제20회 한국정보처리학회 추계학술발표대회 논문집 제10권 제2호 (2003. 11)

액티브 네트워크 기반의 분산 취약성 분석 모델에 관한 연구

한영주*, 양진석*, 김희승**, 김현구*, 장범환***, 정태명*
*성균관대학교 전기전자 및 컴퓨터 공학과
**성균관대학교 컴퓨터 공학과
*** 한국전자통신연구원 정보보호연구본부 능동보안기술연구팀
e-mail : {yjhan, jsyang, hskim, hkkim }@imtl.skku.ac.kr,
bhchang@etri.re.kr, and tmchung@ece.skku.ac.kr

A Study on the Distributed Vulnerability Analysis Model based on Active Networks

Young-ju Han *, Jin-seok Yang*, Hee-seung Kim**, Hyun-ku Kim*,
Beom-Hwan Chang*** and Tai-myoung Chung*
*Dept. of Electrical and Computer Engineering, Sungkyunkwan University
**Dept. of Computer Engineering, Sungkyunkwan University
** *Network Security Dept., Information Security Technology Div., ETRI

요 약

액티브 네트워크는 프로그램을 담은 액티브 패킷을 사용하여 네트워크에 프로그램이 가능하도록 함으로써 네트워크와 서비스에 유연성을 제공하는 새로운 접근방법이다. 그러나 액티브 패킷의 실행 능력은 액티브 노드에 새로운 취약성을 생성하며, 이러한 취약성을 이용한 공격은 액티브 패킷의 이동성을 이용하여 네트워크 전역에 쉽게 전파될 수 있다. 이러한 공격을 미리 방지하기 위해서는 기존의 취약성 분석 모델보다 향상된 모델이 필요하다. 본 논문에서는 액티브 네트워크 기반에서의 취약성 분석 모델의 요구사항을 기술하고, 요구사항을 수용할 수 있는 분산 취약성 분석 모델에 대해서 기술하고자 한다. 분산 취약성 분석 모델은 네트워크에 뛰어난 확장 적응성을 제공할 것이다.

1. 서론

네트워크 기술이 발전함에 따라 네트워크를 기반으로 하는 다양한 어플리케이션과 서비스들이 증가하고 있다. 그러나, 현재 네트워크의 기반구조는 증가하는 네트워크 서비스들을 발빠르게 수용하기에 한계를 가지고 있다. 이러한 문제를 해결하기 위해 제안된 개념이 액티브 네트워크이며, 현재 액티브 네트워크에 대한 다양한 연구가 진행 중이다[1,2].

액티브 네트워크는 네트워크 상의 중간 노드들이 자신을 통과하는 패킷들을 실행할 수 있는 기반구조를 제공한다. 사용자는 이러한 기반구조를 이용하여 최적화된 서비스들을 좀 더 빠르게 배포할 수 있다. 반면, 중간 노드에서의 프로그램 실행은 현존하는 네트워크 기반 구조에 비해 더 많은 보안 위협 요소를 내포한다. 게다가 프로그램을 담은 액티브 패킷의 이동성은 이러한 보안 위협을 네트워크 전역으로 쉽고 빠르게 전파시킬 수 있다. 대부분의 공격은 네트워크나 시스템의 취약성을 검사하는 것부터 시작된다. 취약성 검사는 보안 위협을 미리 알아내 위협 요소를 제거할 수 있도록 하는 유용한 방법이다. 현재 많은 취약성 분석 도구가 개발되어 사용되고 있으나, 액티브 네트워크에 적용하기에는 문제점을 가지고 있다.

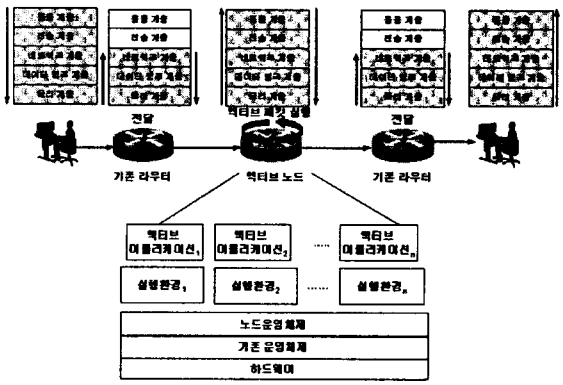
본 논문에서는 액티브 네트워크 상에서 적용 가능한 취약성 분석 모델을 제시하고자 한다. 2장에서는

액티브 네트워크 개념 및 보안 위협과 기존의 취약성 분석 모델에 대하여 알아보고 3장에서는 액티브 네트워크 기반의 취약성 분석 모델의 요구사항을 기술한다. 4,5장에서는 본 논문에서 제안하는 분산 취약성 분석 모델의 개념, 구조 및 프로세싱 과정에 대해 기술한 후 마지막으로 6장에서 결론을 내린다.

2. 연구 배경

2.1 액티브 네트워크

액티브 네트워크는 중간 노드에서 응용 계층까지 조작할 수 있는 네트워크를 말한다. 즉, 중간 노드를 실행 가능하게 만들어 기존 중간 노드들의 "저장-전달" 기능이 아닌 "저장-처리-전달"의 기능을 갖게 하여 유연하고 동적인 네트워크 구조를 제공한다[1]. 이러한 기능을 가지는 중간 노드를 "액티브 노드"라고 하며, 프로그램을 실어 전송할 수 있는 패킷을 "액티브 패킷"이라고 한다. [그림 1]은 액티브 네트워크의 개념을 나타낸다. 액티브 노드는 들어오는 패킷을 수신하여 실행 여부를 결정하고 실행 시에는 실행환경으로 액티브 코드를 전달하여 실행한다. 액티브 노드는 실행 결과에 따라 수신한 패킷을 그대로 보내거나 새로운 코드를 가진 패킷을 생성하여 다음 노드로 전달한다.



[그림 1] 액티브 네트워크의 개념

액티브 노드는 노드운영체제, 실행환경 그리고 액티브 어플리케이션으로 구성된다[4]. 노드운영체제는 패킷 스케쥴링, 자원 관리, 패킷 구분 등을 제공하며, 실행환경은 액티브 패킷이 실행될 수 있는 환경을 제공한다. 마지막으로 액티브 어플리케이션은 특정 실행환경 위에서 동작하는 어플리케이션을 의미한다.

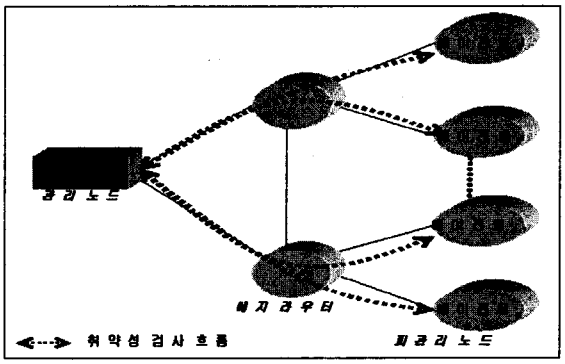
액티브 패킷은 네트워크에 유연성과 확장성을 제공하는 잇점이 있으나, 다음과 같은 보안상의 문제점을 가진다[5].

- 액티브 패킷에 의한 액티브 노드 및 네트워크 자원 그리고 다른 액티브 패킷의 오용
- 액티브 노드에 의한 액티브 패킷의 오용

이러한 문제점들은 서비스 거부(Denial of Service), 손상(Damage)등 다양하고 복합적인 공격을 발생시킬 수 있다.

2.2 기존 취약성 분석 모델에 대한 고찰

현재 사용되고 있는 취약성 분석모델은 크게 호스트 기반 취약성 검사와 네트워크 기반 취약성 검사로 나뉜다. 호스트 기반 취약성 검사의 경우, 검사 호스트의 운영체제, 특정 서비스, 그리고 구성 파일 등을 검사하여 검사 호스트의 저수준(low-level)으로의 직접 접근을 통해 해당 호스트의 취약성 존재 유무를 검사할 수 있으며, 네트워크 기반 취약성 검사의 경우는 보호하고자 하는 도메인 내에 있는 방화벽이나 웹서버등과 같이 중요한 시스템들의 취약성을 원격에서 탐지하여 외부의 침입자로부터 네트워크 상의 구성 요소들을 보호할 수 있는 검사 모델이다[3]. 본 논문에서는 우리가 제안하는 모델의 비교 모델로써 원격 취약성 분석 모델을 주요하게 다룬다. 일반적인 취약성 분석 모델은 개념적으로 "정책설정->목적지 노드 설정->데이터 수집->데이터 추론->보고"의 절차를 거친다[3].



[그림 2] 기존의 취약성 분석 모델

기존의 취약성 분석 모델은 [그림 2]에서와 같이 보통 서버-클라이언트 구조로써 중앙에 정책 설정, 목적지 노드 설정, 취약성 분석 및 보고를 담당하는 매니저가 있고, 피관리 노드 상의 에이전트들은 매니저로부터 요청을 받아 취약성 검사를 수행하고 수행 결과를 매니저에 보고하는 중앙 집중화된 모델을 가진다. 이러한 중앙 집중식 취약성 분석 모델은 보통 주요 정보를 단일 지점에서 관리하므로 설치, 구현, 백업, 갱신, 유지 보수 등이 용이한 반면, "단일 지점 실패(single point of failure)"가 될 가능성이 높다. 이 외의 중요하게 다루어져야 할 중앙 집중식 취약성 분석 모델의 단점은 다음과 같다.

- ***더딘 적응성*** : 새로이 발견된 취약성에 대한 적용을 위해서는 취약성 분석 시스템의 갱신이 필요하게 된다. 이러한 과정은 관리자에 의해 수동적으로 이루어지기 때문에 더딜 수 밖에 없다.
- ***약한 확장적응성*** : 관리 도메인 내에 피관리 노드를 추가하게 되면 추가적인 관리자에 의한 추가적인 에이전트의 설치가 필요하게 되며, 피관리 노드의 수가 증가할수록 매니저가 관리 및 처리해야 할 정보가 많아지기 때문에 병목현상 및 과부하 등의 문제가 발생할 수 있다.

3. 액티브 네트워크 기반의 취약성 분석 모델의 요구사항

액티브 네트워크 기반의 취약성 분석 모델에 필요한 요구사항은 다음과 같다.

▶ *대규모 네트워크에 대한 고려*

액티브 패킷을 통한 공격의 파급력을 고려하여 대규모 네트워크 단위의 취약성 분석이 가능해야 한다.

▶ *확장성 있는 취약성 분석 모델*

전체 취약성 분석 모델의 성능에 영향을 주지 않고 피관리 노드의 추가가 용이해야 한다.

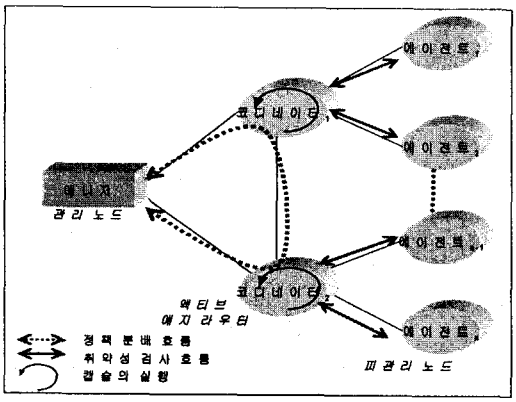
▶ *새로운 취약성에 대한 높은 적응성*

새로운 취약성에 대한 빠른 대응을 해야한다.

4. 분산 취약성 분석 모델의 개념

4.1 개념

분산 취약성 분석 모델은 현재 취약성 분석 모델의 장점은 유지하면서 단점을 극복한 모델이다. 현재 취약성 분석 모델에서는, 매니저가 정책 설정과 취약성 분석 둘 다를 수행한다. 이는 피관리 노드가 증가함에 따라 매니저의 부하를 가중시킬 수 있다. 본 분산 취약성 분석 모델의 기본 목적은 매니저의 부하를 줄이는데 있다. [그림 3]는 이러한 분산 취약성 분석 모델의 개념도를 나타낸 것이다. [그림 3]와 같이 매니저와 피관리 노드 사이에 코디네이터가 추가되었다. 매니저는 코디네이터들을 대상으로 관리 노드에 대한 정책을 설정하여 분배하며, 코디네이터는 매니저로부터 수신한 정책을 기반으로 피관리 노드에 대한 취약성 분석을 수행한다. 코디네이터는 에지 라우터에 위치한다. 즉, 매니저로 집중되었던 부하는 피관리 노드의 최근접에 위치하는 코디네이터로 분산된다.



[그림 3] 분산 취약성 모델의 개념도

4.2 캡슐의 정의

위에서 설명한 매니저에 의한 정책 분배 및 코디네이터에 의한 취약성 검사는 액티브 코드인 캡슐에 의해 수행된다. 캡슐은 데이터 패킷과 실행 코드로 구성된다. 본 모델에서 사용되는 캡슐은 다음의 두가지이다.

▶ ***vp*Capsule(정책 분배 캡슐)**

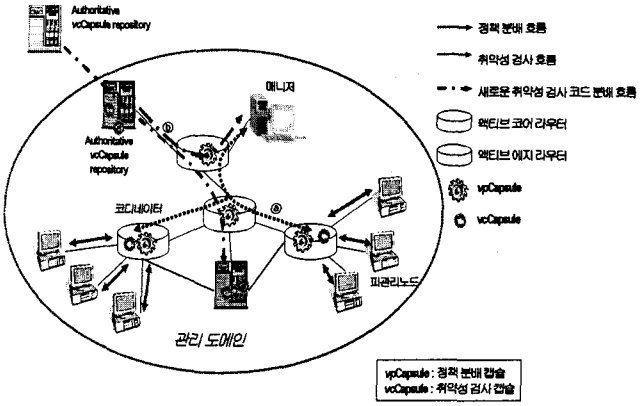
정책은 각각의 취약성을 기준으로 "*어떤 노드를 대상으로 언제 취약성 검사를 수행할지*"에 대한 정보이다. vpCapsule은 이러한 정책을 담아 각각의 코디네이터에게 분배하는데 사용되는 캡슐이다. *vp*Capsule은 매니저가 생성한 후 코디네이터에게 전송된다.

▶ ***vc*Capsule(취약성 검사 캡슐)**

*vc*Capsule은 코디네이터에서 특정 취약성을 검사할 수 있는 검사 코드를 담고 있는 캡슐이다. 이러한 코드의 종류에는 포트 스캐닝(port scanning), HTTP 취약성 검사 코드 등이 있다.

4.3 구조

본 절에서는 분산 취약성 분석 모델의 구조 및 주요 구성요소에 대해 기술하고자 한다. [그림 4]는 분산 취약성 분석 모델이 어떻게 모델링 되었는지를 보이고 있다. 본 모델은 대규모 네트워크를 고려한 관리 영역의 구분을 위해 관리 도메인을 정하고 취약성 분석 정책 적용은 이 도메인 단위로 관리된다. [그림 4]에서 점선 ⓐ는 관리 도메인내 정책 분배 흐름을 나타내며, 직선은 코디네이터에서 서브넷 단위로 이루어지는 취약성 검사 흐름을 나타낸다. 점선 ⓑ는 새로이 발견된 취약성에 대한 취약성 검사 코드의 배포 흐름을 나타낸다. 피관리 노드는 매니저에 관리 도메인 내에서 매니저에 의해 관리되는 종단 노드이다. 피관리 노드는 액티브 노드이거나 일반 노드 일 수 있다.



[그림 4] 분산 취약성 분석 모델의 구조

관리 도메인은 하나의 매니저와 하나 이상의 vcCapsule Repostory 및 코디네이터를 포함한다. 각각의 기능을 살펴보면 다음과 같다.

▶ **매니저(Manager)**

자신이 속해 있는 관리 도메인의 전체 관리에 대한 책임을 가진다. 즉, 피관리 노드, 코디네이터의 관리 및 취약성 리스트 관리를 수행한다. 그리고, 각각의 피관리 노드별 취약성 검사를 위한 정책 설정 및 각각의 피관리 노드가 속한 코디네이터로의 정책 분배를 수행한다. 이러한 정책 전달은 이동 코드인 *vp*Capsule을 통해 분배된다. 또한 코디네이터로부터 취약성 검사 결과를 수신받아 관리자에게 보고한다. 더불어 관리자가 손쉽게 매니저를 제어할 수 있도록 사용자 인터페이스를 제공한다.

▶ **코디네이터(Coordinator)**

코디네이터는 관리 도메인내에 존재하는 액티브 노드로 구성된 에지 라우터 상에서 동작한다. 코디네이터는 매니저로부터 수신한 정책에 따라 자신에게 연결되어 있는 서브넷 상의 피관리 노드들을 대상으로 실질적인 취약성 검사를 수행한다. 코디네이터의 정보는 매니저에 등록되어 관리된다.

▶ ***vc*Capsule Repository**

관리 도메인내에는 여러 개의 *vc*Capsule Repository가 존재한다. 그 중 하나가 "*Authoritative*" 권한을 가지게 된다. *vc*Capsule Repository는 취약성 검사를 수행하는 *vc*Capsule를 보관한다. 새로운 취약성에 대하여 취약성 검사 코드인 *vc*Capsule이 생성되면 맨 처음 Authoritative *vc*Capsule Repository에 등록되며, Authoritative *vc*Capsule Repository가 관리 도메인 내에 있는 *vc*Capsule Repository들로 새로 추가된 *vc*Capsule를 분배한다. 실질적인 취약성 검사가 이루어지는 코디네이터에 실행해야 할 *vc*Capsule이 자신의 캡슐 캐쉬(cache)에 존재하지 않을 경우, 가까운 *vc*Capsule Repository로부터 해당 *vc*Capsule를 수신받아 사용하게 된다.

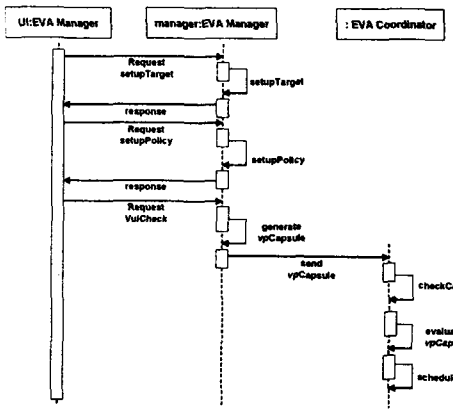
이와 같이 매니저에서 피관리 노드에 최근접한 코디네이터로의 취약성 검사 부하의 분산은 확장 적응성 측면에서 네트워크를 강하게 만든다. 또한, *vc*Capsule Repository에 의한 새로이 발견된 취약성에 대한 검사 코드의 빠른 배포는 네트워크에 향상된 적응력을 제공한다.

5. 분산 취약성 분석 모델에서의 프로세싱 과정

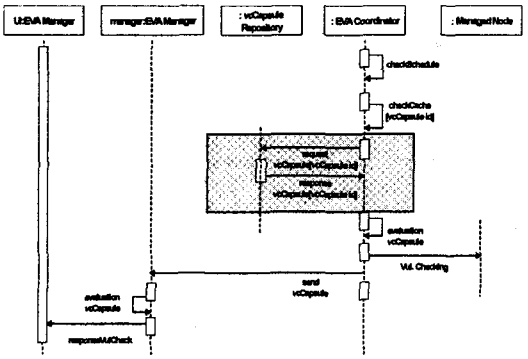
본 모델은 다음의 3가지 프로세싱 과정을 가진다.

- 정책 설정 및 분배 과정
- 취약성 검사 과정
- 새로운 취약성 분석 코드의 배포 과정

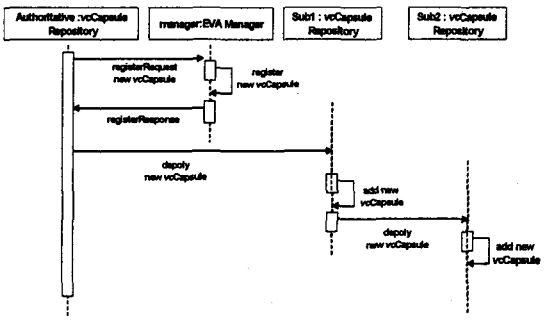
[그림 5,6,7]은 각각의 프로세싱에 대한 순차 다이어그램을 나타낸다. [그림 6]에서 회색 박스 안의 동작과정은 코디네이터가 특정 피관리 노드를 대상으로 취약성 검사를 하려고 할 때 해당 취약성 검사 코드가 자신의 캐쉬에 없을 때 *vc*Capsule Repository로부터 수신받는 과정을 나타낸다.



[그림 5] 정책 설정 및 분배에 대한 순차 다이어그램



[그림 6] 취약성 검사에 대한 순차 다이어그램



[그림 7] 새로운 취약성 분석 코드 배포에 대한 순차 다이어그램

6. 결론

본 논문에서는 액티브 네트워크 기반의 분산 취약성 분석 모델을 제시하였다. 제시한 분산 취약성 분석 모델은 기존의 클라이언트-서버 기반의 중앙 집중식 취약성 분석 모델이 가지는 한계를 극복하고 보다 향상된 확장 적응성과 새로운 취약성에 대하여 빠른 적응성을 보이는 모델이다. 따라서 대규모 네트워크에도 적용 가능할 것으로 예상된다.

현재 우리는 ANTS 실행환경을 기반으로 하는 액티브 네트워크 환경에서 분산 취약성 분석 모델을 적용한 분석 도구를 개발 중에 있으며, 앞으로 다양한 실행환경 위에서 동작할 수 있도록 확장 개발을 진행할 것이다.

참고문헌

[1] D. L. Tennenhouse, et al., "A Survey of Active Network Research", IEEE communications magazine, Jan. 1997.

[2] D. Raz, et al., "An Active Network Approach to Efficient Network Management", IWAN'99, 1999.

[3] Internet Security Systems, Network and Host-based Vulnerability Assessment, Technical White Paper.

[4] K. Calvert, et al., "Architectural Framework for Active Networks", AN Working Group, July 1999.

[5] K. Psounis, "Active Networks: Applications, Security, Safety, and Architectures", IEEE Communications Surveys, First Quarter, 1999.