Journal of The Korea Society of Computer and Information
Vol. 23 No. 9, pp. 57-64, September 2018
www.ksci.re.kr
https://doi.org/10.9708/jksci.2018.23.09.057

Design of a Protected Server Network with Decoys for Network-based Moving Target Defense

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

Abstract

In recent years, a new approach to cyber security, called the moving target defense, has emerged as a potential solution to the challenge of static systems. In this paper, we design a protected server network with a large number of decoys to anonymize the protected servers that dynamically mutate their IP address and port numbers according to Hidden Tunnel Networking, which is a network-based moving target defense scheme. In the network, a protected server is one-to-one mapped to a decoy-bed that generates a number of decoys, and the decoys share the same IP address pool with the protected server. First, the protected server network supports mutating the IP address and port numbers of the protected server very frequently regardless of the number of decoys. Second, it provides independence of the decoy-bed configuration. Third, it allows the protected servers to freely change their IP address pool. Lastly, it can reduce the possibility that an attacker will reuse the discovered attributes of a protected server in previous scanning. We believe that applying Hidden Tunnel Networking to protected servers in the proposed network can significantly reduce the probability of the protected servers being identified and compromised by attackers through deploying a large number of decoys.

▸Keyword: Network-based moving target defense, mutation, cyber security, protected server, decoy

I. Introduction

현재, 반응적/수동적 형태의 정보보호 기술들이 ICT (Information Communication Technology) 인프라의 보안 설정을 정적으로 유지하고 있기 때문에, 대상 시스템의 취약점을 분석할 수 있는 충분한 시간을 공격자들에게 제공해주는 현상이 발생하고 있다 [1]. 이러한 현상을 공격자 우위의 시간적 비대칭성이라고 한다 [1-3].

최근, 이러한 시간적 비대칭성을 극복하기 위하여 MTD (Moving Target Defense) 기술이 개발되고 있다 [3]. MTD는 보호대상 시스템의 다양한 특징들을 시간의 변화에 따라 역동적으로 변경하여 각종 사이버 공격을 차단하는 능동적 사전 보안 기술이다. MTD의 기술을 세분화하면, 첫째, 네트워크 특징 및 설정을 변경하는 네트워크 기반 MTD, 둘째, 시스템 플랫폼 특징을 변경하는 플랫폼 기반 MTD, 셋째, 런타임 환경 또는 응용 프로그램 코드를 변경하는 소프트웨어 기반 MTD, 넷째, 데이터의 포맷과 표현을 변경시키는 데이터 기반 MTD로 분류할 수 있다 [1-3].

본 논문은 이상의 MTD 기술 분류 중에서 네트워크 기반 MTD (NMTD: Network-based MTD)에 포함되는 논문이다.

• 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: 2018. 08. 21, Revised: 2018. 09. 10, Accepted: 2018. 09. 17.
• 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).

NMTD 관련 기술은 전통적인 네트워크 인프라를 기반으로 NMTD를 구현하는 방법과 SDN (Software-Defined Network)을 기반으로 NMTD를 구현하는 방법으로 분류된다. 본 논문에서는 전통적인 네트워크 인프라를 기반으로 NMTD를 구현한 방법들 [4-11] 중에서, 한국전자통신연구원이 개발한 HTN (Hidden Tunnel Networking) [11]이 적용된 보호대상 서버 (Protected Server)들의 네트워크를 구축할 때, 보호대상 서버의 익명성 (Anonymization)을 높일 수 있도록 디코이 (Decoy)를 활용하는 보호대상 서버 네트워크를 설계한다.

디코이 (Decoy)와 허니팟 (Honeypot)은 악의적인 행위를 하려는 공격자가 잘못된 (보호대상 서버가 아닌) 대상을 공격하도록 유도하는 시스템 또는 구성 요소를 의미한다 [12]. 엄밀하게 구분하자면, 허니팟과 디코이는 서로 다른 목적과 동작을 보이지만 [12-13], 여러 논문에서 허니팟과 디코이를 혼용하여 사용하기도 한다. 따라서 본 논문에서는 가상 허니팟 (Virtual Honeypot)이면서 높은 상호작용 허니팟 (High-Interaction Honeypot)으로 분류되는 허니팟을 사용하며, 이를 간략히 디코이라고 표현하도록 한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 전통적인 네트워크 인프라를 기반으로 하는 NMTD 기술들에 대하여 간략히 소개한 뒤, 허니팟과 디코이 용어에 대하여 정리한다. 제 3장에서는 보호대상 서버에 적용될 HTN의 특성 및 동작에 대하여 살펴보고, 제 4장에서는 디코이를 활용하는 보호대상 서버 네트워크의 설계한 뒤, 장단점을 분석한다. 마지막으로, 제 5장에서 결론을 맺는다.

II. Related Works

공격자들은 실제 공격에 앞서 목표 시스템의 취약점을 확보하기 위하여 정찰 단계를 수행하는데, NMTD는 보호 대상의 네트워크 속성을 변화시켜 정찰을 어렵게 만드는 매우 효율적인 방어 방법이다 [1]. 본 장에서는, 전통적인 네트워크 인프라를 기반으로 NMTD를 구현한 방법인 DYNAT (Dynamic Network Address Translation) [4], APOD (Applications that Participate in their Own Defense) [5], NASR (Network Address Space Randomization) [6], RHM (Random Host Mutation) [7], DESIR (Decoy-Enhanced Seamless IP Randomization) [8], HIDE (Host IDEntity anonymization) [9], SSCM (Scalable and Seamless Connection Migration) [10], HTN (Hidden Tunnel Networking) [11]에 대하여 간략히 소개한 뒤, 허니팟과 디코이의 목적 및 동작과 허니팟의 종류에 대하여 정리한다.

DYNAT [4]는 보호대상 서버의 주소와 오픈 포트번호의 노출을 막기 위하여, 송수신되는 패킷에 대하여, 라우팅에 필수적인 목적지 IP 주소의 네트워크 주소 부분을 제외한 나머지 정보인 호스트 ID와 목적지 포트번호를 송신자 네트워크를 벗어날 때 암호화하고 수신자 네트워크에 도착할 때 복호화하는 방식을 사용한다. DYNAT는 이 암호화에 사용되는 키 값 (Keying Parameter)으로 시간에 따라 변화하는 값들을 사용하기 때문에, NMTD 기법으로 분류된다.

APOD [5]는 패킷의 IP 주소와 포트 번호를 지속적으로 변경시킴으로써 공격자를 혼란시키는 방법을 사용한다. 패킷이 송신지 네트워크를 벗어날 때, 미리 설정된 난수 발생기 (Random Number Generator)를 이용하여, 패킷의 IP 주소와 포트번호를 각각 가용한 IP 주소와 포트번호 범위 내에서 무작위로 선택된 값들로 변경하며, 목적지 네트워크에 도달하면, 동일한 난수 발생기의 값에 따라 원래 IP 주소와 포트번호로 복원된다. 일정 시간이 지나면 새로운 주소와 포트번호 값이 선택되도록 난수 발생기의 값이 변경되기 때문에, APOD도 NMTD 기법으로 분류된다.

NASR [6]은 스캐닝을 수행하여 웜 히트리스트 (Worm Hitlist)를 생성하는 공격자에 대응하기 위하여 제안되었다. NASR은 보호대상 서버의 IP 주소를 빈번하게 바꾸는 것이 이미 작성된 웜 히트리스트를 쓸모없게 만들 수 있다는 아이디어에 기반하여, 보호대상 서버들이 일정 시간 간격마다 DHCP (Dynamic Host Configuration Protocol)를 이용하여 새로운 주소를 임대하게 함으로써, 보호대상 서버의 IP 주소 변이를 구현한다.

RHM [7]은 이전의 기법들에 비하여 높은 예측불가능 (Unpredictability)를 제공하기 위하여, LFM (Low Frequency Mutation)과 HFM (High Frequency Mutation)을 사용한다. LFM은 각 보호대상 서버에 할당될 수 있는 IP 주소의 범위를 바꾸는 변이이고, HFM은 LFM에서 할당된 IP 주소 범위에 속하는 하나의 IP 주소를 일정 시간 간격마다 각 보호대상 서버에게 할당하는 변이이다. 따라서 RHM은 HFM을 이용하여 보호대상 서버의 IP 주소를 짧은 시간마다 바꾸면서도, 오랜 기간 모니터링하는 공격자에 대비하기 위하여, LFM을 이용하여 보호대상 서버를 긴 시간마다 한 번씩 다른 네트워크 주소로 이동시키는 NMTD 방법이다.

DESIR [8]은 NMTD에 추가로 허니팟/디코이 기술도 활용한다. 보호대상 서버는 독립적인 노드로 존재하는 Randomization 컨트롤러로부터 일정 시간 간격마다 변경된 IP 주소를 할당받으면서, 새로운 주소를 할당받은 시점에 이미 서비스 중인 연결들을, 이동통신 등에서 연결을 마이그레이션 (Migration)하는 것과 유사한 방법으로, 연결이 끊어지지 않도록 마이그레이션한다. 또한, 정상적인 클라이언트는 인증 서버로부터 보호대상 서버의 현재 IP 주소를 알아낼 수 있지만, 그렇지 못한 공격자의 패킷들은 디코이-베드 (Decoy-bed)내에 위치한 허니팟/디코이에게 전달되도록 하여, 공격자가 허튼 곳에 시간과 자원을 낭비하도록 한다.

HIDE [9]는 이전의 기법들이 자동화된 스캐닝 또는 웜 공격에 대해서는 적합할 수 있지만, 전문적인 지식을 소유한 인간 공격자에 대해서는 효율적이지 못하다는 사실에 근거하여,

NMTD에 추가로 허니팟/디코이 기술 및 보호대상 호스트의 핑거프린트 (Fingerprint)의 익명화와 변이도 활용한다. HIDE의 NMTD 방법은 RHM과 동일하며, DESIR와 유사하게, HIDE는 목적지 주소가 보호대상 서버의 IP가 아닌 패킷들을 모두 허니팟/디코이로 전달한다. 또한 보호대상 서버와 동일한 서비스 목록을 제공하는 다수의 허니팟/디코이를 배치하고 이에 대한 변이를 수행함으로써 보호대상 호스트의 핑거프린트 익명화와 변이를 제공한다.

SSCM [10]은 DESIR의 연결 마이그레이션의 확장성 문제를 해결하기 위하여 제안된 NMTD 방법으로, 주소를 변이하는 호스트가 물리적으로 이동하는 것이 아니기 때문에 해당 호스트는 짧은 시간 동안 새로 할당받은 IP 주소뿐만 아니라 이전 IP 주소를 이용하여 패킷을 송수신할 수 있다는 사실에 근거하여 설계되었다. 따라서 SSCM은 보호대상 서버가 동시에 많은 수의 연결을 서비스하고 있는 경우, 연결 마이그레이션에서 발생하는 서비스 중지 시간을 DESIR에 비하여 월등히 줄일 수 있다. SSCM의 향후 연구 내용으로, SSCM이 연결 마이그레이션을 위해 두 개의 IP 주소를 목적지로 하는 패킷을 모두 수신하는 짧은 기간 동안, 이전 IP 주소를 목적지로 하는 새로운 연결의 패킷들을 허니팟/디코이에게 전달하는 방법의 개발이 언급되었다 [10]. 본 논문에서는 보호대상 서버에 HTN이 적용되었다고 가정하고 있지만, SSCM이 적용된 경우에 본 논문의 보호대상 서버 네트워크를 활용하면, 앞서 언급한 SSCM의 향후 연구 내용에 대한 해결책이 될 수 있다.

HTN [11]은 한국전자통신연구원에서 개발된 NMTD 방법으로 본 연구의 보호대상 서버에서 사용되는 방법이기 때문에, 다음 장에서 보다 상세히 서술한다.

이상의 기법들 중의 몇 가지에서 허니팟/디코이 기술에 대한 언급이 있었는데, 허니팟과 디코이의 용어 및 종류에 대하여 정리하면 다음과 같다.

허니팟과 디코이는 악의적인 행위를 하려는 공격자가 잘못된 대상을 공격하도록 유도하는 시스템 또는 구성 요소를 의미하며, 이러한 유도 과정에서, 시스템 관리자에게 잠재적으로 유용한 정보를 제공한다고 알려져 있다 [12]. 근본적으로 기만(Deception)은 어떤 목적을 위하여 인지 시스템에 오류가 발생하도록 유도하는 행위를 의미하는데, 이러한 행위를 수행하는 시스템 또는 요소가 허니팟과 디코이이다. 허니팟과 디코이가 처음 등장하였던 시기의 연구자들이 언급하는 차이점 대신에 정보보호 기술 관점에서 사용되는 용어로서 두 개의 차이점을 간략히 정리하자면 다음과 같다. 허니팟은 공격자를 낚는 것(To Lure Attackers)을 주 목적으로 하지만, 디코이는 보호대상 기기를 보호하는 것 (To Protect Other Machines)을 주 목적으로 한다 [13].

허니팟은 여러 가지 방법으로 분류될 수 있는데, 그 중 한 가지는 물리적 허니팟 (Physical Honeypots)과 가상 허니팟 (Virtual Honeypots)으로 분류하는 것이다. 물리적 허니팟은 자체 IP 주소가 있는 실제 물리적인 시스템을 의미하고, 가상 허니팟은 가상 기계 (Virtual Machine) 등으로 다른 물리적인 시스템에 의해 실행되는 허니팟을 의미한다. 또 다른 방법으로, 높은 상호작용 허니팟 (High-Interaction Honeypots)과 낮은 상호작용 허니팟 (Low-Interaction Honeypots)으로 분류할 수 있다. 높은 상호작용 허니팟은 공격자가 상호 작용할 수 있는 실제 운영체제와 서비스를 제공하는 허니팟을 의미하고, 낮은 상호작용 허니팟은 네트워크 스택과 같은 일부 기능만 시뮬레이션 되는 허니팟을 의미한다 [14].

이상과 같은 차이점과 다양한 형태의 허니팟이 존재하기는 하지만, 기존의 NMTD 기법들에서 허니팟과 디코이를 혼용하여 사용하기 때문에, 본 논문에서도 가상 허니팟이면서 높은 상호작용 허니팟으로 분류되는 허니팟을 간략히 디코이라고 표현하도록 한다.

III. Hidden Tunnel Networking

HTN [11]은 매우 짧은 주소 변이 주기 (Address Mutation Interval)에서도 보호대상 서버가 주소 변이를 수행할 수 있는 것을 목표로 설계되었다. 주소 변이 주기가 짧다는 것은 이전의 스캐닝 공격을 통해 공격자가 획득한 정보의 유효기간이 짧아진다는 것을 의미한다. 이를 위하여, HTN은 그림 1과 같이 두 단계로 동작한다.

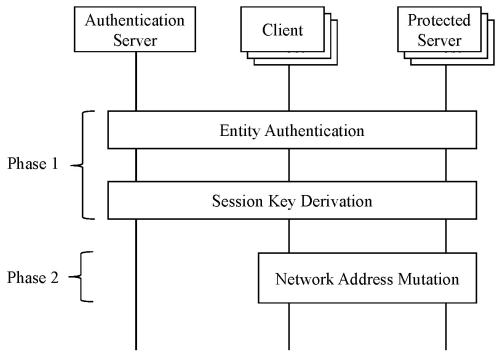

Fig. 1. Two Phases of HTN [11]

그림 1의 첫 번째 단계는 개체 인증 및 세션 키 교환을 수행하는 단계로서, 두 번째 단계를 위한 사전 준비 단계이다. 첫 번째 단계에서 인증에 성공한 클라이언트와 서버는 세션 키를 분배하는데, 세션 키란 익명 주소 생성에 사용되는 비밀 키를 의미한다. 두 번째 단계는 익명 주소 생성 및 변이를 수행하는 단계이며, 모든 클라이언트와 서버는 시간 동기화가 이루어져 있어서, 첫 번째 단계에서 분배한 세션 키를 활용하여 상호 간의 메시지 교환 절차 없이 각자 주소를 생성하고 변이에 사용할 수 있다.

이상과 같은 두 단계에 의해 생성된 익명 주소를 사용하여 클라이언트와 서버가 패킷을 송수신하는 예는 그림 2와 같다.

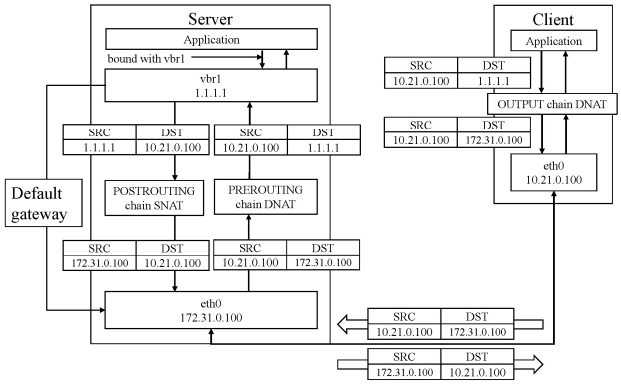

Fig. 2. Packet Flow of HTN [11]

클라이언트의 응용 프로그램이 서버의 숨겨진 IP 주소 (예: 1.1.1.1)로 패킷을 전송하면, 네트워크로 전송되기 전에, 그림 1의 두 번째 단계에서 생성한 서버의 IP 주소 (예: 172.31.0.100)로 목적지 주소 변환 (DNAT)이 수행된다. HTN에서 서버의 숨겨진 IP 주소는 DESIR에서 서버 프로세스가 사용하는 외부에 숨겨진 IP 주소와 매우 유사하다.

서버는 네트워크로부터 자신에게 전송되어 오는 패킷들을 수신하기 위하여, 서버 자신이 변이를 수행할 수 있는 IP 주소 집합 전체에 대하여 Proxy ARP와 동일한 방법으로 목적지 물리주소로 자신의 물리주소를 제공한다. 서버가 성공적으로 패킷을 수신하면 목적지 주소 변환 (DNAT) 기능이 수행되어, 패킷의 목적지 주소가 원래 서버의 숨겨진 IP 주소 (예: 1.1.1.1)로 변환된다. 변환된 패킷을 수신한 서버의 응용 프로그램은 클라이언트에게 응답 패킷을 전송한다. 이 응답 패킷에 대하여 송신지 주소 변환 (SNAT) 기능이 수행되고, 클라이언트의 패킷이 서버의 응용프로그램에 도착할 때까지 처리된 것의 역순으로 서버의 응답 패킷이 처리되고, 결국 클라이언트의 응용 프로그램이 서버의 응답 패킷을 수신하게 된다. 그림 2에서 네트워크로 전달되는 패킷에 사용된 서버의 IP 주소 (예: 172.31.0.100)는 그림 1의 두 번째 단계에서 매우 짧은 주소 변이 주기의 시작 시점에 새로 생성되고 변경된 주소에 해당한다.

그런데, HTN에서와 같이 매우 짧은 주소 변이 주기를 사용하게 되면, DESIR나 SSCM과 같이 연결 마이그레이션을 수행하는 것이 불가능해질 수 있다. 따라서 HTN에서는 RHM의 Long-Lived Connection 처리 방법과 동일하게, 연결을 마이그레이션하는 대신에 서버의 NAT 테이블에 Long-Lived Connection의 (발신지 IP, 발신지 포트, 목적지 IP, 목적지 포트)에 대한 변환 규칙을 삽입하고 연결이 해제될 때까지 유지한다.

다음 장에서는, 보호대상 서버에 HTN 기법이 적용된 경우, 보호대상 서버가 공격자에 의해 식별되고 공격될 확률을 낮추기 위하여 많은 수의 디코이를 포함하는 보호대상 서버 네트워크를 설계한다.

IV. Design of Protected Server Networks

보호대상 서버를 위한 네트워크를 설계하기 위해서는 먼저 해당 네트워크에 할당된 IP 주소의 분배 방안을 마련해야 한다. 이를 위하여, 보호대상 서버에게 할당하는 IP 주소 집합을 정의한 뒤, 디코이가 담당할 IP 주소의 범위를 결정하고, 마지막으로 필요한 디코이의 개수를 계산한다.

먼저 보호대상 서버 네트워크의 전체 IP 주소 집합을 S_{IP}라고 하자. 그리고, 보호대상 서버가 n개 존재하는 경우, 보호대상 서버 $i(1 \le i \le n)$가 주소 변이를 수행할 수 있는 IP 주소 집합을 S_{IP}^i라고 하자. HTN의 경우, 집합 S_{IP}^i는 고정되어 있고 Long-Lived Connection들에 대한 마이그레이션을 제공하지 않는다는 사실에 주의하여야 한다. 따라서 공격자가 트래픽을 모니터링할 수 있다고 가정하면, 집합 S_{IP}와 S_{IP}^i사이에는 $S_{IP} = \cup_{i=1}^{n} S^{i_{IP}}$와 같은 조건이 만족하여야 한다.

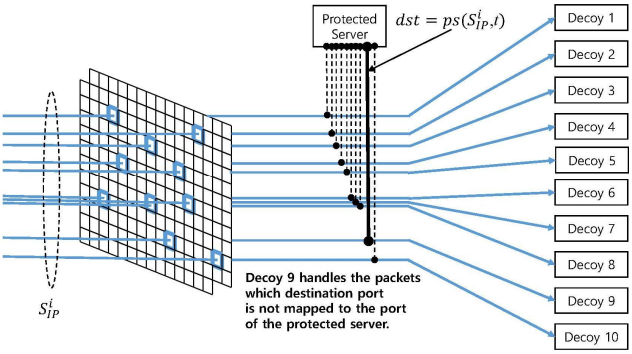

Fig. 3. Concept of Packet Processing by a Protected Server and Its Decoys

이러한 조건이 만족된 네트워크에서 보호대상 서버 i가 집합 S_{IP}^i에 속한 IP 주소들로 지속적으로 주소 변이를 수행하고 Long-Lived Connection이 다수 존재하면, 공격자가 외부에서 트래픽 모니터링을 수행한 결과, 보호대상 서버가 속한 네트워크 전체가 보호대상 서버로 구성되었다고 착각할 수 있다. 예를 들어, 보호대상 서버가 m개의 연결을 유지하고 있고, m개의 연결이 서로 다른 주소 변이 주기에 시작하였다고 가정하면, 서버는 m개의 서로 다른 IP 주소를 사용하여 서비스를 제공하고 있는 것처럼 보이게 된다.

보호대상 서버 i가 집합 S_{IP}^i에 속한 IP 주소들로 지속적인 주소 변이를 수행할 경우, 시간 t에 보호대상 서버 i의 IP 주소를 $ps(S_{IP}^i, t)$라고 하자. 그러면, 공격자가 보호대상 서버의 IP 주소가 아닌 목적지 IP 주소로 패킷을 보내오는 경우 디코이에게 전달하기 위해서는 보호대상 서버 i 각각에 대하여, $|S_{IP}^i| - 1$개의 디코이가 필요하다. 본 논문에서는 보호대상 서버 i를 위한 디코이 그룹을 "디코이-베드 (Decoy-bed) i"라고 표현한다. 그러면, 시간 t에 보호대상 서버 i를 대신하여 디코

이-베드 i가 담당할 IP 주소의 집합은 $S_{IP}^i \setminus \{ps(S_{IP}^i, t)\}$가 된다.

이와 같이 설정하면, 목적지 주소가 $ps(S_{IP}^i, t)$가 아닌 모든 패킷들 중에서 Long-Lived Connection으로 보호대상 서버 i의 NAT (Network Address Translation) 테이블에 등록된 (발신지 IP, 발신지 포트, 목적지 IP, 목적지 포트)를 제외한 모든 패킷을 디코이가 처리하게 된다. 여기에서 주의할 점은 목적지 주소가 $ps(S_{IP}^i, t)$인 패킷이라고 하더라도, 목적지 포트번호가 보호대상 서버 i가 서비스를 제공 중인 포트번호가 아니라면, 이 패킷 역시 디코이가 처리하여야 한다는 것이다. 따라서 앞서 계산한 $|S_{IP}^i|-1$개의 디코이에 추가로, 보호대상 서버 i가 제공 중인 오픈 포트를 제외한 다른 오픈 포트로 전송되어 오는 패킷들을 처리하기 위한 디코이가 한 대 더 필요하게 된다. 결과적으로 보호대상 서버 i를 위한 디코이-베드 i에는 $|S_{IP}^i|$개의 디코이가 필요하다.

이상의 내용을 $|S_{IP}^i|=10$인 보호대상 서버 i와 디코이들에 의한 패킷 처리 개념도로 나타내면 그림 3과 같다. 그림 3에서, 집합 S_{IP}^i에 속한 IP 주소를 목적지로 가지는 패킷은 보호대상 서버 i가 존재하는 네트워크로 진입할 수 있다. 또한, 보호대상 서버의 네트워크로 진입한 모든 패킷들은 각각의 목적지 IP 주소에 따라 해당 디코이로 전달된다. 이 때, 현재 시간 t에 보호대상 서버 i의 IP 주소인 $ps(S_{IP}^i, t)$가 목적지 IP 주소인 패킷에 대하여, 해당 패킷의 포트번호가 보호대상 서버의 서비스 중인 포트번호인 경우에만, 보호대상 서버 i가 이 패킷을 가로채어 NAPT (Network Address & Port Translation)를 통해 서버 프로세스에게 전달한다.

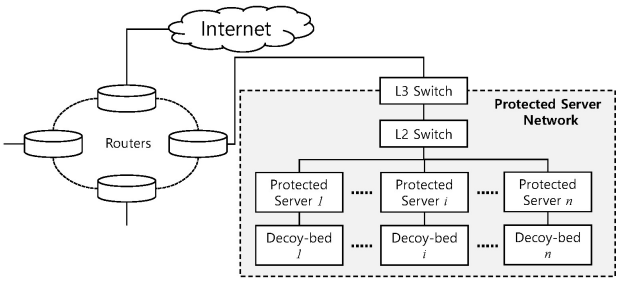

Fig. 4. Network Architecture of Protected Servers and Decoy-beds

일반적으로, 보호대상 서버를 보유하고 있는 기관은 방화벽, 네트워크 침입 탐지 시스템 (NIDS: Network Intrusion Detection System), 라우터 및 스위치 등을 사용하여 보호대상 서버를 인터넷과 연결하고 있다. 그림 4는 방화벽과 네트워크 침입탐지 시스템 등의 위치는 표현하지 않고, 라우터에 연결된 보호대상 네트워크 내에서 보호대상 서버와 디코이-베드가 어떤 구조로 연결되어 있는지를 보여준다.

그림 4에서, 보호대상 서버가 속한 네트워크의 전체 IP 주소 집합 S_{IP}에 속한 IP 주소를 목적지로 하는 모든 패킷들은 L3 스위치와 L2 스위치를 거쳐, 보호대상 서버 네트워크로 전달된다. 또한, 보호대상 서버 i의 집합 S_{IP}^i에 속하는 IP 주소들 중 하나를 목적지로 하는 모든 패킷들은 보호대상 서버 i로 전달된다. 이 때, L3 스위치와 L2 스위치는 보호대상 서버 i가 마치 $|S_{IP}^i|$개의 IP 주소를 모두 사용하고 있는 것으로 판단하고 그에 맞춰 패킷을 포워딩한다.

그림 5는 보호대상 서버 네트워크의 내부 연결 구조를 보여준다. 보호대상 서버들과 L2 스위치 사이에는 집합 S_{IP}에 속한 IP 주소가 사용되고, 보호대상 서버와 디코이-베드 사이에는 사설 IP 주소가 사용된다. 그림 5와 같이, 디코이-베드 i가 사설 IP 주소인 192.168.0.0/24를 사용하는 경우, S_{IP}^i에 속한 각각의 IP 주소는 디코이-베드 i의 특정 디코이와 일대일로 매핑되고, 이러한 매핑 정보는 보호대상 서버 i의 NAT 테이블에 등록된다.

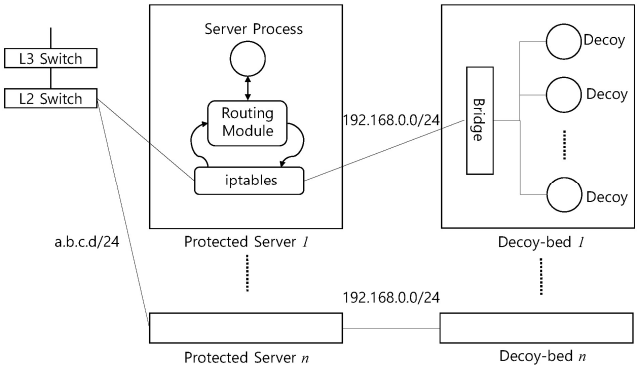

Fig. 5. Architecture of the Protected Server Network

집합 S_{IP}^i에 속한 IP주소와 디코이-베드 i에 속한 디코이의 일대일 매핑 정보가 NAT 테이블에 등록되어 있음에도 불구하고, 앞서 언급한 것과 같이, 목적지 주소가 $ps(S_{IP}^i, t)$이거나 Long-Lived Connections에 속하는 패킷들을 가로채서 보호대상 서버 i의 프로세스로 전달할 수 있는 이유는 NAT 테이블의 위쪽에 위치하는 규칙 (높은 우선 순위의 규칙)에 만족하면 아래쪽 규칙 (낮은 우선 순위의 규칙)을 더 이상 찾지 않는 iptables [15]의 동작 특징 때문이다. 이에 대해서는, 아래의 제안하는 보호대상 서버 네트워크의 첫 번째 장점에서 상세히 서술하도록 한다.

본 논문에서 제안하는 보호대상 서버 네트워크는 다음과 같은 장점을 가진다.

- 보호대상 서버의 매우 빠른 주소 변이 지원
- 디코이-베드 구성의 독립성 제공
- 보호대상 서버 i의 주소 집합 S_{IP}^i의 변경 자유
- 공격자의 수집 정보 재활용 방지

보호대상 서버 네트워크에서 보호대상 서버의 IP 주소 변이

와 디코이-베드 내에서의 디코이 구성 및 관리는 완전히 분리되어 있다. 디코이-베드 i에서 디코이들을 구성하기 위해서는 $|S_{IP}^i|$개의 디코이를 생성한 뒤, 사설 네트워크 주소 중 하나를 할당하면 된다. 그러면, 보호대상 서버는 NAT 테이블에 보호대상 서버 i의 S_{IP}^i에 속한 IP 주소와 디코이에 할당된 사설 IP 주소의 변환 규칙을 등록하면 된다. 이상과 같이 등록된 디코이와의 주소 변화 규칙에 추가로, Long-Lived Connection을 위한 변환 규칙과 현재 시간 t에 보호대상 서버 i의 주소 $ps(S_{IP}^i, t)$에 대한 변환 규칙을 높은 우선 순위를 가지는 규칙으로 추가하면 보호대상 서버 i의 NAT 변환 규칙 설정이 완료된다. 이와 같은 NAT 변환 규칙 구성에서는 보호대상 서버 i의 IP 주소가 $ps(S_{IP}^i, t)$에서 $ps(S_{IP}^i, t')$로 바뀌는 경우, $ps(S_{IP}^i, t)$에 관한 규칙을 삭제하고 $ps(S_{IP}^i, t')$에 관한 규칙을 추가하기만 하면 되는데, 이러한 수정은 1초 미만의 매우 짧은 주소 변이 주기 내에서 수행될 정도로 단순하다. 따라서, 제안하는 보호대상 서버 네트워크는 보호대상 서버의 매우 빠른 주소 변이를 지원한다는 장점을 가진다.

디코이-베드 구성의 독립성도 보호대상 서버 네트워크의 구조와 특성 때문에 얻어지는 장점이다. 디코이와 관련하여 보호대상 서버 i가 수행하는 일은 단지 집합 S_{IP}^i에 속한 IP 주소와 디코이에 할당된 사설 IP 주소의 변환 규칙을 등록하는 것이다. 따라서 디코이-베드 i에서 특정 디코이를 교체 (추가 후 삭제)한다고 하더라도, 보호대상 서버 i에는 어떠한 영향도 발생하지 않는다.

HTN이 보호대상 서버에 적용되는 경우, 보호대상 서버 i의 IP 주소 집합 S_{IP}^i는 시간에 따라 변하지 않는 고정적인 집합이 된다. 그러나 본 논문에서 제안하는 보호대상 서버 네트워크에서는, 향후 집합 S_{IP}^i가 시간에 따라 변경되는 기법이 보호대상 서버에 적용된다고 하더라도, 집합 S_{IP}^i의 변경이 디코이-베드 i에 미치는 영향은 미약하다는 장점을 가지고 있다. 먼저, 집합 S_{IP}^i의 크기는 변하지 않으면서 집합 S_{IP}^i에 속한 IP 주소들만 달라지는 경우에는 디코이-베드에 미치는 영향은 전혀 없다. 다음으로, 집합 S_{IP}^i의 크기가 변하는 경우에는 새로운 디코이를 생성하거나 기존의 디코이를 삭제하는 수준의 관리만 수행하면 된다. 디코이가 도커 (Docker)나 컨테이너 (Container)로 구현되는 경우, 미리 준비된 이미지로부터 디코이를 생성하는 것은 매우 간단한 일이며, 이에 소요되는 시간도 매우 짧다. 또한, 디코이-베드는 사설 IP 주소를 사용하기 때문에, 본 연구에서 제안하는 네트워크를 구축하는 경우, 보호대상 서버 i의 주소 집합 S_{IP}^i의 변경을 자유롭게 결정하고 실행할 수 있다는 장점을 가진다.

HIDE [9]에 따르면, 숙련된 공격자는 IP 주소가 바뀐 후에도 이전에 수집한 핑거프린트를 이용하여 서버를 식별할 수 있다고 한다. 보호대상 서버 i가 제공하는 서비스 목록을 $svc(ps(S^{i_{IP}}, t))$라고 하고, NAT 테이블에 낮은 우선 순위의 변환 규칙으로 시간 t에 $ps(S_{IP}^i, t)$와 매핑되어 있는 디코이가 제공하는 서비스 목록을 $svc(decoy(S_{IP}^i, t))$라고 하자. 그러면, 시간 t에 공격자가 정찰 공격을 통하여 획득한 보호대상 서버 i의 제공 서비스 목록은 $svc(ps(S_{IP}^i, t)) + svc(decoy(S_{IP}^i, t))$가 된다. 서비스 목록 $svc(ps(S^{i_{IP}}, t))$는 보호대상 서버 i의 제공 서비스 구성을 변경하지 않는 한, 시간과 무관하게 고정적인 목록이다. 즉, $svc(ps(S_{IP}^i, t)) = svc(ps(S_{IP}^i, t'))$ for $t \neq t'$이다. 하지만, 새로운 주소 변이 주기에 보호대상 서버는 $svc(decoy(S_{IP}^i, t)) \neq svc(decoy(S_{IP}^i, t'))$ for $t \neq t'$인 새로운 디코이와 NAT 테이블에서 매핑될 수 있다. 따라서, 서로 다른 주소 변이 주기에 속하는 두 개의 시간 t와 t'에 공격자에게 보이는 보호대상 서버 i의 서비스 목록은 각각 $svc(ps(S_{IP}^i, t)) + svc(decoy(S_{IP}^i, t))$와 $svc(ps(S_{IP}^i, t')) + svc(decoy(S_{IP}^i, t'))$가 되어, 서로 다른 서비스 목록이 된다. 즉, 공격자는 이전 주소 변이 주기에서 수집한 정보를 다음 주소 변이 주기에서 재활용할 수 없게 된다. 이러한 특성은 주소 변이 주기가 짧을수록 보다 큰 장점으로 인식될 수 있는데, 앞서 살펴본 바와 같이, 제안하는 보호대상 서버 네트워크는 보호대상 서버의 매우 빠른 주소 변이를 지원하기 때문에, 공격자의 수집 정보 재활용 방지라는 장점을 가진다고 말할 수 있다.

그러나, 제안하는 보호대상 서버 네트워크는 공격자가 RTT (Round-Trip Time) 측정을 통한 디코이 판별이 가능하다는 단점을 가지고 있다. 이러한 단점을 가지는 이유는, 그림 4와 같이, 디코이-베드가 보호대상 서버를 통하여 라우터와 연결되어 있기 때문이다. 만일, 공격자가 보호대상 서버와 디코이가 응답하는 패킷들로부터 RTT를 측정한다면, 보호대상 서버와의 RTT 값이 디코이와의 RTT 값보다 짧다는 사실을 발견할 수 있다. 이와 유사한 문제는 가상 네트워크를 구성하는 기법 [21]에서도 발견되었고 인위적인 링크 지연 (Link Delay)을 생성해야 할 필요성도 언급되었다. 하지만, 인위적인 링크 지연 추가는 정상적인 사용자의 성능을 저하시킬 수 있다는 이유로 적용 여부에 대한 판단을 유보한 상태이다 [21]. 따라서, 본 논문에서도 아직까지는 인위적인 링크 지연 생성을 고려하고 있지 않다.

V. Conclusions

공격자에게 대상 시스템의 취약점을 분석할 수 있는 충분한 시간을 제공하는 ICT 인프라의 정적인 보안 설정을 해결하기 위하여, 보호대상 시스템의 다양한 특징들을 시간의 변화에 따라 역동적으로 변경하는 MTD 기술이 최근 등장하였다. 본 논문에서는, 네트워크 기반 MTD로 제안된 기술들 중에서, HTN이 보호대상 서버에 적용되었을 때, 공격자가 보호대상 서버를

식별하고 공격할 수 있는 확률을 보다 낮추기 위하여 다수의 디코이를 활용하는 보호대상 서버 네트워크를 설계하였다. 네트워크 MTD 방식은 방어하고자 하는 공격자의 방식 또는 유형에 따라 다르게 설계되는데, HTN은 1초 미만의 매우 짧은 시간 간격에서도 보호대상 서버가 주소 변이를 수행할 수 있도록 하여 전문적인 지식을 소유한 인간 공격자가 보호대상 서버를 식별하지 못하도록 하는 것을 목표로 설계된 기법이다. 본 논문에서 설계된 보호대상 서버 네트워크는, 제안 네트워크의 장단점 분석을 통하여 살펴본 바와 같이, 이상의 특징을 가지는 HTN에 최적화된 네트워크라 할 수 있다.

REFERENCES

[1] S. Woo, K. Park, D. Moon, and I. Kim, “Trends in Moving Target Defense based on Network Address Mutation,” Review of Korea Institute Of Information Security And Cryptology, Vol. 28, No. 2, pp. 5-11, April 2018.

[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] H. Okhravi, T. Hobson, D. Bigelow and W. Streilein, “Finding Focus in the Blur of Moving-Target Techniques,” In IEEE Security&Privacy, vol.12, no. 2, pp. 16-26, March 2014.

[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 Workshops - SecureComm 2007, pp. 224-233, 2007.

[14] Niels Provos and Thorsten Holz, “Virtual Honeypots: From Botnet Tracking to Intrusion Detection,”Addison Wesley, 2008.

[15] O. Andreasson, “Iptables Tutorial 1.2.0 - Linux Firewall Configuration,”GNUFree Document, http://www.freetechbooks.com/ iptablestutorial-1-2-0-linux-firewall-configuration-t273.html

[16] S. Achleitner, T. L. Porta, P. McDaniel, S. Sugrim, S. V. Krishnamurthy, and R. Chadha, “Cyber Deception: Virtual Networks to Defend Insider Reconnaissance”, MIST’16, 2016.

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 data processing, IoT, wireless/mobile communications, and distributed services.

Kyung-min Park received his MS degree in computer engineering from Chungnam National University, Rep. of Korea, in 2013. He joined the Electronics and Telecommu nications 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 a senior researcher. 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, data mining, biometrics, and image processing.