Journal of The Korea Society of Computer and Information
Vol. 18, No. 11, November 2013
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2013.18.11.125

SSL VPN기반의 행위·순서패턴을 활용한 접근제어에 관한 연구

장 은 겸*, 조 민 희*, 박 영 신*

A Study on Access Control Through SSL VPN-Based Behavioral and Sequential Patterns

Eun-Gyeom Jang*, Min-Hee Cho*, Young-Shin Park*

요 약

본 논문에서는 SSL VPN을 기반으로 사용자 인증과 사용자 단말의 무결성을 검증할 수 있는 네트워크 접근제어 기술을 제안한다. 사용자 단말이 VPN을 이용해 내부 네트워크에 접속할 때 사용자 인증과 사용자 단말의 보안 패치, 바이러스 백신 등의 보안 서비스를 확인하는 안전성 검사를 수행한다. 그리고 변종의 악성코드를 탐지하기 위해 사용자 단말의 윈도우 API 정보를 통한 행위패턴을 바탕으로 악성코드를 탐지하고, 탐지의 신뢰도를 높이기 위해 순서패턴의 유사도를 비교하여 변종의 악성코드를 탐지하여 외부의 보안 위협으로부터 시스템을 보호한다.

▸ Keywords : VPN, NAC, 행위패턴, 순서패턴, 악성코드

Abstract

In this paper, we proposed SSL VPN-based network access control technology which can verify user authentication and integrity of user terminal. Using this technology, user can carry out a safety test to check security services such as security patch and virus vaccine for user authentication and user terminal, during the VPN-based access to an internal network. Moreover, this system protects a system from external security threats, by detecting malicious codes, based on behavioral patterns from user terminal's window API information, and comparing the similarity of sequential patterns to improve the reliability of detection.

▸ Keywords : VPN, NAC, Behavioral patterns, Sequential patterns, malicious codes

•제1저자 : 장은겸 •교신저자 : 박영신
•투고일 : 2013. 10. 15, 심사일 : 2013. 11. 1, 게재확정일 : 2013. 11. 11.
* 대전대학교 컴퓨터공학과(Dept. of Computer Science, Daejeon University)

I. 서 론

유원격지 사용자와 서비스를 제공하는 시스템 간에 송수신 되는 기밀정보와 개인정보를 보호하기 위해 데이터를 암호화 하여 보안 프로토콜이 적용된 안전한 경로로 통신하는 가상사설망(VPN: Virtual Private Network) 기술이 사용되고 있다.

그러나 VPN 기술은 네트워크를 통해 송수신 되는 데이터의 기밀성과 무결성은 보장하지만, 공격에 취약한 사용자 단말에 대한 보안대책이 부족한 문제가 있다. 즉, 가정이나 출장지 등에서 기업의 내부 네트워크에 접근하는 경우 원격지 컴퓨터에 안티 바이러스나 개인 방화벽과 같은 보안 프로그램이 실행되더라도 운영체제의 보안 패치나 보안제품의 업데이트 등이 수행되지 않으면 공격에 노출될 수 있다[1].

이를 해결하기 위해 사용자 단말이 VPN을 이용해 내부 네트워크에 접속할 때 신원을 확인하는 사용자 인증과 정의된 보안정책을 준수했는지 여부를 검사하여 네트워크의 접속을 통제하는 네트워크 접근제어(NAC: Network Access Control) 기술이 요구되고 있다. NAC는 허가되지 않거나 웜·바이러스 등 악성코드에 감염된 PC나 노트북 등이 기업 내부의 네트워크에 접속하는 것을 원천적으로 차단해 전체 네트워크를 보호할 수 있는 기술이다[2].

따라서, 본 연구에서는 사용자가 원격지에서 접속하는 경우 기밀성을 보장할 수 있는 SSL(Secure Socket Layer) VPN과 내부 네트워크의 무결성을 보장할 수 있는 NAC 시스템을 연동하여 사용자 단말로부터 확산되는 보안위협을 미리 차단할 수 있는 사전 방어적인 네트워크 보안체계를 연구하였다.

II. 관련연구

2.1 SSL VPN 기술

SSL(Secure Socket Layer)은 네트워크 또는 인터넷 상에서 데이터나 메시지 전송을 안전하게 관리하기 위한 기술이다.

그림 1과 같이 SSL VPN은 SSL을 이용한 가상사설망으로 안전한 통신을 위해 클라이언트와 서버 사이에 송수신되는 패킷을 암호화한다.

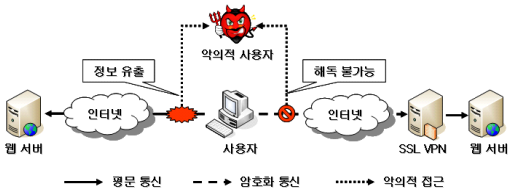

그림 1. SSL VPN을 이용한 암호화 통신
Fig. 1. SSL VPN-based encryption communication

암호화를 위해 먼저 상호인증이 이루어지고 인증된 키와 약속된 암호 알고리즘을 이용하여 보안 터널을 형성하고 이를 통해 통신이 이루어진다.

SSL VPN은 서비스를 요청하는 클라이언트와 서비스를 제공하는 서버간의 안전한 통신을 위하여 상호인증, 기밀성, 무결성 서비스를 제공하며 효율성을 위해 데이터 압축 기능도 제공한다. SSL은 보안서비스를 위해 표 1과 같이 표준화된 알고리즘을 제공하고 있다.

표 1. SSL에서 적용되는 알고리즘
Table 1. Algorithm applying to SSL

기 능	적용 알고리즘
상호인증	RSA, Diffie-Hellman
기밀성	DES, 3DES, AES, ARC4, AREA, SEED
무결성	MD5, SHA-1, HMAC-MD5, HMAC-SHA-1

- 상호인증: 서버와 클라이언트 간의 인증을 위한 세션 키를 생성하고 "Hello"라는 메시지를 교환하여 상호인증을 수행한다.
- 기밀성: 전송되는 메시지에 대한 다양한 형식의 암호화를 통해 타인에게 메시지 내용이 노출되지 않도록 한다.
- 무결성: 전송 메시지의 변조를 방지할 수 있도록 MD5 알고리즘을 이용하여 무결성을 보장한다.

SSL의 구조는 그림 2와 같이 SSL 동작에 대한 관리를 위해 사용되는 SSL Handshake 프로토콜, SSL Cipher Change 프로토콜, SSL Alert 프로토콜과 실질적인 보안 서비스를 제공하는 SSL Record 프로토콜로 이루어져 있다[3].

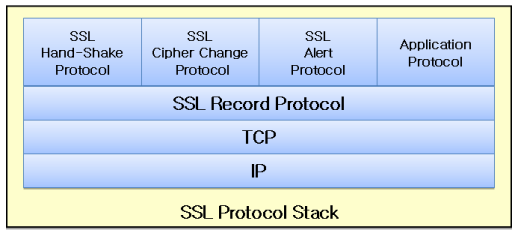

그림 2. SSL 프로토콜의 구조
Fig. 2. SSL protocol's structure

SSL 동작은 다음과 같은 순서로 이루어진다. 클라이언트는 서버에게 Client Hello 메시지를 전송한다. 서버는 클라이언트에게 Server Hello 메시지와 서버 인증서를 전송하고, 만약 클라이언트 인증서가 필요한 경우 인증서 요청도 함께 전송한다.

클라이언트는 암호화에 사용되는 세션 키와 함께 클라이언트에게 지원하는 Cipher Suite를 서버로 전송하고, 서버가 인증서를 요청한 경우에는 클라이언트의 인증서도 함께 전송한다. 클라이언트는 Finished 메시지를 서버로 전송하고 데이터 전송단계로 이동한다. 서버는 Cipher Suite를 받아들이고 Finished 메시지를 클라이언트로 전송한 후 데이터 전송단계로 이동한다. 상호 합의한 Cipher Suite에 의해서 암호화된 메시지를 교환한다.

2.2 NAC 시스템

네트워크 접근제어는 사용자 단말이 네트워크에 접속하는 순간부터 보안정책에 따라 정의된 일련의 절차를 통해 안전성이 검증된 단말이 네트워크에 접속할 수 있도록 하는 보안 플랫폼 기술이다. 즉, NAC는 사용자 인증, 사용자 단말에 대한 보안성 평가, 접근제어를 통합한 기술이다[4].

가트너 그룹은 그림 3과 같이 접근 호스트에 대한 지속적인 평가, 보안문제에 대한 대응, 네트워크 접근허용, 보안정책 준수에 대한 지속적인 모니터링 및 대응 등 업무 순환 절차에 관한 네트워크 보안 모델로 NAC 기술을 정의하고 있다[5]. 가트너 그룹은 NAC의 절차가 보안정책(Policy)에서 출발한다고 정의하였다. 보안정책(Policy)은 네트워크에 접속하기 전에 관리자가 무엇을 강제화할 것인지에 대한 보안 체크리스트이다. 일반적인 기업의 보안정책은 운영체제의 보안 패치가 최신으로 유지되며, 바이러스 백신 프로그램이 정상적으로 구동되고, 백신 업데이트가 제대로 실행되며, 개인 방화벽이 정확히 설정되어 운영되고 있는지를 확인한다.

베이스라인(Baseline)은 네트워크에 접속하려는 사용자 단말의 상태가 먼저 수립된 보안정책과 일치하는지 비교하는 단계이다. 이러한 절차는 안전한 네트워크를 보장하기 위해 LAN, WAN, 무선, IPsec, SSL VPN 등 사용자 단말의 네트워크 연결방식과 무관하게 수행되어야 한다.

베이스라인 평가 결과를 기반으로 접근제어는 사용자 단말에 미리 정의된 수준의 네트워크 접근권한을 부여한다. 예를 들어, 베이스라인을 따르는 사용자 단말의 경우는 전체 네트워크에 대한 접근 권한을 부여하고, 베이스라인을 따르지 않은 사용자 단말의 경우는 네트워크 접근을 차단하거나, 치료(Mitigate)를 위한 특정 네트워크 영역으로 접근하도록 유도할 수 있다.

사용자 단말이 정상적인 절차를 통해 네트워크에 접속한 다음 지속적인 보안정책의 유지 여부를 확인하기 위해 감시(Monitor)를 수행하며 문제가 있는 경우 전체 네트워크 관점에서 적절한 대응을 수행한다. 예를 들어 웜·바이러스의 활동 트래픽이 발생하는 경우 해당 사용자 단말을 네트워크로부터 강제로 분리하고 방화벽과 연계하여 해당 포트로의 접속을 차단한다[6]. 가트너 그룹의 정의 모델을 정리하면 표 2와 같다.

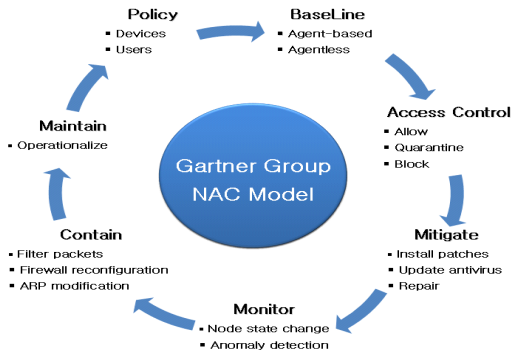

그림 3. 가트너 그룹의 NAC 정의 모델
Fig. 3. NAC model defined by the Gartner Group

표 2. 가트너 그룹의 정의 모델 설명
Table 2. Explanation of the model defined by the Gartner Group

정 의	역 할
정책(Policy)	사용자 인증과 사용자 단말 보안정책
베이스라인 (Baseline)	Agent 기반 또는 Non-Agent 기반
접근제어(Access Control)	허가와 차단은 물론 검역을 위한 완충지대 제공
치료(Mitigate)	정책이 요구하는 OS 또는 PC 보안 프로그램의 설치, 업데이트, 사용자 단말의 무결성 확보를 위한 치료 수단을 제공
감시(Monitor)	사용자 단말에 대한 상태와 비정상 상태에 대한 감시
유지(Contain)	패킷 필터와 ARP 조작 등을 포함한 포트기반 차단
관리(Maintain)	위의 순환이 반복되는 일련의 과정을 유지 관리

2.3 제안 악성코드 탐지방법

현재의 바이러스 백신 시스템은 패턴이 존재하지 않을 경우 탐지가 불가능한 오용 탐지기반의 진단법을 사용하고 있어 패턴의 보유수가 탐지율과 직결되고 있다. 백신 제조사들은 특정위치 및 문자열의 각 패턴생성 메커니즘을 비밀로 관리하고 있으며, 패턴의 보유수가 많아짐에 따라 오용탐지 오류도 자주 발생하고 있다[6,7].

또한, 보안 취약점에 대한 패치가 되지 않았거나 바이러스 백신 패턴에 포함되지 않는 신종 악성코드는 수분 내에 단위 네트워크상의 시스템을 감염시킬 수 있고 다른 지역의 네트워크로 확산될 수 있다. 그러므로 신종 및 변종 악성코드가 발생될 경우 기존 패턴기반의 탐지 및 치료 방법은 한계가 있다. 그리고 대부분의 안티 바이러스 제품들은 악성코드의 변종이 발생할 때마다 장시간의 분석과정을 거쳐 패턴을 생성하고 사용자에게 배포하고 있으나 확산이 진행된 다음 패턴이 적용되는 문제가 있다[8,9].

따라서, 본 논문에서는 사용자 단말의 보호를 위해 악성코드의 행위패턴에 따라 행위기반 탐지를 수행하고, 탐지의 신뢰도 향상을 위해 순서패턴을 이용하여 유사도를 비교한 다음 변종 악성코드를 탐지하는 방법을 제안한다.

행위패턴 생성을 위해 악성코드와 호출되는 함수간의 연관성을 분석하여 윈도우 API를 식별하였으며, API 중에서 주요자원에 접근하는 API를 대상으로 패턴을 생성하였다. 이러한 행위기반 탐지패턴은 주요자원에 접근하는 행위를 패턴으로 생성하고, 순서패턴은 API 호출순서에 의해 생성된다. 또한 탐지의 정확성을 위해 행위기반의 탐지를 바탕으로 악성코드의 원형패턴과 변종 API 순서패턴의 유사도를 측정하여 높은 유사도를 보일 경우 최종적으로 악성코드로 판단한다.

제안하는 방법은 악성코드의 종류 중 최초로 발생되는 원형 악성코드를 분석하여 생성된 행위패턴으로 이후 발생하는 여러 변종에 대해 탐지가 가능하며, 순서패턴에 따른 유사도 비교를 통해 오용탐지의 문제를 해결할 수 있다.

III. 보안 강화 네트워크 접근 제어 시스템

3.1 제안 시스템 구성 및 기능

제안 시스템은 그림 3과 같이 기밀성, 보안성, 무결성을 제공하기 위한 SSL VPN 기반 NAC 시스템으로 구성한다. 제안하는 NAC 시스템은 외부 지사나 내부 직원이 출장지에서 글로벌 망을 통해 내부자원에 접근하는 경우 권한이 있는 사용자가 가상의 사설망을 이용하여 접속하고, NAC 서버에서 사용자 단말의 무결성 검사를 통해 내부자원에 안전하게 접근할 수 있도록 보안서비스를 제공한다.

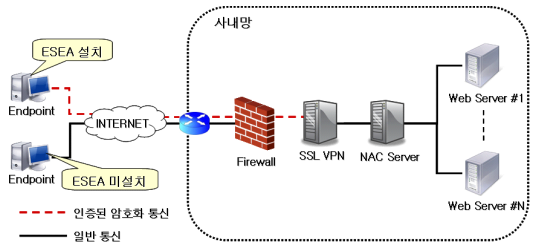

그림 3. SSL VPN 기반 NAC 시스템의 구성
Fig. 3. SSL VPN-based NAC system composition

- ESEA(Endpoint Security Enforcement Agent): ESEA는 SSL VPN과 NAC의 사용자 단말 기능을 통합하여 서버 접속시 SSL VPN의 사용자 인증 수행 후, 암호 알고리즘과 대칭키를 통해 터널링을 구성한다. 인증이 정상적으로 이루어진 후 사용자 단말은 NAC 서버와 연동하여 무결성 검사를 수행한다.
- SSL VPN 게이트웨이: 사용자 단말과 안전한 통신환경을 제공하기 위해 사용자 인증을 제공하고, 터널 구성에 필요한 정형화된 암호 알고리즘과 대칭키 관리를 지원한다. 또한 인증된 사용자가 이용할 수 있는 웹 서비스나 그룹에 대한 접근을 제어기능을 제공한다.
- NAC 서버: ESEA로부터 전송받은 사용자 단말의 상태정보에 대한 무결성 검사를 수행하며 취약점이 탐지된 경우 내부 네트워크의 접근을 제한한다. 또한 사용자 단말의 악성코드를 탐지하기 위해 악성코드의 행위패턴과 순서패턴을 생성하고 시그너처 형태로 변환하여 패턴을 저장한다.

그림 4에서는 개발 시스템의 처리 흐름을 네트워크 접속 시점만으로 나타내고 있지만, 네트워크 접근 이후에도 지속적으로 보안정책을 점검하고 점검결과에 따른 대응절차를 수행하는 순환구조를 이룬다.

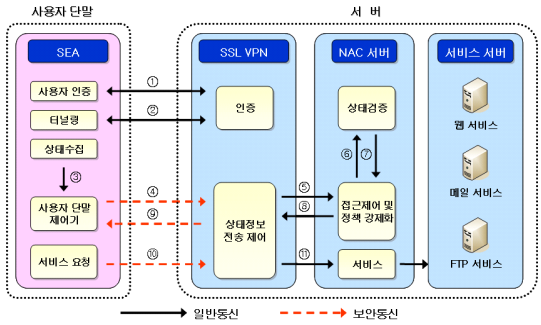

그림 4. 제안시스템 아키텍처 흐름도
Fig. 4. Architecture flow of the proposed system

① SSL VPN 서버와의 접속을 시도한다. 인증서를 통한 인증 과정을 거친다.
② 사용자가 인증되면 지정된 암호 알고리즘을 통해 터널을 구성한다.
③ 사용자 단말의 보안패치, 바이러스 백신 프로그램 정상작동, 백신 업데이트 등 각 보안 상태를 점검하고, PE 파일이 실행될 경우 윈도우 API를 추적하여 해당 정보를 사용자 단말 제어기로 전송한다.
④ 상태수집기로 부터 보안 상태를 최종 수집하여 SSL VPN으로 전송한다.
⑤ 인증된 사용자인지 확인 후 ④번에서 보낸 보안 상태정보를 NAC 서버로 전송한다.
⑥ 전달된 보안 상태정보로부터 사용자 단말의 보안상태 점검을 요청한다.
⑦ 상태 검증을 통해 전달받은 점검 결과를 수집하여 접근 허용여부를 판단하여 사용자 접근을 제어한다.
⑧ 점검 결과 및 접근 허용여부를 SSL VPN으로 전달한다.
⑨ 전달받은 접근권한을 설정하고 결과를 사용자 단말에 전달한다.
⑩ ⑪ 보안상태가 검증된 사용자에 한하여 정상적으로 네트워크 접속을 통해 서비스를 이용할 수 있도록 한다.

3.2 사용자 단말의 보안강화

(1) 사용자 단말의 상태정보

1) 안티 바이러스와 보안패치 정보

사용자 단말이 안전하다는 것을 확인하기 위해 안티 바이러스 정보가 요구된다. 표 3은 사용자 단말의 무결성 검사를 하기 위한 상태정보 항목이며 이 정보는 NAC 서버로 전송되어 평가된다.

표 3. 보안정책 검사 항목
Table 3. Security policy inspection items

보안정책 항목		상세 점검 방안
안티 바이러스	설치유무	설치관련 특정 레지스트리 검사
	실행여부	서비스 또는 프로세스 실행 검사
	업데이트	업데이트 관련 레지스트리 검사
운영체제 보안패치	서비스팩 업데이트	서비스팩 업데이트 관련 레지스트리 검사
	패치유무	해당 보안패치의 레지스트리 검사
	업데이트	업데이트 관련 레지스트리 검사

2) 윈도우 API 정보 추출

윈도우 실행파일인 PE 파일 구조에서 현재 사용하기 위해 내재된 API를 추출하는 방법은 섹션 테이블 내에 있는 IMAGE_IMPORT_DESCRIPTOR 구조체로 부터 시작되는 삽입 주소 테이블(IAT: Imported Address Table)을 이용한다.

그림 5는 Sasser worm의 PE 구조를 분석하여 재구성한 것이다. Sasser의 경우 ADVAPI32.DLL에 있는 AbortSystem- ShurdownA, RegOpenKeyA, RegSetValueA, RegCloseKey를 사용하고 있다.

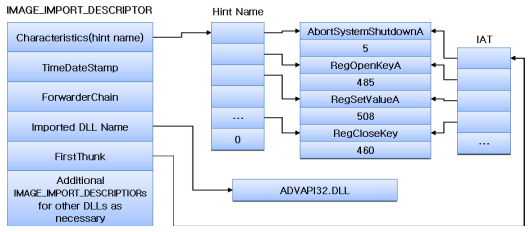

그림 5. IMAGE_IMPORT_DESCRIPTOR 자료구조
Fig. 5. Behavior-based malicious code detection structure

IAT를 얻은 후 각 API를 추적하기 위하여 API가 위치한 상대적 가상 주소 (RVA: Relative Virtual Address)를 계산해야 한다. RVA는 이미지가 해당 프로세스의 가상 주소 공간 내에 적재 되었을 때 그 시작 주소에 대한 상대적 번지이다.

RVA는 식 1과 같이 삽입된 디스크립터 주소, PE 구조 내에 있는 이미지 베이스, API의 위치를 가리키는 오프셋을 합하여 계산한다. RVA는 어셈블리 CALL 명령의 목적지 주소

에서 API를 추출할 수 있는 중요한 요소이다.

RVA = Address(Import Descriptor) + Image_base + Offset (1)

(2) 행위기반 악성코드 탐지

제안하는 행위기반 악성코드 탐지는 행위패턴과 순서패턴을 거쳐 악성코드의 여부가 결정된다. 먼저 행위패턴은 최상단의 패턴구조로 프로세스, 파일시스템, 기타 자원에 관련된 패턴이 정의되며, 일반적으로 바이러스, 웜, 백도어 등 특징이 상이한 악성코드의 패턴이 행위패턴에서 결정된다. 또한 세부적인 유형을 결정하는데 주로 네트워크 또는 자원 중 Mutex를 설정하는 패턴이 사용된다.

순서패턴은 행위패턴을 통해 탐지되는 결과의 정확도를 향상시키기 위해 원형의 API 순서와 추출된 API 호출 순서의 유사도를 비교하여 악성행위의 정도를 판단한다. 특정 유형의 악성코드에 대해 API 순서 유사도 정도가 높을 경우 최종적으로 악성코드로 판별하여 사용자에게 경고하고 네트워크 접근을 차단한다.

악성코드 행위 탐지모듈은 그림 6과 같이 실시간 감시 모드로 동작하면서 모든 프로세스의 악성행위 여부를 조사한다. 조사대상은 프로세스, 네트워크, 레지스트리, 파일 등으로 해당자원에 위치한 악성행위의 연관성을 통해 탐지한다.

실시간 감시모드는 실행되는 PE 파일의 API 정보를 추적하여 API와 인자 필터링을 거친 후 악성코드와 동일한 행위를 수행하는지 판단하여 악성코드일 경우 시그너처 DB의 API 순서와 실행파일에서 생성된 API 순서의 유사도를 비교하여 최종적으로 악성코드 여부를 판단한다.

(3) 악성코드 유사도 분석

본 논문에서 제안하는 악성코드 유사도 비교방법은 그림 7과 같이 ESEA에서 전송한 실행파일의 윈도우 API 호출 순서와 NAC 서버에 등록된 악성코드의 API 순서를 비교하여 유사도를 계산하고 임계값이 넘을 경우 악성코드로 최종 판단한다.

순서패턴은 하나의 집합으로 표현될 수 있으며 집합에서 각 원소들은','로 각 API를 구분한다. 순서패턴의 교집합은 단순히 서로 중복되는 원소들의 개수로만 계산되는 것이 아니라 순서에 대한 처리 절차가 포함되어야 한다. 이러한 문제를 해결하기 위하여 본 논문에서 사용되는 유사도 비교 알고리즘은 동적 프로그래밍 기법을 응용하여 구현하였다.

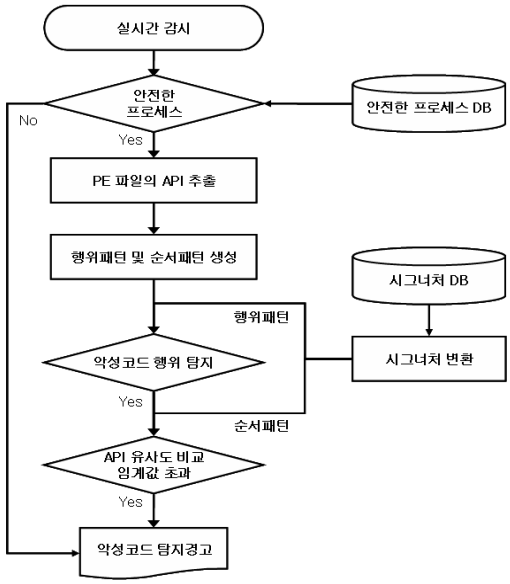

그림 6. 행위기반 악성코드 탐지 구조
Fig. 6. Behavior-based malicious code detection structure

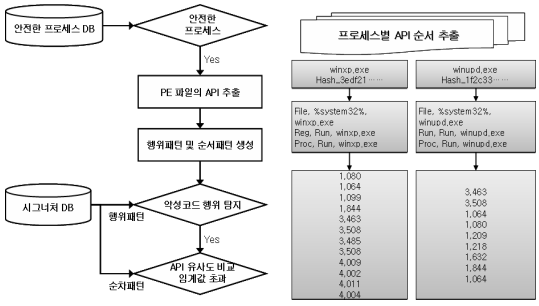

그림 7. 유사도 비교를 위한 구조
Fig. 7. Structure for similarity comparison

시그너처 DB에 정의된 악성코드와 ESEA에서 전송한 각 API 호출 순서가 유사한지 판단하는 API의 유사도 계산식은 식 2와 같다.

$$Sim(S_i, S_j) = \frac{|S_i \wedge S_j|}{|S_i \vee S_j|} \quad (2)$$

그림 8과 같이 교집합 계산을 위하여 먼저 가중치 행렬을 구성하였다. 순서패턴 Si={W, A, N, D, E, R, S}와 순서패턴 Sj={W, A, D, D, E, R, R, Y, S}의 교집합을 구하기 위하여 가중치 행렬을 생성하고 각 교차점의 가중치를 계산하는 방법을 보였다. 최종 교집합의 개수는 이들 가중치들로 구

할 수 있다.

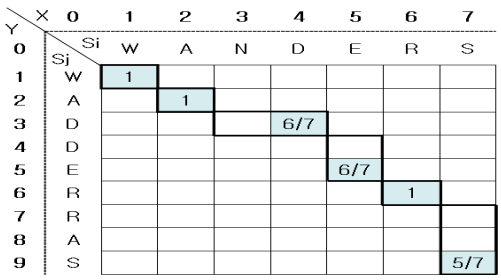

그림 8. 가중 행렬
Fig. 8. Weight Matrix

식 3은 좌표 (X, Y)에서의 가중치(W)를 구하기 위한 함수이다. X와 Y좌표 상에 놓인 두 순서들의 공통된 요소를 교집합의 원소로 정하고, 서로 교차된 지점의 가중치는 바로 이전에 두 순서의 요소가 교차된 지점으로 부터 거리 값에 전체 윈도우의 크기를 고려하여 결정된다. 윈도우의 크기는 패턴의 길이와 동일하며 두 좌표의 거리는 윈도우의 크기를 초과할 수 없으므로 윈도우 크기를 고려하여 윈도우 안에서 어느 정도 가까운 거리에 있는지를 가중치로 설정한다.

$$W(X, Y) = 1 - \frac{((X - X_{pre})(Y - Y_{pre})) - 1}{W_p} \quad (3)$$

Wp = 패턴의 길이, 윈도우 크기

교차점 (X, Y)가 이전의 교차점(Xpre, Ypre)로부터 떨어진 거리는 두 교차점 사이의 빈 블록의 개수에 비례한다. 만일 이전의 교차점이 없으면 초기 교차점은 좌표(0, 0)로 가중치는 0이며, 다음 세 가지 조건을 만족해야 한다.

- 조건 1: 현재 일치된 좌표를 (X, Y)로 놓고, 어떤 임의의 이전 일치된 좌표를 (Xany, Yany)로 놓도록 한다.
- 조건 2: (Xpre, Ypre)가 (X, Y)의 이전 일치된 좌표라고 할 때, Xpre는 X와 같지 않아야 하고 Ypre는 Y와 같지 않아야 한다.
- 조건 3: 만일, 좌표상의 X의 순환이 끝나고 Y의 순환이 끝나지 않았을 경우 X좌표를 0으로 하여 나머지 Y좌표에 대한 일치작업을 시작한다.

이전 교차점(X_{pre}, Y_{pre})을 구하기 위한 규칙은 다음과 같다.

- 규칙 1: 이전 일치된 좌표는 |Xany × Yany| 값이 |X × Y| 값보다는 작은 모든 값들 중에서 최대값을 가지는 좌표여야 한다.
- 규칙 2: 만일, 두 개 이상의 이전 일치된 좌표에서 동일한 |Xany × Yany| 값을 가진다면, 그들 중 |Xany + Yany| 값이 더 큰 좌표를 이전 일치된 좌표로 선택한다.
- 규칙 3: 만일 |Xany × Yany| 값과 |Xany + Yany| 값이 모두 같은 두 개 이상의 이전 일치된 좌표가 발견된다면 다음 두 단계의 계산 과정을 다시 따른다.
- X좌표와 Y좌표가 같을 때에는 고려할 필요 없이 어떤 이전 일치된 좌표를 선택하더라도 가중치는 같다.
- X좌표가 Y좌표와 같지 않을 때에는 각 이전 일치된 좌표로부터 |X - Xany| 값과 |Y - Yany| 값을 계산하여 둘 중 더 큰 값을 추출하여 모든 추출된 값 중 최소값을 갖는 좌표를 이전 일치된 좌표로 선택한다.

앞에서 구했던 가중치를 이용하여 최종 교집합의 수를 계산하는 공식은 다음과 같다.

$$|S_i \wedge S_j| = \sum_{S \in M} \frac{W_p}{D_{\max}} \quad (4)$$

식 4에서 M은 모든 일치된 교차점의 좌표들이고, Ws는 교집합의 원소 S가 가지는 모든 가중치들의 합이며, Dmax는 두 순서패턴에서 원소 S가 중복되는 개수들의 최대값을 의미한다.

IV. 제안 시스템 실험 및 분석

4.1 실험 환경

본 논문에서 제안하는 접근제어 시스템의 실험환경은 그림 9와 같이 크게 내부 네트워크 접속을 요청하는 사용자 단말인 ESEA, 사용자 인증을 위한 SSL VPN 게이트웨이, 그리고 사용자 단말의 무결성 검증을 위한 NAC 서버로 구성된다. 그리고 내부 네트워크는 사설 IP 주소를 사용하였고, 외부 네트워크는 글로벌 IP 주소를 할당하여 실험하였다.

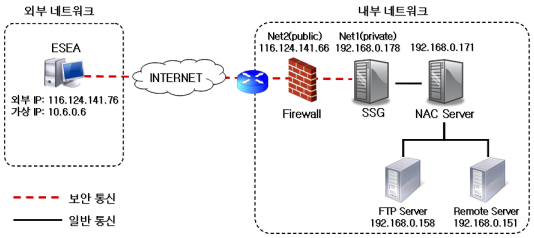

그림 9. SSL VPN 기반 NAC 시스템 운영환경
Fig. 9. SSL VPN-based NAC system operation environment

4.2 실험 시나리오

(1) SSL VPN을 이용한 사용자 인증

SSL VPN에서는 관리자 모드와 사용자 모드로 분리하여 서비스를 제공한다. 관리자 모드에서는 SSL VPN 운영을 위한 시스템 환경설정, 보안정책 설정, 사용자 관리 등을 수행하고, 사용자 모드에서는 ESEA 설치 및 사용자 정보수정 등의 기능을 수행한다.

1) 관리자 모드

관리자 모드에서는 SSL VPN 운영을 위한 시스템 설정, SSL VPN 동작관리, 시스템 보안정책 설정, 관리자 및 사용자에 대한 관리 기능 등을 수행한다.

- 관리자 로그인: 관리자는 일반 PC에서 웹을 통해 SSL VPN의 SMT에 접속한다.
- 시스템 설정: 초기 SSL VPN 구동을 위한 시스템 설정으로 네트워크 IP 및 DNS 등의 정보를 설정한다.
- 보안정책 설정: 내부 서버 및 사용자 그룹을 생성하고, 각 그룹별 보안정책을 설정한다.
- 사용자 관리: 내부 자원에 접근할 수 있는 사용자 ID 및 Password를 관리한다.

2) 사용자 모드

사용자 모드는 사용자가 SSG 접속을 통해 보안서비스를 제공받기 위한 모드로써 보안서비스를 제공받기 위해서는 그림 10과 같은 절차가 필요하다.

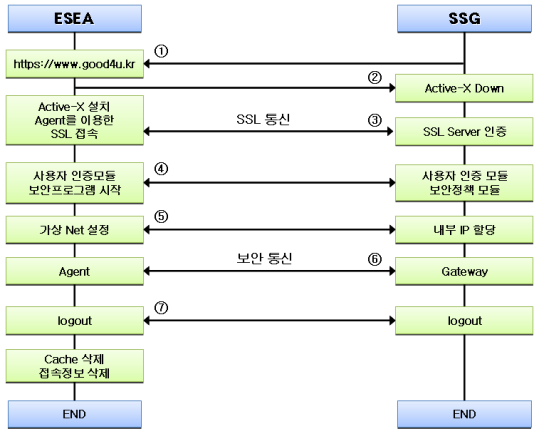

그림 10. 사용자 모드 서비스 시나리오
Fig. 10. User mode service scenario

① SSG 접속: 사용자 단말은 인증 및 보안 프로그램의 설치를 위해 SSG에 접속한다.
② Active-X 설치: 사용자 단말에 ESEA 프로그램 설치 유무를 확인한 후 Active-X를 통해 ESEA 프로그램을 설치한다.
③ SSL Protocol 수행: 사용자 단말은 ESEA 설치한 다음 SSL 프로토콜을 이용하여 보안접속을 시도한다.
④ 인증 절차: ESEA는 SSG에 사용자 인증을 수행한다. 인증을 수행한 다음 보안터널 형성을 위한 환경정보를 제공받는다.
⑤ 내부 사설 IP 할당: ESEA는 인증 수행 후 제공받은 환경정보를 바탕으로 내부 자원에 접근할 수 있는 내부 사설 IP를 할당 받는다.
⑥ 보안서비스 제공: ESEA는 SMT의 보안정책에 따라 SSG와 보안터널을 형성하고 사용자 단말의 무결성 검사를 위해 NAC 서버와 연동된다.
⑦ logout: 보안 통신을 종료한다.
⑧ Cache 제거: 접속 정보를 제거한다.

일반적인 보안서비스 절차에서는 사용자 로그인을 통해 사설 인증서 및 환경정보, 사설 IP를 할당받아 보안터널을 형성하고, 로그아웃을 하면 할당받은 정보를 모두 제거하여 재사용을 방지한다.

(2) 결과 분석

NAC 서버에 구현된 악성코드 탐지 모듈에 대한 분석을 하고자 한다. 패턴생성 및 생성된 패턴의 탐지결과를 분석하기 위해 각각 메일(Mail)형, 취약점(Vulnerability)형, 봇(Bot)형, 메신저(Messenger)형으로 패턴을 생성하여 변종

에 대한 탐지 결과를 분석하였으며, 순서패턴의 유사도 탐지 결과를 비교 분석 하였다.

악성코드의 행위패턴을 통해 1차적으로 악성코드의 여부를 판단할 수 있으나 정상적인 메일 프로그램을 악성행위를 수행하는 것처럼 조작하는 경우에도 악성코드로 탐지될 수 있는 문제가 있다. 예를 들어 윈도우즈에서 기본적으로 제공되는 Outlook Express를 "C:\Windows \System32"로 복사한 후 레지스트리의 시작 위치에 등록하고, 실행할 경우 25번 포트를 이용하여 메일서버로 접속하기 때문에 행위패턴 탐지에서 악성코드로 판별한다.

그러나 순서패턴에 의한 유사도 비교에서 실제 레지스트리에 등록하거나 System32로 파일을 복사하려는 API 순서가 발생하지 않기 때문에 유사도가 0이 되어 악성코드가 아님을 판단을 할 수 있다.

1) 메일(Mail)형

표 4는 Bagle 웜에 대한 API 순서 추출결과이다. Bagle.o는 파일 감염 및 백신 관련제품의 삭제에 대한 패턴이 추가되어 초기의 파일 감염능력이 없거나, 백신 관련제품의 삭제 기능이 없는 웜의 경우 낮은 유사도를 보였으며, 특정 프로그램의 경우 시작위치에 설정되고, 시스템 폴더에 파일을 복사한 뒤 실행하는 등 웜의 패턴을 동일하게 따르는 부분으로 인해 잘못된 탐지가 발생하였다.

표 4. Bagle 웜의 API 순서 추출 결과
Table 4. API sequence extraction result of Bagle worm

순서	원형(a)	b~n	o~z	Bagle.af	Bagle.al~au
1	3.463	3.463	3.463	3.463	3.463
2	3.508	3.508	3.469	3.469	3.469
3	1.064	1.064	3.463	3.463	3.463
4	1.080	1.080	3.508	3.508	3.508
5	1.209	1.209	1.064	1.064	1.084
6	1.218	1.218	1.209	1.080	1.064
7	1.632	1.632	1.218	1.209	1.209
8	1.844	1.844	1.632	1.218	1.218
9	1.064	1.064	1.844	1.632	1.632
10	0.861	0.861	1.064	1.844	1.844
11			0.861	1.064	1.064
12				0.861	0.861
유사도	1.00	1.00	0.63	0.78	0.66

2) 취약점(Vulnerability)형

표 5는 Sasser 웜에 대한 API 순서 추출 결과이다. Sasser 15872.C부터는 특정 파일을 새롭게 생성하고자 하는 동작이 발생하지 않음으로 인해 API 순서에서 삭제되어 유사도에 영향을 미치고 있지만 취약점형의 웜이 일반적으로 수행하는 동작을 그대로 유지하여 0.9이상의 유사도로 변종 Sasser에 대한 탐지가 가능하다.

표 5. Sasser 웜의 API 순서 추출 결과
Table 5. API sequence extraction result of Sasser

순서	원형(15872)	15872.B	15872.C	15872.D	16384	74752
1	1.080	1.080	1.064	1.064	1.064	1.064
2	1.064	1.064	1.099	1.099	1.099	1.099
3	1.099	1.099	1.844	1.844	1.844	1.844
4	1.844	1.844	3.463	3.463	3.463	3.463
5	3.463	3.463	3.508	3.508	3.508	3.508
6	3.516	3.516	3.485	3.485	3.485	3.485
7	3.485	3.485	3.508	3.508	3.508	3.508
8	3.508	3.508	4.009	4.009	4.009	4.009
9	4.009	4.009	4.002	4.002	4.002	4.002
10	4.002	4.002	4.011	4.011	4.011	4.011
11	4.011	4.011	4.004	4.004	4.004	4.004
12	4.004	4.004				
유사도	1.00	1.00	0.95	0.95	0.95	0.95

3) 봇(Bot)형

표 6은 봇형 웜 중 Agobot 웜에 대한 API 순서 추출 결과이다.

표 6. Agobot 웜의 API 순서 추출 결과
Table 6. API sequence extraction result of Agobot Worm

순서	원형	64000.D	441344	138752.B	197120.M	99328	98417
1	1.064	1.064	1.064	1.064	1.064	1.064	1.064
2	1.099	1.099	1.099	1.099	1.099	1.099	1.099
3	3.463	3.463	3.463	3.463	3.463	3.463	3.463
4	3.508	3.508	3.508	3.508	3.508	3.508	3.508
5	1.431	1.431	3.463	1.431	3.463	3.463	1.431
6	1.080	2.651	3.469	1.080	3.469	3.508	1.080
7	2.651	1.632	1.431	2.651	1.431	3.463	2.651
8	1.632	1.844	2.651	1.632	1.080	3.469	4.011
9	1.844	4.011	1.844	1.844	2.651	1.431	4.052
10	4.011	4.052	4.011	4.011	1.632	1.080	4.004
11	4.052	4.004	4.052	4.052	1.844	2.651	4.009
12	4.004	4.009	4.004	4.004	4.011	1.632	
13	4.009		4.009	4.009	4.052	1.844	
14					4.004	4.011	
15					4.009	4.052	
16						4.004	
17						4.009	
유사도	1	0.86	0.74	1	0.86	0.79	0.80

변종은 원형에 비해 새롭게 특정 파일을 생성하고자 하는 동작, 레지스트리에 백신 및 보안제품의 레지스트리 정보를

삭제, 보안 관련 프로세스 종료, 레지스트리 기본 실행 및 서비스 시작 위치에 등록 등 여러 가지 변수에 의해 API 순서 유사도에 영향을 미치고 있다. 봇형 웜은 원형과 비교하여 0.7이상의 유사도를 보이고 있어 변형된 악성코드의 탐지가 가능함을 알 수 있다.

4) 메신저(Messenger)형

표 7은 메신저형 웜의 대표적인 Bropia 웜에 대한 API 순서 추출 결과이다. 메신저형 웜은 웜의 특징보다는 웜을 확산시키기 위한 경로로 사용되는 Dropper의 특징이 강하기 때문에 순서패턴이 단순하다. 동일한 Bropia 변종은 유사도가 1로 정확히 탐지가 가능하며, 유사종인 Sumom도 임계값으로 설정된 0.6 범위 안에 포함되므로 탐지가 가능하다.

표 7. Bropia와 Sumom의 API 순서추출결과
Table 7. API sequence extraction results of Bropia and Sumom

순서	B159744	B188928	S36352	S17429	S23476
1	1.064	1.064	1.064	1.064	1.064
2	1.099	1.099	1.099	1.099	1.099
3	3.463	3.463	3.463	3.463	3.463
4	3.508	3.508	3.508	3.508	3.508
5	1.080	1.080	1.632	1.632	1.632
6			1.844	1.844	1.844
7			1.080	1.080	1.080
유사도	1	1	0.66	0.66	0.66

5) 원형 패턴생성을 통한 변종탐지 결과

악성코드 탐지 패턴생성과 관련하여 최초 원형의 악성코드로 패턴을 생성하는 것은 다음에 발생하는 변종 악성코드를 탐지하는 것과 밀접한 관련이 있다.

그림 11은 최초 원형인 V1 패턴을 생성하여 다른 변종 웜을 탐지하는 유사도를 보인 것이다. V2, V3, …, V7은 단순한 변종이 아닌 순서패턴에 큰 차이를 보이는 변종의 발생을 의미한다. 즉, 악성코드 기능상 추가가 발생하여 API 순서상 큰 변화가 발생한 변종이다. 실험대상인 메일형, 취약점형, 봇형, 메신저형 모두 유사도 측면에서 최초 발생한 원형과 비교하여 0.6이상의 유사도로 변형 악성코드의 탐지가 가능하다.

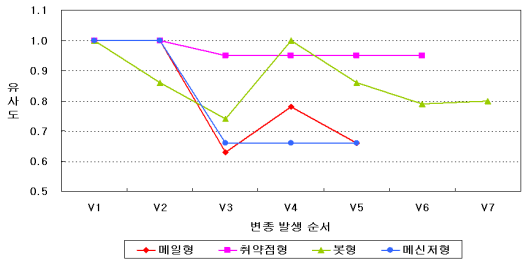

그림 11. 원형 패턴을 통한 변종 악성코드 탐지 결과
Fig. 11. Malicious code detection result through circular patterns

(3) 유사도 비교 분석

1) 유사도 측정 알고리즘 비교

그림 12는 변종 악성코드의 API 순서가 가장 많이 발생한 Agobot을 대상으로 유사도 측정 방식을 비교한 결과이다. 피어슨(Pearson) 상관관계 측정법과 스피어만 (Spearman) 상관관계 측정법은 악성코드의 변형에 따라 유사도 차이가 크게 나타났으며, 제안된 유사도 측정 방법은 변형 악성코드의 유사도가 모두 0.7이상으로 큰 차이가 없다.

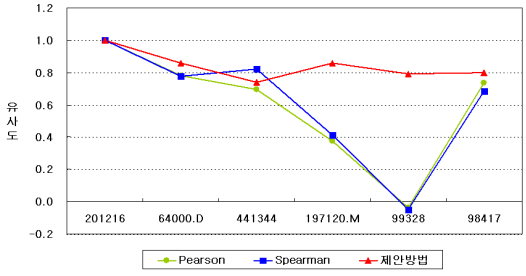

그림 12. 유사도 측정 알고리즘 비교
Fig. 12. Comparison of the similarity measuring algorithm

2) 유사도 임계값에 따른 탐지율

그림 13과 같이 각 유형별 유사도의 임계값 설정하기 위해 1부터 0까지의 순서로 탐지율을 분석하였다.

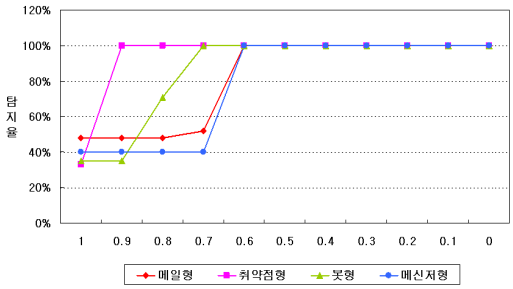

그림 13. 유사도 수치에 따른 탐지율
Fig. 13. Detection ratio depending on the value of similarity

유사도 임계값이 취약점형은 0.9이상, 메일형과 메신저형은 0.6이상, 봇형은 0.7이상일 때 탐지율이 100%임을 확인할 수 있다.

3) 동종의 다른 유형탐지

각 종류의 악성코드에 대해 Bagle, Mydoom, Netsky 등 다양한 유형의 웜들이 발생하고 있으므로 동일한 종류의 악성코드일 경우 순서패턴 유사도를 통해 탐지가 가능한지 확인하고자 한다.

본 논문에서는 가장 많은 변종을 발생시킨 Bagle을 통한 패턴 생성결과를 바탕으로 다른 유형의 웜이 탐지되는 결과를 보인다. Bagle 웜을 기준으로 비교했을 때 표 8과 같이 임계값으로 설정한 0.6을 모두 넘기 때문에 탐지가 가능하다. 또한 Mydoom과 Netsky는 서로 0.9를 넘는 유사도를 보여 한 유형의 패턴으로 여러 종류의 웜에 대한 탐지가 가능함을 알 수 있다.

표 8. 다른 유형의 웜 유사도 비교결과
Table 8. Comparison results of worm similarity of other types

구분	Bagle	Mydoom	Netsky
Bagle	1	0.77	0.70
Mydoom	0.77	1	0.93
Netsky	0.70	0.93	1

V. 결론

본 논문에서는 네트워크 전체의 보안성 향상을 위해 사용자 단말에 ESEA를 설치하고, SSL VPN 기반의 NAC 시스템과 연동하여 사용자 시스템을 안전하게 보호할 수 있는 네트워크 접근제어 기법을 제안하였다.

SSL VPN과 NAC를 활용하여 사용자 인증과 사용자 시스템의 보안 안전성을 분석하여 내부 네트워크 접근을 제어한다. 또한 변종의 악성코드 탐지를 위해 행위기반의 탐지패턴을 생성하여 악성코드를 탐지하였다. 이 방식은 실행 프로그램의 동적 감시를 통해 윈도우 API의 행위패턴과 유사도 비교를 위한 순서패턴을 생성하여 악성코드 여부를 판단한다.

제안된 탐지방법은 악성코드 유형별로 한 개의 행위패턴으로 악성코드를 탐지하고 순서패턴의 유사도 임계값을 설정하여 탐지의 신뢰성을 높였으며, 패턴이 업데이트 되지 않은 시스템에서도 변종의 악성코드를 탐지할 수 있는 장점이 있다.

향후, 모바일이나 무선 네트워크 환경에 맞는 NAC의 연구와 정상 프로그램의 안전한 프로세스 분석을 통해 다양한 변종 악성코드의 행위·순서패턴에 적용한다면 보다 안전한 네트워크 서비스가 될 것으로 기대한다.

참고문헌

[1] Mark S. Kadrich, "Endpoint Security", Addision-Wesley, 2007.
[2] Karen O'Donoghue, "Network Access Control", Interop Labs, May 2006.
[3] Joseph Steinberg, Tim Speed, J. Steinberg, T. Speed, "SSL VPN: Understanding, evaluating and planning secure, web-based remote access", Packt Publishing, 2005.
[4] "Anyclick NAC Technology Paper", UNET System, 2006.
[5] Lawrence Orans, "Gartner's Network Access Control Model", Gartner, August 2005.
[6] Eun-Gyeom Jang, "A study on block-based recovery of damaged digital forensic evidence image", Springer MTA, March, 2012.
[7] Nam-Youl Park, Yong-Min Kim, Bong-Nam Noh, "A Behavior based Detection for Malicious Code Using Obfuscation Technique", KIISC, Vol. 16, No. 3, June. 2006.
[8] Eun-Gyeom Jang, "A Study on Comparison of Road Surface Images to Provide Information on Specific Road Conditions", KSCI, Vol. 17. No. 4, April. 2012.
[9] Hyo-Nam Kim, Jae-Kyoung Park, Yoo-Hun Won, "A Study on the Malware Realtime Analysis

Systems Using the Finite Automata", KSCI, Vol. 18. No. 5, May. 2003.

저 자 소 개

장 은 겸

2007: 대전대학교 컴퓨터공학과 공학박사

2009 ~ 현재: 대전대학교 컴퓨터공학과 겸임교수

관심분야: 시스템 접근제어, 모바일앱, 컴퓨터 포렌식스, DRM

Email : janegu@dju.ac.kr

조 민 희

2001: 목원대학교 정보통신공학과 공학사.

2003: 목원대학교 정보통신공학과 공학석사.

2005: 서원대학교 전자계산교육과 교육학석사.

2012: 대전대학교 컴퓨터공학과 공학박사 수료.

관심분야: 정보보안, 컴퓨터 포렌식스

Email : chominhee@lycos.co.kr

박 영 신

2002: 대전대학교 컴퓨터공학과 공학사.

2004: 대전대학교 컴퓨터공학과 공학석사.

2009: 대전대학교 컴퓨터공학과 공학박사

관심분야: 컴퓨터 포렌식스, 정보보안

Email : good4u23@naver.com