JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 24 No. 10, pp. 25-32, October 2019
https://doi.org/10.9708/jksci.2019.24.10.025

Attack Surface Expansion through Decoy Trap for Protected Servers in Moving Target Defense

Tae-Keun Park*, Kyung-Min Park**, Dae-Sung Moon**

*Professor, Dept. of Applied Computer Engineering, Dankook University, Yongin, Korea
**Researcher, Information Security Research Division, ETRI, Daejeon, Korea
**Principal Researcher, Information Security Research Division, ETRI, Daejeon, Korea

[Abstract]

In this paper, we propose a method to apply the attack surface expansion through decoy traps to a protected server network. The network consists of a large number of decoys and protected servers. In the network, each protected server dynamically mutates its IP address and port numbers based on Hidden Tunnel Networking that is a network-based moving target defense scheme. The moving target defense is a new approach to cyber security and continuously changes system's attack surface to prevent attacks. And, the attack surface expansion is an approach that uses decoys and decoy groups to protect attacks. The proposed method modifies the NAT table of the protected server with a custom chain and a RETURN target in order to make attackers waste all their time and effort in the decoy traps. We theoretically analyze the attacker success rate for the protected server network before and after applying the proposed method. The proposed method is expected to significantly reduce the probability that a protected server will be identified and compromised by attackers.

▸ **Key words**: Network-based moving target defense, attack surface, cyber security, decoy trap

[요 약]

본 논문에서는 보호대상 서버 네트워크에 디코이 트랩을 통한 공격 표면 확장의 적용 방법을 제안한다. 보호대상 서버 네트워크는 많은 수의 디코이들과 보호대상 서버로 구성되며, 각 보호대상 서버는 Hidden Tunner Networking이라는 네트워크 기반 이동 표적 방어 기법에 따라 IP 주소와 포트 번호를 변이한다. 이동 표적 방어는 공격을 막기 위하여 지속적으로 시스템의 공격 표면을 변경하는 사이버 보안에서의 새로운 접근방법이다. 공격 표면 확장은 공격을 막기 위해 디코이와 디코이 그룹을 활용하는 접근방법이다. 제안하는 방법에서는 공격자가 디코이 트랩에서 공격자의 모든 시간과 노력을 허비하도록 커스텀 체인과 RETURN 타켓을 사용하여 보호대상 서버의 NAT 테이블을 수정한다. 본 논문에서는 제안하는 방법이 적용되기 전과 후에 보호대상 서버 네트워크에서의 공격자 성공률을 수식으로 계산한다. 제안하는 방법은 보호대상 서버가 공격자에 의해 식별되고 공격당할 확률을 현저히 줄일 것으로 기대된다.

▸ **주제어**: 네트워크 기반 이동 표적 방어, 공격 표면, 사이버 보안, 디코이 트랩

• First Author: Tae-Keun Park, Corresponding Author: Tae-Keun Park
*Tae-Keun Park (tkpark@dankook.ac.kr), Dept. of Applied Computer Engineering, Dankook University
**Kyung-Min Park (kmpark@etri.re.kr), Information Security Research Division, ETRI
**Dae-Sung Moon (daesung@etri.re.kr), Information Security Research Division, ETRI
• Received: 2019. 09. 18, Revised: 2019. 10. 07, Accepted: 2019. 10. 07.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

네트워크 및 시스템 보안 향상을 위한 일반적인 접근 방법은 공격 표면 (Attack Surface)을 줄이는 것이었다. 하지만, 복잡한 시스템의 경우, 공격 표면을 줄이려는 접근 방법이 오히려 부적절한 결과를 유발할 수 있다 [1]. 따라서 최근에는 공격 표면을 줄이는 기술보다 공격 표면을 지속적으로 변경하는 기술에 대한 관심이 증가하고 있다. MTD (Moving Target Defense) 기술은 공격 표면을 지속적으로 변경하는 기술이다 [2-3].

MTD는 시간의 변화에 따라 보호대상 시스템의 다양한 특징들을 변경하여 각종 사이버 공격을 차단하는 능동적 사전 보안 기술이다 [3]. MTD의 기술을 세분화하면, 네트워크의 특징과 설정을 바꾸는 네트워크 기반 MTD 기술, 시스템 플랫폼의 특징을 바꾸는 플랫폼 기반 MTD 기술, 런타임 환경 또는 응용 프로그램 코드를 바꾸는 소프트웨어 기반 MTD 기술, 및 데이터 포맷과 표현을 바꾸는 데이터 기반 MTD 기술로 분류할 수 있다 [2-3].

이상의 기술 분류 중에서, 본 논문은 네트워크 기반 MTD (NMTD: Network-based MTD) 기술에 관한 논문이다. NMTD로 분류되는 기존의 기법들은 전통적인 네트워크 인프라에서 NMTD를 구현한 기법들과 SDN (Software-Defined Network)에서 NMTD를 구현한 기법들로 분류될 수 있다.

본 논문에서는 전통적인 네트워크 인프라에서 동작하도록 설계된 NMTD 방법들 [4-11] 중에서, HTN (Hidden Tunnel Networking) [11] 기법이 적용된 보호대상 서버 (Protected Server)에 대한 보안을 증강하고자 한다. 구체적으로, 보호대상 서버들과 디코이 (Decoy)들로 구성된 보호대상 서버 네트워크 [3]에 디코이 트랩 (Decoy Trap)을 통한 공격 표면 확장 (Attack Surface Expansion) 방법 [1]을 적용함으로써, 단위 시간 내 공격자 성공률 (Attacker Success Rate)을 낮추고자 한다.

디코이와 허니팟 (Honeypot)은 악의적인 공격자로 하여금 보호대상 서버가 아닌 잘못된 대상을 공격하도록 유도하는 구성요소 또는 시스템을 의미한다 [12]. 엄밀하게 구분하면 허니팟과 디코이의 목적과 동작에 차이가 존재하지만 [12-13], 여러 문헌에서 디코이와 허니팟을 엄격하게 구분하지 않고 혼용하고 있다. 따라서 본 논문에서는 디코이라는 용어를 사용한다.

본 논문의 구성은 다음과 같다. 전통적인 네트워크 인프라에서 NMTD를 구현한 기법들에 대하여 제 2장에서 간략히 소개한다. 제 3장에서 보호대상 서버와 디코이로 구성된 보호대상 서버 네트워크에 대하여 살펴본 뒤, 제 4장에서 디코이를 활용한 기존의 공격 표면 확장 방법들에 대하여 정리한다. 제 5장에서는 보호대상 서버 네트워크에 디코이 트랩을 통한 공격 표면 확장을 적용하는 새로운 방법을 제안하고, 공격자 성공률 측면에서 제안한 방법의 성능을 분석한다. 제 6장에서 결론에 대하여 기술한다.

II. Related Works

DYNAT (Dynamic Network Address Translation) [4]는 보호대상 서버가 사용 중인 포트 번호와 주소의 노출을 막기 위하여, 라우팅에 필수적인 목적지 IP 주소의 네트워크 주소 부분 이외의 정보들 (예: 호스트 ID와 목적지 포트번호)을 암호화하고 복호화한다. 암호화와 복호화를 위하여 클라이언트와 서버는 초기 비밀 값을 설정하며, 일정 시간마다 비밀 값을 동기화된 형태로 변경함으로써 주소 변환이 발생하도록 한다. 암호화와 복호화에 사용되는 Keying Parameter들이 시간에 따라 변화하기 때문에, DYNAT는 NMTD 기법으로 분류된다.

APOD (Applications that Participate in their Own Defense) [5]는, 미리 설정된 난수 발생기 (Random Number Generator)를 이용하여, 패킷이 송신지 네트워크를 벗어날 때 패킷의 포트번호와 IP 주소를 각각 가용한 포트번호와 IP 주소 범위 내에서 무작위로 선택된 값들로 변경한다. 예를 들어, 클라이언트 측의 호핑 델리게이트 (Hopping Delegate)는 패킷 헤더의 (real address, real port)를 (fake address, fake port)로 변환한다. 그리고 패킷이 목적지 네트워크에 도달할 때, NAT 게이트웨이가 클라이언트 측의 호핑 델리게이트와 동기화된 난수 발생기의 값을 사용하여 원래 IP 주소와 포트번호 (real address, real port)를 복원한다. 시간이 지남에 따라 난수 발생기의 값이 바뀌기 때문에, APOD 역시 NMTD 기법으로 분류된다.

NASR (Network Address Space Randomization) [6]은 빈번하게 보호대상 서버의 IP 주소를 바꾸는 것이 공격자에 의하여 작성된 웜 히트리스트 (Worm Hitlist)를 쓸모없게 만들 수 있다는 아이디어를 기반으로 설계되었다. NASR에서 보호대상 서버들은 DHCP (Dynamic Host Configuration Protocol) 서버로부터 일정 시간 간격마다 새로운 주소를 임대한다. NASR-enabled DHCP 서버는 refresh timer, soft-change timer 및 hard-change timer에 기반하여 보호대상 서버의 IP 주소가 빈번하게 변경될 수 있도록 한다. Refresh timer는 보호대상 서버가 IP 주소 요청을 하도록 만들기 위한 타이머이고, soft-change timer는 사용 중인 연결이 존재하지 않는다고 보고된 보호대상 서버의 IP 주소의

변경 주기를 알려주는 타이머이며, hard-change timer는 사용 중인 연결이 존재하더라도 보호대상 서버의 IP 주소를 무조건 변경해야 하는 주기를 알려주는 타이머이다. 이를 통하여 NASR은 보호대상 서버의 IP 주소 동적 변이를 구현한다.

RHM (Random Host Mutation) [7]은 높은 예측불가능(Unpredictability)의 제공을 위하여 HFM (High Frequency Mutation)과 LFM (Low Frequency Mutation)을 사용하는 NMTD 기법이다. LFM은 각 보호대상 서버에 할당 가능한 IP 주소의 범위를 바꾸는 변이를 의미한다. 그리고 HFM은 일정 시간 간격마다 각 보호대상 서버에게 LFM에서 할당된 IP 주소 범위에 속하는 IP 주소 하나를 할당하는 변이를 의미한다. RHM은 짧은 시간마다 보호대상 서버의 IP 주소를 바꾸기 위하여 HFM을 이용하고, 긴 시간 동안 정찰하는 공격자에 대비하기 위하여 LFM을 이용한다.

DESIR (Decoy-Enhanced Seamless IP Randomization) [8]은 NMTD 개념 뿐만 아니라 디코이 개념도 활용한다. 보호대상 서버는 일정 시간 간격마다 변경된 IP 주소를 독립적인 노드로 존재하는 Randomization 컨트롤러로부터 할당받는다. 그리고 새로운 주소를 할당받은 시점에 보호대상 서버는 이미 서비스 중인 연결들을 마이그레이션 (Migration)한다. 이때 사용되는 방법은 모바일 네트워크에서 연결을 마이그레이션하는 방법과 유사하다. DESIR이 적용된 환경에서 공격자가 아닌 정상적인 클라이언트는 보호대상 서버의 현재 IP 주소를 인증 서버로부터 알아낼 수 있다. 그러나 DESIR는, 정상적인 클라이언트가 아닌 공격자에 의해 전송된 패킷들을 디코이-베드 (Decoy-bed)내의 디코이에게 전달되도록 함으로써, 공격자가 자신의 시간과 자원을 엉뚱한 곳에 낭비하도록 한다.

HIDE (Host IDEntity anonymization) [9]는 많은 지식을 소유하고 있는 인간 공격자에 대한 방어를 위하여, NMTD 개념에 추가로 보호대상 호스트의 핑거프린트 익명화와 변이 (Fingerprint Anonymization and Mutation) 및 디코이 기술도 활용한다. HIDE의 기본적인 NMTD 방법은 RHM과 동일하다. 차이점으로, HIDE는 보호대상 서버의 IP 주소가 목적지 주소가 아닌 패킷들을 디코이로 전달한다. 또한 보호대상 서버와 동일한 서비스들을 제공하는 많은 수의 디코이를 배치한 뒤 이에 대한 변이를 수행함으로써 HIDE는 보호대상 호스트의 핑커프린트 익명화와 변이를 제공한다.

SSCM (Scalable and Seamless Connection Migration) [10]은 DESIR의 연결 마이그레이션 확장성 문제를 해결하기 위하여 제안된 NMTD 방법이다. 주소 변이를 수행한 호스트가 다른 물리적 공간으로 실제 이동하는 것은 아니기 때문에, 해당 호스트는 어느 정도 짧은 시간 동안에 새로 할당받은 IP 주소뿐만 아니라 이전 IP 주소로도 패킷을 보내고 받을 수 있다는 사실에 근거하여 SSCM이 설계되었다. 그 결과, SSCM은 보호대상 서버가 많은 수의 연결을 동시에 서비스하고 있는 경우, DESIR에 비하여 연결 마이그레이션에 의하여 유발되는 서비스 중지 시간을 월등히 줄일 수 있다.

HTN (Hidden Tunnel Networking) [11]은 1초 이내의 매우 짧은 주소 변이 주기에서도 보호대상 서버의 주소 변이 수행을 가능케 하는 것을 목표로 설계되었다. 주소 변이 주기가 짧아지면, 공격자가 이전의 스캐닝 공격을 통해 획득한 정보의 유효기간도 짧아진다. 이를 위하여, HTN에서는 첫 번째 단계에서 인증에 성공한 클라이언트와 서버가 익명 주소 생성에 사용되는 세션 키를 분배한다. 다음으로 두 번째 단계에서 익명 주소를 생성하고 변이를 수행한다.

III. Protected Server Networks

본 장에서는 연구 [3]에서 제안된 보호대상 서버 네트워크 (Protected Server Networks)에 대하여 살펴본다. 보호대상 서버 네트워크의 전체 IP 주소 집합을 S_{IP}라고 하고, 보호대상 서버 $i(1 \le i \le n)$가 주소 변이를 수행할 수 있는 IP 주소 집합을 S_{IP}^{i}라고 하자. [3]에서 제안된 보호대상 서버 네트워크에서, 집합 S_{IP}와 S_{IP}^{i}는 $S_{IP} = \cup_{i=1}^{n} S^{i_{IP}}$와 같은 관계를 가진다.

보호대상 서버 i가 집합 S_{IP}^{i}에 속한 IP 주소들로 지속적인 주소 변이를 수행하고 있는 동안, 시간 t에 보호대상 서버 i의 IP 주소를 $ps(S_{IP}^{i}, t)$라고 하자. 그림 1은 $|S_{IP}^{i}| = 10$인 보호대상 서버 i와 디코이들에 의한 패킷 처리 개념도를 보여준다.

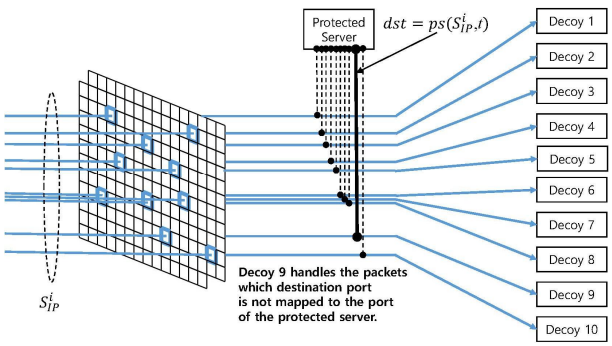

Fig. 1. Concept of Packet Processing by a Protected Server and Its Decoys [3]

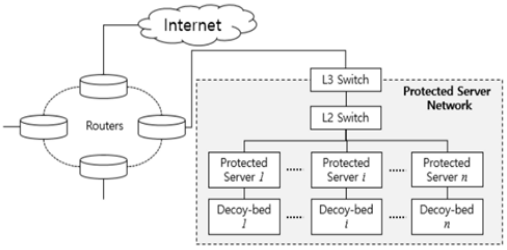

Fig. 2. Network Architecture of Protected Servers and Decoy-beds [3]

그림 1에서 집합 S_{IP}^{i}에 속한 IP 주소를 목적지로 가지는 패킷들은 보호대상 서버 i가 존재하는 네트워크에 도착한다. 도착한 패킷들은 패킷에 포함된 목적지 IP 주소에 따라 해당 디코이로 전달된다. 이 패킷들 중에서 현재 시간 t에 보호대상 서버 i의 IP 주소인 $ps(S_{IP}^{i}, t)$가 목적지 IP 주소이고 보호대상 서버 i의 현재 서비스 중인 포트번호가 목적지 포트 번호로 지정된 패킷들만 보호대상 서버 i가 가로챌 수 있도록 NAPT (Network Address & Port Translation)가 설정된다.

그림 2는 보호대상 서버 네트워크 내에서 디코이-베드와 보호대상 서버가 어떤 구조로 서로 연결되어 있는지를 보여준다. 그림 2에서 방화벽과 네트워크 침입탐지 시스템 등의 위치는 표현되지 않았다. 그림 2에서, 보호대상 서버 네트워크의 전체 IP 주소 집합 S_{IP}에 속한 IP 주소를 목적지로 하는 모든 패킷들은 L2 및 L3 스위치를 통과한 뒤, 보호대상 서버 네트워크로 전달된다. 또한, 보호대상 서버 i의 집합 S_{IP}^{i}에 속하는 IP 주소들 중 하나가 목적지 주소로 지정된 모든 패킷들은 보호대상 서버 i로 전달된다. 이 때, L2 및 L3 스위치는 보호대상 서버 i가 $|S_{IP}^{i}|$개의 IP 주소를 모두 사용하고 있다고 판단하고, 보호대상 서버 i에게 패킷들을 포워딩한다. 해당 패킷들을 수신한 보호대상 서버 i는, 그림 1과 관련하여 설명한 바와 같이, 정상적인 사용자의 패킷들만 NAPT를 통해 서버 프로세스에게 전달한다.

IV. Attack Surface Expansion Approaches

본 장에서는 디코이를 활용한 기존의 공격 표면 확장 방법들 [1]에 대하여 정리한다. 연구 [1]은 가상화된 인프라를 보호하기 위한 세 가지 공격 표면 확장 방법을 제안하였다.

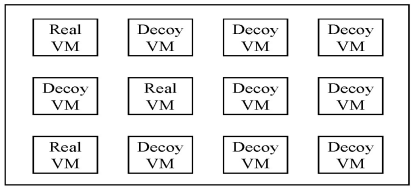

Fig. 3. Attack Surface Expansion through Decoy Pool [1]

첫 번째 방법은 디코이 풀 (Decoy Pool)을 통한 공격 표면 확장 방법이다. 이 방법에서는, 그림 3과 같이 DVM (Decoy Virtual Machine)들이 RVM (Real Virtual Machine)들과 동일한 형태로 배포된다. DVM은 본 논문에서의 디코이에 해당하며, RVM은 본 논문에서의 보호대상 서버에 해당한다. 이 방법은 DVM 사이에 RVM을 숨겨서 전체 공격 성공률을 낮추는 것을 목표로 한다.

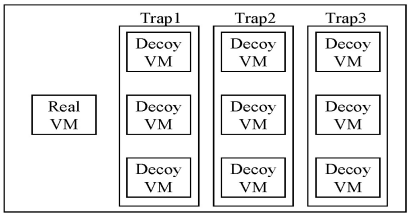

Fig. 4. Attack Surface Expansion through Decoy Trap [1]

두 번째 방법은 디코이 트랩 (Decoy Trap)을 통한 공격 표면 확장 방법이다. 이 방법을 적용하기 위해서는 공격자가 단위 시간 내에 탐색하고 공격할 수 있는 최대 회수 K 값을 알아야 한다. 그러나 최대 회수 K의 값을 미리 알 수 있다는 가정은 많은 응용 분야에서 현실적인 것이 아님에 유의할 필요가 있다 [1]. 그림 4와 같이, 이 방법에서는 디코이의 그룹이 트랩으로 설정되고, 트랩 하나의 크기는 앞서 언급한 값 K와 동일하며, 전체 트랩은 오직 하나의 RVM을 보호하기 위하여 존재하도록 설정된다.

앞서 언급한 바와 같이, 공격자가 단위 시간 내에 탐색하고 공격할 수 있는 최대 회수 K 값을 미리 알 수 있다는 가정이 현실적이지 못하기 때문에, 세 번째 방법이 제안되었다 [1]. 그림 5에서와 같이, 디코이 클러스터 (Decoy Cluster)는 R개의 DVM으로 구성된다. R의 값이 K보다 같거나 크면 디코이 클러스터는 디코이 트랩이 된다. 그렇지 않은 경우, 공격자는 디코이 클러스터의 전체 DVM을 탐색하고 공격한 후, 제한된 시간 내에 계속해서 다른 RVM 또는 다른 디코이 클러스터를 탐색하고 공격할 수 있다.

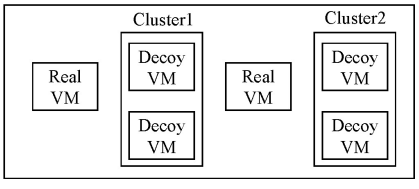

Fig. 5. Attack Surface Expansion through Decoy Cluster[1]

[1]의 저자들이 수식과 시뮬레이션을 통해 이상의 세 가지 방법의 성능을 분석하였다. 구현 측면에서는 디코이 풀을 통한

공격 표면 확장 방법이 가장 간단하지만, 보안 측면에서는 디코이 트랩을 사용하는 공격 표면 확장 방법이 가장 높은 보호 기능을 제공하며, 디코이 클러스터를 사용하는 공격 표면 확장 방법이 중간 수준의 보호 기능을 제공함을 보였다.

V. Attack Surface Expansion in the Protected Server Networks

앞 장에서 살펴본 바와 같이, 디코이 풀, 디코이 트랩, 디코이 클러스터를 통한 공격 표면 확장 기법들 중에서 가장 뛰어난 보호 기능을 제공하는 것은 디코이 트랩을 통한 공격 표면 확장 기법이다. 하지만, 공격자가 단위 시간 내에 탐색하고 공격할 수 있는 최대 회수 K 값을 미리 알 수 있다는 가정은 비현실적이다 [1]. 이 때문에, [1]의 저자들에 의해 현실적으로 적용 가능한 기법으로 디코이 클러스터를 통한 공격 표면 확장 기법을 제안하였다.

그런데, 디코이 클러스터를 통한 공격 표면 확장 기법을 표현한 그림 5를 살펴보면, 보호대상 서버 네트워크 구성도를 보여주는 그림 2와 매우 유사함을 발견할 수 있다. 만일 두 개의 구성과 동작이 동일하다면, 보호대상 서버 네트워크에 대한 단위 시간 내 공격자 성공률 (Attacker Success Rate)은 디코이 클러스터를 사용하는 공격 표면 확장 방법의 공격자 성공률과 같아진다. 그러나, 만일 보호대상 서버 네트워크에서의 단위 시간 내 공격자 성공률을 디코이 트랩를 사용하는 공격 표면 확장 방법에서의 공격자 성공률과 같은 수준이 되도록 만들 방법을 찾을 수 있다면, 보호대상 서버들에 대한 보안을 증강할 수 있다.

본 장에서는, 공격자가 단위 시간 내에 탐색하고 공격할 수 있는 최대 회수 K 값을 미리 알 수 있다는 비현실적인 가정을 제거하고, [1]에서 제안된 디코이 트랩을 통한 공격 표면 확장 방법과 동일한 수준의 보안을 [3]에서 제안한 보호대상 서버 네트워크에서 확보할 수 있는 방법을 제안한다.

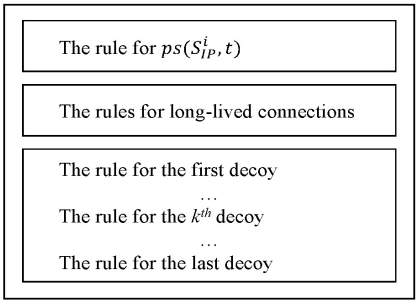

Fig. 6. Rules in the NAT table for a Protected Server

그림 6은 [3]에서 제안한 보호대상 서버 네트워크의 NAT 테이블의 규칙 구성을 보여준다. 그림 6에서와 같이, 보호대상 서버 i를 보호하기 위하여 설치된 디코이-베드 i에 속한 디코이에 대한 일대일 매핑 정보 (주소 변환 규칙)는 보호대상 서버 i의 NAT 테이블에서 가장 아래쪽에 위치해야 한다. iptables [14]의 동작 특징에 따르면, NAT 테이블의 위쪽에 위치하는 규칙 (높은 우선순위의 규칙)중 하나의 조건을 만족하는 경우, 아래쪽 규칙 (낮은 우선순위의 규칙)에 대해서는 처리를 수행하지 않기 때문이다. 따라서, 그림 1과 같이, 목적지 주소가 $ps(S_{IP}^i, t)$이거나 장기 연결(Long-Lived Connection)에 속하는 패킷들을 보호대상 서버 i가 가로채도록 하기 위해서는 NAT 테이블의 규칙들이 그림 6과 같은 순서대로 관리되어야 한다. [3]에서 제안한 보호대상 서버 네트워크의 서버들은 HTN [11]에 따라 NMTD를 수행하는데, HTN이 장기 연결(Long-Lived Connection)을 위한 규칙들을 요구하기 때문에, 그림 6에는 장기 연결(Long-Lived Connection)을 위한 규칙들이 포함되어야 한다. [3]에서는 $ps(S_{IP}^i, t)$를 위한 규칙과 장기 연결(Long-Lived Connection)을 위한 규칙 중 어느 규칙의 우선순위가 높은 지에 대하여 서술하지 않았지만, 그 규칙들의 우선순위가 보호대상 서버의 동작에 영향을 미치지 않기 때문에, 그림 6에서는 $ps(S_{IP}^i, t)$를 위한 규칙의 우선순위가 가장 높은 것으로 표시하였다.

보호대상 서버 네트워크가 디코이 트랩을 통한 공격 표면 확장 방법과 같이 동작하기 위해서는 그림 6과 같은 보호대상 서버의 NAT 테이블을 수정하여야 한다. 그림 7은 수정된 NAT 테이블에서 규칙들의 실행 조건이 검사되어야 할 순서를 보여준다.

그림 7에서 NAT 테이블의 규칙 검사가 시작되면, 가장 먼저 패킷의 송신자가 디코이에 접근한 적이 있어서 공격자로 의심받고 있는지 여부를 검사하여야 한다. 만일, 한 번이라도 디코이에 접근한 적이 있어서 공격자로 의심받고 있다면, 그 송신자의 모든 패킷들은 디코이에게 전달되도록 하여야 한다. 그렇지 않은 경우, 송신자는 공격자가 아니거나 한 번도 디코이에게 접근하지 않은 공격자일 수 있다. 이를 구분하기 위하여, 보호대상 서버의 IP 주소 $ps(S_{IP}^i, t)$를 목적지로 하는 패킷인지 또는 장기 연결(Long-Lived Connection)에 속하는 패킷인지를 확인할 필요가 있다. 따라서 해당 시점에 공격자로 의심되지 않는 송신자의 패킷에 대해서는 그림 6과 동일한 순서로 NAT 규칙 검사가 실시될 필요가 있다.

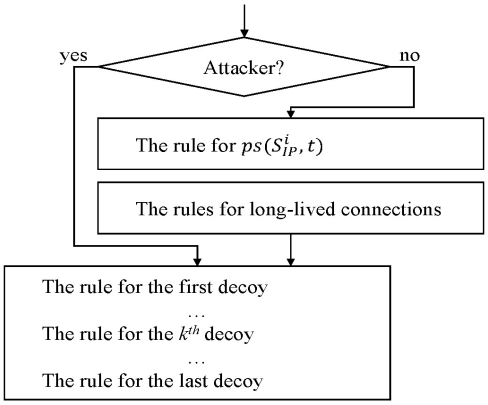

Fig. 7. Flow Chart for the Modified NAT Table

앞서 언급한 바와 같이, iptables로 작성된 NAT 테이블의 규칙들은 우선순위에 따라 검사되며, 실행 조건을 만족하는 규칙이 존재할 때 그 규칙만 수행되고 낮은 우선순위의 규칙들은 무시된다. 따라서 그림 7과 같은 순서로 규칙 검사가 수행되도록 하기 위해서는 커스텀 체인 (Custom Chain)과 iptables의 RETURN 타겟 (target)을 활용하여야 한다.

커스텀 체인은 iptables 명령어에서 "-N" 또는 "--new-chain" 옵션을 사용하여 생성된다 [14]. 예를 들어, "iptables -N rx-check"와 같은 명령을 실행하면, "rx-check"라는 커스텀 체인이 만들어진다.

RETURN 타겟은 해당 규칙의 조건이 만족할 때, 규칙 검사가 실행 중인 체인에서의 규칙 검사를 중지하고, 해당 체인의 상위 체인 (Superior Chain)에서 규칙 검사를 계속하게 만든다. 만일 RETURN 타켓에 의해 중지되는 체인이 가장 상위의 체인이라면, 해당 패킷은 디폴트 정책에 의하여 처리된다 [14]. 이상의 동작은 프로그래밍 언어의 함수에서 리턴 문을 사용하였을 때의 동작과 유사하다. 예를 들어, 수신한 패킷에 대하여, 앞서 언급한 "rx-check"의 규칙들을 검사하는 도중에, 어떤 규칙의 조건이 만족되었는데, 그 규칙의 타겟이 RETURN인 경우, "rx-check"를 빠져나와 "rx-check"의 상위 체인에서 규칙 검사를 계속하게 된다.

```
(1) iptables –N rx-check
(2) iptables –A rx-check –m set --match-set attackerlist src –j RETURN
(3) iptables –A rx-check <rule for ps(S_IP^i, t)>
(4) iptables –A rx-check <rule for the first long-lived connection>
    ...
(5) iptables –A rx-check <rule for the last long-lived connection>
```

```
(6) iptables –A PREROUTING –j rx-check
(7) iptables –A PREROUTING <rule for the first decoy>
    ...
(8) iptables –A PREROUTING <rule for the first decoy>
```

Fig. 8. *iptables* Commands for the Modified NAT Table

커스텀 체인과 RETURN 타켓을 사용하여 그림 7을 iptables 명령어로 표현하면 그림 8과 같다. 그림 8의 (1)번 줄에서 커스텀 체인으로 "rx-check"를 생성한다. (2)번 줄의 "attackerlist"는 한 번이라도 디코이에게 접근한 적이 있는 송신자 주소가 포함된 목록이다. 따라서, (2)번 줄의 명령어에 의하여, 송신자 주소가 "attackerlist"에 포함되어 있다면 "rx-check"를 종료하고 "rx-check"의 상위 체인으로 돌아간다. 그림 8의 (6)번 줄에서 확인할 수 있듯이, "PREROUTING" 체인으로부터 "rx-check"로 점프 ("-j" 옵션)한 것이기 때문에, (2)번 줄에서 RETURN을 하게 되면, 다음으로 검사될 규칙은 (7)번 줄의 규칙이 된다. 송신자 주소가 "attackerlist"에 포함되어 있지 않다면, (3)번 줄의 규칙 검사가 수행된다. (3)번 줄의 규칙은 보호대상 서버의 IP 주소 $ps(S_{IP}^{i}, t)$를 목적지로 하는 패킷인지를 판단하는 규칙이며, (4)번부터 (5)번까지의 규칙들은 장기 연결(Long-Lived Connection)에 속한 패킷인지를 판단하는 규칙이다. 이상의 규칙들 중 하나에 해당하지 않으면, (7)번부터 (8)번까지의 규칙들에 의해, 패킷은 디코이 중 하나에게 전달된다.

그러나, 그림 8의 규칙만으로는 디코이 트랩을 통한 공격 표면 확장 방법과 동일한 수준의 성능을 보장할 수 없다. 디코이 트랩을 통한 공격 표면 확장 방법과 동일한 수준의 단위 시간 내 공격자 성공률을 제공하기 위해서는, 보호대상 서버들 간 "attackerlist"의 정보 교환을 통해, 보호대상 서버 네트워크에 존재하는 모든 디코이-베드가 하나의 보호대상 서버를 위한 것처럼 동작할 수 있도록 설정하여야 한다. 만일 디코이-베드 i의 디코이에 접근한 적이 있는 공격자의 주소를 보호대상 서버 i가 나머지 보호대상 서버들과 공유할 수 있다면, 그 공격자가 다른 보호대상 서버 j에 접근하려 할 때, 그 공격자를 디코이-베드 j의 디코이에 묶어둘 수 있기 때문이다. 보호대상 서버들 간 "attackerlist" 정보 교환 문제는 보호대상 서버들을 연결하는 별도의 관리 네트워크를 구축함으로써 해결될 수 있다. 오픈스택 구축 및 관리에 사용되는 관리 네트워크 (Management Network) [15]와 유사한 형태로 보호대상 서버들만의 독립적인 관리 네트워크가 필요하다.

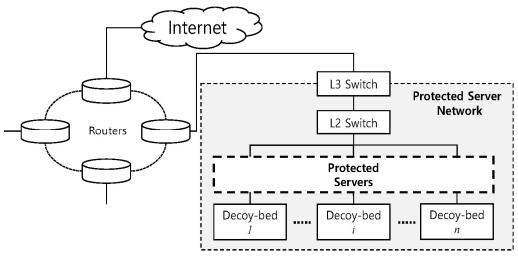

Fig. 9. Proposed Network Architecture of Protected Servers and Decoy-beds

그림 9는 디코이 트랩을 통한 공격 표면 확장 방법을 보호대상 네트워크에 적용한 후의 개념적인 네트워크 구성도를 보여준다.

그림 2 및 그림 9와 같은 보호대상 서버 네트워크에서 단위 시간 내 공격자 성공률을 계산하기 위하여 사용되는 기호의 의미는 다음과 같다. 첫째, 보호대상 서버와 디코이의 전체 수를 N이라 하고, 보호대상 서버과 디코이의 수를 각각 N_R과 N_D라 한다. $N=N_R+N_D$이다. 둘째, 공격자가 보호대상 서버에 접근하였을 때, 공격이 성공할 확률을 q라 한다. 셋째, 공격자가 단위 시간 내에 탐색하고 공격할 수 있는 최대 회수는 K라 하고, K의 값은 미리 예측할 수 없다고 가정한다. 넷째, 디코이-베드 i에 속한 디코이의 수는 N_D^i이다. 모든 보호대상 서버 i에 대하여, N_D^i의 값은 모두 동일하다고 가정한다.

그림 2와 같은 보호대상 서버 네트워크에서 단위 시간 내 공격자 성공률은 수식 (1)과 같이 계산할 수 있다.

$$p=\frac{N_R}{N}\times q+\frac{N_D}{N}\times \left(\frac{N_R}{N-N_D^i}\times q+\frac{N_D-N_D^i}{N-N_D^i}\times \left(\frac{N_R}{N-2N_D^i}\times q+\frac{N_D-2N_D^i}{N-2N_D^i}\times(...)\right)\right) \quad (1)$$

수식 (1)은 연구 [1]에서 제안한 디코이 클러스터를 사용하는 공격 표면 확장 방법의 공격자 성공률 수식과 동일한 형태를 가진다. 수식 (1)에서 중첩 계산 (Nested Calculation) 횟수는 $\lfloor (K-1)/N_D^i \rfloor$ 이다. 수식 (1)에서 공격자가 보호대상 서버를 발견하였다고 하더라도 공격에 성공할 확률 q는 높지 않다. 왜냐하면, 그림 2에서 보호대상 서버가 포트 주소에 대해서도 NMTD를 수행하고 있기 때문이다. 즉, 시간 t에 서비스 중인 포트 주소 이외의 포트 주소로 공격자가 접근하면, 보호대상 서버는 공격자의 패킷이 디코이로 흘러가도록 만들기 때문에 공격자는 디코이에 빠지게 된다.

이에 반하여, 그림 9와 같이, 본 논문에서 제안하는 보호대상 서버 네트워크에서 단위 시간 내 공격자 성공률은 수식 (2)와 같다. 왜냐하면, 공격자가 보호대상 서버 중 하나를 공격하여 성공하지 못하면, “attackerlist” 정보 교환을 통하여 보호대상 서버들이 해당 공격자의 모든 패킷을 디코이로 흘러가도록 만들기 때문이다.

$$p=\frac{N_R}{N}\times q \quad (2)$$

공격자가 단위 시간 내에 탐색하고 공격할 수 있는 최대 회수 K가 디코이-베드 i에 속한 디코이의 수 N_D^i보다 작거나 같을 때, 수식 (1)은 수식 (2)와 같아진다. 그러나 본 논문에서 제안하는 보호대상 서버 네트워크에서 단위 시간 내 공격자 성공률인 수식 (2)는 값 K에 영향을 받지 않는다.

따라서 본 논문에서 제안하는 방법으로 보호대상 서버 네트워크를 구축한다면, 공격자가 단위 시간 내에 탐색하고 공격할 수 있는 최대 회수 K 값을 미리 알 수 있다는 비현실적인 가정 없이도, 디코이 트랩을 통한 공격 표면 확장 방법과 동일하게, 항상 일정하고 높은 수준의 보안을 제공할 수 있다. 달리 표현하자면, 보호대상 서버 네트워크에 보호대상 서버가 다수 존재하더라도, 임의의 디코이-베드로 한 번이라도 접근한 적이 있는 공격자에게는 보호대상 서버가 전혀 보이지 않도록 만들 수 있다.

VI. Conclusions

네트워크 및 시스템 보안을 향상시키기 위하여, 공격 표면을 지속적으로 변경하는 MTD 기법들과 디코이를 활용하여 공격 표면을 확장하는 방법들이 제안되고 있다. 본 논문에서는, 네트워크 기반 MTD로 제안된 HTN 기법을 사용하는 보호대상 서버에 대한 보안 향상을 위하여, 보호대상 서버 네트워크에 디코이 트랩을 통한 공격 표면 확장을 적용할 수 있는 방법을 제안하였다. 제안하는 방법에서는 공격자가 모든 시간과 노력을 디코이 트랩에서 허비하도록 커스텀 체인과 RETURN 타겟을 활용하여 보호대상 서버 NAT 테이블에서의 규칙 검사 순서가 변경되도록 하였다. 또한 보호대상 서버들이 공격자 목록에 대한 정보 교환을 수행하도록 하였다. 제안하는 방법의 성능 분석을 위하여, 단위 시간 내 공격자 성공률을 계산함으로써 제안하는 방법이 적용되기 전에 비하여 적용된 이후의 공격자 성공률이 현격히 줄어듦을 보였다. 특히, 공격자가 단위 시간 내에 탐색하고 공격할 수 있는 최대 회수 값을 미리 알 수 있다는 비현실적인 가정 없이도, 제안하는 방법이 기존 디코이 트랩을 통한 공격 표면 확장 방법과 동일한 수준으로 동작할 수 있음을 보였다.

ACKNOWLEDGEMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-00213, Development of Cyber Self Mutation Technologies for Proactive Cyber Defense).

REFERENCES

[1] T. Al-Salah, L. Hong, and S. Shetty, “Attack Surface Expansion Using Decoys to Protect Virtualized Infrastructure,” Proceedings of the 2017 IEEE International Conference on Edge Computing, pp. 216-219, June 2017.

[2] K. Kang, T. Park, and D. Moon, “Analysis of Threat Model and Requirements in Network-based Moving Target Defense,” Journal of The Korea Society of Computer and Information, Vol. 22, No. 10, pp. 83-92, October 2017.

[3] T. Park, K. Park, and D. Moon, “Design of a Protected Server Network with Decoys for Network-based Moving Target Defense,” Journal of The Korea Society of Computer and Information, Vol. 23, No. 9, pp. 57-64, September 2018.

[4] D. Kewley, R. Fink, J. Lowry and M. Dean, “Dynamic Approaches to Thwart Adversary Intelligence Gathering,” Proceedings of the DARPA Information Survivability Conference and Exposition, pp. 176-185, August 2001.

[5] M. Atighetchi, P. Pal, F. Webber and C. Hones, “Adaptive Use of Network-Centric Mechanisms in Cyber-Defense,” Proceedings of the sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, pp. 183-192, 2003.

[6] S. Antonatos, P. Akritidis, E. P. Markatos, K. G. Anagnostakis, “Defending against histlist worms using network address space randomization,” Computer Networks, vol.51, no.12, pp.3471-3490. 2007.

[7] J. H. Jafarian, E. Al-Shaer and Q. Duan, “An Effective Address Mutation Approach for Distructing Reconnaissance Attacks," IEEE Transactions on Information Forensics, vol.10, no.12, pp. 2562-2577, 2015.

[8] J. Sun and K. Sun, “DESIR: Decoy-enhanced seamless IP randomization," Proceedings of the IEEE INFOCOM, 2016.

[9] J. H. Jafarian, A. Niakankahiji, E. Al-Shaer and Q. Duan, “Multi-dimensional Host Identity Anonymization for Defeating Skilled Attacks," Proceedings of the 2016 ACM Workshop on Moving Target Defense, pp. 47-58, 2016.

[10] T. Park, K. Kang, and D. Moon, “A Scalable and Seamless Connection Migration Scheme for Moving Target Defense in Legacy Networks,” IEICE Trans. Inf. & Syst., In Press, Vol.E101-D, No.11, November 2018.

[11] K. Park, S. Woo, D. Moon, K. Koo, I. Kim, and J. Lee “Pseudonym Address based Hidden Tunnel Networking for Network Address Mutation,” KOREA Patent App. No. 10-2018-0076029, 2018.

[12] Fred Cohen, “The Use of Deception Techniques: Honeypots and Decoys”, Fred Cohen & Associates, at http://all.net/journal/deception/Deception_Techniques_.pdf, accessed 23 March 2018.

[13] K. Borders, L. Falk, and A. Prakash, “OpenFire: Using Deception to Reduce Network Attacks”, 2007 Third International Conference on Security and Privacy in Communications Networks and the

Authors

Tae-keun Park received his B.S., M.S., and Ph.D. degrees in Computer Science and Engineering from POSTECH, Pohang, Korea in 1991, 1993, and 2004, respectively. He joined POSTECH PIRL in 1993 and moved to SK Telecom in 1996. From 2000 to 2001 and from 2001 to 2002, he worked for 3Com Korea and Ericsson Korea, respectively. In 2004, he joined in the department of Multimedia Engineering, Dankook University, Korea. He is currently on the faculty of the department of Applied Computer Engineering at Dankook University. His research interests include network security, IoT, wireless/mobile communications, and distributed services.

Kyung-min Park received his B.S., M.S., and Ph.D. degree in Computer Engineering from Chungnam National University, Rep. of Korea, in 2010, 2013, and 2019. He joined the Electronics and Telecommunications Research Institute(ETRI), Daejeon, Rep. of Korea, in 2017. His research include network protocols & security, network middleware, and distributed computing.

Dae-sung Moon received his MS degree in computer engineering from Pusan National University, Rep. of Korea, in 2001. He received his PhD degree in computer science from Korea University, Seoul, Rep. of Korea, in 2007. He joined the Electronics and Telecommunications Research Institute(ETRI), Daejeon, Rep. of Korea, in 2000, where he is currently working as the director of Network&System Security Research Section. He has also been a Chief major professor with the Department of Information Security Engineering, University of Science and Technology, Daejeon, Rep. of Korea. His research interests include network security, machine learning, biometrics, and image processing.