情 報 保 護 學 會 誌
第18卷 第5號, 2008. 10

인터넷 뱅킹 보안을 위한 웹 공격의 탐지 및 분류

박 재 철*

요 약

인터넷 뱅킹은 인터넷을 통해 금융 업무를 처리하는 시스템으로, 시·공간적 제약이 없어 이용자가 크게 증가하고 있지만 인터넷을 기반으로 한 웹 공격으로 인하여 많은 위협을 받고 있다. 인터넷 뱅킹은 서비스를 제공하는 은행에 따라 사용자 인터페이스와 처리 방법이 매우 다양하므로, 인터넷 뱅킹 시스템을 목표로 한 웹 공격을 탐지하기 위해서는 해당 인터넷 뱅킹 서비스의 특징을 반영할 수 있는 고유의 패턴을 생성해야 한다. 본 논문에서는 서열 정렬 알고리즘을 이용하여 인터넷 뱅킹 이용에 대한 정상 및 비정상 패턴을 자동으로 생성하여 웹 공격을 탐지하고 분석하는 방법을 제안한다. 제시한 방법의 성능 평가를 위하여, 모의 인터넷 뱅킹 프로그램을 설치한 후 정상적인 이용과 웹 공격을 시도한 자료를 구분하여 수집하고 유사도를 측정하였다. 실험결과 제안된 기법이 오탐율이 낮고 탐지 성능 또한 뛰어남을 확인하였다. 그리고 전문가의 도움 없이 정상 패턴과 비정상 패턴을 생성할 수 있어 효율적으로 변형된 공격이나 새로운 공격을 차단하고 비정상 행위에 판단에 대한 근거를 제시할 수 있음을 보였다.

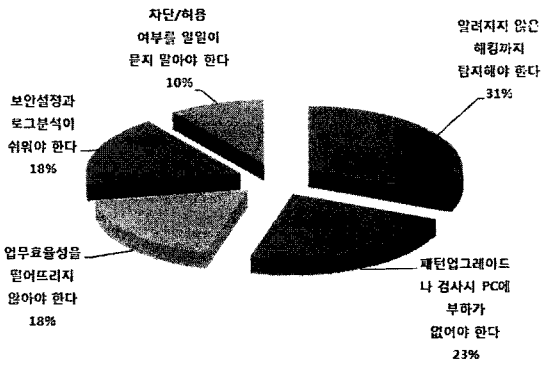
Ⅰ. 서 론

인터넷 뱅킹은 웹 인터페이스의 편리함으로 일상생활에서 가장 친숙한 금융 거래 매체가 되었으며, 그 이용이 나날이 증가하고 있다. 우리나라의 경우 인터넷 뱅킹과 CD/ATM, 텔레뱅킹 등 전자금융이 전체 은행 거래 중 80%에 이르고 지속적인 성장률을 보이고 있다[1]. 한국은행의 '국내 인터넷 뱅킹 서비스 이용 현황'에 따르면 2008년도 1/4분기 인터넷 뱅킹 이용 건수는 하루 평균 약 2,118만 건으로 전분기 대비 3.6%나 증가한 것으로 나타났다[3]. 한편 금융감독원은 인터넷 뱅킹 보안 사고가 전년도에 비해 8배나 증가했다고 발표하여 편리한 인터넷 뱅킹이 보안 측면에서는 많은 불안 요소를 안고 있다는 사실을 다시 한 번 확인케 했다[2]. 이렇게 인터넷 뱅킹 이용자가 증가함에 따라 금융 사기와 공격 기법도 지능적으로 진화하고 있고, 피해자와 피해 액수가 늘어나면서 인터넷 뱅킹 보안은 심각한 사회 문제가 되고 있지만 인터넷 뱅킹 보안에 대한 연구는 미흡한 실정이다.

인터넷 뱅킹 보안을 위한 방법으로 금융 기관에서는 방화벽(firewall), 침입탐지시스템(IDS), 침입차단시스템(IPS) 등과 같은 보안 장비를 이용하여 보안 관리를 수행하고 있다. 하지만 인터넷 뱅킹을 운용하기 위해서는 HTTP 포트 80과 HTTPS(SSL) 포트 443 트래픽 전송을 허용해야 하기 때문에 IP와 포트를 이용하는 기존의 보안 장치들은 응용계층에서 이루어지는 웹 공격에 대해 제한된 보호만을 제공할 수 있다[9]. 즉, 웹 서비스(80포트)의 개방성으로 인하여 방화벽과 침입탐지시스템은 그 기능을 발휘할 수 없으며 금융 기관에 따라 각기 다른 인터넷 뱅킹 서비스를 제공하므로 이에 따른 패턴 생성과 탐지가 더욱 어렵다[6,14].

이러한 상황을 입증하는 자료로 2007년 기무사와 KISA가 공동주관으로 실시한 설문조사(국방정보보호 컨퍼런스 참석자 200명 대상)에 따르면, 크래커(cracker)가 주 공격 대상으로 삼는 곳은 웹 서버와 DB 서버라는 응답이 70%로 가장 많았고 기존 보안 장치의 문제점으로는 변종이나 신종 공격에 무방비라는 응답이 52%, 패턴 업그레이드를 제 때 하지 않으면 공격 차단을 못한다는 답변이 25%로 그 뒤를 이었다. [그림 1]은 설문조사에 나타난 새로운 보안 장치에 대한 요구 사항을 보여주고 있다. 그림에서 알 수 있듯이 알려지지 않은 공격을 시스템 성능에 영향을 끼치지 않고 효율적으

* 전남대학교 정보보호학과 (jchori@gmail.com)



〔그림 1〕 새로운 보안 장치에 대한 요구 사항

로 탐지해야 한다는 답변이 대부분을 차지하고 있다.

웹 공격을 탐지하기 위한 방법은 크게 오용 탐지와 비정상행위 탐지로 구분된다. 침입 고유 패턴(intrusion signature) 방식의 오용 탐지[16,20]는 알려진 공격의 특성을 분석하므로 정확성이 높고 분석이 용이하다는 장점이 있다. 하지만 알려져 있는 공격만을 탐지하므로 변조되거나 새로운 공격에 대한 대처 능력이 떨어지는 한계가 있다. 반면 비정상행위 탐지는 정상적인 행위에 대한 프로파일을 생성하여 공격을 탐지하므로 새로운 공격에 대한 탐지에 좋은 방법이지만 인터넷 뱅킹의 다양성으로 인해 프로파일 생성이 어렵고 많은 긍정오류가 발생하게 된다. 결과적으로 웹을 기반으로 하는 인터넷 뱅킹 보안은 네트워크 계층이 아닌 응용 계층에서의 침입탐지시스템이 필요하며 해당 인터넷 뱅킹의 특성을 반영한 정상 패턴과 공격 패턴을 자동으로 생성하여 신종 공격에 빠르게 대응할 수 있어야 한다.

인터넷 뱅킹의 위협 요소에 대한 보안은 피싱, 파밍, 금융사기를 막기 위한 클라이언트 보안과 악성코드를 이용한 서버 공격을 막는 서버 보안으로 구분할 수 있다. 현재의 인터넷 뱅킹은 서버에 중요한 기능이 집중되어 있고 중요한 개인 정보가 저장되어 있기 때문에 본 연구에서는 서버 보안에 초점을 맞추었다. 본 논문은 다음과 같이 구성되어있다. 2장에서는 인터넷 뱅킹의 구조와 구성요소에 대해 분석하고 웹 공격의 특성과 탐지 방법에 대해 논한다. 3장에서는 응용 계층에서 웹 공격 탐지를 위한 방법론으로 정상 패턴과 비정상 패턴 생성에 이용되는 서열 정렬 알고리즘에 대해 언급한다. 4장에서는 본 논문에서 제안한 정상과 비정상 패턴의 성능을 실험하고 그 결과를 분석한다. 마지막으로 5장에서는 연구 결과를 정리하고 결론을 맺는다.

Ⅱ. 관련 연구

웹 공격과 이에 대한 탐지 방법을 설명하기에 앞서 인터넷 뱅킹의 구조와 각각의 기능이 어떻게 작동하는지 알아 볼 필요가 있다. 웹 공격은 인터넷 뱅킹의 각 요소가 갖고 있는 고유의 취약점을 이용할 뿐만 아니라 요소 간의 연결과정에서 새롭고 복합적으로 나타나는 취약점 또한 이용 가능하기 때문에 이에 대한 이해가 중요하기 때문이다.

2.1 인터넷 뱅킹 구조와 구성 요소

인터넷 뱅킹은 웹 브라우저, 웹 서버, 응용 프로그램, 데이터베이스의 요소들로 구성된다. 따라서 인터넷 뱅킹 보안을 위해서는 각 요소의 취약성뿐만 아니라 그들의 상호 작용으로 인한 취약성을 예측하고 분석해야한다. 인터넷 뱅킹을 구성하는 요소들은 고유의 취약점을 가지고 있으므로 모든 계층에서 위·변조가 가능하며 이를 완벽하게 제거하는 것은 불가능하다. 더욱이 각 요소의 취약점에 대해 보안 관리를 한다고 하여도 이들의 연계 과정 중 발생하는 취약점이 모든 영역에 영향을 미칠 수 있으므로 보안 요구사항 또한 광범위하다. 시스템 관리자 입장에서 보았을 때 HTTP 프로토콜은 다수의 요청과 응답 패킷이 오가게 되므로 많은 양의 운용 기록을 생성하게 되고 이에 따라 자동화된 자료 분석 방법과 공격 탐지 시스템이 요구된다. 하지만 네트워크 기반의 다른 공격과 달리 웹 기반 공격은 시스템 운용 기록을 이용하는 호스트기반 침입탐지[7]나 IP와 포트를 이용한 네트워크 기반 침입탐지[5,10]만으로는 완벽한 보안이 어렵다고 전문가들은 분석한다. 이러한 현실에서 인터넷 뱅킹 고유의 취약점을 의도적으로 악용하는 외부의 공격이나 내부의 비정상적인 서비스 요청에 대해 기존의 방화벽과 침입탐지시스템의 기능에 추가적으로 웹에 특화된 보안관리 시스템이 요구되고 있다.

2.2 인터넷 뱅킹 취약점 분석

다양한 계층으로 이루어진 인터넷 뱅킹이 응용 프로그램의 보안성을 높이는 것만으로 안전하지 않다는 것은 분명하다. 악의적인 목적을 가진 공격자는 앞서 언급한 인터넷 뱅킹의 모든 구성 요소의 취약점을 고려해 공격을 시도할 것이고, 하나의 웹 공격이 여러 공격 유

형과 중복되어 각 구성요소에 영향을 미치기 때문에 어떤 계층에 취약점이 있는지 파악조차 어렵게 된다. OWASP는 기업 및 공공 기관이 이러한 문제에 대처할 수 있도록 가장 심각하면서 즉각적인 개선 조치가 필요한 웹 애플리케이션 10대 취약점 목록[14]을 발표하였다. 이 목록은 웹 애플리케이션의 취약점 현황을 반영하기 위해 지속적인 갱신이 이루어지고 있으며 해당 취약점 보안을 위한 유용한 정보 및 대처 방안들을 포함하고 있다[15]. [표 1]은 모의 인터넷 뱅킹 사이트인 Hacme Bank[12]에 웹 공격을 시도하고 취약점을 분석한 결과이다. OWASP에서 제시한 상위 4가지 취약점에 해당하는 공격이 인터넷 뱅킹에 미치는 영향과 결과를 확인할 수 있다.

- XSS : Cross Site Scripting(CSS) 공격이라고도 불리며 콘텐츠 암호화나 검증 절차 없이 사용자가 입력한 자료를 응용 프로그램에서 받아들이거나, 웹 브라우저로 보낼 때 발생한다. 공격자는 희생자의 브라우저 내에서 스크립트를 실행하여 사용자 세션 가로채기, 웹 사이트 손상, 웜 전파가 가능하다.
- 인젝션 취약점 : 사용자가 입력한 자료가 시스템 명령어나 DB 질의문의 일부분으로 응용 프로그램에 보내질 때 발생하며 악의적인 명령어를 실행시키거나 자료의 변경이 가능하며 운영 체제 명령어를 삽입하여 외부의 다른 시스템에 우회 접근하는 것도 가능하다.
- 악성파일 실행 : 원격 파일 삽입(RFI)에 취약한 코드는 공격자가 악의적인 코드와 자료의 삽입을 허용함으로써 발생하며 PHP, XML, 그리고 사용자로부터 파일명이나 파일을 받아들이는 프레임워크에 영향을 준다.
- 직접 객체 참조 : 개발자가 파일, 디렉터리, 데이터베이스 기록 혹은 키 같은 내부 구현 객체에 대한 참조를 URI 혹은 폼 매개변수로 노출시킬 때 발생하며 공격자는 이러한 참조를 조작해서 승인 없이 다른 객체에 접속한다.

2.3 웹 공격 탐지 방법

이번 절에서는 웹 공격을 탐지하는 방법에 대해 살펴보고, 이러한 방법이 인터넷 뱅킹 보안에 충분하지 않은 이유를 논한다.

2.3.1 오용 탐지

오용 탐지[11]는 시스템과 응용 프로그램의 알려진 취약점을 패턴으로 정의하고 이와 일치하는 경우를 침입으로 간주한다. 하지만, 감사 기록(auditing log)에 대한 의존도가 높고 공격 패턴에 근거한 탐지만이 가능하므로 변형되거나 알려지지 않은 새로운 공격은 탐지하지 못한다. 인터넷 뱅킹을 목표로 한 웹 공격은 해당 금융기관에서 개발한 응용 프로그램 고유의 구조적 특성을 이용하기 때문에 오용 탐지의 패턴처럼 정형화되고 일반적인 공격 패턴 생성이 어렵다. 기존의 오용 탐지 기반 침입탐지 방법 중 웹 공격 탐지에 초점을 둔 연구는 많지 않으며, 대부분 네트워크 공격 탐지 기법에 웹 공격과 관련된 패턴을 추가하여 활용하였다.

대표적인 오용 탐지 기반 침입탐지시스템인 Snort[17]

〔표 1〕 Hacme Bank의 취약점과 공격 결과

취약점	공격 방법	입력 필드	공격 결과
인젝션 취약점	로그인 인증 우회	사용자 계정, 암호	임의의 사용자 계정으로 로그인
	테이블 정보 변경	사용자 계정	DB 정보 획득, 임의의 사용자 등록
	시스템 명령 삽입	사용자 계정	외부에서 저장 프로시저 이용
악성 파일 실행	로그인 실패횟수 변경	사용자 계정, 암호	계정 및 암호 추측(Bruteforce) 공격
	원격 파일 실행	URI	사용자 인증 우회, DB 정보 획득
직접 객체 참조	사용자 권한 상승	계좌 정보 페이지	사용자 등급 변경
	입력 값 검증 우회	계좌 이체 페이지	계좌 이체 금액 변조
	인가되지 않은 접근	대출 신청 페이지 (이자율)	대출 이자율 변경
	인가되지 않은 접근	대출 신청 페이지 (계좌)	대출 계좌, 상환 계좌 정보 변조
XSS	사용자 세션 가로채기	게시판 본문	관리자와 다른 사용자 정보 취득

는 네트워크 영역 전체를 탐지 대상으로 하고 HTTP 패킷의 내용(content)으로부터 알려진 공격 패턴을 찾아 공격 여부를 판단한다. Snort는 웹 공격에 관련된 시그니처만 1000개 이상 갖고 있고 프로그램 원시 코드(source code)와 공격 시그니처가 공개되어 있기 때문에 침입탐지시스템을 연구하는 목적으로 다양하게 사용된다. 하지만, 시그니처에 기술된 패턴에 인코딩과 공백문자 삽입 등 약간의 변형을 가할 경우 탐지가 어렵고 공격자가 잘못된 경고를 나타내게끔 사전에 알려진 시그니처로 패킷을 생성하여 전송할 수도 있다[21]. 이 경우 침입탐지시스템이 잘못된 오류메시지를 과다하게 생성하여 무력화될 수 있으며, 그 과정에서 실제의 공격 행위들이 감춰질 수 있다. 즉, Snort는 취약점 공개 사이트 등을 통해 이미 잘 알려져 있는 공격들만 탐지할 수 있기 때문에 웹 콘텐츠에 따라 패턴이 다른 공격들이나, 패턴을 정형화하여 시그니처로 만들 수 없는 공격들은 이러한 방법으로는 탐지할 수 없다. 또한 시그니처에 특정 문자열을 찾아 공격을 판정하는 이러한 방법은 매개변수 값이 암호화되거나 동적으로 생성되는 인자 값을 예측하기 어렵고 더욱이 인터넷 뱅킹은 금융 업무의 특성상 복잡한 산술 연산자가 많이 나타나기 때문에 특수문자의 이용을 제한할 경우 정상적인 서비스 제공에 많은 제약을 가하게 된다.

2.3.2 비정상행위 탐지

비정상행위 탐지는 정상적인 사용자 모델을 학습시키고 이 모델에 벗어나는 경우를 침입으로 간주하는 방법으로, 변형되거나 알려지지 않은 공격을 탐지할 수 있지만 구현 비용이 크고 긍정 오류(false positive)가 많이 발생되는 단점이 있다. Kruegel[5]은 웹 브라우저를 통해 서버로 전달된 GET 요청 인자의 ASCII값 빈도를 이용해 마코프 모델(Marcov model)을 생성하고 웹 공격을 탐지하였다. 제안된 모델은 크게 PPU(Packet Processing Unit)와 SPU(Statistical Processing Unit)로 구성된다. PPU는 시스템으로 전송된 네트워크 패킷 헤더의 스트링 값을 이용하여 프로토콜과 서비스를 분류하고 SPU는 일정 시간 동안 나타난 요청 타입, 요청 길이 그리고 패킷 자료 분포를 분석한다. 유사도 측정을 위하여 패킷 자료에서 요청 문자의 분포를 내림차순으로 정렬한 후, 가중치를 적용하여 산출한 점수를 통하여 비정상행위를 판단하였다. 하지만 이 방법은 GET 방식으로 전달되는 자료만 이용하기 때문에 특정 CGI 취약점을 이용한 공격이나 로그인 인증 우회, 허용되지 않는 페이지로의 접근, 인자 값의 악의적인 조작과 같은 공격 탐지가 어렵다는 단점이 있다. 즉, 패킷의 바디 영역을 이용하는 POST 방식은 제외되었기 때문에 네트워크 공격과 일부 웹 공격은 탐지할 수 있지만 인터넷 뱅킹을 목표로 하는 응용 계층의 웹 공격 탐지에 한계가 있다.

Ⅲ. 제안방법

웹 공격을 탐지하기 위한 오용 탐지와 비정상행위 탐지 방법은 앞서 언급하였듯이 각각 장점과 단점을 갖고 있으며, 본 논문에서는 2가지 방법을 혼합하여 사용한다. 구체적으로 기술하자면 변형되거나 새로운 공격을 탐지 할 수 있도록 정상적인 이용에 대한 트랜잭션을 학습하여 정상 패턴을 생성하고, 웹 공격의 특성 파악과 분석을 위해 공격 코드를 추출하여 비정상 패턴을 생성하는 방법을 함께 사용한다.

3.1 매개변수 자료 수집

웹 공격의 유사도 측정을 위해 본 논문에서는 인터넷 뱅킹의 매개변수를 실험 자료로 사용하였다. 수집된 자료는 키워드 치환 행렬을 사용하여 알파벳 문자로 이루어진 서열로 가공하고 축약한다. 키워드 치환 행렬은 웹 공격에 주로 이용되는 특수문자, 시스템 명령어, 중요한 함수명, 디렉터리, 매개변수명과 같은 키워드를 등록시키고 두 문자로 이루어진 알파벳 문자열로 대응시켜 불필요한 자료를 제거하는 기능을 한다. 따라서 실제 정상적인 사용자의 요청에서 호출되지 않는 매개변수는 수집 자료에 포함되지 않게 되므로 키워드 치환 행렬을 통해 생성되는 알파벳 문자열은 인터넷 뱅킹의 구조와 특징을 가장 잘 나타내는 키워드 서열이라고 할 수 있다.

악의적인 사용자가 인터넷 뱅킹에 비정상적인 요청을 할 때에는 변수 이름을 새로 정의하기 보다는 주로 정해진 매개변수 인자에 질의문과 특수문자를 사용하여 공격하게 된다. 정상적인 입력 값과 공격 코드가 삽입된 입력 값을 비교해 보면 악의적인 요청의 경우 사용자 입력 값에 질의어와 논리연산에 사용되는 특수문자가 나타나거나 주석문과 같은 불필요한 코드가 삽입되어 있다. 또한 전체적인 구조측면에서 볼 때 매개변수의 순서가 재배치되어 정렬 구조가 다르고 보안 장치를 회피

하거나 우회하기 위해 입력 값을 인코딩하여 전송하게 된다. 공격자는 인터넷 뱅킹을 공격하기 위해 대부분의 인자 값들을 조작할 수 있으며 매개변수와 입력 값의 삽입, 삭제, 변경 등 여러 가지 방법을 통한 공격이 가능하다. 또 두 종류 이상의 SQL 구문들을 조합하여 응용 프로그램이 하나의 완전한 SQL 구문으로 인식하도록 값을 도출하는 공격 또한 가능하기 때문에 웹 서버나 데이터베이스의 접근 기록을 이용하여 공격 여부를 확인하는 것은 매우 어렵다[20]. 결국 웹 공격을 효과적으로 탐지 위해서는 웹 서버가 전달한 사용자 요청을 응용 프로그램이 해석하지 않은 상태에서 확인하는 작업이 필요하다. 따라서 본 논문에서 패턴 생성을 위해 적용한 서열 정렬 방법을 언급하기 전에 유사도 측정을 위해 척도로 사용하고 있는 URI의 질의 문자열 그리고 인자 값의 전달 방법에 대해 간략히 예를 들어 기술한다.

```
프로토콜://서버:포트/경로/실행파일?매개변수=인자 값
http://www.hacmebank.com:8080/Account/Transfer.aspx?
    amount=attack
```

모든 URI는 좌측에서 우측으로 인식되며 프로토콜의 구성요소는 서버를 연결하는 방법과 관련 개체를 순서대로 인식하는 방법을 정의한다. 위의 URI는 'www.hacmebank.com'의 주소로 8080 포트를 이용하는 Account 디렉터리에 위치한 'Transfer.aspx' 파일이 실질적으로 실행 가능한 서버 측 스크립트 프로그램으로, 매개변수 'amount'가 호출되어 'attack'을 인자 값으로 처리하는 것을 나타낸다. 일련의 많은 인터넷 뱅킹은 이러한 질의 문자열을 통해서 다양하게 작동을 하기 때문에 질의 문자열과 매개변수를 분석하는 것은 매우 중요한 일이며 이들을 통하여 특정 인터넷 뱅킹에 필요한 요소들과 역할을 파악할 수 있다. 매개변수는 '+' 기호 또는 '?' 기호에 의하여 URI로 설정될 수 있으며 그 형식은 "변수명=값"의 형식이다. 다수의 매개변수들을 구분하기 위해 "&"기호가 사용되며 이를 질의 문자열(query string)이라고 한다. 질의 문자열은 HTML 폼(FORM)을 통해 사용자와 인터넷 뱅킹이 상호작용을 하게 되며 사용자 입력 값은 GET과 POST라는 두 가지 방법으로 전달된다. GET 인자는 HTTP 헤더로 전달되지만, POST 인자는 HTTP 바디의 일부분으로 전달되는 기술적인 차이점이 있다. 덧붙여 GET 방식은 문자열을 저장할 수 있는 질의 문자열의 길이 제한으로 인해 많은 양의 입력 자료를 서버에 전송할 수 없으므로 표준 입력을 사용하는 POST 방법이 널리 사용되고 있다. 정리하자면 POST 방식은 사용자의 요구 사항들을 표준 입력을 통해서 받는다는 점을 제외하고 처리 과정은 GET 방식과 동일하다[4]. 일반적으로 대부분의 인터넷 뱅킹에서는 URI에 자료가 노출되는 GET 방법보다 POST 방법이 안전하다고 생각하고 사용되지만 POST 요청 방식 역시 암호(PASSWORD) 필드의 값과 숨겨진 필드의 내용들 모두 평범한 텍스트 형태로 보내지며, 차이는 단지 URI의 한 부분으로 전송되지 않는다는 것이다[18]. 따라서 전송 자료를 조작하거나 HTTP 헤더가 아닌 URI에 직접 입력 값을 삽입하여 GET 방식처럼 자료를 전송하여도 매개변수와 사용자 입력 값은 전달된다. 결과적으로 응용계층에서 GET/POST 방법으로 전달되는 매개변수와 URI의 분석은 악의적인 사용자가 공격을 시도하기 위한 과정일 뿐만 아니라 취약성 분석 및 웹 해킹 탐지를 위한 핵심이다.

3.2 정상 및 비정상 패턴 생성

이번 절에서는 인터넷 뱅킹의 정상 패턴과 비정상 패턴을 생성하고 유사도를 측정하기 위해 사용하는 서열 정렬 방법에 대해 기술한다. 서열 정렬은 서열의 유사도를 검사하기 위하여 사용하는 방법을 말하며, 이 때 정렬이란 유사한 부분과 다른 부분을 구분지어 표시한 두 서열 간의 상호배열을 의미한다. 서열 정렬의 대표적인 방법으로는 전체 서열을 비교하는 전역 정렬(global alignment)과 부분열의 일치에 초점을 두는 지역 정렬(local alignment)이 있다[8]. 웹 공격 탐지를 위해 전역 정렬 방법을 사용할 경우 서열의 전체 길이에 맞추어 정렬하기 때문에, 정상적인 요청의 구조가 변조되었을 때 쉽게 파악할 수 있다. 반면 공격 코드의 분석을 위해 지역 정렬 방법을 사용할 경우 연속적인 부분열의 일치를 찾아 정렬하므로 특정 공격에 자주 사용하는 특수문자의 기능을 유추할 수 있다.

3.2.1 전역 정렬을 이용한 정상 패턴 생성

인터넷 뱅킹의 정상적인 사용자 요청을 알파벳 문자로 치환하여 서열의 구조가 만들어지면 웹 공격 탐지를 위해 전역 정렬 방법을 적용하여 유사도를 측정할 수 있다. 전역 정렬은 서열 전체를 통하여 가장 최적화된

```
NLGPSTKDFGKISESREFDNQ
 |       ||||      |
QLNQLERSFGKINMRLEDALV
```

〔그림 2〕 전역 정렬의 예

전장정렬을 탐색하는 방법이다. 정상 서열과 비정상 서열과의 정렬에서 어느 정도 유사성이 있는지 알아내기 위해 전역 정렬하는 예를 [그림 2]에서 보이고 있다.

전역 정렬은 니들만-분쉬(Needleman-Wunsch) 알고리즘이 대표적이며 서열의 구조 탐색이 가능하다. 두 개의 서열을 행렬로 구성하고 행렬의 왼쪽 위에서 오른쪽 아래로 부분 서열의 고득점 정렬들을 탐색하여 최적 경로를 구하는 것이다. 경로 탐색의 경우 행렬을 따라 최고 정렬 값을 기록하는 경로를 추적하고 이들 경로에 따른 측정 점수를 합산하여 연결하는 알고리즘을 사용하여 최적의 전역 정렬이 구해지며 그 식은 아래와 같다[13].

$$S_{ij} = \max\{ S_{i-1,j-1} + s(a_i b_j), \max_{x \geq 1}(S_{i-x,j} - g_x), \max_{y \geq 1}(S_{i,j-y} - g_y) \} \quad (1)$$

구체적인 예로 일반적인 사용자가 동일 인터넷 뱅킹에 서비스 요청을 할 경우, 동적으로 변수명과 인자 값이 생성되더라도 해당 매개변수의 구조는 크게 변하지 않는다. 하지만 공격자가 인터넷 뱅킹의 사용자 요청을 조작하면 매개변수의 순서, 빈도 그리고 공격 코드 삽입으로 인해 전체 길이의 변화가 생기게 된다. 따라서 정상적인 매개변수 구조를 벗어나는 키워드 서열은 공백 삽입에 의해 전역 정렬에 일치되지 않는 결과로 나타난다.

일치 점수 p와 불일치 점수 q로 정의된 키워드 일치 행렬 IdMat Si,j(p,q)을 이용하여 두 서열 Sa, Sb 의 전역 정렬 유사도를 나타내는 Global Similarity GloSIM은 아래의 식 (2)와 같이 정의되며, 두 서열의 유사도를 판단하기 위해 사용하는 키워드 일치 행렬(Identity Matrix) IdMat은 두 서열의 코드가 일치하면 1을, 불일치하면 0을 부여하여 정렬 점수를 가산한다.

$$GloSIM(S_a, S_b) = Global\ alignment\ S_a\ and\ S_b\ with\ IdMat\ S_{i,j}(p,q) \quad (2)$$

웹 공격 탐지를 위한 전역 정렬 수행 시 일치율은 일치되는 키워드 수를 공백을 포함한 전체 길이로 나눈다. 정상 패턴 N과 서열 S의 공격여부를 판단하기 위한 전역 정렬 일치도 GI 는 아래의 식 (3)과 같이 백분율로 계산된다.

$$GI = \frac{alignment\ match\ code(NCode\ Letter, SCode\ Letter)}{NCode\ Letter + gap} \times 100 \quad (3)$$

3.2.2 지역 정렬을 이용한 비정상 패턴 생성

지역 정렬은 두 개의 서열에서 어떤 부분이 높은 상동성을 갖는지 알아내기 위해 짧은 단위의 부분 서열들을 탐색한다. 동일 기능이나 유사한 기능에 대한 효율적인 상세 정렬방법으로 높은 지역적 상동성을 찾기 위하여 가장 많이 이용하는 방법이다. [그림 3]은 지역 정렬의 예이다.

```
NLGPSTKDDFGKILGPSTKDDQ
        ||||
QNQLERSSNFGKINQLERSSNN
```

〔그림 3〕 지역 정렬의 예

공격이 시도된 사용자 요청 자료를 살펴보면 동종의 공격에 사용되는 키워드가 비슷하고 변조된 매개변수 또한 동일한 경우가 많다. 이는 공격자가 매개변수를 변조할 때 사용하는 특수문자의 조합과 순서가 일정한 패턴을 갖고 있기 때문이며 공통적으로 나타나는 공격 키워드를 정렬하여 보면 공격 코드의 패턴을 찾을 수 있다. 따라서 비정상적인 사용자 요청에 대한 특성을 파악하고 공격을 분류하기 위해서는 부분열의 일치에 초점을 두는 지역 정렬 방법이 효과적이다. 지역 정렬은 스미스-워터만(Smith-Waterman) 알고리즘이 널리 사용되며 니들만-분쉬법과 다른 점은 최대 공통 문자열을 2차원 행렬상의 어느 지점에서나 구할 수 있도록 음(-)이 아닌 양수를 이용하여 출발점을 독립화시킨 점이다. 아래의 식 (4)는 스미스-워터만 알고리즘을 정리한 식이다[19].

$$S_{ij} = \max\{ S_{i-1,j-1} + s(a_i b_j), \max_{x \geq 1}(S_{i-x,j} - g_x), \max_{y \geq 1}(S_{i,j-y} - g_y), 0 \} \quad (4)$$

일치 점수 p와 불일치 점수 q로 정의된 키워드 일치 행렬 IdMat Si,j(p,q)을 이용하여 두 서열 Sa, Sb 의 지역 정렬 유사도를 나타내는 지역 정렬 방법의 점수 측정 방법 Local Similarity LocSIM은 식 (5)와 같이 정의된다.

$$LocSIM(S_a, S_b) = Local\ alignment\ S_a\ and\ S_b\ with\ IdMat\ S_{i,j}(p,q) \quad (5)$$

비정상 패턴 A와 서열 S의 공격 분석을 위한 지역 정렬 일치도 LI 는 아래의 식 (6)과 같이 백분율로 계산된다. 결과적으로 웹 공격 탐지를 위한 일치율은 서열 전체 구조의 변화를 파악하기 위해 일치되는 키워드 수를 정상 패턴의 공백을 포함한 전체 길이로 나누고, 공격 분석을 위한 일치율은 코드의 부분적 기능을 유추하기 위해 일치되는 키워드 수를 공백을 제외한 공격 패턴의 전체 길이로 나누는 방법을 사용한다.

$$LI = \frac{Alignment\ Mmatch\ Ccode\ (A\ Code\ Letter, SCode\ Letter)}{A\ Code\ Letter} \times 100 \quad (6)$$

한편, 공격 키워드의 특성을 파악하기 위해서는 공격 코드가 포함된 서열을 정렬하여 공격에 사용되는 공통적인 키워드를 찾을 필요가 있는데 앞서 설명한 정상서열과 비정상서열의 전역 정렬을 이용하여 일치되지 않는 코드를 추출한다. [표 2]는 전역 정렬을 통한 공격 코드 추출 과정을 보여주고 있다.

매개변수와 사용자 입력 값은 환경변수에 의해 일부는 고정된 것이고 일부는 고정되지 않은 것이다. 인터넷 뱅킹의 구조는 각각 다른 종류의 웹 서버와 데이터베이스를 이용하기 때문에 비정상적인 사용자 요청으로부터 이상 문자를 추출하기 위해 불필요한 부분을 제거하는 것은 공격 패턴 생성을 위해 매우 중요하다. 또한 플랫폼에 독립적이면서 알려진 공격 패턴을 통해 변조되거나 새로운 공격을 학습하기 위해서도 불필요한 코드의 제거는 필수적이다. 따라서 모든 인터넷 뱅킹에 공통적으로 사용되는 공격 키워드만을 추출하여 패턴을 만들고 그 결과를 참조하여 정렬하면 특정 인터넷 뱅킹의 매개변수 구조에 영향을 받지 않고 모든 종류의 인터넷 뱅킹에 적용이 가능하다.

3.2.3 패턴 프로파일 생성

[표 3]은 수집된 매개변수 정보, 변환된 알파벳 문자 서열, 변환된 서열의 전체 길이, 문자열 발생 상대 빈도로 이루어진 정상 패턴과 비정상 패턴 프로파일의 구성 예를 보여준다. 웹 공격 탐지를 할 경우에는 정상 패턴과 전역 정렬하여 유사도를 측정하고, 공격을 분석할 경우에는 비정상 패턴과 지역 정렬하여 유사도를 측정하게 된다. 패턴 서열은 중복을 피하기 위해 두 서열의 일치도가 100%가 아닐 경우만 프로파일에 포함시킨다. 이 때 정렬 대상이 되는 최적 서열 선택은 서열에 나타날 수 있는 문자열의 발생 확률을 이용한다. 즉, 서열의 전체 길이에 따른 문자열 발생의 상대 빈도를 측정하여 문자열의 발생 확률을 구하고 프로파일에서 최적의 유사 서열을 선택할 수 있도록 한다. 수집된 자료의 프로파일 서열을 P, 검사 대상 서열을 S라고 하였을 때 최적 서열 찾기 Optimal Sequence Detection (OSD)는 아래의 식 (7)과 같다.

$$OSD^*(P,S) = \min_{0 \le KeyProDiff \le 1} \{OSD_{KeyProDiff}(P,S)\} \quad (7)$$

서열의 전체 길이인 P CodeLength와 S CodeLength를 측정한 후, 두 문자로 이루어진 한 쌍의 문자열 P CodeProbability와 S CodeProbability의 서열에 따른 문자열의 발생 확률을 측정하고 각 서열에서 나타난 문자들의 발생 확률을 같은 문자열별로 비교하여 그 차이를 합한다. 결론적으로 CodeProbabilityDiff는 0에 가까울수록 최적의 서열 쌍이 된다.

〔표 2〕 공격 코드 추출 과정

1	정상 / 비정상 매개변수 자료	정 상	function=Loan & txtUserName = alice & txtPassword = 1234
		공 격	function=Loan & txtUserName = ' or 1 = 1 - - & txtPassword =
2	코드 서열 전체 정렬	정 상	FN EQ LO AM TT UN EQ -- -- -- -- -- AM TT PW EQ
		공 격	FN EQ LO AM TT UN EQ AP QR EQ HY HY AM TT PW EQ
3	비일치되는 키워드 추출	공격코드	AP QR EQ HY HY
		키 워 드	' or = - -

〔표 3〕 정상/비정상 패턴 프로파일의 구성 정보

	매개변수 정보	변환된 서열	길이	문자열 상대 빈도
정 상	&function=Welcome&__EVENTTARGET=_ctl0:lnkBtnFundsTransfer&__EVENTARGUMENT=&__VIEWSTATE=dDwtOTYzNTkyNzt0PDtsPGk8MT47P....	AM EQ AM EQ VS CQ AM EQ AM EQ PL LS PL PL PL PL PL PL SL PL EQ EQ	23	AM(0.38) CQ(0.03) EQ(0.38)...
	&function=AccountTransfer&__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=dDwtOTYzNTkyNzt0PDtsPGk8MT47PjtsPHQ8O2w8aTw1Pj....	AM EQ CN AM EQ AM EQ AM EQ PL CD PL PL PL CD PL CD PL PL CD PL PL PL CD PL PL SU SU SU PL AS PL AM EQ	48	AM(0.24) BN(0.02) CD(0.24)...
	….	….	….	….
비정상	' having 1=1--	SQ HV EQ HY HY	5	EQ(0.20) HY(0.40) HV(0.20)...
	../../../../bin/ps -	PE PE SL PE PE SL PE PE SL PE PE SL BN SL PS HV	16	BN(0.17) HV(0.06) PE(0.50)...
	….	….	….	….

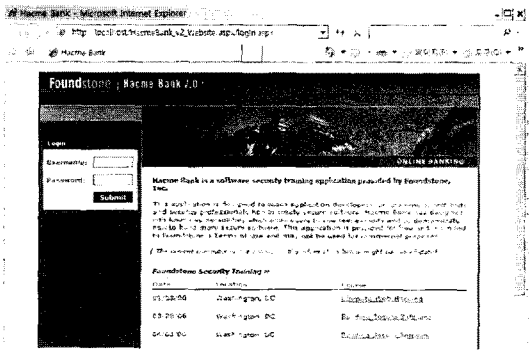
Ⅳ. 실험 및 결과 분석

이번 장에서는 인터넷 뱅킹 보안을 위해 정상 및 비정상 패턴을 생성한 실험 환경과 본 논문에서 제안한 웹 공격 탐지 성능과 취약점 분석 기법의 정확성을 검증하고 그 결과를 분석한다.

4.1 실험 환경

실험을 위해 모의 인터넷 뱅킹의 트랜잭션을 수집하고 사용자 요청 자료를 추출하였다. 실험에 사용된 인터넷 뱅킹 서비스는 사용자 웹 브라우저, 웹 서버, 인터넷 뱅킹, 데이터베이스로 구성되어 동작된다. 사용자 요청을 위한 입력 값은 앞서 언급했던 GET과 POST 방법을 이용하여 다양하게 부호화되어 웹 서버에 전송되는데, 본 논문에서는 POST 자료를 GET 형식(질의 문자열)에 맞추어 수집하고 텍스트 파일에 기록하였다.

[그림 4]는 실험에 사용한 Hacme Bank 2.0의 사용자 인터페이스이다. 구체적인 환경은 MS .NET Framework v1.1, MS IIS 웹 서버, Microsoft SQL Server 2000이다. 운영체제는 윈도우 XP(Window XP sp3)이고 ASP와 C# 언어로 개발된 모의 인터넷 뱅킹(Hacme Bank v2.0)을 설치하였고 사용자 웹 브라우저는 인터넷 익스플로러(Microsoft Internet Explorer 7.0)이다.



〔그림 4〕 모의 인터넷 뱅킹인 Hamme Bank 2.0의 사용자 인터페이스

[표 4]는 웹 공격 탐지와 분석을 위한 실험 자료 집합이다. 실험 자료의 구성은 인터넷 뱅킹에 대한 정상적인 사용과 비정상적인 사용으로 구분하였고 비정상적인 사용은 OWASP에서 중요 취약점으로 선정한 10가지 취약점 중 직접적인 공격에 해당하면서 가장 많이 발생하는 웹 공격 방법 4가지(XSS, 인젝션 취약점, 악성 파일 실행, 직접 객체 접근)로 구분하여 수집하였다. 정상 패턴 생성을 위하여 400개의 정상적인 사용자 요청 자료와 공격 패턴 생성을 위한 120개(공격 별 30개)의 비정상 자료를 수집하였다. 또한 생성된 패턴의 정확성을 검증하기 위해 정상 사용 자료 50개와 비정상 패턴 검증을 위해 패턴 생성에 사용되지 않은 공격 코드 80개(공격 별 20개)를 수집하여 구성하였다.

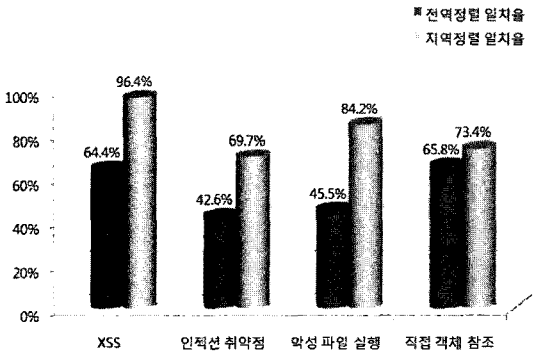
〔표 4〕 실험 자료 집합

	정 상	비 정 상			
		XSS	인젝션 취약점	악성 파일 실행	직접 객체 접근
학습 자료	400	30	30	30	30
검증 자료	50	20	20	20	20

4.2 실험 결과

전체적인 실험 구성은 웹 공격 탐지를 위해 정상 서열과 비정상 서열의 유사도를 측정하는 실험과, 공격 분석을 위해 공격 코드의 유사도를 측정하는 실험으로 구성된다. 유사도를 측정할 때 지역 정렬 방법과 전역 정렬 방법을 달리 적용하고 그 결과를 비교함으로써, 본 논문에서 유사도 측정 방법을 달리한 이유를 확인할 수 있다.

실험을 위해 400개의 정상 자료를 알파벳 문자로 이루어진 서열로 변환하고 중복되는 서열을 제외한 93개의 정상 패턴을 생성한 후 각 공격 서열과 정렬하여 일치율을 구하였다. [그림 5]는 정상 서열과 비정상 서열을 전역 정렬과 지역 정렬의 방법으로 정렬하고 일치율의 평균을 구한 실험 결과이다. 정상과의 일치율이 높을수록 오탐율은 높아지는 반면 공격 탐지율은 낮아지는데, 실험 결과를 살펴보면 4가지 공격 모두 전역 정렬을 이용한 유사도 측정에서 정상과의 일치율이 낮은 걸 볼 수 있다. 지역 정렬은 변조된 매개변수의 순서에 영향을 받기 때문에 공격자가 매개변수의 위치를 서열의 앞이나 끝부분으로 변경하여 요청을 할 경우 유사도가 높게 나타나게 된다는 것을 알 수 있었다.



〔그림 5〕 정상 서열과 공격 서열과의 일치율

[표 5]는 정상 서열과 비정상 서열을 전역 정렬과 지역 정렬로 유사도를 측정하고 95%의 일치율을 임계값으로 탐지율을 계산한 결과이다. 비정상적인 사용자 요청을 탐지하기 위해서는 전역 정렬을 이용하여 유사도를 측정하는 방법이 지역 정렬 보다 높은 탐지율을 보였다. 자료 분석 결과 지역 정렬은 연속적인 부분열을 찾는 특성 상 매개변수의 길이가 길수록 높은 일치율을 보이고 오탐율이 높아졌다. 한편 전역 정렬을 이용한 실험 결과 중에 악성 파일 실행과 직접 객체 참조에서 정상과의 일치율이 100%가 되어 탐지하지 못하는 경우가 발생하였다. 공격을 시도한 비정상 자료 확인 결과, XSS와 인젝션 취약점을 이용해 사용자 인증을 무력화시킨 후 은행 이자율과 고객 등급의 변조를 시도한 공격이었다. 이러한 공격은 숫자나 알파벳으로 이루어진 인자 값만을 변경하는 방법으로 해당 인자 값이 키워드 치환이 되지 않기 때문에, 본 논문에서 제안한 방법으로 탐지하기 어려웠다. 이처럼 매개변수 인자를 변조하는 공격은 기존의 IDS와 방화벽으로도 차단하기 어렵기 때문에 선행되는 로그인 인증 우회와 같은 공격을 차단하고 그 보다 앞서 인터넷 뱅킹 개발 과정에서 입력 값 검증을 철저히 하여 보완할 필요가 있다.

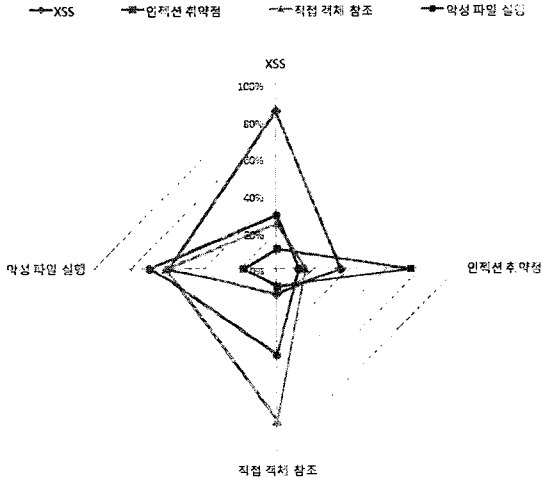
[표 6]은 추출한 공격 패턴의 정확성과 탐지율을 알아보기 위해 지역정렬로 실험한 결과이다. 비정상 집합 자료에서 공격 코드를 추출한 후 생성된 공격 패턴과 취약점 별 테스트 서열을 지역 정렬하고 유사도를 측정하였다. 유사도를 계산하는 방법은 3장의 식 (6)에서 언급한 일치 코드 개수를 생성된 서열 패턴의 전체 길이로 나누어주는 방법을 사용하였다. 같은 종류의 공격을

〔표 5〕 전역 정렬과 지역 정렬의 탐지율 비교

	XSS	인젝션 취약점	악성 파일 실행	직접 객체 접근
전역 정렬	94.8%	99.3%	81.5%	72.4%
지역 정렬	33.6%	23.8%	44.4%	36.7%

〔표 6〕 학습된 비정상 패턴과 테스트 서열의 지역 정렬 결과

		테스트 서열			
	평 균 일치율	XSS	인젝션 취약점	직접 객체 참조	악성 파일 실행
학습된 패턴	XSS	84.9% (87.5%)	35.2%	14.2%	59.3%
	인젝션 취약점	10.7%	73.8% (91.7%)	10.3%	18.2%
	직접 객체 참조	24.7%	15.8%	83.5% (83.3%)	60.9%
	악성 파일 실행	28.8%	12.1%	47.7%	69.1% (72.7%)



〔그림 6〕 비정상 패턴을 통한 공격 분류 결과

수행한 테스트 서열 코드와 공격패턴 서열 코드의 일치율이 다른 공격 코드들 보다 높다는 것은 공격의 분류가 성공적으로 되었다는 것을 나타낸다. 실험 결과 XSS 공격이 84.9%, 직접 객체 참조 공격이 83.5%로 동일한 공격 코드를 많이 사용하여 유사도가 높은 것으로 나타났고, 악성 파일 실행이 69.1%로 가장 낮은 유사도를 보였다. 괄호 안의 값은 공격 분류의 정확성을 나타낸 값으로 인젝션 취약점이 91.7%로 다른 공격과 가장 구분되며 악성 파일 실행이 72.7%로 가장 낮았다.

[그림 6]은 학습된 비정상 패턴과 테스트 서열의 지역 정렬 결과를 도표로 나타낸 것이다. 각 공격 영역의 분포를 통하여 해당 공격 코드가 다른 공격 코드와 어느 정도 유사한지 비교할 수 있다. XSS 공격과 직접 객체 참조 취약점을 이용한 공격이 가장 넓은 분포를 보이고 있어 다른 공격과 동일한 코드를 많이 사용한다는 것을 알 수 있고 인젝션 취약점에 사용되는 공격 코드는 상대적으로 다른 공격 영역에 좁게 분포함을 알 수 있다. 공격 코드 분석 결과 인젝션 취약점을 이용한 공격은 SQL 질의어나 시스템 명령어와 같이 다른 취약점에 잘 이용되지 않는 키워드를 사용했기 때문이며 악성 파일 실행은 ../와 같이 경로 이동과 관련된 키워드 등 다른 공격에도 자주 사용되는 코드가 많음을 확인하였다.

V. 결 론

본 논문에서는 인터넷 뱅킹 보안을 위해 웹 공격을 탐지하고 분석하는 방법을 제시하였다. 인터넷 뱅킹의 구조적인 특징을 갖는 매개변수를 순차적으로 추출하여 알파벳으로 이루어진 정상 서열 패턴을 만든 후, 유사도를 측정하여 공격을 탐지하였다. 그리고 비정상 자료에서 공격 코드를 추출하여 비정상 패턴을 만든 후, 공격에 공통적으로 사용되는 코드의 유사도를 측정하여 공격의 특성을 파악하였다.

본 논문에서는 기존 연구에서 고려되지 않았던 POST 방식의 자료까지 이용함으로써 GET 방식에 제한적이었던 웹 공격 탐지 영역을 확장시켰고 인터넷 뱅킹의 정상 패턴을 자동으로 생성하여 관리의 효율성을 높이고 비정상행위 탐지의 단점인 긍정오류를 줄였다. 공격 코드 추출을 통해 생성된 비정상 패턴은 서열 정렬을 이용하여 유사도를 측정하기 때문에 오용 탐지의 단점인 부정오류 발생율을 낮추고 공격의 분석과 분류를 위해 공격의 유사도를 정량적인 수치로 제공하였다. 하지만 인터넷 뱅킹의 입력 값 검증이 약할 경우 정상 패턴의 개수가 많아져 오탐 발생 가능성이 높아질 수 있고 숫자나 단순한 문자열 변조는 탐지할 수 없다는 한계가 있다. 하지만 기존 보안장비와의 충돌 없이 응용계층에서 독립적인 운용이 가능하고 자동화된 패턴 생성으로 관리의 효율성을 높일 수 있어 인터넷 뱅킹 보안을 위한 효과적인 방법이 되리라 확신한다.

참고문헌

[1] 김인순, 김용석, "흔들리는 전자금융 - 급증하는 보안 사고", 전자신문, <http://www.etnews.co.kr/news/detail.html?id=200702200143>, 2007.2

[2] 박용주, "인터넷뱅킹 보안사고 급증세", 연합뉴스, <http://www.newshankuk.co.kr/news/news_view.asp?articleno=w2007082909171996701>,

2007.8

[3] 한국은행, "2008 1/4분기 국내 인터넷뱅킹서비스 이용현황", 한국인터넷진흥원, <http://isis.nida.or.kr/board/service/bbsView.jsp?bbs_id=1&item_id=403>, 2008.7

[4] Andrews Mike, and Whittaker James A., "How to break Web software : functional and security testing of Web applications and Web services", Addison-Wesley Professional, 2006

[5] Christopher Kruegel, and Giovanni Vigna, "Anomaly detection of web-based attacks", In Proceedings of the 10th ACM conference on Computer and communications security CCS '03, ACM Press, 2003.10

[6] D. Aucsmith, "Creating and maintaining software that resists malicious attack", Distinguished Lecture Series Atlanta, GA, 2004.9

[7] Dorothy E. Denning, "An intrusion-detection model", IEEE Transactions on Software Engineering, 13(2):222-232, 1987.

[8] Eisenstein E., and Schachman H. K., "Determining the roles of subunits in protein function", IRL Press, 1989

[9] Jaechul Park, and Bongnam Noh, "Web Attack Detection: Classifying Parameter Information according to Dynamic Web page", International Journal of Web Services Practices, Vol.2 No.1-2:68-74, 2006

[10] M. Bykova, S. Ostermann, and B. Tjaden, "Detecting network intrusions via a statistical analysis of network packet characteristics", In Proceedings of the 33rd Southeastern Symposium on System Theory, 2001

[11] M. Roesch, "Snort-lightweight intrusion detection for networks", In Proceedings of USENIX LISA'99, 1999

[12] McAfee, Inc. "Hacme Bank v2.0", <http://www.foundstone.com>, Foundstone Inc., 2008.7

[13] Needleman, S. B., and Wunsch C. D., "A general method applicable to the search for similarities in the amino acid sequence of two proteins", J. Mol. Biol., 48:443-453, 1970

[14] OWASP, "Top ten most critical web application vulnerabilities", <http://www.owasp.org/index.php/Top_10_2007>, 2008

[15] OWASP, "Vulnerability", <http://www.owasp.org/index.php/Category:Vulnerability>, 2008

[16] Pete Finnigan, "Oracle Security Step-by-step", SANS Institute, <http://www.securityfocus.com>, 2002

[17] SNORT, "Snort-The Open Source Network IDS", <http://www.snort.org>, 2008

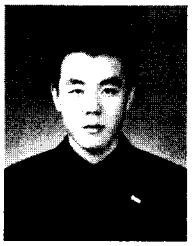
[18] Stuart McClure, Saumil Shah, and Shreeraj Shah, "Web Hacking: Attacks and Defense", Addison- Wesley Professional, 2002.8

[19] Temple F. Smith, and Michael S. Waterman, "Identification of Common Molecular Subsequences", Journal of Molecular Biology, 147: 195-197, 1981

[20] Victor Chapela, "Advanced SQL injection", OWASP, <http://www.owasp.org/docroot/owasp/misc/Advanced_SQL_Injection.ppt>, 2005

[21] William Yurcik, Samuel Patton, and David Dos, "An achillesi heel in signature-based IDS: Squealing false positives in snort", In RAID '01, 2001.

〈著者紹介〉



박 재 철 (Jae-Chul Park)

2003년 8월 : 전남대학교 정보통신협동과정 석사
2008년 8월 : 전남대학교 정보보호협동과정 박사
<관심분야> 웹 애플리케이션 보안, 침입탐지, 컴퓨터 포렌식, 데이터마이닝