情報保護學會論文誌
第 22 卷 第 4 號, 2012. 8

효율적인 인터넷 범죄수사를 위한 범행호스트 탐지 및 범죄행위 입증기술*

김 형 석,[1†] 김 은 진,[2] 김 휘 강[1‡]
[1]고려대학교 정보보호대학원, [2]경기대학교 국제산업정보학과

Network Forensic Evidence Generation and Verification Scheme*

Hyungseok Kim,[1†] Eunjin Kim,[2] Huy Kang Kim[1‡]
[1]Graduate School of Information Security, Korea University,
[2]Department of International Industrial Information, Kyonggi University

요 약

인터넷범죄를 수사함에 있어 가장 중요한 점은 범행호스트의 범행을 입증하는 것이다. 그러나 범죄자들은 범행부인을 위해 범행호스트의 IP주소를 변경하거나 패킷의 출발지 IP주소를 조작한다. 또한 악의적인 어플리케이션을 이용하여 범행기록을 남기지 않는다. 본 논문은 인터넷범죄수사의 한계를 극복하기 위한 Network Forensic Evidence Generation and Verification Scheme을 제안한다. 이 기술은 인터넷범죄수사를 위해 패킷의 생성위치와 전송과정에 주소필드가 조작되지 않았음을 보장하는 증거를 생성하여, 범행기록이 부재하여도 패킷을 통해 입증한다. 그리고 증거생성에 의한 라우터의 성능저하를 최소화하기 위해 Timestamp SecretKey Distribution Scheme과 Flow-Based Selection Scheme을 추가 제안한다. 마지막으로 제안한 기술들을 활용하기 위한 시스템을 구현하고 패킷전송률을 실험한다.

ABSTRACT

One of the most important point in the Internet crime investigation is tracing back and pointing out a criminal host. However, criminals can forge a crime record stored in the crime host, or can utilize malicious applications in order not to leave a crime record. In addition, criminals can change the source IP address of a crime host and deny their involvement. In this study, we suggests the Network Forensic Evidence Generation and Verification Scheme (NFEGVS) to rectify the current limitation of Network Forensic technologies. This scheme can prove who and when the crime has occurred. In addition, this prevents leaking of symmetric key for guaranteeing certification and integrity of Forensic Evidence by proposing the Timestamp Secret Key Distribution Scheme, and minimizes performance degradation of router when generating forensic evidence with the Flow-Based Selection Scheme. In this paper, we implement the proposed scheme and evaluate overall performance of the proposed system.

Keywords: Network forensic, Internet Crime, IP traceback, Packet Marking, Network Security

접수일(2012년 01월 25일), 수정일(2012년 04월 20일), 게재확정일(2012년 08월 01일)
* 본 연구는 지식경제부 및 정보통신산업진흥원의 "대학 IT 연구센터 육성·지원사업"의 연구결과로 수행되었음 (NIPA-2012-H301-12-4008)
† 주저자, hskim5048@korea.ac.kr
‡ 교신저자, cenda@korea.ac.kr

I. 서 론

인터넷범죄란 인터넷을 범죄수단으로 사용하는 범죄를 의미한다. 오늘날 인터넷범죄의 수는 지속적으로 증가하고 있으며, 이에 대응하기 위해 네트워크 포렌식 기술이 다양하게 연구되고 있다[1,2]. 네트워크 포렌식 기술은 누가, 언제, 어떻게 범행을 했는지 네트워크 단에서 수집된 정보를 바탕으로 시스템 내의 디스크 포렌식 및 시스템 정보들과 교차 분석하여 입증해 내는 일련의 작업들이라 할 수 있다. 그러나 현재의 네트워크 포렌식 기술은 피해를 받은 시스템에 기록된 로그를 통해 어떻게 범행이 이뤄졌는지 파악할 수 있으나, 누가 언제 범행했는지 입증하는 것에는 한계를 가지고 있다. 그 이유는 범죄에 이용된 호스트의 IP 주소를 변경한다면, 범죄시점에 해당 IP를 사용한 호스트를 입증하기 어렵기 때문이다. 또한 범죄에 이용된 패킷의 출발지 IP 주소를 조작한다면 패킷의 생성위치를 은닉할 수 있고, 패킷을 조작하지 않았다 하더라도 인증과 무결성을 보장하지 않는 TCP/IP 패킷의 특성을 악용한다면 전송과정에 조작된 패킷이라 주장할 수 있기 때문이다. 뿐만 아니라 범죄자는 범죄에 이용된 호스트에 저장된 로그를 삭제하는 등 범행기록을 조작할 수 있으며, 범행기록 자체를 남기지 않는 애플리케이션을 이용할 수 있기 때문이다.

본 논문에서는 이러한 문제를 해결하고자, 새로운 형태의 네트워크 포렌식 기술인 Network Forensic Evidence Generation and Verification Scheme (NFEGVS)를 제안한다. 본 제안은 범죄에 이용된 IPv4 패킷에 누가 언제 범행했는지 입증할 수 있도록 하였으며, 이를 위해 범죄에 이용된 호스트가 생성한 패킷이 범행대상시스템으로 전달되기 위해 경유하는 라우터에서 패킷의 생성위치와 주소필드의 무결성을 입증하기 위한 포렌식증거를 생성하고, 패킷의 목적지에서 패킷에 저장된 포렌식증거의 적합성을 검증할 수 있도록 하였다. 이 기술은 네트워크 트래픽 기록을 분석하여 증거로 이용할 경우 범행호스트의 기록이 삭제되어 존재하지 않더라도 패킷에 생성한 증거를 통해 범행을 입증할 수 있는 장점이 있다. 이를 위해 패킷이 위변조되지 않았음을 담보하여 증거능력을 인정받고 증명력을 높이고자 한다. 또한 이 기술은 소프트웨어로 구현되어 별도의 장치로 구성이 가능하여, 라우터와 범행대상시스템에 적재되지 않아도 적용이 가능하다.

본 논문에서는 NFEGVS를 위해 2가지 기술을 추가로 제안한다. 첫 번째는 대칭키 알고리즘의 비밀키를 공유하기 위한 Timestamp SecretKey Distribution Scheme (TSKDS)으로 증거의 생성과 검증 시 고속연산이 가능한 대칭키 알고리즘을 활용하기 위하여 비밀키의 유출문제를 극복하는 기술이다. 두 번째는 증거를 생성하는 라우터의 성능저하를 최소화하기 위해 flow 기반으로 증거생성용 패킷을 선별하는 Flow-Based Selection Scheme (FBSS) 기술이다.

본 논문의 2장에서는 관련연구와 문제점을 제시하고, 3장에서는 Network Forensic Evidence Generation and Verification Scheme, 4장에서는 Timestamp SecretKey Distribution Scheme, 5장에서는 Flow-Based Selection Scheme을 소개한다. 6장에서는 제안한 기술들을 활용하기 위한 시스템을 구현하고 활용방안을 제시하며, 7장에서는 구현된 시스템을 실험한다. 마지막으로 8장에서 본 논문의 결론 및 향후 연구계획을 제시한다.

II. 관련연구

2.1 현재의 네트워크 포렌식 기술

네트워크 포렌식 기술은 [표 1]에서 나타나듯이 범행데이터 수집기술과 IP Traceback 기술로 분류할 수 있다. 범행데이터 수집기술이란 패킷이나 어플리케이션의 로그를 수집하여 범행을 입증하는 기술로 수집위치에 따라 범행호스트 수집기술과 범행대상시스템 수집기술로 분류할 수 있다. 범행호스트란 범행에 이용된 호스트를 의미하며, 범행대상호스트란 범행에 의해 피해를 입은 시스템을 말한다. 범행호스트 수집기술로는 Web Historian[3]이나 Index.dat analyzer[4]와 같이 범행에 이용된 어플리케이션에서 작성한 로그를 수집하는 방식이 있다. Web Historian은 Internet Explorer, FireFox, Chrome과 같은 웹브라우저에서 생성한 히스토리파일을 분석하는 웹 포렌식도구로 사용자가 접속한 웹사이트 목록을 제공한다. Index.dat analyzer는 Internet Explorer를 통해 접속한 기록을 보관하는 Index.dat를 분석하여, 임의로 삭제된 기록을 탐지해 내는 웹포렌식 도구이다.

범행대상시스템 수집기술은 범행대상시스템에서 패킷이나 어플리케이션에서 작성한 로그를 수집하는 방식을 말한다. eMailTrackerPro[5]는 수신한 이메

〔표 1〕 네트워크 포렌식 기술의 분류

분류	기술		설명
범행데이터 수집기술	수집위치: 범행호스트	Web Historian〔3〕	- 웹브라우저에서 생성한 히스토리파일을 분석하는 웹 포렌식도구
		Index.dat analyzer〔4〕	- Internet Explorer에서 생성한 접속기록을 관리하는 데이터베이스인 Index.dat 파일을 분석하는 웹 포렌식도구
	수집위치: 범행대상시스템	eMailTrackerPro〔5〕	- 수신한 이메일 헤더를 통해 전송자 위치를 탐지하는 메일 포렌식도구
		TCPDUMP〔6〕, Wireshark〔7〕	- 패킷스니핑 도구
IP Traceback	Link Testing〔8,9〕		- 상위링크를 추적
	Messaging〔10〕		- 라우터에서 목적지시스템으로 ICMP Message 전송
	Logging〔11〕		- 라우터를 경유하는 패킷 로깅
	Packet Marking	Probability Packet Marking 〔12,13〕	- 공격경로를 구성하기 위해 모든 라우터가 마킹수행 - 라우터들은 확률 P에 의해 대상 패킷 선정
		Deterministic Packet Marking 〔14,15〕	- ISP의 진입점라우터만 마킹수행

일의 헤더를 통해 메일송신자의 위치를 탐지하는 이메일 포렌식도구이다. TCPDUMP〔6〕와 Wireshark〔7〕은 대표적인 패킷스니핑 도구로 패킷의 헤더와 페이로드를 통해 누가 어떻게 범죄를 발생시켰는지를 분석할 때 이용한다.

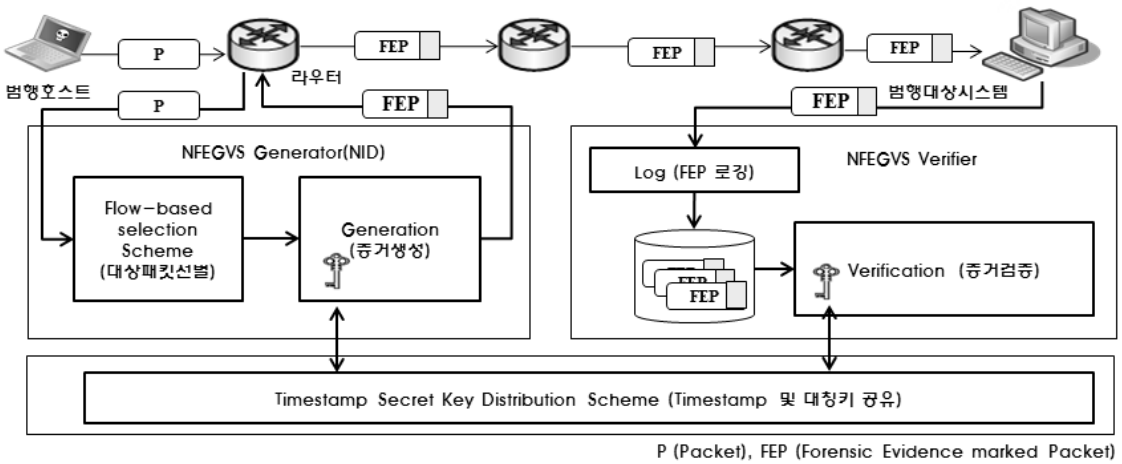
IP Traceback 기술은 라우터를 활용하여 출발지 IP 주소가 조작된 패킷의 근원지를 탐지한다. IP Traceback 기술에는 피해시스템에서 수신한 패킷의 상위링크를 추적하는 Link Testing〔8,9〕과 라우터에서 패킷의 목적지로 ICMP Message를 전송하는 Messaging〔10〕 그리고 라우터에서 경유하는 패킷을 저장하는 Logging〔11〕으로 분류할 수 있다. Link Testing과 Logging은 네트워크 프로토콜을 변경하지 않는다는 장점이 있지만 라우터 자원에 의해 심각한 제약을 받으며〔16〕, Messaging은 ICMP Message가 본래의 의도와 달리 Distributed Denial of Service(DDoS) 공격을 가중시킬 수 있는 단점이 있다〔16〕. IP Traceback의 또 다른 형태의 기술로는 Packet Marking(PM)이 있다.

이 기술은 라우터에서 경유하는 패킷에 흔적을 남기는 방법으로 공격경로 구성여부에 따라 Probability Packet Marking(PPM)〔12,13〕 과 Deterministic Packet Marking(DPM)〔14,15〕으로 분류할 수 있다. 공격경로란 범행호스트에서 생성된 패킷이 라우터들을 경유하며 범행대상시스템까지 전송된 경로를 말한다. PPM은 패킷이 경유하는 모든 라우터에서 추적을 위한 데이터를 생성하여 공격경로를 구성한다. 이 기술은 라우터의 성능저하를 방지하기 위해 라우터마다 확률 P를 통해 추적대상 패킷을 선별하여 선택적으로 마킹을 한다. 하지만, A. Belenky와 N. Ansari는 범행호스트를 탐지하기 위해 공격경로를 구성할 필요성에 의문을 제기하고〔14〕, 진입점라우터에서만 추적데이터를 생성하는 DPM기술을 제안하였다. 공격경로구성은 네트워크 전체의 성능을 저하시키는 반면 범죄호스트를 추적하는 효과는 미비하기 때문이다〔14〕. 또한 PPM과 DPM은 네트워크 프로토콜과의 호환성을 유지하며 추적을 위한 32비트 IP 주소를 저장해야 하는 문제가 있다. S. Savage는 fragment 비율이 0.25% 보다 낮으므로 IP header의 identification(ID) 필드에 저장하는 방식을 제안하였다〔12〕.

2.2 범행데이터 수집기술의 문제점

범행데이터 수집기술은 패킷이나 어플리케이션 로그를 수집하여 범행이 어떻게 발생했는지를 탐지하는 기술이다. 그러나 이 기술은 누가 언제 범행을 발생시켰는지 입증하기 어렵다. 범죄자는 범행대상시스템에서 수집한 로그를 다음과 같은 이유로 부인할 수 있기 때문이다.

첫 번째, 범행호스트의 IP주소는 범죄자에 의해 변경가능하다. 따라서 범죄자가 범행 후 범행호스트의 IP 주소를 변경할 경우, 패킷을 통해 확인한 IP주소로는 범행을 입증하기 어려울 뿐만 아니라, IP주소를 변경하지 않았다 하더라도 범행을 부인할 경우 입증하기 어렵다. 두 번째, 패킷의 출발지 IP 주소는 전송과



〔그림 1〕 NFEGVS 구성

정에 변경가능하다. TCP/IP 패킷은 무결성이 보장되지 않으며, Network Address Translation(NAT)〔17〕과 같이 정상적인 절차에 의해서도 변경될 수 있다. 따라서 범죄자는 이를 악용하여 범행 부인이 가능하다. 세 번째, 패킷의 출발지 IP주소는 범행호스트에서 조작생성이 가능하다〔18〕. TCP/IP 패킷은 인증이 보장되지 않으므로 출발지 IP 조작을 통해 범행을 은닉하거나 타인에게 전가할 수 있다. 네 번째, 범죄자들은 범행호스트의 범죄기록을 조작할 수 있다. 충분히 숙련된 범죄자들은 로그를 작성하지 않는 어플리케이션을 사용하여 범행흔적을 남기지 않을 뿐 아니라, 남겨진 로그에 대해서도 충분히 조작하거나 삭제할 수 있다.

2.3 IP Traceback 기술의 문제점

현존하는 IP Traceback 기술들은 2가지 문제점을 가지고 있다. 첫 번째, IP Traceback 기술은 추적범위가 제한적이다. IP Traceback은 패킷이 어떤 라우터를 경유했는가를 통해 범행호스트의 위치를 탐지하나, 라우터는 서로 다른 네트워크로 연결된 장비로 어떤 네트워크에서 패킷을 전송했는지 입증할 수 없다. 두 번째, IP Traceback 기술은 추적데이터의 인증과 무결성을 보장하지 않는다〔13〕. 따라서 범죄자는 이 기술에서 제시하는 증거를 부인할 수 있을 뿐 아니라, 증거를 조작하여 범행을 은닉할 수 있다.

III. Network Forensic Evidence Generation and Verification Scheme (NFEGVS)

본 논문에서는 라우터를 통해 경유하는 패킷에 범행을 입증하는 증거를 생성하고, 범행대상시스템에서 증거의 적합성을 검증하는 Network Forensic Evidence Generation and Verification Scheme (NFEGVS)을 제안한다. 이 기술에서 범행호스트가 아닌 라우터에서 네트워크 패킷에 증거를 생성하는 이유는 다음과 같다.

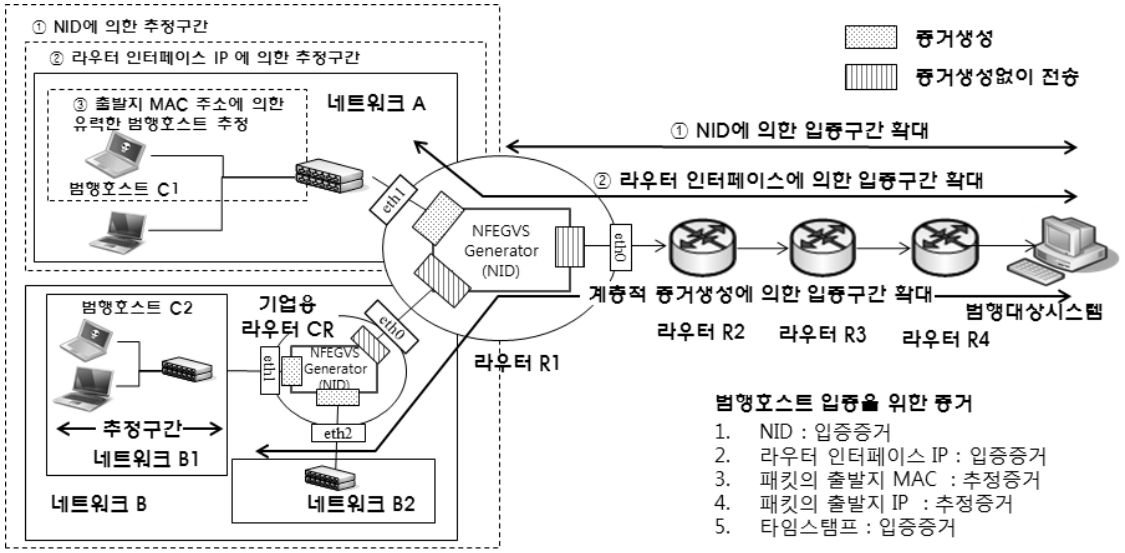
범죄자에게 점유되어 있는 범행호스트는 증거가 조작될 수 있으므로 신뢰하기 어렵다.

범행호스트에서 관련증거가 삭제되어도 네트워크 단에서 입증이 가능하다.

NFEGVS는 〔그림 1〕에서 나타나듯이 라우터에 적재되어 증거를 생성하는 NFEGVS Generator 모듈과 범행대상시스템에 적재되어 증거의 적합성을 검증하는 NFEGVS Verifier 모듈 그리고 Timestamp SecretKey Distribution Scheme으로 구성되어 있다.

3.1 NFEGVS 증거의 증명력 강화

범행호스트가 아닌 라우터에서 증거를 생성하는 방법은 범죄자에 의한 관련증거 조작을 방지하는 장점을 가진다. 그러나 범행패킷 생성위치와 증거생성위치가 상이하여 제한적인 추적범위를 가지게 된다. 그 이유는 증거를 생성한 위치부터 범행을 입증할 수 있기 때



〔그림 2〕 범행호스트 입증을 위한 증거

문이다. 본 논문에서는 이 문제점을 극복하고 증거의 증명력을 강화하기 위해 다음과 같은 3가지의 목표를 설정한다.

- 패킷 주소필드의 무결성을 보장하여, 범행호스트의 범행부인에 대응한다.
- 범행호스트의 위치를 입증하는 구간을 확대하여, 범죄수사 범위를 집중시킨다.
- 인증과 무결성을 보장한 증거를 생성하여 증명력을 높인다.

3.2 패킷의 주소필드 무결성 보장

NFEGVS는 범행호스트의 범행부인에 대응하기 위해 범행패킷의 주소필드의 무결성을 보장한다. TCP/IP 프로토콜은 출발지 호스트에서 생성한 패킷을 목적지 호스트까지 전달하기 위해 3가지 주소를 사용한다[19]. 3가지 주소란 network interface layer간 hop-to-hop delivery를 위한 MAC 주소와 internet layer간 source-to-destination delivery를 수행하는 IP 주소 그리고 transport layer간 process-to-process delivery를 수행하는 port이다. 그리고 protocol을 추가로 이용한다. Protocol은 전송을 위한 주소필드는 아니지만, transport layer를 구분한다. 이 주소들은 범행패킷이 생성된 위치와 최종적으로 전달될 위치 그리고 어떤 서비스를 이용하기 위한 패킷인지를 나타내는 중요한 정보이다.

따라서 NFEGVS는 인증과 무결성을 보장하지 않는 TCP/IP 프로토콜의 특성을 범죄자가 악용하지 못하도록, 범행패킷 주소필드의 무결성을 보장한다. 이를 위해 NFEGVS Generator는 유입된 패킷으로부터 주소필드를 수집하여 주소데이터(AD; Address Data)를 구성한다. 라우터를 경유할 때마다 변경되는 출발지 MAC주소나 범행호스트 은닉에 이용되는 출발지 IP주소와 같은 필드들은 다음 절의 입증구간을 확대하는 증거로 활용되며, 해당 값들은 패킷에 저장되어 AD와 함께 주소필드의 무결성을 보장하게 된다. 따라서 AD는 패킷의 주소필드 중 MAC주소와 출발지 IP 주소를 제외한 72비트의 목적지 IP주소, 출발지 port, 목적지 port, protocol로 구성된다.

3.3 범행호스트 입증구간 확대

범행패킷을 생성하는 위치와 증거를 생성하는 위치가 상이한 것은 범행호스트의 위치를 탐지하기 위한 입증구간과 추정구간 문제를 발생시킨다. 입증구간이란 생성된 증거를 이용하여 범행호스트가 존재하는 위치를 입증할 수 있는 구간을 의미하며, 이를 위해 수집된 증거를 입증증거라 정의한다. 추정구간이란 생성된 증거를 이용하여 입증할 수는 없으나 유력한 범행호스트가 존재하는 위치를 추정할 수 있는 구간을 의미하며, 이를 위해 수집된 증거를 추정증거라 정의한다.

범행대상호스트에서 수집한 범행패킷이나 IP

Traceback 기술로 마킹된 패킷의 경우 입증구간은 존재하지 않고 추정구간만 존재하게 된다. 그 이유는 인증과 무결성이 보장되지 않기 때문이다.

본 논문에서는 이 문제를 극복하기 위해 범행호스트의 위치를 탐지하는 입증구간을 확대하고 추정구간을 축소시켜, 추정오류에 의해 발생 가능한 비효율적인 범죄수사를 개선한다. 이를 위해 NFEGVS는 다음과 같은 5가지의 정보를 증거로 활용한다.

첫 번째, NFEGVS Generator별로 할당되는 고유한 식별자인 NID(NFEGVS Identification)를 활용한다. NFEGVS는 생성된 증거의 신뢰성을 높이기 위해 인증과정을 수행한 NFEGVS Generator에 고유한 NID를 부여한다. 만약 〔그림 2〕의 범행호스트 C1에서 범행대상시스템을 공격했다면, NID는 C1이 네트워크 A나 B에 존재한다는 것을 입증한다. 이때 추정구간은 네트워크 A와 B 전체가 된다.

두 번째, 패킷이 유입된 라우터의 인터페이스 IP 주소를 입증증거로 활용한다. 〔그림 2〕의 C1이 공격한 경우, 라우터 R1의 인터페이스 eth1 주소를 증거로 활용하면, C1이 네트워크 A내에 존재한다는 것을 입증하여 네트워크 B는 수사범위에서 제외된다. 이때 추정구간은 네트워크 A가 된다.

세 번째, 패킷의 출발지 MAC 주소를 추정증거로 활용한다. MAC주소는 Network Interface Card (NIC) 제조사에서 고유하게 설정되어 배포되는 유일성을 가지는 주소이다. 또한 패킷의 MAC주소는 전송과정에 hop-to-hop delivery〔19〕에 의해 라우터를 경유할 때마다 변경되므로 범죄대상시스템에서는 파악할 수 없는 중요한 정보이다. 이런 특성은 범죄자가 출발지 IP 주소만큼 출발지 MAC 주소를 조작할 필요성을 느끼지 못하게 한다. 〔그림 2〕의 C1이 공격한 경우, 패킷의 MAC주소를 증거로 활용하면 네트워크 A내에서 유력한 범행호스트 C1을 추정할 수 있어, 수사대상의 우선순위를 선정할 수 있다. 그러나 패킷의 출발지 MAC주소는 출발지 IP 주소와 마찬가지로 조작이 가능하고, 중간에 라우터가 존재한다면 변경이 가능하므로 추정증거로 활용한다.

네 번째, 패킷의 출발지 IP 주소를 추정증거로 활용한다. 범행호스트의 IP 주소는 변경될 수 있고, 패킷의 출발지 IP 주소는 조작이 가능하지만, 유력한 범행호스트를 추정하므로 수사대상의 우선순위를 선정할 수 있다.

다섯 번째, 범행이 발생한 시간은 범행호스트의 위치를 입증하는 증거는 아니지만 인터넷범죄수사에서 중요한 요건 중 하나이다. 그러나 범행호스트나 범행대상시스템의 시간은 사용자에 의해 변경이 가능하여 신뢰할 수 없을 뿐만 아니라, TCP/IP 패킷에는 시간을 기록하는 필드가 없다. 따라서 본 논문에서는 NFEGVS에서 제공하는 타임스탬프를 범죄시간으로 활용한다.

NFEGVS Generator는 5가지의 증거로 구성된 152비트의 추적데이터(TD; Tracking Data)를 범행패킷에 저장하여 최종목적지로 전송한다.

3.4 증거생성위치 선정

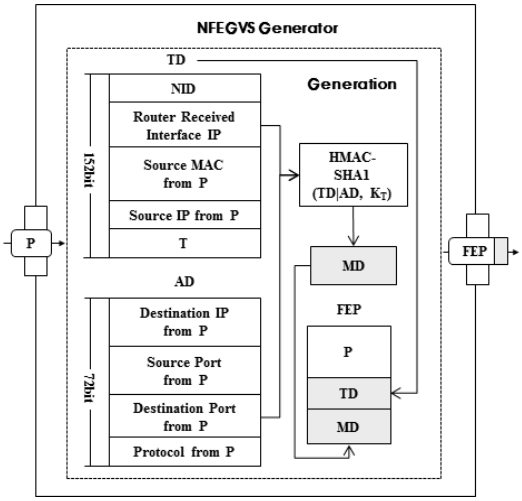
NFEGVS를 이용하여 증거생성위치를 선정하는 문제는 증거의 신뢰성을 결정하는 중요한 문제이다. 이 문제는 다음과 같은 두 가지 요건에 의해 결정된다.

첫 번째, 증거생성위치는 범행호스트와 인접할수록 증명력을 높일 수 있다. 〔그림 2〕의 라우터 R3에서 증거를 생성한다면, 입증구간은 R3에서 범행대상시스템까지가 되고 추정구간은 R3로 패킷을 전송할 수 있는 모든 네트워크가 되어 수사범위가 광범위해진다.

두 번째, 증거생성위치는 신뢰할 수 있는 라우터에서 생성해야 한다. 범죄자가 사용하는 사설공유기는 범행호스트처럼 증거가 조작될 수 있다. 따라서 증거는 신뢰할 수 있는 기관에서 관리하는 라우터에서 생성되어야 한다.

따라서 본 논문에서는 ISP의 진입점라우터를 기점으로 네트워크 환경구성에 따라 계층적으로 증거를 생성한다. 이를 위해 NFEGVS Generator 모듈은 라우터의 인터페이스별로 증거를 생성할 수 있게 설계하였다.

〔그림 2〕의 진입점라우터 R1은 eth0를 통해 공용네트워크와 연결되고, eth1을 통해 ADSL과 같은 개인호스트들로 구성된 네트워크 A와 연결되며, eth2를 통해 기업용 네트워크인 네트워크 B와 연결되어 있다. 만약 R1의 eth2에서 범행호스트 C2에서 생성한 범행패킷에 증거를 생성한다면, 추정구간은 네트워크 B 전체가 된다. 그러나 R1의 eth2에서는 유입되는 패킷에 증거생성 없이 전송하고, 기업용 라우터 CR의 eth1에서 증거를 생성한다면 〔그림 2〕에 나타나듯이 입증구간은 CR까지 확대되고 추정구간은 네트워크 B1으로 축소되어 효율적인 범죄수사를 진행할 수 있다.



〔그림 3〕 Generation 처리절차

3.5 NFEGVS Generation

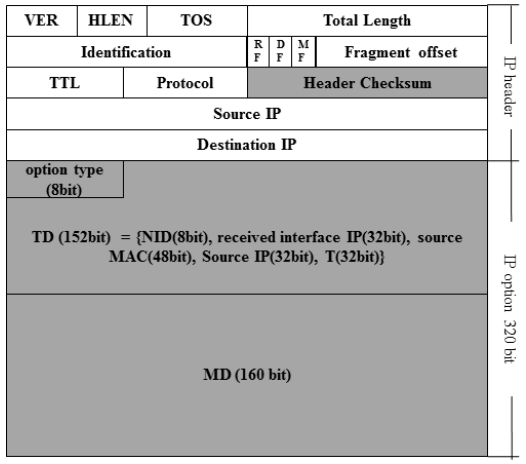
Generation은 〔그림 3〕에 보이는 것처럼 NFEGVS Generator 모듈에서 인증과 무결성을 보장한 증거를 생성한다. 인증과 무결성 보장방법에는 대칭키 알고리즘을 사용하는 HMAC〔20〕과 공개키 알고리즘을 사용하는 디지털서명이 존재한다. D.X. Song과 A. Perrig는 디지털서명의 늦은 연산처리 속도를 근거로 HMAC을 사용하는 것을 제안하였다〔13〕. 본 논문에서는 Generation을 HMAC-SHA1과 DSA로 구현한 후 실험을 통해 패치전송률을 비교한다.

Generation은 패킷이 유입되면 패킷과 라우터로부터 152비트의 TD와 72비트의 AD를 구성하고, 본 논문에서 추가로 제안한 Timestamp SecretKey Distribution Scheme으로부터 표준시간인 타임스탬프(T; Timestamp)와 T에 대한 비밀키 K_T를 제공받아 160비트의 메시지 다이제스틀 다음과 같이 생성한다.

$$MD = HMAC - SHA1(TD|AD, K_T) \quad (1)$$

생성된 MD는 TD와 함께 패킷에 저장되어 다음경로로 전달된다.

본 논문에서는 TD와 MD를 저장하기 위해 패킷의 IP Option 필드를 사용한다. IP option 필드는 20바이트의 IP 헤더를 debugging과 같은 용도로 60바이트까지 유연하게 사용하기 위한 필드이다〔21〕. NFEGVS는 option type만을 사용하는 case 1 형식〔21〕을 이용하고, 새로운 option type인 00100001을 정의한다. 신규 정의된 00100001은 copied flag 0, future use class 01 그리고 number 00001로 현재 사용되지 않은 예약된 option type으로 구성된다〔21〕. NFEGVS는 152비트의 TD, 160비트의 MD 그리고 8비트의 option type을 〔그림 4〕에 나타나듯이 IP option 필드에 저장한 후 IP 헤더의 checksum을 갱신한다.

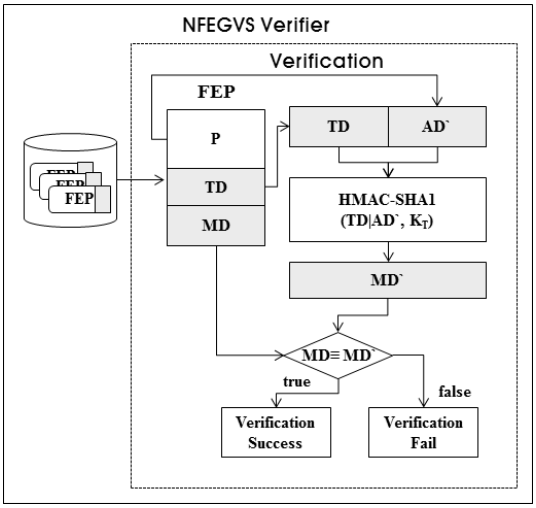


〔그림 4〕 IP option 필드에 증거저장

IP option 필드의 사용은 다음과 같은 2가지 문제를 발생시킬 수 있다. 첫 번째, 320비트의 증거저장에 의해 패킷최대크기가 초과되는 문제이다. 본 논문에서는 이를 방지하기 위해 초과 발생 시 패킷 P를 P와 P`로 분리한 후 선행패킷인 P에 증거를 저장한다. 두 번째, IP option 필드를 사용 중인 패킷에 저장하는 문제이다. 본 논문은 이를 위해 패킷을 P와 P`로 분리한 후 P`에 증거를 저장한다. 본 논문에서는 IP option 필드에 증거가 저장된 패킷을 FEP(Forensic Evidence marked Packet)이라고 정의한다.

3.6 NFEGVS Log & Verification

증거가 저장된 패킷들은 라우터들을 경유하며 최종목적지인 범행대상시스템에 전달된다. NFEGVS는 증거의 보관과 검증을 위해 〔그림 1〕에 나타나듯이 범행대상시스템에 NFEGVS Verifier 모듈을 적재한다. 이 모듈은 유입되는 패킷들로부터 FEP를 선별하여 저장하는 Log와 저장된 FEP의 적합성을 검증하는 Verification으로 구성되어 있다. Log는 패킷의 IP option type이 NFEGVS에서 정의한 00100001인



〔그림 5〕 Verification 처리절차

패킷을 확인한 후 외부저장장치에 저장한다.

Verification은 〔그림 5〕에 나타나듯이 저장장치에 기록된 FEP의 적합성을 검증한다. 이를 위해 주소데이터 AD`={Destination IP, Source Port, Destination Port, Protocol}를 구성하고, TD에 저장된 NID와 T를 Timestamp SecretKey Distribution Scheme으로 전송하여 비밀키 $K_T$ 전달받는다. 전달받은 $K_T$를 이용하여 다음 수식과 같이 MD`를 계산한 후, FEP에 저장된 MD와 비교하여 적합성을 검증한다.

$$MD' = HMAC\text{-}SHA1(TD|AD', K_T) \quad (2)$$

IV. Timestamp SecretKey Distribution Scheme (TSKDS)

NFEGVS는 대칭키 알고리즘인 HMAC-SHA1을 통해 증거를 생성하고 검증한다. 대칭키 알고리즘은 공개키 알고리즘에 비해 빠른 연산처리가 가능하나 〔13〕, 비밀키 유출문제가 발생할 수 있다. 본 논문에서는 비밀키 유출문제를 해결하여 빠른 연산처리를 수행하는 HMAC-SHA1을 통해 인증과 무결성을 보장하고자 한다. 이를 위해 Timestamp SecretKey Distribution Scheme(TSKDS)을 제안한다.

이 기술은 〔그림 6〕에 나타나는 것처럼 Key Distribution Server(KDS)를 이용하여 표준 타임스탬프인 T와 비밀키인 K_T를 공유한다. 대칭키 공유절차는 다음과 같다. 첫 번째, KDS는 유효주기(ET; Effective Timestamp)마다 NID별로 새로운 T와 K_T를 생성하여, 암호화통신을 통해 NFEGVS Generator로 전송한다. NFEGVS Generator는 전달받은 K_T를 통해 증거를 생성하고, FEP에 NID와 T를 저장하여 패킷의 목적지로 전송한다. ET마다 비밀키를 생성하는 이유는 비밀키 공유과정이 많은 수의 패킷에 증거를 생성해야 하는 NFEGVS Generator의 성능에 미치는 영향을 최소화하기 위해서다. 두 번째, NFEGVS Verifier의 Log는 KDS로부터 정기적으로 T를 전달받아, 다음 수식과 같이 FEP에 저장된 T의 적합성을 검증한다.

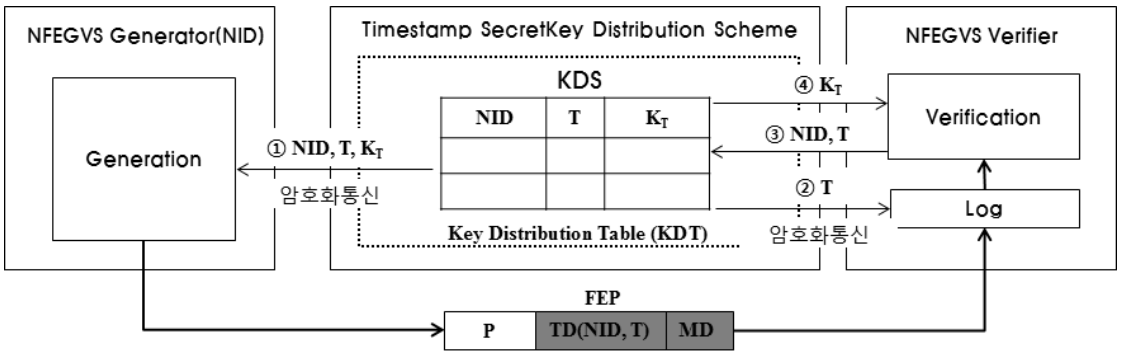
$$T-(ET+\alpha) \le FEP.T \le T \quad (3)$$

α는 패킷이 라우터들을 경유하며 최종목적지까지 전송하는데 소요되는 평균전송시간을 나타내며, 연구결과에 따르면 패킷전송시간은 450ms 보다 적게 소요된다〔27〕. 따라서 FEP에 저장된 T는 ET와 α를 더한 시간을 경과하지 않았다면 적합성을 검증받는다. 또한 ET는 범죄자가 암호화통신으로 공유되는 대칭키를 알아내고 증거를 조작하는데 소요되는 시간보다 적게 설정한다면 대칭키가 유출되어 발생하는 문제로부터 안전할 수 있다. 적합한 ET의 값은 향후 연구를 통해 진행할 예정이다. 세 번째, NFEGVS Verifier의 Verification은 FEP의 NID와 T를 KDS로 전달하여 비밀키를 요청한다. 네 번째, KDS는 요청한 K_T를 전송하고, Verification은 증거를 검증한다.

V. Flow-Based Selection Scheme (FBSS)

NFEGVS Generator에서 유입되는 모든 패킷에 암호알고리즘을 이용하여 증거를 생성한다면, 라우터는 심각한 성능저하가 발생한다. 만약 한 개의 패킷이 라우터를 통해 전송되는 평균시간을 α라 하고 증거를 생성하는 평균시간을 β라고 가정한다면, 10,000개의 패킷이 라우터를 통해 전송될 경우 전송시간은 10,000α의 시간이 소요되며, 동일한 수의 패킷에 증거를 생성하여 전송한다면 10,000(α+β)의 시간이 소요하게 된다. 따라서 10,000β 만큼 시간지연이 발생하게 된다.

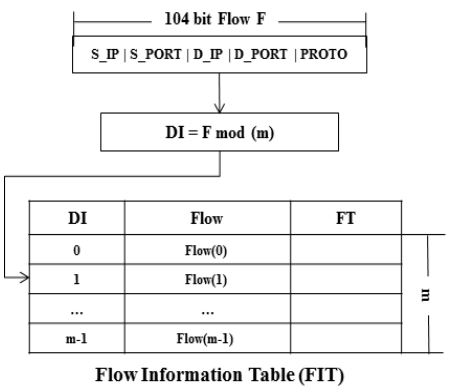
본 논문에서는 증거생성에 의해 발생하는 성능저하를 최소화하기 위해 Flow-Based Selection Scheme(FBSS) 방법을 제안한다. FBSS는 〔그림 1〕에 나타나듯이 NFEGVS Generator에 추가되어



〔그림 6〕 TSKDS 처리절차

증거생성 대상패킷을 선별한다. FBSS는 〔그림 7〕에서 보이듯이 m개의 크기를 가지는 Flow Information Table(FIT)을 이용하여 동일한 flow에 대한 증거의 중복생성을 방지하여 라우터의 성능저하를 최소화한다. flow란 출발지에서 생성된 패킷을 목적지까지 전달하기 위해 패킷의 주소필드로 구성된 트래픽 분류방법으로 DDoS를 효율적으로 탐지하는 방법 중 하나이다〔22〕. 만약 10,000개의 패킷이 1,000개의 flow로 구성되었다면 FBSS는 1,000개의 패킷에 대해서만 증거를 생성한다. 따라서 FBSS를 활용하여 10,000개의 패킷에 증거를 생성하고 전달하기위해서는 10,000α+1,000β 시간이 소요되어 시간지연을 최소화할 수 있다.

FBSS는 〔그림 8〕처럼 증거를 생성한 flow를 FIT에 등록한다. 이때 FIT에서 m의 크기를 결정하는 것은 중요한 이슈가 된다. 그 이유는 104비트의 flow로 표현 가능한 집합 $G \in \{g \mid 0 \le g \le 2^{104}\}$ 이므로 순차 검색한다면 시간복잡도는 $O(2^{104})$가 된다. 따라서 본 논문에서는 $0 \le m \le 2^{104}$인 m을 설정 가능하게 하였다. 향후 라우터의 성능, 라우터가 관리하는 호스트의 수 그리고 발생하는 flow의 분포를 대상으로 최적화된 m의 값을 계산하는 연구를 진행할 것이다. 이 경우 m의 크기를 가지는 FIT의 순차 검색에 대한 시간복잡도는 O(m) 이 된다.



〔그림 7〕 FIT의 다이제스트 인덱스 연산

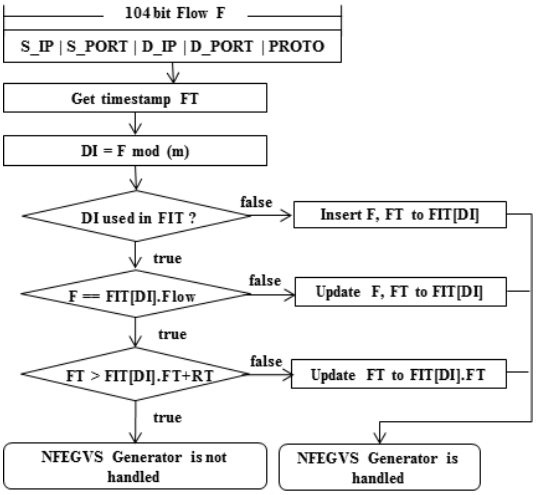
FBSS는 FIT의 효율적인 검색과 관리를 위해 2가지 방법을 이용한다. 첫 번째, 〔그림 7〕에 보이듯이 해시로 계산된 다이제스트 인덱스(DI; Digest Index)를 FIT의 인덱스로 활용한다. FBSS는 패킷 P가 유입되면 flow F를 구성하고 다음과 같이 다이제스트 인덱스를 계산한다.

$$DI = Fmod(m) \tag{4}$$

DI는 FIT의 인덱스로 활용되며 시간복잡도는 O(1)로 개선된다. 그러나 m이 2^{104}보다 작은 경우 서로 다른 flow가 동일한 DI를 가지는 충돌이 발생한다.

두 번째, 충돌을 해결하기 위해 FIT〔DI〕.Flow와 F를 비교한다. 만약 동일하다면 이미 증거를 생성한 flow이므로 해당 패킷에 대해서 증거를 생성하지 않는다. 그러나 동일하지 않다면 FIT에 등록된 flow를 새로운 F로 갱신한 후 증거를 생성한다. 따라서 개선된 시간복잡도는 해시연산과 flow 비교연산을 추가한 O(2)가 된다.

마지막으로 FBSS는 FIT에 등록된 flow들에 대해 지속적으로 증거가 생성되지 않는 문제를 해결하기 위해 보존시간(RT; Retention Timestamp)을 활용한다. 보존시간이란 FIT에 등록된 flow가 유지되는 시간이다. 이를 위해 flow는 FIT에 등록될 때 FIT 등록 타임스탬프(FT; FIT insert Timestamp)를 저장한다. 만약 FIT에 등록된 flow가 유



〔그림 8〕 FBSS 증거생성 대상패킷 선별 알고리즘

입되었다면, FIT〔DI〕.FT가 보존시간인 RT를 경과했다면 FIT를 갱신하고 해당 패킷에 증거를 생성한다.

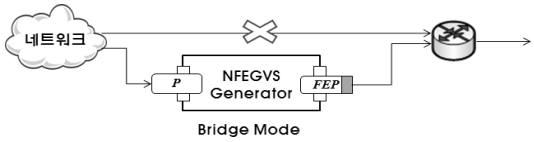
VI. 시스템 구현 및 활용방안

6.1 NFEGVS Generator

본 논문은 제안한 기술들을 통해 증거를 생성하는 NFEGVS Generator를 리눅스 2.6 기반의 어플리케이션 게이트웨이로 구현하였다. 이를 위해 libipq〔23〕 라이브러리를 활용하여 커널영역의 패킷을 어플리케이션영역에서 처리할 수 있게 하였다. 또한 OpenSSL〔24〕의 Crypto 라이브러리를 이용하여 HMAC-SHA1과 DSA 방식의 인증과 무결성이 보장된 증거를 생성하였다.

6.2 NFEGVS Verifier

본 논문은 범행대상시스템에 적재되어 증거를 검증하는 NFEGVS Verifier를 리눅스 2.6 기반의 libpcap〔25〕 라이브러리를 사용하여 구현하였다. libpcap은 어플리케이션 레벨의 패킷 캡쳐 라이브러리이다. NFEGVS Verifier는 pcap_next_ex 함수를 통해 패킷을 수신하면 증거가 저장된 패킷 FEP를 선별한 후 pcap_dump 함수를 이용하여 NID별로



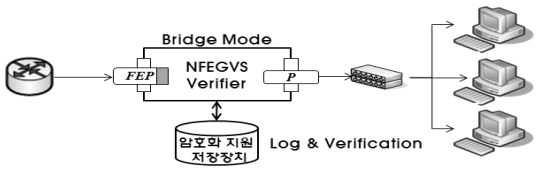
〔그림 9〕 NFEGVS Generator 별도장치 구성

외부저장장치에 기록한다. 기록된 패킷은 Verification 모듈에 의해 적합성이 검증된다. Verification 모듈은 OpenSSL〔25〕의 Crypto 라이브러리를 이용하여 HMAC-SHA1과 DSA 방식의 인증과 무결성이 보장된 증거를 검증하였다.

6.3 활용방안

본 논문에서는 제안한 NFEGVS Generator와 Verifier 모듈을 소프트웨어로 구현하여 실현 가능한 기술임을 입증하였다. 그러나 NFEGVS Generator와 같이 라우터에 적재해야하는 기술들은 라우터 제조사에서 채택하지 않는다면 실용화되기 어렵다. 따라서 NFEGVS Generator는 라우터에 영향을 미치지 않기 위해 bridge 방식의 게이트웨이로 구현한 별도의 장치로 구성될 수 있다. 이 경우 기존 구성환경에 적용하기 쉽고, 라우터에 적재될 필요가 없는 장점이 있다. 〔그림 9〕는 별도의 장치로 구현된 NFEGVS Generator를 기존 네트워크에 구성하는 방법을 도식화 한 것이다.

NFEGVS Verifier 모듈 역시 운영체제 제조사에서 채택하지 않는다면, 사용자들이 개별적으로 설치해야 한다는 단점이 있다. 따라서 NFEGVS Verifier는 NFEGVS Generator와 마찬가지로 〔그림 10〕과 같은 bridge 방식의 게이트웨이로 구현할 수 있다. 이 경우 개인 PC에 설치할 필요가 없을 뿐만 아니라, 암호화를 지원하는 외부저장장치에 기록하여 PC에서 기록할 경우 발생 가능한 FEP의 조작을 방지할 수 있다.



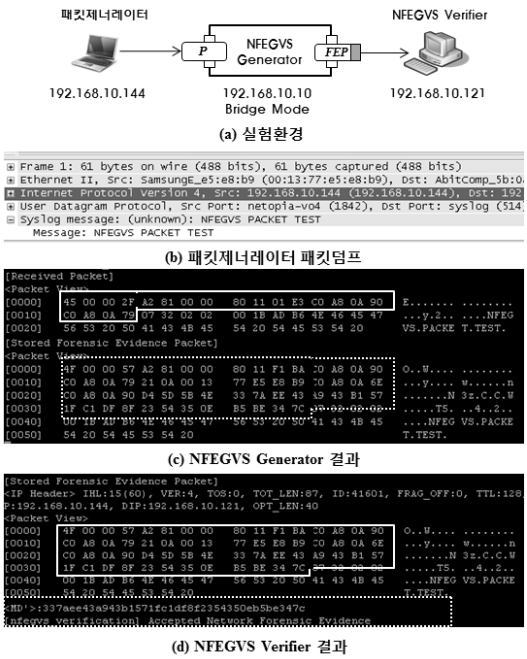
〔그림 10〕 NFEGVS Verifier 별도장치 구성

VII. 실험

7.1 구현결과

〔그림 11〕의 (a)는 소프트웨어로 구현된 NFEGVS Generator와 NFEGVS Verifier의 실험환경을 나타낸다. NFEGVS Generator는 bridge로 구성되었으며, NFEGVS Verifier 시스템과 패킷제너레이터와 연결하였다. 실험대상인 NFEGVS Generator의 시스템사양은 Intel Xeon 2CPU, 4MB memory, 500G HDD, 1000M Giga Ethernet NIC 2개로 구성되었다.

NFEGVS Generator는 패킷제너레이터에서 생성한 패킷 P를 수신하면 증거를 생성한 후 FEP로 전환하여 패킷의 목적지인 NFEGVS Verifier 시스템으로 전송한다. 〔그림 11〕의 (b)는 패킷제너레이터에서 생성한 패킷 P를 Wireshark를 이용하여 출력한 화면이다. 〔그림 11〕의 (c)는 NFEGVS Generator에서 수신한 P와 생성한 증거를 저장한 FEP의 내용을 출력한 화면이다. 그림의 상단에 실선으로 표시된 부분은 수신한 P의 IP 헤더를 나태내고, 점선으로 표시된 부분은 증거가 저장된 FEP의 IP 헤더를 나타낸다. 〔그림 11〕의 (d)는 NFEGVS Verifier에서 FEP를 검증한 결과를 나타낸다. 그림 상단의 실선으로 표시된 부분은 FEP의 IP 헤더를 나타내고, 하단의 점선으로 표시된 부분은 HMAC-SHA1을 통해 생성한 MD′와 검증결과를 나타낸다.



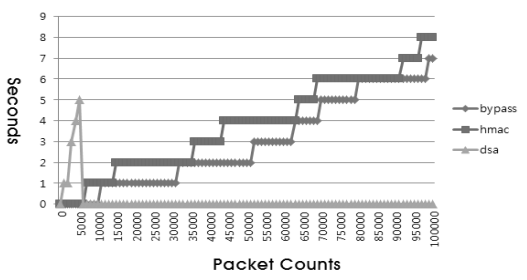
〔그림 11〕 NFEGVS Generator와 Verifier 구현결과

7.2 NFEGVS Generator 패킷전송률 실험

본 논문에서는 NFEGVS Generator의 패킷전송률을 실험하였다. 실험방법은 mausezahn〔26〕 도구를 통해 100,000 개의 패킷을 초당 30,000개 이상씩 4초 이내로 NFEGVS Generator에 전송했을 때, 어플리케이션 레벨에서 bypass한 경우와 대칭키 알고리즘인 HMAC-SHA1을 이용하여 증거를 생성한 경우 그리고 공개키 알고리즘인 DSA를 이용하여 증거를 생성한 경우 패킷을 전송하는데 소요되는 시간을 측정하였다.

어플리케이션 레벨 bypass란 커널영역에서 패킷을 전달하는 것이 아닌 libipq를 통해 패킷을 어플리케이션영역으로 전달한 후 bypass하는 것을 의미하는 것으로, 어플리케이션 게이트웨이로 작성된 NFEGVS Generator의 암호화 시 패킷전송률을 비교하기 위한 지표로 활용된다.

〔그림 12〕의 실험결과를 보면 어플리케이션 레벨에서 bypass한 경우 100,000개의 패킷을 전송하는데 7초가량 소요되었다. 대칭키 알고리즘인 HMAC-SHA1의 결과는 동일한 수의 패킷을 전송하는데 어플리케이션 bypass보다 1초 더 소요된 8초를 나타낸다. 그러나 공개키 알고리즘인 DSA의 경우 초당 1,000개의 패킷을 전송하다 5초 경과한 후 시스템불능 상태에 도달하였다. 그 이유는 어플리케이션에서 패킷에 암호화알고리즘 처리가 지연되어, 커널에서 유입된 패킷들이 어플리케이션 영역으로 전달되지 못하고 메모리가 소모되었기 때문이다.



〔그림 12〕 패킷전송률 실험결과

VIII. 결론 및 향후 연구과제

본 논문에서 현재의 네트워크 포렌식 기술과 방법들을 분류하였고, 현재의 네트워크 포렌식 기술들의 한계를 살펴보았으며, 새로운 네트워크 포렌식 기술의 필요성을 제기하였다. 본 논문에서는 패킷을 이용하여 범행호스트의 위치 및 주소필드의 무결성을 보장하는 증거를 생성하고 검증하여, 누가 언제 범행했는지 입증하는 Network Forensic Evidence Generation and Verification Scheme을 제안하였다. 이 기술은 라우터를 이용하여 범행호스트에 저장된 기록이 조작되어도 네트워크 단에서 범행을 입증할 수 있다는 장점을 가지고 있다. 또한 이 기술은 소프트웨어로 구현되어 실용화 할 수 있음을 입증하였으며, NFEGVS Generator와 Verifier를 별도의 장치로 구성이 가능하여 라우터와 범행대상시스템에 적재되지 않아도 이 기술을 이용할 수 있는 장점이 있다.

증거의 증명력을 높이기 위해 사용한 암호알고리즘별 패킷전송률을 실험을 통해 확인하였으며, 암호알고리즘을 적용하지 않은 경우와 비교 실험하여 HMAC-SHA1의 적용가능성을 확인하였다. 또한 Flow-Based Selection을 추가 제안하여 flow별로 패킷을 선별하여 증거를 생성하여 패킷전송 성능저하를 최소화하였다. 본 논문에서 제안한 기술을 어플리케이션 게이트웨이가 아닌 ASIC(Application-specific Integrated Circuit)으로 개발된다면 보다 우수한 성능이 보장하게 될 것이다.

마지막으로 빠른 연산처리가 가능한 대칭키 알고리즘을 사용하기 위해 Timestamp SecretKey Distribution Scheme을 추가로 제안하여 비밀키 유출문제를 극복하였다.

본 논문은 향후 추가연구를 통해 인터넷공유기와 같이 신뢰할 수 없는 게이트웨이 내부에 존재하는 범행호스트를 탐지하고 범행을 입증하는 기술로 발전시킬 계획이며, IPv6 지원방법에 대해서도 연구를 진행할 예정이다.

참고문헌

〔1〕 E. Casey, "Network traffic as a source of evidence: tool strengths, weaknesses, and future needs," Digital Investigation, vol. 1, no. 1, pp. 28-43, Feb. 2004.

〔2〕 N. Meghanathan, S.R. Allam, and L.A. Moore, "Tools and techniques for Network Forensics," International Journal of Network Security and its Applications, vol. 1, no. 1, pp. 14-25, 2009.

〔3〕 Web Historian, http://www.mandiant.com/products/free_software/web_historian

〔4〕 Index.dat Analyzer, http://majorgeeks.com/Index.dat_Analyzer_d5259.html

〔5〕 eMailTrackerPro. http://www.emailtrackerpro.com

〔6〕 TCPDUMP, http://www.tcpdump.org

〔7〕 Wireshark, http://www.wireshark.org

〔8〕 Z. Gao, and N. Ansari, "Tracing cyber attacks from the practical perspective," Communications Magazine IEEE, vol. 43, no. 5, pp. 123-131, May. 2005.

〔9〕 R. Stone, "CenterTrack: An IP Overlay Network for Tracking DoS Floods," Ninth USENIX Security Symp (Security '00), pp. 199-212, 2000.

〔10〕 S.M. Bellovin, "ICMP traceback Messages," Internet Draft: draft-bellovin-itrace-00.txt, Mar. 2000.

〔11〕 A. C. Snoeren et al, "Hash-based IP traceback," in Proc. ACM SIGCOMM, vol. 31, no. 4, pp. 3 - 14, Oct. 2001

〔12〕 S. Savage et al., "Network Support for IP traceback," ACM/IEEE Trans. Networking, vol. 9, no. 3, pp. 226-237, Jun. 2001.

〔13〕 D.X. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP traceback," Proc. IEEE INFOCOM '01, pp. 878-886, 2001.

〔14〕 A. Belenky and N. Ansari, "IP traceback with Deterministic Packet Marking," IEEE Comm. Letters, vol. 7, no. 4, pp. 162-164, Apr. 2003.

〔15〕 Y. Xiang, W. Zhou and M. Guo. "Flexible deterministic packet marking: an IP traceback system to find the real source of attacks," IEEE Transactions on Parallel and Distributed Systems, vol.

20, no. 4, pp. 567-580, Apr. 2009.
〔16〕 H. Aljifri, "IP traceback : A New Denial-of-Service Deterrent," IEEE Security and Privacy, vol. 1, no. 3, pp. 24-31, May. 2003.
〔17〕 George Tsirtsis and Pyda Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)," RFC 2766, Feb 2000.
〔18〕 L. T. Heberlein and M. Bishop, "Attack class: Address spoofing," in Natl. Information Systems Security Conf, pp. 371-378, Oct. 1996
〔19〕 B. A. Forouzan. TCP/IP Protocol Suite, 4th Ed., McGraw Hill, 2009
〔20〕 H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," RFC 2104, Feb 1997.
〔21〕 J. Postel, "Internet protocol," RFC 791, Sep 1981.
〔22〕 netflow, http://www.cisco.com
〔23〕 libipq, http://www.netfilter.org/projects/index.h tml
〔24〕 OpenSSL, http://www.openssl.org
〔25〕 libpcap, http://www.tcpdump.org
〔26〕 mausezahn, http://www.perihel.at/sec/mz
〔27〕 C. Fraleigh, et al, "Packet-level traffic measurements from the sprint IP backbone," IEEE Network, vol. 17, no. 6, pp. 6-16, Nov. 2003.

〈著者紹介〉

김 형 석 (Hyung-Seok Kim) 학생회원
2001년 2월: 인하대학교 컴퓨터공학과 졸업
2012년 7월: 고려대학교 정보보호학과 석사
2012년 8월~현재: 고려대학교 정보보호학과 박사과정
〈관심분야〉 네트워크 포렌식, 네트워크 보안, 시큐어코딩

김 은 진 (Eunjin Kim) 정회원
1999년 2월: KAIST 산업경영학과 졸업
2001년 2월: KAIST 경영공학과 석사 졸업
2007년 8월: KAIST 경영공학과 박사 졸업
2008년 9월~현재: 경기대학교 국제산업정보학과 조교수
〈관심분야〉 경영정보시스템, 보안경제학

김 휘 강 (Huy Kang Kim) 종신회원
1998년 2월: KAIST 산업경영학과 학사
2000년 2월: KAIST 산업공학과 석사
2009년 2월: KAIST 산업및시스템공학과 박사
2004년 5월~2010년 2월: 엔씨소프트 정보보안실장, Technical Director
2010년 3월~현재: 고려대학교 정보보대학원 조교수
〈관심분야〉 온라인게임 보안, 네트워크 보안, 네트워크 포렌식