유비쿼터스 컴퓨팅 시스템의 생존성 개선을 위한 정량적 분석 모델링 기법

최 창 열[†] · 김 성 수[††]

요 약

유비쿼터스 컴퓨팅 시스템은 하나 이상의 컴퓨터가 네트워크로 상호 연결된 프로세서 시스템이다. 하지만, 기존의 보안 유지 방법은 정성적 탐지 및 대응책으로 공격이 발생한 이후 대응에만 치중하여 능동적 차원의 보안 유지 방법에 대한 연구가 부족하다. 따라서, 본 논문은 정량적인 분석을 통해 범용적 인프라의 개선뿐만 아니라 특정 인프라 공격에 대해서 탐지 및 대응할 수 있는 방법에 대해 제안한다. 이를 위해 시스템의 고정적 요소 정보, 임의의 요소 정보, 공격 유형 모델링을 근간으로 시스템의 보안성을 정량적으로 분석할 수 있도록 생존성에 대한 정의 및 모델링 기법을 사용하였다. 그리고 제안한 기법의 검증을 위해 TCP-SYN 공격과 Code-Red 웜 공격에 대한 생존성 분석을 수행하였다.

키워드 : 유비쿼터스 컴퓨팅, 생존성, 정량적 보안 정책, 공격 유형 모델링

A Quantitative Assessment Modeling Technique for Survivality Improvement of Ubiquitous Computing System

Changyeol Choi[†] · Sungsoo Kim[††]

ABSTRACT

Ubiquitous computing system is about networked processors, which is constructed with one or more computers interconnected by the networks. However, traditional security solution lacks a proactive maintenance technique because of its focusing on developing the qualitative detection and countermeasure after attack. Thus, in this paper, we propose a quantitative assessment modeling technique, by which the general infrastructure can be improved and the attacks on a specific infrastructure be detected and protected. First of all, we develop the definition of survivality and modeling technique for quantitative assessment modeling with the static information on the system, random information, and attack-type modeling. In addition, the survivality analysis on TCP-SYN attack and Code-Red worm attack is performed for validating the proposed technique.

Key Words : Ubiquitous Computing, Survivality, Quantitative Security Policy, Attack-type Modeling

1. 서 론

장비의 소형화 및 처리 능력 확대에 더불어 컴퓨팅 시스템을 구축하기 위한 비용이 절감되면서, 다양한 산업 분야에서 컴퓨팅 능력을 활용하고 있다. 더욱이 유비쿼터스 컴퓨팅 환경이 도래하면 언제 어디서나 컴퓨팅 자원을 사용할 수 있게 되고, 이를 위해 도처에 컴퓨팅 장치들이 탑재되어 유·무선으로 서로 연결된 네트워크를 통해 상호작용을 한다[1, 2]. 하지만, 유비쿼터스 컴퓨팅 시스템과 같이 네트워크로 상호 연결을 통해 형성된 대규모 인프라는 악의적인 공격이나 이로 인한 시스템 결함 발생 피해가 일부 서비스 정지가 아닌 전체 시스템의 마비를 일으킬 수 있다. 이를 해결하기 위한 방법으로써, 공격 방법 분석을 근간으로 유사 공격을 차단하려는 방화벽 개발 또는 네트워크 전체적인 관점에서보다는 특정 위치에 국한된 탐지와 대응 방법과 같이 정성적인 탐지에 의한 해결 방안에 대한 연구가 활발히 진행되었다. 이와 같이 대규모 인프라 공격에 대한 기술 연구는 지난 수년 동안 집중적으로 이뤄졌으며, 최근 유비쿼터스 컴퓨팅 시스템 개발이 활발해지면서 센서 네트워크와 같은 특정 속성을 갖는 컴퓨팅 환경에서의 공격에 대한 연구도 최근 활발히 진행되고 있다. 하지만 대다수 기존 연구들은 공격 방법에 대한 원천적인 연구보다는 실제의 공격 경험을 토대로 한 탐지 및 대응 방

※ 이 논문은 2005년도 두뇌한국21사업에 의하여 지원되었음.
※ 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅 및 네트워크 원천기반기술개발 사업의 지원에 의한 것임.
† 준 회 원 : 아주대학교 대학원 박사과정
†† 종신회원 : 아주대학교 정교수
논문접수 : 2005년 3월 31일, 심사완료 : 2005년 7월 18일

법의 연구가 주류였다. 탐지 및 대응 방법에 관련된 대부분의 연구들에서는 공격 탐지와 대응 시점이 공격이 진행된 이후에 이뤄지고, 이를 위해 단시간에 수집된 소량의 패킷 정보를 이용하는 정성적인 분석을 사용하고 있다. 더욱이 네트워크 전체적인 관점에서보다는 특정 위치에 국한된 탐지와 대응 방법들이 연구되어 왔다.

지금까지 진행된 컴퓨팅 시스템의 보안성을 향상하기 위한 연구들은 대체로 세가지로 분류할 수 있다. 첫 번째는 공격 자체에 대한 분석[3, 4]이며, 두 번째는 경험을 통해 얻은 공격 특성에 맞춘 공격에 대한 실제적인 방어 방법[5, 6]이고, 마지막으로 컴퓨팅 인프라의 개선[7, 8]이다. 특히, 초기에는 주로 보안 시스템 아키텍처에 관한 연구와 공격이 발생됐는지 여부를 탐지하는데 초점을 맞춘 연구들이 이뤄졌으나, 최근에는 공격 발생 탐지 자체보다는 공격을 위한 작업 흐름 혹은 공격 패킷들을 정상 패킷과 구분하려는 관점에서의 탐지와 대응 방법에 대한 연구가 활발히 진행되고 있다. 그런데 기존 보안 유지 기법은 대부분 기밀성 유지를 위한 암호화 기법에 집중된 연구 결과나 특정 공격 기법에 대한 연구 결과와 같이 정성적인 분석에만 의존하여 탐지, 예방 및 치료하려고 하였기 때문에 이미 알려진 공격에 대해서는 효과적으로 대응할 수 있으나 아직까지 알려지지 않은 공격이나 알려진 공격의 변형에 대한 대응책은 마련하지 못하였다.

따라서, 알려진 공격이지만 그에 대한 탐지 및 대응책을 피하기 위해 공격 방법을 변형한 공격에 대해서 지속적으로 피해를 입는 사례가 발생하였다. 이와 같은 정성적인 분석 방법에만 의존하는 대응 기법의 단점을 보완하기 위해 기존의 결함허용(Fault-Tolerance) 기술과 최근의 보안 유지기술이 결합된 형태로 해당 시스템이 부분적으로 공격에 의해 손상되더라도 최소한의 필수 서비스를 지속적으로 수행하는 개념 개발에 관한 연구가 진행 중이다[9, 10]. 또한, 미국 DARPA의 OASIS(Organically Assured and Survivable Information System) 프로그램 중 HAQUIT(Hierarchical Adaptive Control of Quality of service for Intrusion Tolerance) 프로젝트[11]의 경우 사용자 성능이 25% 이상 저하되는 것을 방지하면서 일정시간 동안의 공격에 대해서 견딜 수 있는 것을 목표로 하고 있다. 이밖에, Willow 프로젝트[12]에서는 대규모 분산 시스템의 생존성(Survivability)을 지원하는 시스템을 개발한 사례가 있다. 한편, [13, 14]에서는 네트워크 기반 컴퓨팅 환경에서 시스템의 생존성을 정량적으로 정의한 기존의 다양한 방법들을 체계적으로 분석하였지만, 일반적으로 적용될 수 있는 명확한 정의가 존재하지 않는다. 또한 정량적 분석을 통해 시스템의 보안성을 평가해보려는 다수의 노력이 최근 이뤄졌지만, 사전에 정의된 네트워크 토폴로지 기반 분석[15], 응용 서비스만을 대상으로 하는 분석[16], 특정 공격에 대한 정성적인 특성만을 고려한 분석[17] 또는 특정 대상 운영체제만을 고려한 분석[18]이기 때문에 이질적인 환경에서 토폴로지 및 컴포넌트가 동적으로 변하면서 다양한 서비스를 제공해야 되는 유비쿼터스 컴퓨팅 시스템에는 적합하지 못한 분석 모델이다.

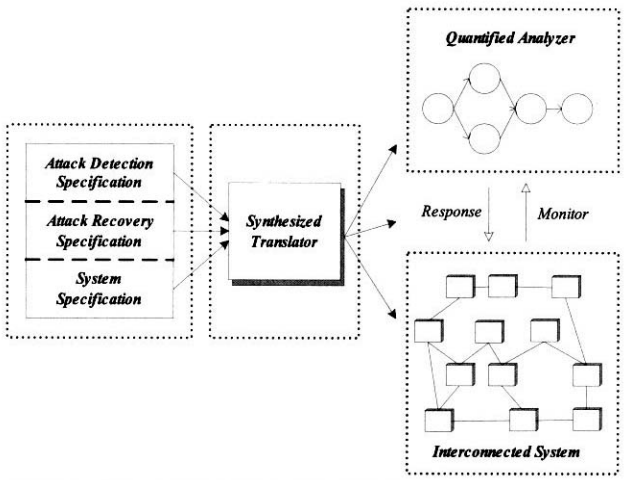
본 논문의 사전 연구 결과로 [19, 20]에서는 생존성을 시스템이 갖추어야 하는 규격이라는 관점에서 유비쿼터스 컴퓨팅 환경적 요소를 고려하여 생존성에 대한 정의를 제안하였다. 이는 변종 공격을 탐지하지 못했던 기존의 정성적인 방법을 보완하기 위한 적합한 대안이라 평가되고 있으며, 공격 도구의 설치 과정을 정상적인 통신 흐름과 구분하기 어려웠던 정성적인 탐지의 단점을 극복할 수 있는 방법으로 대두되었다. 더욱이 공격이 진행되고 있는 중에도 공격을 탐지할 수 있으므로 공격에 대한 대응 시점을 앞당길 수 있는 장점이 있으나, 구체적으로 유비쿼터스 컴퓨팅 시스템의 구성 및 동작 방법에 대한 고려는 간과된 이론적 검증만 수행한 연구결과이다. 따라서 본 논문의 목적은 위와 같은 기존 연구의 문제점을 해결하기 위한 것으로, 자동화된 악성 코드 기반 공격 속도가 네트워크 관리자 중심의 대응속도를 훨씬 능가하였다는 것을 감안하여 네트워크 중심의 자동 대응 패러다임 마련을 위해 수학적인 분석 모델링을 제시하는 것이다. 또한 공격 유형을 공격 속도 특성에 부합되는 분포도와 시스템의 구성 노드 중 공격에 대응하고 있는 노드수로 분류하여, 지역적인 탐지 결과에 따른 수동적 대응 및 단기적 정보에 의존한 탐지에 따른 높은 오류율을 감소하기 위한 임의 구현 방법을 제공하기 위한 것이다. 이를 위해 2장에서는 생존성 개념을 적용하기 위한 유비쿼터스 컴퓨팅 시스템의 구성 및 설계 방법에 대해 설명하고, 3장에서는 대규모 인프라 공격에 대한 공격 유형 모델링을 근간으로 시스템의 보안성을 평가할 수 있는 생존성 모델링에 대해 제시하며, 4장에서는 본 논문에서 제안한 기법의 실효성 검증을 위해 전통적인 TCP-SYN 공격과 Code-Red 웜 공격에 대한 시스템의 생존성을 분석한다. 마지막으로 5장에서는 본 논문의 연구 결과를 살펴보며 향후 연구방향에 대해 제시한다.

2. 시스템 구성 및 설계

본 장에서는 본 논문에서 제안하고자 하는 유비쿼터스 컴퓨팅 시스템의 생존성 개선 기법을 적용하기 위한 시스템 구성 및 설계 방법을 설명하고 중앙집중적인 관리가 아닌 각 노드간의 협업을 통하여 제안한 개선 기법을 활용하기 위한 방안에 제시한다.

(그림 1)은 생존성 개선 기법을 적용하기 위한 유비쿼터스 컴퓨팅 시스템의 구성도이며, 물리적으로 각 노드는 유무선 네트워크 장비로 상호 연결되어 서비스를 수행할 수 있는 시스템을 형성한다. 또한 각 노드는 대규모 인프라 공격을 해결하기 위해 정량적인 분석을 할 수 있는 모듈(*Quantified Analyzer*)을 탑재하고 협업을 통한 공격 탐지 및 방어가 가능하도록 서로의 모니터링 정보를 공유한다. 그리고 각 노드의 시스템 정보(*System Specification*)를 분석한 결과를 근거로 하여 공격을 탐지하기 위해 공격 유형(*Attack Detection Specification*)을 분류하고 해당 공격 유형에 방어하기 위한 대응책(*Attack Recovery Specification*)을 정의한다. 마지막으로 협업을 수행

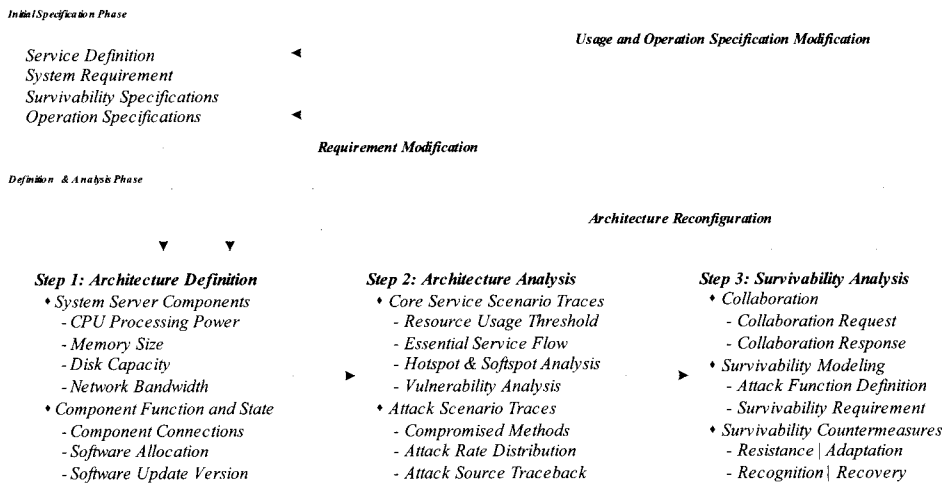
함에 있어 이질적인 시스템 성질을 해결하기 위해 정보 및 코드 변환기(*Synthesized Translator*)를 각 노드에 탑재한다.



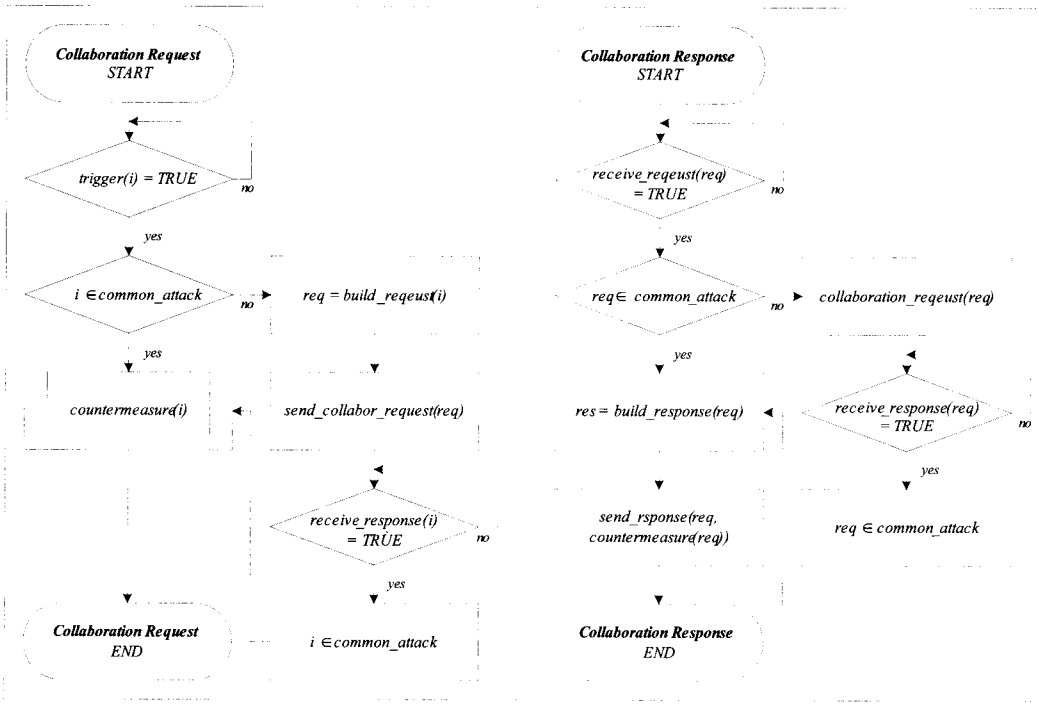
(그림 1) 생존성 개선 기법을 적용하기 위한 유비쿼터스 컴퓨팅 시스템의 구성도

(그림 2)는 3장에서 설명할 생존성 모델링 및 공격 유형 모델링의 적용 대상이 되는 유비쿼터스 컴퓨팅 시스템을 설계하기 위한 방법으로 크게 세 개의 단계로 구분된다. 이에 앞서 초기 명세 과정(*Initial Specification Phase*)에서는 사용자가 접속한 유비쿼터스 컴퓨팅 시스템을 통해 제공받고 싶은 서비스, 필요로 하는 시스템의 성능, 요구되는 생존성에 대한 명세 및 운영 요구사항에 대해서 명시를 한다. 다음에, 1 단계인 구조 정의 단계(*Architecture Definition*)에서는 초기 작성된 명세서를 기반으로 시스템 구조의 스타일 및 형태가 선택적으로 결정되며 이 형태를 위한 개개 노드의 사양을 도출(*System Server Components*)한 후 이를 관리하고 서비스를 지원하기 위한 미들웨어 및 소프트웨어(*Component Function and State*)를 설치한다. 이후, 2 단계인 구조 분석 단계(*Architecture Analysis*)에서는 공격이 존재하거나 진행중이라 할지라도 필수적으로 지원되어야만 하는 서비스, 사용 가능한 자원의 임계값, 제공할 서비스의 취약점을 분석한 결과를 토대로 핵심 서비스 시나리오(*Core Service Scenario Traces*)를 선택 및 결정하고 이 서비스에 대한 공격 유형과 부합되는 분포도 생성, 공격 방법, 공격 진원지의 역추적 등 공격 유형 모델(*Attack Scenario Traces*)을 도출한다. 마지막으로 생존성 분석 단계(*Survivability Analysis*)에서는 지엽적인 분석 결과를 토대로 하는 대응책의 높은 오류율을 제거하기 위해서 상호 연결된 컴퓨팅 시스템 내 노드간 협업(*Collaboration*)을 수행하고 이를 위한 생존성 분석 모델(*Survivability Modeling*)을 도출하며, 해당 공격에 대한 대응책(*Survivability Countermeasures*)을 마련한다. 그리고 생존성 분석 결과는 공격 가능성이 발생되었을 때 이에 대비하기 위한 시스템 구조로 재구성을 수행하게 된다(*Architecture Reconfiguration*). 이를 위해 시스템 및 서비스의 요구사항 및 명세서를 변경하게 된다(*Requirement Modification*). 또한 명세서에 기반하여 유비쿼터스 컴퓨팅 시스템 구조를 도출 및 정의할 때 사용 가능한 자원에 따라 시스템 및 서비스의 요구사항 및 명세서가 변경될 수 있다(*Usage and Operation Specification Modification*).

(그림 3)은 생존성 개선을 위한 분산된 노드간에 협업을 통한 공격 대응 알고리즘을 설계한 것으로 크게 협업 요청을 하는 부분(*Collaboration Request*)과 요청에 대한 응답을 하는 부분(*Collaboration Response*)으로 나눌 수 있다. 협업을 요청하게 되는 노드는 자원상태 모니터링 결과 이상한 증후로 인하여 공격 발생 가능성이 존재하게 되면 공격 여부를 판단하기 위한 모듈이 작동하며 이때 해당 노드에 기 모델링되어 있는 공격이면 이에 대한 대응책을 작동시킨다. 하지만 해당 노드에 알려진 공격 유형이 아니지만 노드의 오동작을 유발하려는 조짐이 보이거나 일정 기간 패킷 전송율이 의심스러우면 시스템의 다른 노드에게 분석을 요청하게 된다. 협업 요청을 받은 노드는 자신에게 알려진 공격 유형과 요청이 들어온 명세서를 비교한 후, 기 모델링된 공격 유형이면 해당



(그림 2) 고신뢰 유비쿼터스 컴퓨팅 시스템 구조 설계 방법



(그림 3) 분산 협업을 통한 공격 대응 방법

대응책을 전송하고 아니면 또 다른 노드에게 분석 요청을 하게 된다. 이와 같은 협업은 네트워크로 상호 연결된 컴퓨팅 시스템 내 모든 노드에서 이뤄지며 시스템에 정의되어 있지 않은 공격 유형에 대한 분석 요청의 무한 사이클(cycle)을 제거하기 위해 협업 요청 패킷에 송신자의 주소를 MD5[21]와 같은 기술을 통해 덧붙여 보낸다. MD5란 입력 데이터 (길이에 상관없는 하나의 메시지)로부터 128 비트 메시지 축약을 만듦으로써 데이터 무결성을 검증하는데 사용되는 알고리즘으로 시스템의 협업 요청을 한 노드수가 증가하게 되더라도 전송할 메시지의 크기는 항상 동일하게 된다. 따라서 자신에게 알려지지 않은 공격 유형이기 때문에 다른 노드에게 분석 요청을 하는 송신자의 수가 증가하게 된다고 하여 협업 요청 패킷에 덧붙여 보내게 되는 주소의 길이가 증가하게 되는 것이 아니고 항상 일정 크기의 메시지를 전달할 수 있는 알고리즘이다. 그러므로 해당 알고리즘을 통한 협업 요청은 본 논문에서 제안하고자 하는 기법의 적용 대상인 유무선 네트워크로 상호 연결된 유비쿼터스 컴퓨팅 시스템에 적합하다고 할 수 있다.

3. 정량적 분석 모델링

3.1 생존성 및 공격 모델링

인프라 공격을 탐지 및 대응하기 위한 정량적인 분석은 노드 i가 공격을 받아 오동작 및 결함이 발생하는 시간을 기준으로 이뤄진다. 해당 시간은 고정적인 요소(Fixed Effect, θ)와 임의의 요소(Random Effect, u)에 의해서 영향을 받는다. 고정적인 요소는 시스템의 CPU 처리속도, 메모리 크기, 디스크의 용량, 네트워크 대역폭 등이며, 임의의 요소는 방화벽, 설치된 OS 버전, 보안 메커니즘 갱신 정보, 공격자의 능력 등이다. 전형적으로 고정적인 요소와 임의의 요소는 노드의 위험요소(Risk Factor, η_i)의 벡터로 아래와 같이 표현되며, x_i와 z_i는 네트워크로 상호 연결된 컴퓨팅 시스템을 구성하기 위해 사용되는 시스템 명세서에 명시된 임의의 값이다.

$$\eta_i = x_i\beta + z_i u \tag{1}$$

노드에 대한 속성 분석에서 임의의 요소에 대한 정의는 측정에 따라 달라지기 때문에 복잡한 공분산(Covariance) 구조를 모델링하기 위해서는 유연성을 제공해주어야 한다. 따라서 임의의 요소에 대한 분포도는 다변량(Multivariate) 분석을 통한 정규분포를 따른다고 가정하는 것이 일반적이다. 노드의 위험요소가 정의되면, 적어도 주어진 시간 t까지 노드가 공격에 대응하여 정상적인 서비스를 수행할 수 있는 확률을 얻을 수 있는데, 이 확률값을 노드의 생존성(Survivality)이라 한다. 또한 공격을 받고 노드가 오동작을 하는데 까지 걸리는 시간(T_i)이 주어지고, 이에 대한 밀도함수(Density Function), $f(t;\eta_i)$와 누적분포함수(Cumulative Distribution Function), $F(t;\eta_i)$에 대해 정의할 수 있으면 생존성에 대한 분석을 위한 함수 $S(t;\eta_i)$를 모델링할 수 있다.

$$S(t;\eta_i) = \Pr(T_i \geq t) = 1 - F(t;\eta_i) = \int_t^{\infty} f(w;\eta_i)dw \tag{2}$$

다시 말해서, 생존성은 주어진 시간 동안 수행하던 작업을 지속할 수 있는 확률로써 정의할 수 있다. 따라서 생존성에 대한 함수, $S(t)$는 인자값으로 시간(t)이 주워졌을 때, 시스템

에 공격으로 인한 피해가 발생하지 않을 확률로 정의할 수 있다. 시스템이 공격으로 피해가 발생하지 않은 시간은 일반적으로 임의의 변수로 모델링하며, 시스템 초기 상태에서의 생존성은 1이 된다. 또한 다수 노드($i \geq 2$)로 이뤄진 시스템의 경우 전체 시스템의 생존성, $S_R(k,n,t)$은 단일 노드의 생존성, $S(t;\eta_i)$에 의해 결정되며, n개 노드 중 k개의 노드가 공격으로부터 영향을 받지 않은 상태에 머무를 확률은 아래와 같이 구할 수 있다.

$$S_R(k,n,t) = \sum_{i=k}^{n} \binom{n}{i} S(t;\eta_i)(1-S(t;\eta_i))^{n-i} \quad (3)$$

즉, n개 노드의 집합에서 k개를 선택하여 조합할 수 있는 모든 경우의 수를 이항 계수(Binomial Coefficient)로 사용하여 적어도 k개 노드가 공격에 대응하고 있는 확률값을 얻을 수 있는 것이다. 그리고 시스템의 생존성 분석을 위한 모델은 공격 유형에 대한 모델을 통해 얻을 수 있다. 공격 유형 모델이란 주어진 시간에 한 노드가 공격을 받아 오동작을 일으킬 위험에 대해서 측정하는 것이며, 이에 대한 함수는 아래와 같이 얻을 수 있다.

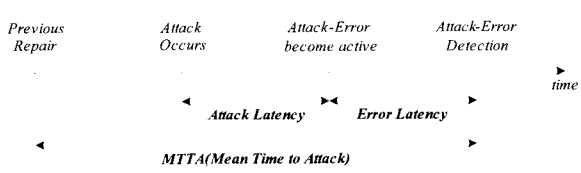
$$\lambda(t;\eta_i) = \lim_{\Delta t \to 0} \frac{\Pr(T_i < t + \Delta t \mid T_i > t)}{\Delta t} = \frac{f(t;\eta_i)}{S(t;\eta_i)} \quad (4)$$

또한, Δt 동안 오동작을 일으킬 확률이 거의 $\lambda(t;\eta_i)\Delta t$와 같다는 것을 통해 또 다른 방법으로 공격 유형 함수를 구할 수 있다. 그리고 공격 유형 함수의 정의로부터 알 수 있듯이 음이 아닌 값을 가진다. 더욱이 시간 t에 공격으로 인한 오동작이 발생할 위험이 없다면 양의 값을 가진다. 그러므로 노드의 생존성 함수와 공격 유형 모델 함수와의 관계를 나타내면 아래 식과 같이 표현할 수 있다.

$$S(t;\eta_i) = e^{-\Lambda(t;\eta_i)}, \quad where\ \Lambda(t;\eta_i) = \int_0^t \lambda(w;\eta_i)dw \quad (5)$$

3.2 공격 유형 모델링

(그림 4)는 본 논문의 아이디어의 출발점을 정립하기 위한 개념으로써 복잡도 이론(Complexity Theory)을 적용하여 공격 유형을 분류하기 위한 기본적인 개념이다. 이는 최초 공격을 시작하여 한 노드의 오동작을 일으킬 때까지의 공격 시간(MTTA, Mean Time to Attack)을 기준으로 분류한 것으로 결함허용 시스템에서 신뢰도 분석을 수행하는 MTTF (Mean Time to Failure)와 유사한 개념이다.



(그림 4) 생존도의 기본적인 개념 정립을 위한 MTTA

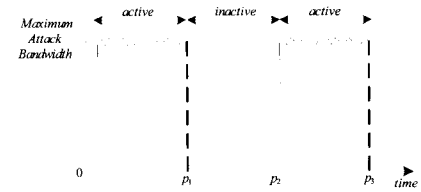
다시 말해서 제안한 기법의 적용범위를 정의하기 위해서 공격자가 한 노드에 대해 공격을 수행하였더라도 해당 노드가 오동작을 하지 않거나 노출되지(Compromised) 않아서 공격자의 의도대로 동작하지 않는다면 그것은 공격이 성공하지 못한 상태로 생존성을 유지하고 있다고 가정한다. 따라서 보안 정책은 정성적인 분석 방법이나 정량적 분석 방법 모두 공격을 감지하거나 현재 노드의 상태를 확인하는 모니터의 정확성에 의존적이다. 더욱이 분산 서비스 거부 공격 같은 경우 단일 노드 내에서 공격 여부를 판단하기란 쉽지 않다. 따라서 메시지 기반 분산 시스템의 경우 노드 간에 협업 및 감시를 통해서 공격 여부를 판단하는 메커니즘이 제안되었다 [22]. 따라서, 분산 협업 시스템의 노드 중 일정시간 (t)까지 완전히 공격자의 의도에 따라 공격을 당한 노드의 수($C(t)$)와 공격 특징과 부합되는 분포도를 가지고 [22]에서 제시한 공격 유형 4가지를 재정의하여, 정량적 분석을 위한 모델링으로 활용한다. 공격 특성을 표현하기 위한 분포도는 결함허용 시스템에서 신뢰도를 분석할 때 사용하는 일반적인 분포도를 사용한다.

• 상수 시간 공격 (Constant Time Attack)

공격율을 최대한으로 달성한 후 공격이 끝날 때까지 최대 공격율을 유지하는 특성을 가지는 것이다. 즉, 시스템을 구성하고 있는 모든 노드의 오동작을 일으키거나 노출시켜 공격자의 의도대로 노드를 작동할 수 있는데 걸리는 시간이 상수 시간을 필요로 하는 공격이다. 이와 같은 공격 유형의 시나리오는 모든 노드의 운영체제 및 보안 메커니즘이 동일하고 공격자가 해당 취약점을 파악하고 있는 경우이다. 이 공격 유형의 특징과 부합되는 분포도는 와이블 분포(Weibull Distribution)이다. 따라서 $C(t)$값은 공격에 필요로 하는 시간(p) 이전까지는 0 이며, 그 이후에는 시스템의 총 노드수(n)가 된다.

$$C(t) = \begin{cases} 0 & if\ t \leq p, \\ n & otherwise \end{cases}$$

• 급격한 추이 변동을 보이는 공격(Pulsing Attack)



(그림 5) 급격한 추이 변동을 보이는 공격의 특성

(그림 5)는 해당공격의 특성을 보여주는 것으로 공격 여부에 대한 시스템 관리자의 감시를 피하기 위해서 공격율을 임의의 기간(p_1, p_2, p_3)에 따라 최대 공격율을 유지하는 경우(active period)와 공격을 하지 않는 경우(inactive period)를 반복하는 특징을 가진 공격이다. 이 공격 유형의 $C(t)$값은 시스템 전체 노드를 동시에 공격하는 것이 아니고 일부만 공격

하므로 공격에 필요로 하는 시간(p_1) 이전까지는 0 이며, p_1 이후에는 시스템의 일부 노드수(a)가 오동작을 일으키거나 노출되며, p_3 이후에 모든 노드수($n = a + b$)가 된다. 따라서 한 상태에서 다른 상태에 전이하기 위해 단계(r-stage)가 필요한 어랑 분포(Erlang Distribution)의 특징과 유사하다.

$$C(t) = \begin{cases} 0 & if\ t \le p, \\ a & if\ p_1 < t \le p_3 \\ b & otherwise \end{cases}$$

• 증가 추이 공격(Increasing Time Attack)

공격율을 점진적으로 높여가며, 노출되는 노드 수를 하나씩 증가시키는 특징을 보이는 것으로, 시스템의 모든 노드에 상이한 운영체제가 설치되어 있거나 보안 메커니즘의 갱신 정보가 서로 달라 새로운 노드를 공격하기 위해서는 다른 노드를 공격하기 위해 걸린 시간이 동일한 경우에 해당되는 공격 유형이다. 따라서 노드의 취약성이 모두 다르므로 시스템의 모든 노드를 공격하기 위한 시간은 점증적으로 증가하게 되며, 이와 같은 특성을 갖는 분포도는 하이포 지수 분포(Hypoexponential Distribution)와 유사하다. 또한 $C(t)$값은 한 노드를 공격하는데 걸리는 시간(p)와 공격 지속 시간(t)의 관계로 나타낼 수 있다.

• 점진적인 추이 변동을 보이는 공격(Gradual Pulse Attack)

점진적인 추이 변동을 보이는 공격은 급격한 추이 변동을 보이는 공격의 특성과 증가 추이 공격의 특성을 포함하고 있는 특징을 가진다. 따라서 공격을 수행하는 기간 동안에는 공격율을 점진적으로 증가시키고 일정 기간 동안 공격을 진행하지 않는다. 따라서 이 공격의 특성은 어랑 분포와 하이포 지수 분포를 일정 기간(period)에 해당 분포를 적용한 후 조합하여 표현할 수 있다.

본 장에서는 네트워크로 상호 연결된 유비쿼터스 컴퓨팅 시스템에서 수행되는 보안 유지 메커니즘은 정성적인 대응책과 더불어 정량적인 분석 모델을 통해 유지 관리 대응전략을 수립하는 것이 공격 및 변종 공격으로부터 예방적인 보안 유지 기법으로 활용 가능하다는 것을 수학적으로 검증하기 위해 새로운 시스템 보안관련 측정요소인 생존성에 대해 정의하였다. 또한 시스템의 생존성을 정의하기 위해서는 특정 공격에 대한 분석이 아니라 범용적 특성에 따라 분류한 공격 유형 모델이 시스템 보안 향상을 위한 중요한 특성임을 파악하였다.

4. 사례 연구

4.1 TCP-SYN 공격

TCP-SYN 공격[22]은 합법적인 연결 요청이 처리되는 것을 방해하기 위해 TCP/IP 프로토콜의 단점을 이용하여 백로그(backlog) 큐에 불필요한 정보를 채우는 것을 목표로 하는 공격이다. 공격을 성공적으로 이루기 위해, 컴퓨터가 SYN과 ACK 해당 비트를 셋팅하여 응답할 때, 응답이 존재하지 않거나 가용하지 않은 호스트에게 전해지는 것 같이, 공격자는 공격 대상 컴퓨터에게 SYN 비트를 셋팅하여 연결 요청을 여러 번 보내야만 한다. 그러면, 해당 노드는 합법적으로 연결할 때보다 오랜 시간 동안 응답을 기다리게 되고 위조된 연결에 의해 백로그 큐의 메모리 공간에 불필요한 정보가 늘어나게 된다. 이와 같이 알려진 TCP/IP 프로토콜의 단점을 이용하는 TCP-SYN 공격과 같이 알려진 시스템 취약성을 가지고 공격하는 공격 유형은 상수 시간 공격(Constant Time Attack)에 해당된다. 따라서 주어진 시간에 한 노드가 공격을 받아 오동작을 일으킬 위험이 동일하므로 공격 위험 함수는 와이블 분포로 표현할 수 있으며, 식 (1)과 (4)를 이용하여 위험 요소를 포함한 공격 유형 모델은 식 (6)과 같이 정리될 수 있다.

$$\lambda(t;\eta_i) = \rho\lambda(\lambda t)^{\rho-1} \tag{6}$$

여기서 ρ는 공격위험율을 뜻하며, 생존성의 변화 정도를 분석할 수 있는 파라미터이다. 즉, ρ가 작은 값을 가지면, 생존성을 분석하기 위한 함수 곡선이 급격하게 떨어지고 결국 시스템이 동작하지 못하게 된다. 또한 큰 값을 가지면 일정 수준을 유지하다가 갑자기 떨어지게 된다. 또한 λ는 공격율을 뜻하며, 정량적 모델링에서 교차점을 하기 분석하기 위한 역할이며, 식 (6)으로 정의된 공격 유형 모델을 가지고 시스템의 생존성을 분석하기 위해서는 식 (5)를 근간으로 계산하고 이를 최종적으로 정리하면 아래와 같다.

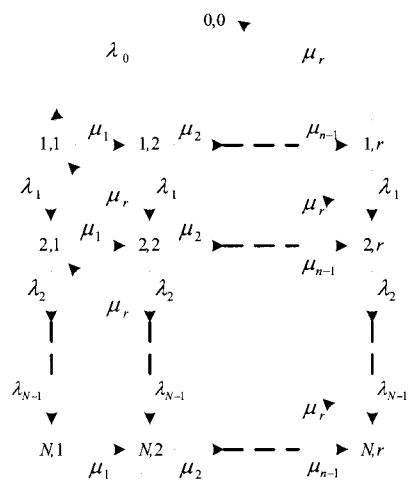
$$S(t;\eta_i) = e^{-(\lambda t)^{\rho}} = e^{-\exp[\rho\ln(t)+\rho\ln(\lambda)]} = e^{-\exp[\rho\ln(t)+\eta]}$$
$$= e^{-t^{\rho}e^{\eta}}\ ,\ \text{where}\ \eta = \rho\ln(\lambda) \tag{7}$$

따라서, 분석 그래프의 기본 모양(basic shape)은 $\eta>0$이면 위험요소가 증가하고, $\eta<0$이면 위험요소가 감소하는 것처럼 위험요소가 다양하게 변한다고 할지라도 일정하다는 가정이 합당하다는 것을 뜻한다. 그리고 한 노드의 공격 시간이 전체 시스템의 노드가 공격에 노출될 때까지의 시간과 같으므로 한 노드를 노출시킬 때까지 필요한 시간 이전까지는 $C(t)$ 값은 0 이고, 해당 시간 이후에는 모든 노드가 노출되게 된다.

4.2 Code-Red 웜 공격

Code-Red 웜 공격[23]은 2001년 7월 12일에 임의 상수 생성기(Random Number Generator)에서 고정된 시드(seed)값을 사용한 Code-Red 웜(CRv1)이 처음 마이크로소프트사의 IIS (Internet Information Server) 웹 노드를 공격한 이후, 19일에는 변형된 Code-Red 웜(CRv2)이 약 14시간 이내에 359,000 노드 이상이 연결된 인터넷을 감염시켜, 손해비용만 26억불에 달하는 것으로 추산된다. Code-Red 웜 공격 단계를 3가지로 구분할 수 있는데, 처음에 공격할 노드에게 긴 URL을 보냄으

로써 인계값 스택 오버플로우(Parameter Stack Overflow)를 유발시킨다. 다음에, 해당 URL의 존재하는 잘못된 반환 주소값으로 스택 안에 있던 정상적인 주소를 덮어쓴다. 그런 후 마지막으로 웹 노드에서 주소값을 얻기 위해 스택의 내용을 읽을 때 잘못된 주소가 지정하고 있는 Code-Red 웜이 악성 코드부분 실행시킨다. 따라서 Code-Red 웜의 경우 다단계 진행 상태를 포함할 수 있는 어랑 분포(Erlang Distribution), 하이포 지수분포(Hypoexponential Distribution), 또는 하이퍼 지수분포(Hyperexponential Distribution)로 정량적인 분석 모델링이 가능하다. 그러나 Code-Red 웜의 경우 점차 감염된 노드수가 늘어날수록 다음 피해를 받을 노드수가 기하급수적으로 증가하기 때문에 증가 추이 공격(Increasing Time Attack)에 해당되며, 이와 같은 특성을 기반으로 식 (1)과 (4)를 정의하기 위해서는 하이포 지수분포를 따라 표현하는 것이 적당하다. 또한 이와 같은 경우 다수 노드로 구성된 시스템의 상태를 분석하기 위해서는 식 (3)과 식 (6)을 근간으로 (그림 6)과 같은 상태 전이 모델링을 정의하고 분석해야 한다. 다시 말해서, 한 노드가 감염되어 오동작을 일으키거나 공격자의 의도대로 컴퓨터가 작동되기 위한 단계를 r이라 하고 감염된 노드수를 N이라고 할 때, 분석 모델은 (그림 6)과 같다. 따라서 (그림 6)은 공격 유형 모델로 하이포 지수 분포를 사용하며, 생존성 함수와 공격을 당한 노드의 수를 분석하기 위해 사용되는 상태 전이도이다. 여기서 한 노드를 공격하는데 걸리는 시간은 $\mu_i (i=1...r)$로 표현되며, 공격 지속 시간은 $\lambda_i (i=1...N)$로 표현된다. 이를 상태 전이도에 표기하기 위해서 (감염된 노드수, 공격 단계)의 쌍으로 시스템의 상태를 표현하며, 초기 및 정상 상태는 (0,0)이다. 예를 들어, (2,3)의 의미는 현재까지 한 노드 1개가 공격에 노출되었으며, 다른 한 노드가 공격을 받고 있는 중인 상태를 나타낸다. 최종적으로 (N,r) 상태에 도달했다는 의미는 시스템의 모든 노드가 공격에 대해 노출되었다는 것을 뜻한다. 다시 말해서, λ_i는 공격율이고, μ_i는 공격 단계 성공률이며, μ_r은 감염된 노드가 치료를 위해 필요한 서비스율을 나타낸다.



(그림 6) Code-Red 웜에 대응하기 위한 생존성 모델링

하지만, 공격 중간 단계에서는 노드가 오동작을 일으키거나 공격자의 의도대로 노드가 동작하지 않으므로 공격 여부에 대한 판단이 불가능한 것으로 가정한다. (그림 6)의 상태 전이도를 근간으로 현재 시스템이 공격에 대해 노출되었는지, 노출되었다면 어느 정도의 노드가 공격에 노출되었는지에 대한 정보를 전이도의 각 상태에 머물 확률을 분석할 수 있으며, 이는 공격 유형 모델링을 근간으로 구할 수 있다. 다시 말해서, (그림 6)에서 모든 상태가 안정 상태(steady-state)일 때의 균형 방정식(balance equation)을 구하면 아래와 같다.

$$\mu_r P_{n,r} = \lambda_{n-1} P_{n-1}$$

$$\mu_r P_{1,i} = (\lambda_1 + \mu_{i+1}) P_{1,i+1}$$

$$\mu_r P_{n,i} = (\lambda_n + \mu_{i+1}) P_{n,i+1} - \lambda_{n-1} P_{n-1,i+1}$$

$$\text{, where } < n > \equiv \{(n,i) \mid i = 1, \ldots, r\}$$

위의 균형 방정식과 각 상태에서 머물 확률의 총합이 1이 되는 보존(conservation) 방정식을 결합한 연립 방정식을 풀면, 시스템이 평형일 때, 각 상태에 머물 확률을 다음과 같이 얻을 수 있으며(식 (8) 참조), 식 (5)를 도출하기 위해 식 (8)을 근간으로 한 노드라도 정상 상태에서 동작하기 위한 시스템의 생존성을 계산하면 $S(t;\eta_i) = 1 - P_{N,r}$와 같이 구할 수 있다.

$$P_{n,i} = \lambda_o G^*(1;i,r-1) P_0 \tag{8}$$

$$P_{n,i} = \lambda_{n-1} \left\{ G^*(n;i,k-1) P_{n-1} - \sum G^*(n;i,k-2) P_{n-1,k} \right\},$$

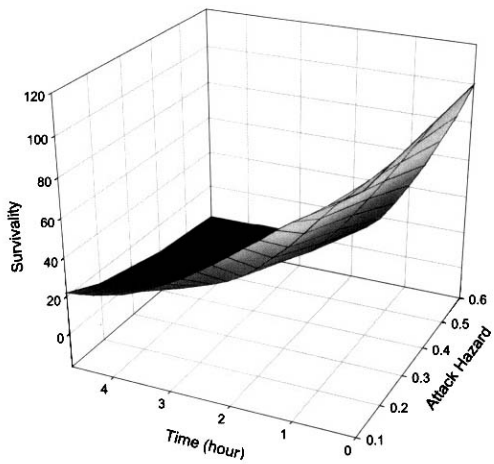
$$\text{where } G^*(n;i,k-1) \equiv \prod_{j=1}^{k-1} \frac{(\lambda_n + \mu_{j+1})}{\mu_j}$$

4.3 생존성 분석 결과

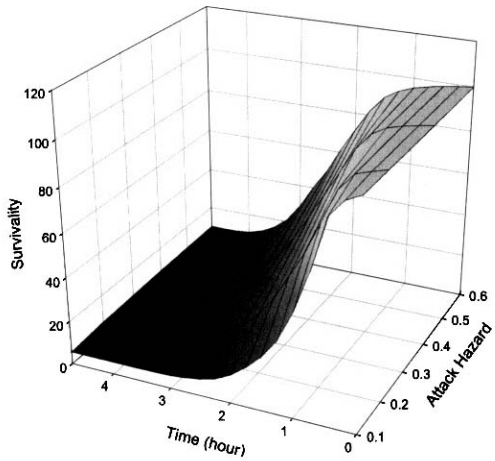
(그림 7)은 TCP-SYN 공격과 Code-Red 웜 공격의 생존성 분석 결과로 공격 위험 요소에 따라 미치는 영향을 보여주는 것으로, 생존성 모델링 함수와 공격 위험 요소 함수에 따른 공격 유형 모델링과 식 (7), (8)을 근간으로 분석한 결과이다. 또한 공격 위험 함수의 위험 요소는 0에서 0.6 사이의 평균값으로 변화하며, 대상으로 하는 공격의 시컨스를 생성하기 위해 의사난수 생성법(Pseudo-Random Number Generator)을 사용하고, 분석을 위한 기본 파라미터 설정은 [22, 23]에 따라 <표 1>과 같다.

〈표 1〉 테스트 파라미터 [22, 23]

파라미터	기본값
관찰 주기 (Observation interval)	1 초
최소 패킷 전송율 (Minimum Packet Rate)	2 Kbps
양단간 송수신된 최대 패킷 수	3
정상 패킷과 공격 패킷 비율	0.5
최대 감염율(Infection Rate)	2,000 host/minute
알려지지 않은 호스트 유형(Unknown host types)	55 %



(a) TCP-SYN 공격



(b) Code-Red 웜 공격

(그림 7) 사례 연구를 통한 생존성 분석 결과

TCP-SYN 공격에 대한 분석의 경우, 공격 위험 요소가 존재한다면 시스템 운영 시간이 늘어날 수록 점진적으로 생존성이 저하되지만, Code-Red 웜 공격의 경우 일정 시간 공격에 대해 견딜 수 있지만 임계점이 지나면 생존성이 급격히 감소함을 알 수 있다. TCP-SYN 공격의 경우 공격 위험 요소가 0.5이면서 시간당 공격율이 분당 20회 이상이만 되더라도 시스템의 전체 생존 노드수가 50% 이하로 떨어진다. 하지만 공격 위험 요소가 0.1로 낮아진 경우에도 분당 26회 이상이면 시스템의 생존 노드수가 50% 이하로 떨어진다. 해당 결과는 TCP-SYN 공격은 알려진 취약성을 이용하므로 공격 위험 요소에는 민감하게 반응하지 않고 공격 빈도수에 상대적으로 민감하게 변화하는 것을 알 수 있다. 또한 TCP-SYN 공격의 경우 시간에 대해 점진적으로 변화하는 이유는 TCP/IP 프로토콜 자체에서 백로그 큐에 저장된 정보 중 타임아웃이 지나면 응답이 없는 것으로 판단하고 지우는 메커니즘이 존재하며, 공격 자체도 정성적인 분석이 용이하여 시스템이 일정 수준 지속적으로 관리할 수 있기 때문이다. 하지만, Code-Red 웜 공격의 경우, 시스템의 전체 생존 노드수가 50% 이하로 떨어지는 경우가 TCP-SYN 공격에 비해 상대적으로 짧은 시간에 발생하며, 이 변화는 공격 위험 요소가 증가할수록 좀더 심해진다. 다시 말해서 공격에 대해 손상을 입은(compromised) 노드수가 초기에는 전체 시스템의 생존성에 미미한 영향을 미칠 정도이지만, 기하급수적으로 노드수가 증가하게 되므로 결국 정성적인 분석을 통해 대비할 수 있더라도 궁극적인 치료를 수행하지 않으면 전체 시스템의 생존성을 유지하기 어렵기 때문이다. 하지만 Code-Red 웜 공격과 같이 상이한 플랫폼 상에서 변형적인 공격이 쉬운 공격은 즉각적인 정성적 공격 대응책 마련이 어렵다. 따라서 기존에 알려진 증가 추이 공격의 시스템 생존성 변화 그래프를 근간으로 비교하여 그래프의 형태(shape)나 변화율(rate) 또는 교차점(intersect)이 기존 공격과 유사하게 나타나면 이에 대한 대응책을 적용하도록 하면 보다 빠르게 공격에 대해 대비할 수 있다. 결론적으로 TCP-SYN 공격과 Code-Red 웜 공격과 같이 공격 유형의 특성을 (그림 7)과 같이 정확히 표현하고 이를 분석할 수 있는 생존성 모델링을 활용할 경우 알려지지 않거나 변형된 공격의 특징을 보다 쉽게 파악할 수 있으므로 이에 대한 대응책을 보다 쉽게 마련할 수 있다.

5. 결 론

본 논문에서 제안한 기법은 유무선 네트워크로 상호 연결된 여러 대의 노드로 구성된 유비쿼터스 컴퓨팅 시스템의 보안 유지 방법을 개선시키는 기법에 관한 것으로 컴퓨팅 시스템의 공격 발생 이후에 정성적인 탐지 및 대응책으로 대처하는 기존의 보안 유지 방법에 비해, 정량적인 분석을 통해 범용적 인프라의 개선뿐만 아니라 특정 인프라 공격에 대해서 탐지 및 대응 가능한 능동적 차원의 보안 유지 방법이다. 이를 위해서 시스템의 고정적 요소 정보, 임의의 요소 정보, 공격 유형 모델링을 근간으로 시스템의 보안성을 정량적으로 분석할 수 있도록 생존성에 대한 정의 및 모델링 기법을 사용하였다. 다시 말해서, 제안한 기법은 시스템의 구조에 대한 정의, 생존성을 위한 필요조건, 공격 모델링을 위한 명세서, 운영을 위한 요구조건 등의 운영 파라미터로부터 대규모 인프라 공격에 대응하기 위한 보안 유지 정책을 수립하고 단시간에 공격 탐지가 이뤄지게 하는 방식으로 고신뢰성을 달성하게 된다. 따라서, 시스템의 공격 발생 이후에 수동적으로 대처하거나 정성적인 분석에 의존한 지엽적인 대응책에 비해, 공격 유형 모델링과 협업을 통한 대응으로 공격 탐지 시점을 앞당길 수 있는 능동적/예방적 차원의 보안 유지가 가능하게 된다. 더욱이 정량적 분석 방법은 관련 시스템에 탑재된 보안 메커니즘 및 대응전략의 높은 오류율을 제거하고 지엽적 또는 단기간 정보에 의존하지 않으므로 최근 발생한 대규모 인프라 공격에 효율적으로 대응할 수 있다. 향후에 본 논문에서 제시한 유비쿼터스 컴퓨팅 시스템의 구성 및 설계 방법을 통해 대표적인 정성적 분석 기법과 제안한 정량적 분석 기법을 구현할 것이다.

참 고 문 헌

[1] M. Weiser, "Some Computer Science Problems in Ubiquitous Computing," Communications of the ACM, Vol.36, pp.75 - 84, 1993.

[2] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," IEEE Personal Communications, pp.10-17, 2001.

[3] Al. Mankin, et al., "On Design and Evaluation of Intention-Driven ICMP Traceback," Proceedings of the IEEE International Conference on Computer Communication and Networks, pp.159-165. Oct., 2001.

[4] D. Schnackengerg and K. Djahandari, "Cooperative Intrusion Traceback and Response Architecture," Proceedings of the 2nd DARPA Information Survivability Conference and Exposition (DISCEX II), pp.56-68, June, 2001.

[5] J. Cabreraa, et al., "Proactive Detection of Distributed Denial of Service Attacks using MIB Traffic Variables-A Feasibility Study," The 7th IFIP/IEEE International Symposium on Integrated Network Management, pp.609-622, May, 2001.

[6] D. Kashiwa, E. Chen, and H. Fuji, "Active Shaping: a Countermeasure against DDoS attacks," Proceedings of the 2nd IEEE European Conference on Universal Multiservice Networks, pp.171- 179, Apr., 2002.

[7] K. Wan and R. Chang, "Engineering of a Global Infrastructure for DDoS attacks," Proceedings of the 10th IEEE International Conference on Networks, pp.419-427, Aug., 2002.

[8] V. Paxson, "An Analysis of Using Reflectors in Distributed Denial-of-Service Attacks," ACM SIGCOMM Computer Communication Review, Vol.31, No.3, pp.38-47, 2001.

[9] F. Wang, R. Uppalli, and C. Killian, "Analysis of Techniques for Building Intrusion Tolerant Server Systems," Proceedings of Military Communications Conference, pp.729-734, Oct., 2003.

[10] A. Avizienis, J. Laprie, and B. Randell, "Fundamental Concepts of Dependability," Proceedings of the 3rd Information Survivability Workshop, pp.7-12, Oct., 2000.

[11] J. Reynolds, et al., "On-line Intrusion Detection Attack Prevention Using Diversity Generate-and-Test, and Generalization," Proceedings of the 36th Annual Hawaii International Conferences on System Sciences, pp.335-342, Jan., 2003.

[12] J. Knight, et al., "The Willow Architecture: Comprehensive Survivability for Large-Scale Distributed Applications," Technical Report CU-CS-926-01, Department of Computer Science, University of Colorado, 2001.

[13] V. Westmark, "A Definition for Information System Survivability," Proceedings of the 37th Annual Hawaii International Conferences on System Sciences, Jan., 2004.

[14] B. Madan, et al., "Modeling and Quantification of Security Attributes of Software Systems," Proceedings of the International Conference on Dependable Systems and Networks, pp.505-514, June, 2002.

[15] F. Cohen, "Simulating Cyber Attacks, Defenses, and Consequences," Fred Cohen & Associates, 1999.

[16] M. Gupta, A. Chaturvedi, and S. Mehta, "The Experimental Analysis of Information Security Management Issues for Online Financial Services," Proceedings of the 21st ACM International Conference on Information Systems, pp.667-675, 2000.

[17] B. Littlewood, et al., "Towards Operational Measures of Computer Security," Journal of Computer Security, pp. 211-229, 1993.

[18] R. Ortalo, et al., "Experiments with Quantitative Evaluation Tools for Monitoring Operational Security," IEEE Transaction on Software Engineering, Vol.25, pp.633-650, 1999.

[19] C. Choi, S. Kim, and W. Choi, "Survivality Modeling for Quantitative Security Assessment in Ubiquitous Computing Systems," Lecture Notes in Computer Science, Springer, Vol.3043, No.1, pp.207-214, May, 2004.

[20] 최창열, 김성수, "유비쿼터스 컴퓨팅의 신뢰성 모델링을 위한 정량적 분석법," 2004년 한국정보과학회 춘계학술발표대회, 한국정보과학회, 제31권 1호, pp.622-624, 2004.

[21] R. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321, Internet Engineering Task Force, 1992.

[22] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the Source," Proceedings of the 10th IEEE Inter- national Conference on Network Protocols, pp.312-321, 2002.

[23] S. Hunter and W. Smith, "Code-Red: a Case Study on the Spread and Victims of an Internet Worm," Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, Analysis and Synthesis, pp.273-284, 2002.

최 창 열

e-mail : clchoi@ajou.ac.kr

1999년 아주대학교 정보통신공학과(공학사)

2000년 아주대학교 정보통신전문대학원(공학석사)

2002년~현재 아주대학교 정보통신전문대학원 박사과정

관심분야 : 유비쿼터스 컴퓨팅, 오토노믹 컴퓨팅, 고가용성 시스템, 미들웨어 등

김 성 수

e-mail : sskim@ajou.ac.kr

1982년 서강대학교 전자공학과(공학사)

1984년 서강대학교 전자공학과(공학석사)

1995년 Texas A&M University 전산학과(공학박사)

1983년~1996년 삼성전자 수석연구원

2002년~2003년 Texas A&M University 교환교수

1996년~현재 아주대학교 정교수

관심분야 : 유비쿼터스 컴퓨팅 및 네트워크, 오토노믹 컴퓨팅, 결함허용 시스템 등