들은 접근이 가능한 장소에 별도로 가지고 있다.

본 가상시스템의 부속시스템 역시 서비스에 필요한 모든 명령을 접근이 가능한 장소에 별도로 저장하여야 한다. 그러나 본 유인용 부속시스템은 일반 시스템의 모든 명령을 지원해야 하므로 본 부속시스템은 일반 OS가 갖는 대부분의 파일을 필요로 한다. 또한 침입자가 불필요한 의심을 품지 않도록 하기 위해서도 일반 시스템에 있는 모든 파일들을 가지고 있어야 하므로 결국 본 가상시스템은 거의 완전한 두 개의 시스템으로 구성되어야 한다. 한편 주 시스템과 부속시스템이 특정 자원을 공유하는 경우도 있는데 이러한 공유는 필수적 공유와 선택적 공유 등 두 가지로 구분된다.

이벤트 생성기는 가상커널 모듈에 두고 해커들이 이 가상커널 모듈에 접근하여 가상시스템으로 이루어지는데 보안커널 모듈은 능동적인 새 프로파일을 생성하고 가상시스템은 주 시스템과 똑 같은 루트 파일시스템으로 구성된다.

3.2 CPN을 이용한 Honeypot 모델 설계

침입 탐지 모델은 Jensen의 CPN을 기반으로 구성되며, CPN으로 모델링 하고 세뮬레이션을 수행할 수 있도록 해주는 프로그램인 Design/CPN이라는 도구를 사용하여 모델링 한다.

CPN을 이용하여 Honeypot을 설명할 수 있는데, CPN을 이용하면 침입패턴을 효과적으로 표현할 수 있고, 토큰(token)이 컬러(color)를 가짐으로 인하여 Honeypot을 설명하는데 있어 유리하며, 컬러(color)에 있어 n개의 튜플 스트링으로 구성할 수 있으므로 n개의 변수를 가진 패턴을 표현하는데 유리하다. 따라서 침입과 역추적 패턴의 표현 및 처리가 가능하다는 이점이 있다. 또한 CPN을 이용하면 침입패턴 유형의 계층구조를 제공하여, 침입패턴의 그룹핑이 가능하고, 침입패턴의 상세한 표현이 가능하다는 장점이 있어 CPN을 이용한 Honeypot 모델을 분석한다.

3.2.1 CPN으로 표현한 Honeypot 침입탐지 모델

계층적 단계를 거치는 CPN은 가상터미널에 들어온 해커들을 유인하여 이들이 침입한 후 어떤 활동을 하였는지를 각각 침입패턴과 추적패턴을 통하여 분석한 후 이들이 어떻게 백도어 등을 설치하고 철수하기까지의 내용을 엿보다가 그때부터 침입탐지 및 추적을 하는 과정을 담고 있다.

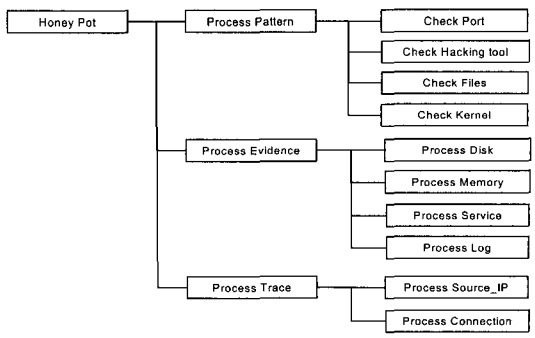

그림 3.2 Honeypot 모델의 전체구조

그림 3.2와 같이 Honeypot 모델은 크게 프로세스 유형(Process Pattern)과 프로세스 증거확보과정(Process Evidence), 프로세스 추적과정(Process Trace)으로 나눌 수 있고, 각각은

다시 상세하게 나뉘어진다.

가. 침입여부 판정모듈(Process Pattern Module)

침입여부 판정모듈은 부정행위자가 침입을 하였을 경우 이를 각 단계별로 포트, 해킹툴, 파일, 커널 등에 대하여 DB 등에 입력시킨 자료에 의하거나 스스로 부정행위로 판정된 침입에 대하여는 이를 파악하여 판정하는 모듈 이다.

1) 포트 체크 모듈(Check Port Module)

포트 체크 모듈은 포트를 체크하여 잘못 들어온 포트에 대하여서는 이들을 제어하도록 한 모듈이다. 포트 체크 모듈에는 inetd.conf와 Established port를 체크하는 등 비정상적으로 침입한 부정행위자를 감시하는 것을 표현한다.

2) 해킹 툴 체크 모듈(Check Hacking Tool Module)

해킹 툴 체크모듈은 rootkit에 대한 Check, Scan_tool을 Check 하는 것과 bof_tool을 Check 하는 것으로 나누어 체크할 수 있다.

3) 파일 체크 모듈(Check File Module)

파일 체크 모듈은 파일을 체크하여 이를 통한 부정행위자의 유입을 탐지하는 모듈로서 파일 체크모듈은 시스템에 들어있는 파일들을 체크하는 System_file과 인터넷을 통하여 침입하였을 경우 계정에 대한 침입여부를 판정할 수 있는 Account_file이 있다.

4) 커널 체크 모듈(Check Kernel Module)

커널 체크 모듈은 커널에 대해서도 체크를 하여 이들에 대한 침입여부까지도 판정한다. 커널 체크 모듈에는 lsmod를 체크하는 경우와 /proc를 체크하는 경우가 있다.

나. 증거확보 모듈(Process Evidence)

증거확보 모듈은 부정행위자를 추적하는 것 뿐만 아니라 부정행위자들을 추적하였을 경우 이들의 처벌을 위한 증거를 확보하는 차원이기도 하고, 컴퓨터수사에 있어서는 기본적인 문제를 해결하는 차원에서 중요하다.

증거확보 모듈에는 5개 과정 즉, Classify Evidence, Process Disk, Process Memory, Process Service, Process Log 과정이 있다.

1) 디스크 이미지 증거확보 모듈(Process Disk Module)

디스크 이미지 증거확보 모듈은 그림 3.9와 같이 커널까지 거쳐 나타난 결과를 증거로 확보하도록 일정한 유형을 가지고 부정행위자를 판정한다.

2) 메모리 증거확보 모듈(Process Memory Module)

메모리 증거확보 모듈은 역시 디스크 이미지와 유사하게 부정행위자에 대한 메모리 이미지에 대한 증거확보이다.

3) 서비스 증거확보 모듈(Process Service Module)

4) 로그파일 증거확보 모듈(Process Log Module)

로그파일 증거확보 모듈은 증거까지 확보된 후 utmp, wtmo, loging, sulog, history 등 여러 단계를 거쳐 나타나는 행위로 부정행위자를 역추적하는 방법이다. 이는 수사 목적상 가장 중요한 증거확보로 수사에 있어서 침입시간, 침입내용, 침입 후 설치된 파일 내지 프로그램 등을 로그파일에서 찾을 수 있다.

다. 침입자 추적 모듈(Process Trace)

침입자 추적 모듈은 증거확보가 된 후에도 좋고 증거확보가 되는 과정에서도 가능한 방법으로 출발지 IP 주소를 체크하는 과정인 IP주소 추적모듈(ProcessSource_IP)과 추가 커넥션의 존재여부를 체크하는 과정인 커넥션 추적모듈(Process Connection)로 나눌 수 있다.

1) IP 주소 추적모듈(Process Source_IP)

IP 주소 추적모듈은 증거확보과정 후 이를 추적하는 것으로 IP를 가지고 실질적인 주소를 추적하여 부정행위자를 체포하는 과정으로 특히 수사기관에서 현재 많이 사용하는 방법이다.

2) 커넥션 추적모듈(Process Connection)

커넥션 추적모듈은 IP를 가지고 로그파일이나 IP 등에 연결된 사항을 파악하는 방법이다. 커넥션 추적모듈은 패턴이 확보된 후 어떤 처리과정을 거쳐 나가는 지를 추적하여 Source가

어디에 있는지를 연결과정을 파악하여 추적하는 것이다.

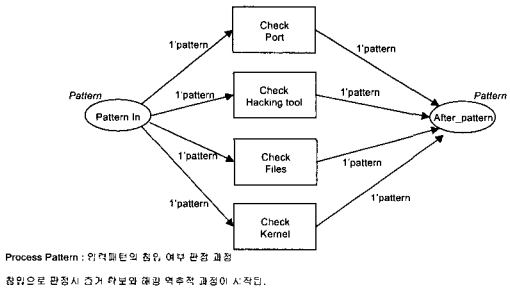

그림 3.3 침입여부 판정(Process Pattern)

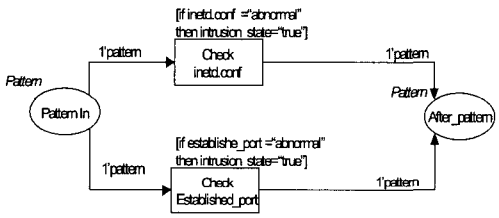

그림 3.4 포트 체크 모듈(Check Port)

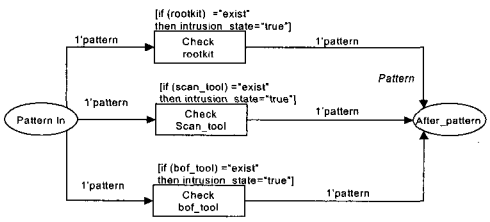

그림 3.5 해킹툴 체크 모듈(Check Hacking Tool)

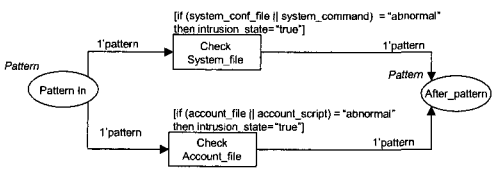

그림 3.6 파일 체크 모듈 (Check File)

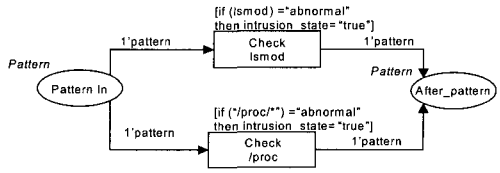

그림 3.7 커널 체크 모듈 (Check Kernel)

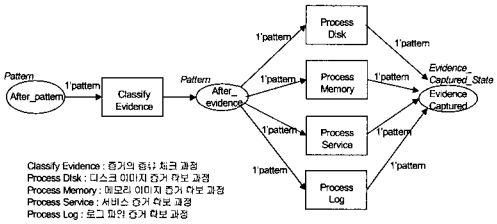

그림 3.8 증거확보 모듈(Process Evidence)

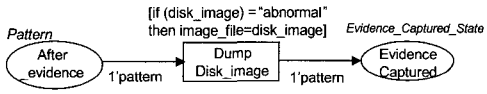

그림 3.9 디스크 이미지 증거 확보 모듈 (Process Disk)

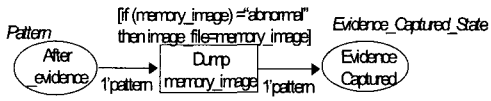

그림 3.10 메모리 이미지 증거확보 모듈(Process Memory)

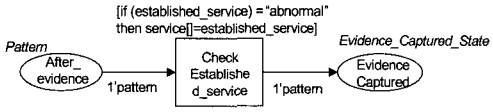

그림 3.11 서비스 증거확보 모듈(Process Service)

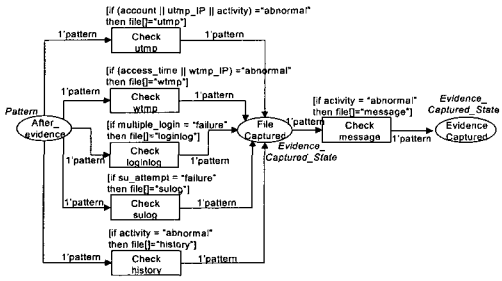

그림 3.12 로그파일 증거확보 모듈(Process Log)

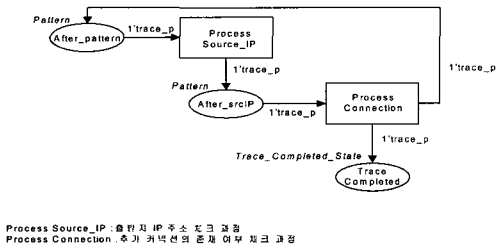

그림 3.13 침입자 추적 모듈(Process Trace)

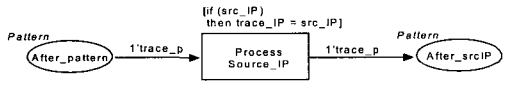

그림 3.14 IP 주소 추적 모듈(Process Source IP)

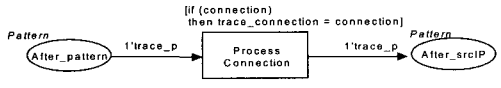

그림 3.15 커넥션 추적 모듈(Process Connection)

Ⅳ. 실험 및 결과 분석

4.1 제안된 CPN을 이용한 Honeypot 시스템 분석

제안한 CPN을 이용한 Honeypot 모델의 침입탐지는 주로 DDoS (Distribute Denial of Service) 즉, 분산서비스 거부공격을 중심으로 분석한다. 분산환경에서의 서비스 거부공격은 엄밀히 말해서 해킹이라고 할 수는 없겠으나 그 피해내용 면에서 훨씬 강력한 부분이 많다. 이 공격을 받으면 그것을 추적하여 처리하는 것이 현재기술로는 어렵기 때문이다. DDoS 공격방법은 해커가 일시에 중간에 있는 숙주 PC를 통하여 많은 내용을 보내어 상대방 서버를 Down 시키거나 작동을 어렵게 함으로써 상대방에게 많은 지장을 초래할 수 있다.

최근 발견되고 있는 분산환경에서의 서비스 거부 공격도구는 trinoo와 가장 최근형태인 TFN(Tribe Flood Network)가 대표적이다. 이중에서 TFN을 중심으로 다음과 같이 분석하였다.

이들의 공격단계는 중간 마스터 호스트의 root 권한 획득과 DDoS 데몬설치 및 작동하는 단계인 준비단계, 패킷의 크기, spoof mask 설정(spoof mask는 패킷의 src ip 주소를 임의의 다른 ip 주소로 변경하는 것을 말함)등 설정단계, 다양한 형태의 공격, udp flooding, syn flooding, icmp packet flooding, icmp smurf 공격단계, 일정한 공격을 한 후의 공격중지 단계인 것이 대부분이다. [8,9,10]

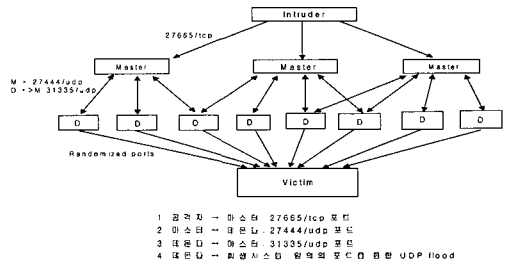

그림 4.1 분산서비스 환경에서의 서비스 거부공격 형태

TFN은 분산도구로 많은 소스에서 하나 혹은 여러 개의 목표 시스템에 대해 서비스 거부 공격을 수행한다. UDP Flood 공격을 할 수 있을 뿐만 아니라 TCP SYN Flood 공격, ICMP echo 요청공격, ICMP 브로드캐스트 공격(smurf 공격)을 할 수도 있다. 공격자는 마스터에 대하여 일정한 루트(root)권한을 획득한 다음 이를 매개로 하여 공격하고자 하는 데몬들

을 공격하여 또다시 루트권한을 획득한 후 최종 피해자에게 공격을 하기 위해 대기하다 최종 피해자를 공격한 후 빠져나간다.

이들과 관련한 포트별 내용은 그림 4-1과 같이 구성된다.

4.2 성능평가 및 결과분석

침입 탐지 모델은 CPN에 의해 모델링된다. CPN은 다음과 같이 세 가지 방법에 의해 분석된다. 첫 번째 방법은 시뮬레이션으로 프로그램 디버깅이나 실행과 유사하다. 이 방법은 시스템 모델의 행위에 대한 통계치를 통해 CPN을 실행시키는 것을 의미한다. 두 번째 방법은 사건 발생 그래프(occurrence graph)를 사용하는 것으로 이 방법은 상태 공간(state spaces)이나 도달성 그래프(reachability graph)라고도 불린다. 도달성 그래프에 대한 기본적인 개념은 노드가 도달 가능한 시스템 상태를 나타내고 아크가 변경 가능한 상태를 나타내는 방향성 그래프를 구축하는 것이다. 이 그래프를 이용하여 모델의 완벽한 분석이 가능하다. 세 번째 방법은 place invariant를 사용하는 것으로 정상적인 프로그램 검증에서 원형을 사용하는 것과 유사하다. 사용자는 모든 도달 가능한 시스템 상태의 만족을 증명하는 방정식들의 집합을 만든다. 이 방정식들은 시스템 모델의 속성을 증명하는 데 사용된다.

본 논문에서는 첫 번째의 방법인 시뮬레이션을 하고, 여러 가지 시스템 모델의 행위에 대한 통계치를 통해 CPN을 실행시키는 것을 의미한다. 이와같은 결과는 미리 시뮬레이션을 할 대상을 정하고 측정할 가지 수를 정한다음 컴퓨터를 이용하여 분석하는 것으로 아래와 같이 실험하였으며, 결과를 보면 다음과 같다.

제안한 Honeypot 모델에 대한 성능을 평가하기 위하여 여러 가지 조건을 설정하여 시뮬레이션을 실시하였다. 제안한 Honeypot 모델에 대한 성능 평가를 위하여 Linux 기반에서 Design/CPN을 이용하여 분석하였다.

또한 최근에 나타난 해킹의 방법인 DDoS의 환경에서 침입탐지 및 역추적과 관련하여 분산서비스 거부(DDoS)의 시나리오를 만든 다음 다음과 같이 다섯 가지 방법으로 분석하였다. 공격대상 호스트에서 udp 패킷을 이용한 탐지 및 추적, 데몬에서 특정포트를 이용한 탐지 및 추적, 셋째 데몬에서 특정스트링을 이용한 탐지 및 추적, 마스터에서 특정포트를 이용한 탐지 및 추적, 마지막으로 마스터에서 특정파일을 이용한 탐지 및 추적을 실시하였다.

표 4.2 Design/CPN을 이용한 실험결과

구분	마스터	데몬	공격대상 호스트	공격 내용	udp 패킷	특정 스트링	특정 포트 (데몬)	특정 포트 (마스터)	합계
탐지 #1	2	4	7	check inetd.conf	1		1	1	3
탐지 #2	5	6	13	check established port		1			1
탐지 #3	8	7		classify evidence	1	1	1	1	4
탐지 #4	11	9		dump memory_image		1			1
탐지 #5		10		check utmp	1		1	1	3
탐지 #6		12		check message	1		1	1	3
				process source_ip	1	1	1	1	4
				process connection	1	1	1	1	4
	마스터	데몬	공격대상 호스트						
탐지 횟수	4	6	2						

4.2.1 분산서비스 거부(DDoS) 공격의 시나리오

분산서비스 거부공격의 형태는 그림 4.1을 활용하여 실험한다.

① 서버해킹
② 서버에 마스터설치
③ 클라이언트 해킹
④ 클라이언트에 데몬 설치
⑤ 공격자가 마스터에게 초기명령 전송 (27665/tcp)
⑥ 마스터가 데몬들에게 초기명령 전송(27444/u에)
⑦ 데몬들은 마스터에게 응답(*HELLO*)전송 (31335/u에)
⑧ 마스터는 데몬들의 목록 저장
⑨ 마스터는 브로드캐스트 주소로 데몬들의 요청요구
⑩ 데몬 들은 마스터에게 요청응답(*PONG*)전송
⑪ 공격자가 마스터에게 공격명령 전송 (27665/tcp)
⑫ 마스터가 데몬들에게 공격명령 전송(27444/u에)
⑬ 데몬들이 공격대상 호스트에 udp flooding 공격시작(Random 포트)

4.2.2 침입에 대한 탐지 및 추적 실험

1. 공격대상 호스트에서 udp 패킷을 통한 DDoS 탐지 및 추적

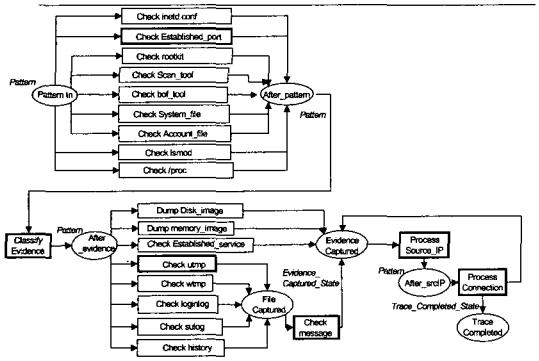

그림 4.2 udp 패킷을 이용한 탐지 및 추적

공격대상 호스트에서 N'udp 패킷을 이용하여 침입한 DDoS 공격이 있었을 경우 수많은 Check 모듈중 Check Established_port 에 체크되었을 경우, After_pattern에 의해 검증되고 이는 다시 증거를 확보하는 과정으로 이송된다. 이 과정에서 다시 utmp에서 체크되고, File Capture가 되고 이후 곧 Check Message 과정을 거쳐 증거가 확보된 후, IP를 추적하고 Process Connection 과정을 거쳐 역추적 하게 된다. 이를 통하여 udp 패킷을 통한 DDoS 탐지 및 역추적이 가능하게 된다.

2. 데몬에서 특정포트를 이용한 DDoS 탐지 및 추적

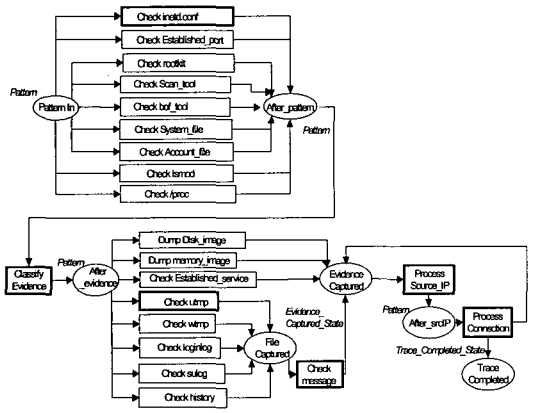

그림 4.3 데몬에서 특정포트를 이용한 탐지 및 추적

초기 데몬에서 특정포트를 27444/u에 +31335/u에 포트를 이용하여 침입한 DDoS 공격이 있었을 경우 Check 모듈중 Check Inetd Conf에서 패턴을 체크한 다음, After_pattern에 의해 검증되고 이는 다시 증거를 확보하는 과정으로 이송된다. 이과정에서 다시 utmp에서 체크되고, File Capture가 된 후 곧 Check Message 과정을 거쳐 증거확보가 된 후, IP를 추적하고 커넥션 추적(Process Connection) 과정을 거쳐 역추적 하게 된다. 이를 통하여 데몬에서 특정포트를 통하여 침입하였을 경우의 DDoS 탐지 및 역추적이 가능하게 된다.

3. 데몬에서 특정 스트링을 이용한 DDoS 탐지 및 추적

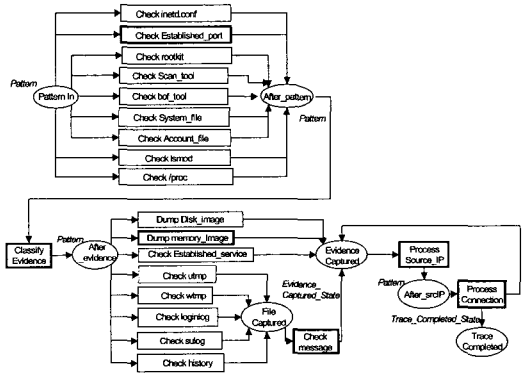

그림 4.4 데몬에서 특정 스트링을 이용한 탐지 및 추적

데몬에서 특정 스트링을 이용하여 침입한 DDoS 공격이 있었을 경우 그림 4.4 에서와 같이 체크 모듈중 Check Established_port에 체크되고, After_pattern에 의해 검증되고 이는 다시 증거를 확보하는 과정으로 이송된다. 이 과정에서 다시 Dump Memory_image를 거쳐 증거확보가 되고, 또 다른 증거확보과정인 Check Message 과정을 거친 후, IP를 추적하고 커넥션 추적(Process Connection) 과정을 거쳐 역추적 하게된다. 이를 통하여 탐지 및 역추적이 가능하게 된다.

4. 마스터에서 특정포트를 이용한 DDoS 탐지 및 추적

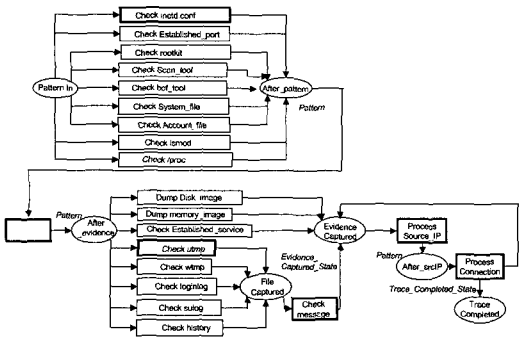

그림 4.5 마스터에서 특정포트를 이용한 탐지 및 추적

마스터에서 특정 포트 27665/tcp+27444/u를 이용하여 침입한 DDoS 공격이 있었을 경우 그림 4.5에서와 같이 Check 모듈중 Check inetd.conf에 체크되고, After_pattern에 의해 검증되고 이는 다시 증거를 확보하는 과정으로 이송된다. 이 과정에서 그 패턴을 인식하는 Check utmp를 거쳐 증거확보가 되고 Check Message 과정을 거친 후, IP를 추적하고 Process Connection 과정을 거쳐 역추적 하게 된다. 이를 통하여 탐지 및 역추적이 가능하게 된다.

5. 마스터에서 특정파일을 이용한 DDoS 탐지 및 추적

마스터에서 특정 파일을 이용하여 침입한 DDoS 공격이 있었을 경우 그림 4.5에서와 같이 Check 모듈중 Check rootkit 에 체크되고, After_pattern에 의해 검증되고 이는 다시 증거를 확보하는 과정으로 이송된다. 이 과정에서 다시 Check History 과정을 거쳐 증거확보가 되고, 파일을 capture하여 Check Message 과정을 거친 후, IP를 추적하고 Process Connection 과정을 거쳐 역추적 하게된다. 이를 통하여 탐지 및 역추적이 가능하게 된다.

4.2.3 침입탐지 실험결과

본 실험은 Linux 기반에서 Design/CPN을 이용하여 분석하였고 실험 기준은 마스터, 데몬, 공격대상 호스트를 만들고 이들로부터 4.3.2 침입에 대한 역추적 실험에서와 같은 공격내용을 정하여 실험한 결과는 표 4.2와 같다.

1. 공격단계별 각 호스트의 침입탐지 단계

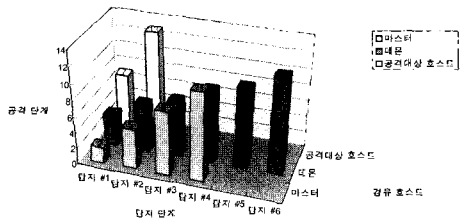

그림 4.6 DDoS 공격단계별 각 호스트의 침입탐지단계

그림 4.6은 공격단계에서 마스터, 데몬, 공격대상호스트를 기준으로 "공격단계별 각 호스트의 침입탐지단계"의 발생그래프를 나타낸다. 입력패턴에서 각각은 탐지#1에서 탐지#6의 순서대로 발생했다는 것을 알 수 있다.

DDoS 공격단계별 각 호스트의 침입탐지 단계를 분석할 수 있고, 경유 호스트별로 공격단계 분석결과 마스터에서 공격대상 호스트로 가는 동안 마스터에서 초기 탐지가 가능하며, 공격대상 호스트로 올라갈수록 초기 탐지가 불가능함을 알 수 있다.

2. DDoS공격에 대한 각 호스트의 침입탐지회수

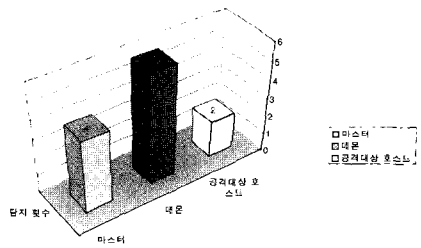

그림 4.7 DDoS 공격에대한 각 호스트의 침입탐지 회수

그림 4.7에서도 보듯이 발생그래프로 나타낸 "DDoS 공격에 대한 각 호스트의 침입탐지 회

수"를 분석하면, 경유호스트별로 침입탐지 회수는 마스트 4회, 데몬이 6회, 공격대상 호스트가 2회이다. 실험 결과 데몬에서 쉽게 침입탐지가 가능하다는 것이 되고, 침입탐지회수가 많다는 것은 그 만큼 많이 이용되는 것을 나타낸다. 따라서 이 단계에서 제안한 모델인 Honeypot 모델을 이용하는데 있어 그만큼 중요성이 있다고 하겠다.

3. DDoS 공격패턴별 각 호스트의 각 트랜지션의 사용회수

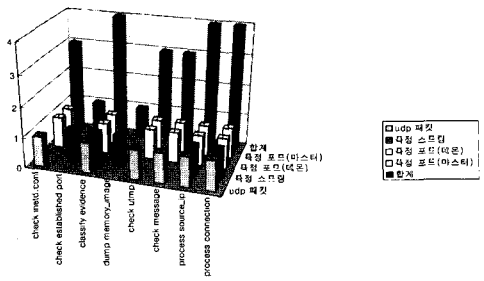

그림 4.8 DDoS 공격패턴별 각 호스트의 각 트랜지션의 사용회수

그림 4.8에서 DDoS 공격패턴별 각 호스트의 각 트랜지션의 사용회수를 살펴보면, 각 호스트별로 사용한 트랜지션의 회수를 분석한 결과 사용한 침입패턴에 따라 초기 트랜지션이 다르게 사용되고 트랜지션의 합계를 분석한 결과 꼭 필요한 트랜지션이 존재함을 알 수 있다. 또한 다양한 형태로 분석한 결과 공격패턴의 다양한 형태의 분석이 가능하고 분석한 침입경로를 통해 역추적이 가능함을 보여준다. 제안한 Honeypot 모델을 통하여 분석하면 침입패턴의 특성에 대한 분류가 가능하고, 침입패턴의 모델링, 매칭과정의 모델링이 가능하며, 다중호스트를 통한 분산서비스 거부공격의 탐지가 가능하다. 그리고 침입패턴의 정형화 형태의 기반을 제공할 수 있으며 침입패턴의 분석을 위한 학습모델의 기반을 제공하는 것이 가능하다.

V. 결 론

침입자에 대한 탐지는 컴퓨터 자원의 무결성이나 비밀성, 이용성 등을 저해하는 행위들을 탐지하는 기술로서 인터넷의 급격한 성장으로 정보의 공유가 활발해지면서 역으로 그 역기능이 심각해지고 있는 상황에서 호스트나 네트워크의 보안을 위해 가장 필요로 하는 기술이지만 안전성이나 성능을 평가할 만한 방법이 부족한 상태이다.

기존의 모델은 실질적으로 해커들이 침입한 후 그 로그파일을 지우거나 backdoor를 설치하는 경우 이에 대한 탐지가 쉽지 않았고, 특히 여러 개의 호스트들을 통하여 침입하는 다중침입의 경우에 대해서는 탐지가 어렵거나 불가능하여 침입 패턴을 구체화하는 등 그 나름대로 추적하는 기술을 향상시키기 위하여 많은 노력을 하고 있으나 아직도 서론에서도 언급하였듯이 많은 해킹 등 컴퓨터범죄가 발생하고 있는 실정이다. 이에 대한 대안의 하나로 최근부터 논의 되어온 유인을 통하여 범죄를 근절하고자 하는 노력의 일환으로 Honeypot을 제안하였다.

본 논문에서는 침입자를 유도하는 침입유형을 정의하고 이에 대한 침입패턴의 특성에 따른 분류와 침입 패턴의 모델링 및 매칭과정의 모델링이 가능함을 설명하였으며, 다중 호스트를 통한 다중침입의 탐지가 가능하다는 것을 실험을 통해 증명하였다. 또한 침입 패턴의 정형화형태의 기반 및 침입패턴의 기계적 학습을 위한 학습모델의 기반을 제공하였다. 그리고 부정행위자인 침입자가 Honeypot이 시스템에 있는지 없는지를 알 수 없도록 하는 모델을 제안하였으며 Denning, Shi도, Kummar 모델과 비교 분석하였다.

침입탐지와 관련하여 기존 연구를 참조, 침입탐지 모델에 근거한 침입기법을 이용한 추적시스템, 침입자 추적요구사항, 침입자 역추적모듈, 역추적 미행시나리오, 역추적 시스템에 대하여 설명하였다.

그리고 제안한 모델인 Honeypot 모델을 설계하는데 있어 모델의 기본요소를 살펴보고, Honeypot 모델의 구성도를 작성하여 이를 CPN을 이용하여 설계와 분석을 하였고, 최근에 많이 발생하는 해킹 중 DDoS공격을 어떻게 역추적 하는지를 호스트에서 udp 패킷을 통하거나, 데몬에서 특정포트를 이용한 경우, 데몬에서 특정스트링을 이용한 경우, 마스터에서 특정포트를 이용한 경우, 마스터에서 특정파일을 이용한 경우 등을 실험을 통하여 탐지 및 추적하

는 과정을 구성하였다.

실험과정에서 Honeypot 모델을 통하여 DDoS를 분석한 결과 장점은 첫째 침입패턴의 특성에 대한 분류가 가능하고, 둘째 침입패턴의 모델링, 매칭과정의 모델링이 가능하며, 셋째 다중호스트를 통한 분산서비스 거부공격의 탐지가 가능하고, 넷째 침입패턴의 정형화 형태의 기반을 제공할 수 있으며, 다섯째 침입패턴의 분석을 위한 학습모델의 기반을 제공하는 것이 가능하다.

향후 연구 방향은 새로운 침입 패턴을 추가하는 방안과 실시간 침입 탐지를 수행하는 모델, 침입 패턴의 기계학습을 수행하는 발전모델, 비정상적인 패턴, 오용 침입 패턴과 침입 패턴 분류 계층의 상관관계에 대한 연구 및 모델을 가지고 실제 상황에서 활용될 수 있는 기본적인 연구가 계속 진행되어야 할 것이다.

참 고 문 헌

[1] 서상용,정부홍,장성형 "부정행위자유인을 위한 유틸리티기반 가상시스템개발", 한국정보보호센터, 1999

[2] 송주석,강명호,김수형,조형재 "전산망 시스템 침입자 역추적 기본시스템 개발", 한국정보보호센터, 1997.

[3] 박정현 "최근 해킹바이러스 동향 및 대응 기술", 한국정보보호센터 제 7회 정보통신망 정보보호워크숍, 2001. 4.

[4] 마복순,정현진,송성훈,임채호,원유헌 "네트워크보안관리를 위한 이동 에이전트 설계", 정보과학회 학술발표 논문집, 제 25권 2호, 1998.10.

[5] Kurt Jensen, Colored Petri nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1: Basic Concepts, EATCS monographs on Theoretical Computer Science, Springer-Verlag, 1992.

[6] Sandeep Kumar, Classification and Detection of Computer Intrusions, Ph.D Thesis, Purdue University, August 1995.

[7] Analysis of CP-nets, http://www.daimi.aau.edu/CPnets/intro/analysis.htm, December 1997.

[8] Simson Garfindel & Gene Spafford, "Practical UNIX & internet Security", 2nd ED, O'Reilly & Associates, Inc. 1996.

[9] W. Richard Stevens, "Advanced Programming in the UNIX Enviroment", Addison Wesley, 1 992.

[10] Common Vulnerabilities and Exposures, unified hack database http"\\cve.mitre.org

현병기(Byoung-Ki Hyun) 정회원

1998년: 중앙대학교 경영정보과 (공학석사)
2001년: 충북대학교 컴퓨터공학과 (공학박사)
현재: 대검찰청 사무관

<주관심분야> 컴퓨터 보안

구경옥(Kyung-Ok Gu) 정회원

1998년: 한양대학교 대학원(공학석사)
1986년~1993년: 정부전산정보관리소 전산주사
현재: 영동대학 사무자동화과 조교수

<주관심분야> 멀티미디어통신 서비스

조도은(Do-Eun Cho) 정회원

2001년: 세명대학교 교육대학원 교육학석사
현재: 충북대학교 컴퓨터공학과 박사과정

<주관심분야> 정보 보안

조용환(Yong-Hwan Cho) 정회원

한국통신학회 논문지 제 23권 9호 참조
현재: 충북대학교 전기전자 컴퓨터공학부 교수

<주관심분야> .Net Framework, 멀티미디어통신, 트래픽공학, Mobile PKI, 정보통신정책