악성 URL 탐지 및 필터링 시스템 구현
(An Implementation of System for Detecting and Filtering Malicious URLs)

장 혜 영 † 김 민 재 †† 김 동 진 ††
(Hye-Young Chang) (Min-Jae Kim) (Dong-Jin Kim)
이 진 영 †† 김 홍 근 ††† 조 성 제 ††††
(Jin-Young Lee) (Hong Kun Kim) (Seong-Je Cho)

요 약 2008년도 SecruityFocus 자료에 따르면 마이크로소프트사의 인터넷 익스플로러를 통한 클라이언트 측 공격(client-side attack)이 50%이상 증가하였다. 본 논문에서는 가상머신 환경에서 능동적으로 웹 페이지를 방문하여 행위 기반(즉, 상태변경 기반)으로 악성 URL을 분석하여 탐지하고, 블랙리스트 기반으로 악성 URL을 필터링하는 시스템을 구현하였다. 이를 위해, 우선 크롤링 시스템을 구축하여 대상 URL을 효율적으로 수집하였다. 특정 서버에서 구동되는 악성 URL 탐지 시스템은, 수집한 웹페이지를 직접 방문하여 머신의 상태 변경을 관찰 분석하고 악성 여부를 판단한 후, 악성 URL에 대한 블랙리스트를 생성·관리한다. 웹 클라이언트 머신에서 구동되는 악성 URL 필터링 시스템은 블랙리스트 기반으로 악성 URL을 필터링한다. 또한, URL의 분석 시에 메시지 박스를 자동으로 처리함으로써, 성능을 향상시켰다. 실험 결과, 게임 사이트가 다른 사이트에 비해 악성비율이 약 3배 많았으며, 파일생성 및 레지스트리 키 변경 공격이 많음을 확인할 수 있었다.

키워드 : 클라이언트 측 공격, 악성 URL 탐지, 악성 URL 필터링, 가상머신, 블랙리스트, 메시지 박스

Abstract According to the statistics of SecurityFocus in 2008, client-side attacks through the Microsoft Internet Explorer have increased by more than 50%. In this paper, we have implemented a behavior-based malicious web page detection system and a blacklist-based malicious web page filtering system. To do this, we first efficiently collected the target URLs by constructing a crawling system. The malicious URL detection system, run on a specific server, visits and renders actively the collected web pages under virtual machine environment. To detect whether each web page is malicious or not, the system state changes of the virtual machine are checked after rendering the page. If abnormal state changes are detected, we conclude the rendered web page is malicious, and insert it into the blacklist of malicious web pages. The malicious URL filtering system, run on the web client machine, filters malicious web pages based on the blacklist when a user visits web sites. We have enhanced system performance by automatically handling message boxes at the time of ULR analysis

· 이 연구는 단국대학교 대학원 연구보조장학금의 지원, 한국정보보호진흥원의 2008년 위탁과제의 지원, 2008년 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원(KRF-2008-313-D00821)을 받아 수행되었음

† 학생회원 : 단국대학교 정보컴퓨터학과
hychang@dankook.ac.kr
†† 학생회원 : 단국대학교 컴퓨터학과
6500cc@gmail.com
kdjorang@gmail.com
windofme@gmail.com
††† 종신회원 : 한국인터넷진흥원 공공정보보호단장
hgkim@kisa.or.kr
†††† 정 회 원 : 단국대학교 컴퓨터학부 교수
sjcho@dku.edu
(Corresponding author임)

논문접수 : 2009년 7월 22일
심사완료 : 2010년 1월 28일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제16권 제4호(2010.4)

on the detection system. Experimental results show that the game sites contain up to three times more malicious pages than the other sites, and many attacks incur a file creation and a registry key modification.

Key words : Client-side attack, Detecting Malicious URLs, Filtering Malicious URLs, Virtual Machine, Blacklist Message Box

1. 서 론

방화벽(Firewall) 및 침입 탐지/방어 시스템(IDS/IPS), 바이러스/웜 필터링 시스템 등 서버 측 공격을 방어하는 기술이 많이 일반화되었다. 따라서 기존 서버 위주의 전통 공격 경로가, 서버보다 상대적으로 방어수준이 떨어지는 사용자 PC에 악성코드(malware, 악성 소프트웨어)를 심는 클라이언트 측 공격(client-side attack)으로 공격 경로가 전이되고 있다[1-4]. SecruityFocus 자료에 따르면 마이크로소프트사의 인터넷 익스플로러를 통한 클라이언트 측 공격이 50%이상 증가하였다[5]. 일반 사용자 PC는 업데이트가 되지 않아 취약한 수많은 애플리케이션들을 포함하고 있는데, 그 중 웹 브라우저는 인터넷 트랜잭션에서 가장 취약한 링크라고 볼 수 있다. 즉, 해킹당한 웹 사이트를 사용자가 한번만 방문하더라도 해당 브라우저의 취약점이 악용될 수 있다. 최근, 공격자의 목적은 웹 페이지에 공격코드(혹은 공격 스크립트)를 삽입할 수 있는 취약성을 가진 웹 애플리케이션을 찾는 것이다. 삽입된 공격코드는 감염된 페이지를 방문하는 사용자에 대한 공격 수단으로 악용된다. 대부분의 경우, 이 공격이 성공되면 drive-by download, 즉 악성코드의 자동 설치로 이어진다. 공격자는 설치된 악성 소프트웨어를 이용하여, 해킹한 컴퓨터 시스템을 원격으로 제어하여 DDoS등의 공격으로 악용하기도 하며, 또 뱅킹 패스워드와 같은 주요 정보를 탈취하거나 스팸메일을 보내며, 또 다른 악성 프로그램들을 설치하기도 한다. 세계 최대 검색업체 구글(Google)의 Provos 연구원 등이 10개월간 10억 개의 URL을 분석한 결과, 3백만개 이상의 악성 URL들이 drive-by download를 구동하고 있다고 2008년 2월 보고하였다[1,2]. 또한 2007년 5월 14일 자료에 의하면, 5월 구글 검색엔진에 포함된 수십억 개 웹사이트 중 450만개를 추려 심층 분석한 결과, 10%인 45만개 사이트가 사용자 동의 없이 임의의 코드가 PC에 설치되는 drive-by download 기능을 가지고 있었다. 조사 대상 웹페이지 중 70만개는 사용자의 PC를 해킹에 취약하게 만드는 코드를 포함하고 있다고 한다. 이처럼, 'drive-by download' 위협이 계속 증가하고 있다. 더욱이 시그니처 기반으로 동작하는 백신의 경우, 빠르게 증가하고 있는 새로운 악성코드나 변종 악성코드를 탐지하지 못하여 피해가 더욱더 늘고 있다. 본 논문의 악성 URL 분석시스템은 시그니처 기반이 아닌 동적으로 상태변화를 모니터링하고 분석함으로써 변종 악성코드, 새로운 악성코드까지도 탐지할 수 있다는 장점이 있다.

본 논문의 "악성 URL 탐지 및 필터링 시스템"은 URL을 수집하여 분석하고 탐지하여 블랙리스트를 관리하는 "악성 URL 탐지시스템"과 그 블랙리스트 기반으로 사용자가 접근할 URL의 악성유무를 알려주는 "악성 URL 필터링 시스템"으로 크게 분류할 수 있다. 즉, "악성 URL 탐지 시스템"은 웹 페이지들을 수집한 다음, 그 웹 페이지들의 악성 여부를 분석·탐지하여 악성 웹 페이지들을 블랙리스트로 구성하여 준다. 이 시스템은 가상머신 환경에서 능동적으로 의심스러운 웹 페이지를 방문하여 악성여부를 행위기반(상태 변경 기반)으로 판단하고, 악성 웹 페이지들을 블랙리스트로 만들어 관리한다. 이때, 대상 URL들의 주소를 효율적으로 수집하기 위하여 크롤링을 적용하고, 탐지의 신뢰성을 확보하기위해 자가진단기능을 제공한다. "악성 URL 필터링 시스템"은 일반 사용자가 브라우저를 통해 웹 페이지를 방문할 때 악성 URL에 의한 클라이언트 공격을 필터링하여 준다. 이 시스템은 사용자 컴퓨터에 설치되어, 시스템이 관리하는 블랙리스트를 주기적으로 갱신하고, 웹브라우저에서 사용자가 접속한 URL들 중에서 악성 URL을 블랙리스트 기반으로 필터링하여 차단한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련 깊은 클라이언트 허니팟의 연구들에 대해 살펴보고, 3장에서는 악성 URL 탐지 및 필터링 시스템에 대해 설명한다. 4장에서는 제안하는 시스템을 구현하고 실험하여 결과를 분석하고, 5장에서 논문의 결론을 맺는다.

2. 관련연구

2.1 클라이언트 허니팟

서버 측 공격은 서버가 제공하는 서비스의 취약성을 목표로 이루어지지만, 클라이언트 측 공격은 악의적인 서버와 상호작용하거나 악의적인 데이터를 처리하는 클라이언트 어플리케이션의 취약성을 목표로 이루어진다.

서버 허니팟(honeypot)은 서버 측 취약점을 노출시켜 악의적인 클라이언트들로부터의 공격을 수동적으로 기다린다. 이에 반해 클라이언트 허니팟(client honeypot)은 클라이언트가 취약한 소프트웨어를 이용하여 잠재적

인 악성 웹 서버를 능동적으로 방문한다는 점에서 차이가 있다[6]. 클라이언트 허니팟은 공격이 발생했는지를 파악하기 위해 서버와 상호작용한다. 서버 허니팟과 유사하게, 클라이언트 허니팟도 상호작용 수준에 따라 High Interaction과 Low Interaction으로 분류되는데, 이는 서버가 클라이언트 허니팟 상에서 활용할 수 있는 기능적인 상호작용(Functional Interaction)수준을 의미한다.

본 논문에서 적용하는 'High Interaction 클라이언트 허니팟'은 실제 클라이언트를 가진 실제 시스템에 필적하는 충분한 기능을 갖추어 악성 서버(공격자)의 행위 정보를 수집할 수 있는 시스템을 말한다. 'High interaction 클라이언트 허니팟'의 예로는 미국 워싱턴 대학의 Spycrawler[3], MS 허니멍키(HoneyMonkey)[7], 뉴질랜드 빅토리아 대학의 Capture-HPC 및 허니넷 프로젝트(Honeynet Project)[4], MITRE의 허니클라이언트(HoneyClient)[5], Vrije University Amsterdam의 SHELIA 등이 있다. 본 논문의 악성 웹 페이지 분석·탐지 시스템 구축을 위해 Capture-HPC엔진을 사용하였다.

2.2 Capture-HPC

Capture-HPC는 뉴질랜드의 허니넷 프로젝트와 공동으로 웰링턴의 빅토리아 대학이 개발한 오픈소스 클라이언트 허니팟이다[4]. Capture-HPC는 레지스트리 수정 여부, 파일 시스템 수정 여부, 프로세스 생성/파괴 정보 분석 등에 근거하여 악성 웹 서버를 탐지하는데 초점을 두고 있다. 그러나 피싱 서버, 클라이언트 머신으로부터 브라우저 히스토리와 같은 사용자의 민감한 데이터를 획득하려는 웹 서버, 악의적 실행파일(명시적으로 다운로드되어 사용자에 의해 실행되는 실행파일)을 보유하고 있는 웹 서버 등을 탐지하지 않는다.

Capture-HPC의 특징은 다음과 같다. 첫째, 상태 변화에 따라 반응하는 이벤트기반 모델을 사용하여 상태 변화를 탐지함으로써 수행 속도가 빠르다. 둘째, 중앙 서버가 네트워크를 통해 수많은 클라이언트를 통제할 수 있도록 하여 확장성이 좋게 설계되었다. 셋째, 다른 클라이언트들에 적용할 수 있는 프레임워크로 되어 있다.

2.3 국내관련 연구

2.3.1 MC-Finder

한국인터넷 진흥원의 MC-Finder는 악성코드은닉 사이트 탐지 프로그램(도구)으로 규칙 기반의 정적 분석 기법을 사용한다. 즉, 개인 사용자 PC에 악성코드 다운로드를 유발하는 정보가 삽입된 사이트에 대한 점검을 수행한다. 그림 1은 MC-Finder의 실행화면이며, Mc-Finder는 규칙의 업데이트를 통해서 국내 웹사이트들이 악성코드 유포지나 경유지로 악용되는 것을 탐지, 차단한다. 수동 검사나 리스트 자동검사, 주기 반복검사 등

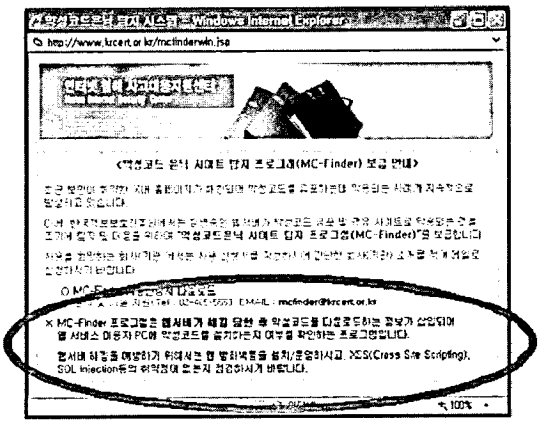


그림 1 KISA의 악성코드은닉 탐지 시스템

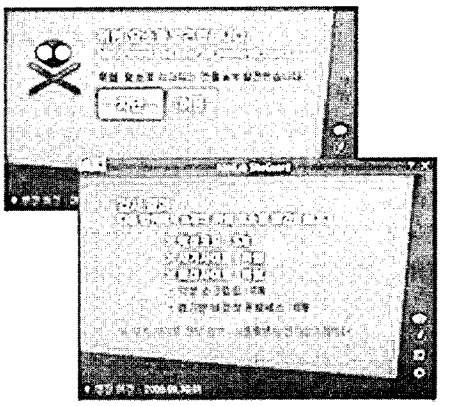


그림 2 안철수 연구소의 사이트가드

여러 형태의 검사 방법을 제공하며, 로그 등을 통하여 점검 결과를 파악가능하다. 본 논문과의 차이점으로 MC-Finder는 규칙 기반 정적 분석 기법을 적용하며, 본 논문의 시스템은 상태 변화 기반으로 일종의 동적 분석 기법을 적용한다는 점이다. 상태 변화 기반의 탐지 기법의 장점은 난독화된 코드 공격 및 알려지지 않은 공격까지도 탐지할 수 있다는 것이다.

2.3.2 사이트가드

악성코드 감염의 80% 이상이 웹사이트 접속을 통해 감염되므로 Ahnlab에서는 웹을 통한 위험을 예방하고 차단하고자 그림 2와 같이 사이트가드(siteguard)프로그램을 개발하였다. 사이트가드는 V3 엔진을 기반으로 실시간 악성코드를 검사하여 악성 사이트나 배포 사이트로 감지될 경우 접속을 차단한다. 주요 기능으로서 피싱 및 사기 사이트에 대한 점검, 안전 다운로드 점검, 악성 스크립트 감지 등을 수행한다. 사용자가 페이지 이동시마다 페이지의 위해여부를 판단해 주는 인터페이스를 제공

한다. 악성 스크립트 검사를 설정할 경우 iframe injection을 휴리스틱으로 감시할 수 있다. 사이트가드는 사용자가 사이트를 이동할 시에 V3엔진 기반의 악성 여부 탐지기가 검사를 수행하며 정기적으로 업데이트를 제공한다. 현재, 서울시 전자상거래센터에서 사기 사이트로 등록된 사이트를 차단하며, 구글, 네이버, 다음 검색 시 검색 결과 페이지의 안전 유무를 표시하여 준다[6].

3. 악성 URL 탐지 및 필터링 시스템

3.1 전체 시스템 구성

악성 웹 사이트에 의한 보안 위협을 최소화하기 위해 본 논문에서는 악성 URL 탐지 및 필터링 시스템을 제안한다. 그림 3과 같이 본 논문에서 제안하는 시스템은 크게 "관리자 부분"(크롤러가 수집한 URL들 중 악성 URL을 탐지하여 블랙리스트로 관리하는 시스템)과 "사용자 부분"(블랙리스트 기반으로 악성 URL을 필터링하는 클라이언트 시스템)으로 구성된다. 즉, 관리자 부분은 크롤러가 수집한 웹 페이지(URL)들을 분석하면서 악성 웹 페이지를 탐지한 후, 악성 웹 페이지들만 블랙리스트 저장소에 관리하는 모듈이다. 이 때 크롤러는 대상 웹 페이지들을 효율적으로 수집하여 준다. 사용자 부분은 관리자 부분이 생성한 블랙리스트를 주기적으로 업데이트하면서, 사용자가 브라우저를 통해 접속하고자 하는 URL들 중 그 블랙리스트에 있는 악성 URL을 필터링하는 모듈이다.

이를 세부적으로 살펴보면 그림 4와 같다. 그림 3의

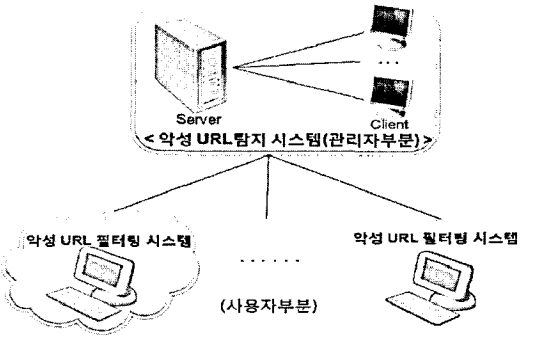


그림 3 악성 URL 탐지 및 필터링 시스템의 전체 구성도

관리자 부분이 그림 4의 "악성 URL 탐지 시스템"이고 이 시스템은 "악성 URL 크롤링 시스템"과 "악성 URL 탐지 및 분석 시스템"으로 구체화되어 있다. 한 개의 사이트에는 많은 URL 링크들이 복잡하게 연결되어 있다. 이들 링크를 따라 분석 대상 URL들을 효율적으로 수집하는 것이 필요하다. 본 논문에서는 웹 크롤러를 개발하여 웹 페이지들을 수집하고 중복된 URL들을 삭제하는 등의 과정을 통하여 분석 대상 URL 리스트를 추출하는 하였다. 악성 URL 크롤링 시스템에서 추출된 URL 리스트는 악성 URL 탐지 및 분석 시스템으로 전송되며, 각 URL을 실제 방문한 후 자신의 상태 변화 여부를 분석하여 각 대상 URL들이 악성인지 아닌지 판단한다. 악성 웹 페이지로 판정된 URL들을 블랙리스트로 구축하여 관리한다. "악성 URL 필터링 시스템"은 일반 사

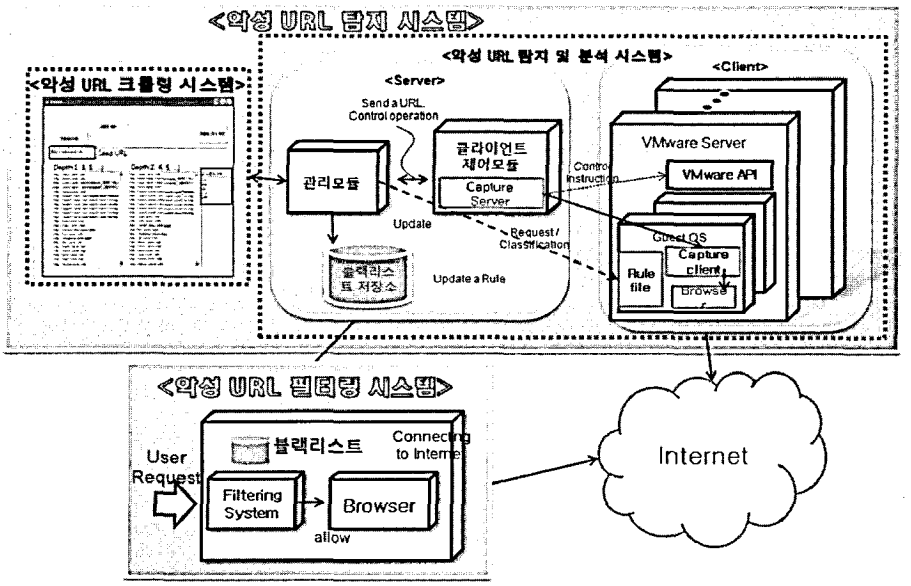


그림 4 악성 URL탐지 및 필터링 시스템의 세부 구성도

용자 컴퓨터에 구축되며, "악성 URL 탐지 시스템"이 구축한 블랙리스트를 다운받아 설치하여 주기적으로 업데이트한다. 사용자가 임의의 URL을 방문할 때마다 그 URL이 블랙리스트에 존재하는지 여부를 조사하여 악성 URL로의 접근을 필터링할 수 있다.

3.2 악성 URL 크롤링 시스템

2008년 1월 Netcraft의 조사결과 155,583,825개의 웹 사이트가 인터넷에 존재한다고 발표하였다. 2007년 평균적으로 웹사이트가 갖는 웹페이지의 수 273건을 계산하면 약 425억개의 웹 페이지가 존재하겠지만, 현재 웹사이트의 개수가 늘었을 뿐 아니라 WEB2.0으로 바뀌면서 웹문서의 개수는 감히 상상조차 할 수가 없다. 이렇게 많은 페이지들을 접할 수 있지만 필요한 정보를 빠른 시간에 찾아 가공하기란 쉽지 않다. 이때, 크롤링(crawling)을 사용하여 웹 페이지들을 수집할 수 있다. 즉, 크롤링이란 웹 크롤러(crawler)가 새로운 웹 페이지 또는 업데이트된 웹 페이지를 찾아 검색 엔진에 추가하는 과정을 말한다. 이렇게 웹 페이지를 읽어오는 프로그램을 웹 크롤러라고 하며 웹 로봇, 봇, 또는 스파이더라고 한다.

본 논문의 URL 크롤링 시스템은 시작 URL(Seed URL)을 정적분석하여 내부에 링크되어 있는 파생 URL을 수집한다. 웹 페이지 내부에서 링크를 표현하는 방법은 크게 3가지가 있다.

- href="ADDRESS"
- location="ADDRESS"
- frame src="ADDRESS"

ADDRESS에 해당하는 부분이 특정 웹 페이지 내부에 링크되어 있는 파생 URL이 된다. 웹 페이지의 소스를 파싱하여 ADDRESS에 해당하는 주소 부분만을 리스팅한다. 이때 depth를 설정하여 파생URL의 범위를 정할 수 있고, 이미지나 PDF로 링크되는 URL을 제외시키기 위해 제외 규칙도 적용하여 효율적으로 크롤링하게 하였다.

3.3 악성 URL 탐지 및 분석 시스템

그림 4의 "악성 URL 탐지 및 분석 시스템"은 능동적으로 대상 웹 페이지를 방문하면서 시스템 내부에 인가되지 않은 자원 상태 변경이 발생하였는지를 분석한다. 이때 Capture-HPC의 엔진(악성 여부를 분석해 주는 모듈)을 이용하여 실시간으로 시스템 상태를 검사한다. 이는 클라이언트-서버 방식으로 작동하며, 서버가 제시한 웹 페이지들을 가상머신 상의 클라이언트가 차례로 방문하여 클라이언트 머신의 상태 변화를 조사하고 악성 URL을 판단한다. 서버 부분은 클라이언트를 제어(시작, 종료, 클라이언트에 웹 사이트와 상호작용을 명령)하는 모듈(Capture Server)과 이 제어 모듈들을 전체적으로 관리하는 관리 모듈(Manager Server), 악성 컨텐츠라고 판단된 URL들을 DB화시켜 저장하는 블랙리스트 저장소(Black List Repository)로 구성된다.

클라이언트 부분은 가상머신 상에 적재되어 실행되는 윈도우즈 이미지와 그 안에서 실제 동적 웹 페이지를 분석하고 탐지하는 기능으로 구성된다. 클라이언트는 서버의 제어를 받아 동작하며, 브라우저를 통해 서버로부터 전달받은 대상 URL을 방문하여 그 웹 서버와 상호작용한다. 동시에 비인가 상태 변화(이벤트) 여부를 감시하여 동적인 상태 변화가 있다면 서버에게 그 정보를 전송(보고)한다. 이때, 악성 공격이나 악의적 프로그램의 실행을 고립화(sand boxing)하기 위해, 가상머신(Virtual Machine, VM)에서 웹 페이지를 렌더링하고, 악성 행위는 렌더링 전후의 시스템 상태를 점검하여 판단한다. 가상머신 기반으로 수행하는 이유는 웹 페이지의 불법 행동을 고립시켜 분석하고 탐지할 수 있기 때문이다. 즉 악성 URL 방문으로 인해 현재의 가상머신이 오염되면 그 오염의 원인을 파악한 후 그 가상머신을 종료시킨다. 이때 다른 가상머신이 있다면 각각의 가상머신은 서로 독립적인 머신으로 동작하므로 공격으로 인한 피해가 전파되지 않는다.

서버는 클라이언트로부터 전송되어온 상태 변화가 악성이라고 판단되면, 해당 URL을 블랙리스트에 추가한 다음, 가상머신의 이미지를 리버팅시켜(가상머신의 무결성을 확보한 후) 클린 상태에서, 다음 웹 서버(URL)와 상호작용을 시작할 수 있게 한다. 서버는 블랙리스트 저장소를 운영하면서 일반 사용자에게 블랙리스트를 제공한다.

악성 웹 컨텐츠 탐지 시, 판단 근거는 규칙 파일(Rule file)의 명세를 따르고 이 명세는 관리서버에 의해 업데이트 된다. 감시 대상 상태변화는 파일 생성 및 제거, 프로세스 생성 및 파괴, 레지스트리 항목 생성 등이다. 관리서버는 탐지 서버로 부터 악성 URL 리스트를 전달받게 되며 이렇게 수집된 악성 URL은 MD5 해시 값으로 블랙리스트 저장소에 DB 형식으로 저장된다.

3.4 규칙 파일

대상 웹 페이지 방문 시에 가상머신 상의 파일 시스템, 레지스트리, 프로세스의 상태 변화를 모니터링하고 발생된 이벤트들에 대해 생성된 리포트를 분석하여, 악성 웹 페이지를 판정한다. 이때 규칙파일을 이용하여 이벤트 유형(event type)과 객체 이름에 의해 특정 이벤트를 무시(omission, 생략)하거나 포함(inclusion)할 수 있다. 무시할 규칙은 + 부호로 시작되어 표시되고, 명시적 포함 규칙은 - 부호로 표시된다. 예를 들어 그림 5과 같이, 방문한 URL에서 .bat, .cmd, .exe 파일에 쓰기 이벤트가 모니터링되면 그 URL을 악성으로 간주하

```
- Write   .*  .+\.bat
- Write   .*  .+\.cmd
- Write   .*  .+\.exe

+ SetValueKey C:\\Program Files\\Internet Explorer\\iexplore\.exe HKCU\\Software\\MySoftware\\IEToy\\.+
```

그림 5 규칙파일의 예

여 블랙리스트에 포함한다. 본 연구에서는 규칙 파일에 새로운 규칙을 추가하였다. 예로 수많은 URL들을 빠르게 분석하기 위해, IE-Toy를 사용하여 특정 웹 페이지 브라우징 시 나타나는 메시지박스를 자동으로 처리하였다. 그림 5의 마지막 줄에 있는 규칙은 IE-Toy의 레지스트리 값이 iexplore에 의해 변경되는 이벤트를 무시하라는 의미로 추가되었다.

3.5 자가진단 기법

행위 분석 기반의 동적 탐지 기법의 장점은 알려지지 않은 악성 프로그램도 탐지할 수 있으며, 주요 시스템 상태 변화를 모니터링할 수 있기 때문에 정확하다는 장점이 있다. 단점으로는, 실제로 해당 URL의 페이지들을 웹 브라우저로 직접 열어 보고 상태를 분석, 모니터링하기 때문에 악성 프로그램에 의해 분석·탐지용 OS 이미지가 오염될 수 있다는 것인데, 이는 가상머신을 사용하여 극복할 수 있다[1,2,4,7]. Capture-HPC 엔진이나 MS, 구글의 관련 연구팀도 자신들의 논문에서 이를 언급하고 있다. 따라서 자가진단 기법이 없더라도 본 실험의 신뢰성은 다른 논문들[1,2,4,7]에서 검증되었다고 볼 수 있다.

본 연구에서는 좀 더 많은 신뢰성을 확보하기 위해, Guest OS 및 가상머신이 공격받아 정상 동작하지 못하는 상황까지도 고려하여 자가진단 방법을 도입하였다. 일종의 이중체크 기법이다. 자가진단은 행동기반 동적 분석기의 감염 가능성을 고려하여 진단하는 것을 목적으로 한다. 즉, 자가 진단을 통해서 운영체제 이미지의 오염을 검증한다. 자가진단 방법으로, 안전하다고 알려진 URL들(예로, 정보보호진흥원 URL, 정보보호온라인 학습장 URL, 국가보안기술연구소 URL, 구글 URL, MS URL 등)을 일정 주기마다 모니터링하여 그 사이트가 안전하다고 판단되는지 여부를 판단한다. 한 방법으로 그림 6에 나타난 것처럼 10개의 페이지마다 한 개씩 안전한 사이트를 삽입하여 해당 로그 결과를 분석하여 악성URL 탐지 및 분석 시스템의 오염여부를 확인할 수 있다. 만약 다른 페이지가 아닌 안전한 사이트에서 악성 행위가 발견되면 서버가 오염된 것으로 판단하여 새로운 OS 이미지를 리버팅하고, 서버 관리자에게 그 사실을 알린다. 본 논문에서는 안전한 사이트의 예로 구글(www.google.co.kr), 한국인터넷진흥원(www.kisa.or.kr), 정보보호 온라인 학습장(www.sis.or.kr) 등으로 실험하였다.

그림 6 10개의 URL 페이지마다 안전한 사이트를 삽입

그림 6의 예에서는 10개의 URL 페이지마다 자가진단용 안전한 URL을 넣었는데, 상황에 따라 N개의 페이지를 넣음으로 상황에 맞게 설정할 수 있으며, 또는 한 도메인마다 하나의 안전한 URL 페이지를 적용하여 분석·탐지 시스템의 오염 여부를 진단할 수 있다.

3.6 블랙리스트 기반의 악성 URL 필터링 시스템

사용자 PC에 설치되는 악성 URL 필터링 시스템은 사용자가 웹서핑을 하기 위해 원하는 URL을 웹 브라우저의 주소창에 입력했을 때 그 URL이 악성 URL인지의 여부를 점검한다. 이를 위해서 "악성 URL 탐지 시스템"이 작성한 블랙리스트를 다운받아 사용자 자신의 시스템에 설치한다. 또한, 블랙리스트 DB는 최신의 내용을 유지해야 하므로 주기적으로 업데이트를 요청하여 갱신한다. 업데이트 방식은 효율적인 동기화 및 부하 경감을 위하여 각 URL 별 추가, 삭제, 변경의 상태를 구분하여 업데이트할 수 있게 하였다.

사용자가 웹 브라우저의 주소창에 URL을 입력하면 실제 웹페이지에 접속하기 전에 악성 여부를 판별하는 작업을 진행한다. 즉, 해당 URL이 블랙리스트에 있다면, 접속을 필터링하여 실제 페이지를 보여주기 전에 경고 메시지를 출력하여 준다.

4. 실험 및 실험결과 분석

본 논문은 인텔의 Core2Quard, 8GB RAM서버에

Microsoft Server 2003을 구동시키고, Capture HPC 2.5.1엔진을 설치하였다. 탐지시스템의 서버의 실행을 위해 선 마이크로시스템즈의 Java JRE를 설치하고, 탐지시스템의 클라이언트가 구동될 Guest OS를 위해 WMware 서버1.0.6을 설치하였다. 또한, 마이크로소프트 비주얼 C++ Redistributable Libraries(SP1)와 IE Toy 1.8버전을 설치하였다.

4.1 메시지 박스를 자동처리함으로써 악성 URL 탐지 시스템의 성능 개선

본 논문에서는 대상 URL들의 분석·탐지를 고속화하기 위해 메시지 박스를 자동으로 처리하였다. 어떤 URL을 접근하게 되면 메시지 박스가 뜨는데, 경우에 따라 메시지 박스의 확인 버튼을 누르지 않으면 더 이상의 브라우징이 진행되지 않아 해당 웹 페이지에 대한 분석이 에러 처리되거나, 브라우징이 완료되지 않아 상당한 시간 동안 분석·탐지를 진행하지 못하고 지연되는 경우가 발생한다. 예를 들면 그림 7과 같이 특정 URL 방문 시 스크립트 오류 발생에 대한 메시지 박스가 뜨게 된다. 악성 URL 여부에 대한 분석과정을 지연없이 진행하기 위해서는 사용자가 직접 No 버튼을 눌러야 한다. 또한, 그림 8과 같이 어떤 URL을 방문하면 즐겨찾기 추가 유도 화면이 뜨게 된다. 이처럼 메시지 박스를 자동으로 제거하는 것이 필요하다.

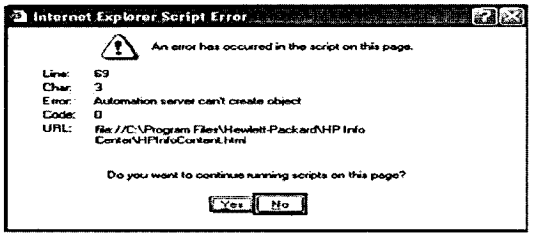


그림 7 스크립트 오류 발생 메시지 박스

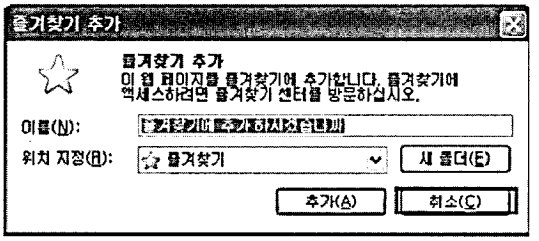


그림 8 즐겨찾기 추가 유도 메시지 박스

본 논문에서 IE Toy를 적용함으로써 브라우저에서 발생하는 대부분의 메시지 박스를 제거하였다. 메시지 박스 자동처리했을 때와 그렇지 않을 때를 비교하기 위해 Group1과 Group2로 나누어 실험하였다. 이때, 악성 URL의 분석에 따른 오버헤드가 존재하므로 정상적인 URL만을 대상으로 하였다. Group1은 메시지박스를 띄우는 URL 100개와 메시지박스가 없는 400개의 사이트로 구성되어 있고, Group2는 메시지박스를 띄우는 URL 150개와 그렇지 않은 350개의 URL로 구성되어 있다. 이때의 결과는 표 1과 같다. URL중 20%가 메시지 박스가 있는 Group1에서는 메시지박스를 처리했을 때 약 4.3배만큼 성능이 좋아졌고, 30%의 메시지 박스가 있는 URL Group2에서는 약 6.7배로 성능이 좋아졌다. 이로써 메시지박스를 처리했을 때 탐지속도를 높일 수 있었고, 메시지 박스를 띄우는 URL이 많을수록 성능이 더 좋아졌다.

4.2 악성 URL 필터링 시스템의 실제 예

사용자 PC에 구축되는 악성 URL 필터링 시스템은, 먼저 악성 URL 탐지 시스템이 관리하는 블랙리스트를 다운받아 설치하고 주기적으로 업데이트한다. 또한, 사용자 웹 브라우저와 연동하여 사용자가 접속하는 URL에 대해 그 블랙리스트 기반으로 악성 유무를 판단하고, 악성 URL 접근 시에 경고 메시지를 출력하여 차단하도록 구축되었다. 이를 위해, 그림 9와 같이 사용자가 입력하는 URL이나 마우스로 클릭하는 링크 URL을 커널 수준에서 가로챈(hooking) 후 URL에 대한 MD5 해시 값을 계산하여, 블랙리스트에 있는 해시 값들과 비교하도록 구현했다. 일치하는 해시 값이 블랙리스트에 있다면, 접근하려는 웹 사이트가 악성이므로 접근하지 못하게 그림 10과 같이 경고 메시지를 주고 차단할 수 있게 하였다.

사용자 인터페이스는 웹 브라우저에 플러그인 형태인 툴바로 구현되었고 'On/Off 버튼'을 이용하여 분석·탐지 기능을 켜고 끌수 있다. 그림 10은 툴바 인터페이스와 경고 안내 페이지를 보여주고 있다. 악성 URL에 대한 경고 및 안내 메시지는 웹 브라우저의 내용 창에 로컬에 미리 준비되어 있는 안내 웹 페이지를 보여준다.

표 1 메시지박스 처리에 따른 총 검사 시간의 변화 (단위:초)

Group / 메시지박스 처리여부	Group1 [100/500] (20% 메시지박스 URL)	Group2 [150/500] (30% 메시지박스 URL)
자동 처리하지 않음	1303.62244	2033.55285
자동 처리함	303.58362	303.58412
성능향상률	4.294배	6.698배

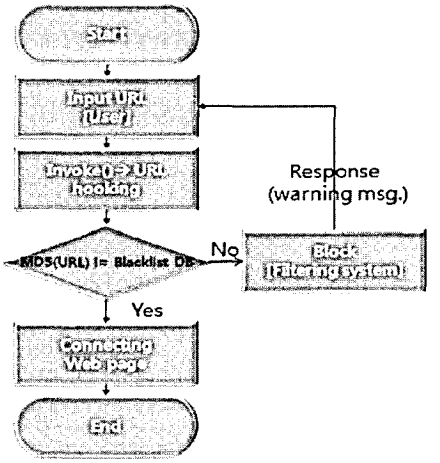


그림 9 악성 URL 필터링 시스템 흐름도

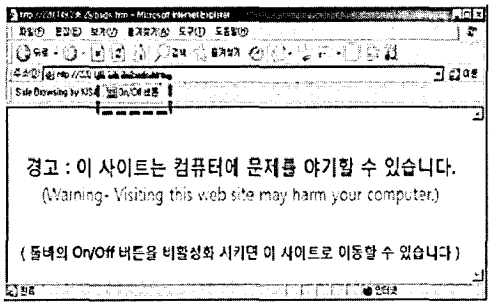


그림 10 악성 URL 페이지 필터링의 예

4.3 URL/호스트들의 분류에 따른 실험 결과

어떤 종류의 웹 사이트를 통해 악성 컨텐츠가 배포되는지 확인하기 위해서 와레즈 사이트, 게임 사이트, 성인 사이트, 쇼핑몰 사이트, 정보 및 커뮤니티 사이트의 5개로 분류하여 실험해 보았다. 각 카테고리 페이지는 Google과 Naver 검색 엔진을 통하여 무작위로 수집하여 시작 URL(Seed URL)을 찾았다. 하위의 파생 페이지는 시작URL을 직접 클릭하여 링크를 이동하면서 파생 URL들을 수집하여 실험하였다. 그 결과 게임사이트가 다른 사이트에 비해 악성비율이 약 3배 높은 것을 확인할 수 있었다.

표 2와 같이 총 51,856개의 URL 중 악성이라고 분류된 89개의 로그를 정적분석한 결과, False Alarm이 5개 발생하였다. 이중 4개는 idew.exe의 실행으로, idew.exe는 인터넷 익스플로러가 문제가 생겨 종료할 때 실행되는 파일로 tmp파일을 열어 기록한다. 방문한 URL에서 .bat, .cmd, .exe파일을 실행시키면 무조건 악성URL로 분류하라고 3.4절에서 규칙을 정했었는데 idew.exe와 같이 정상적으로 실행되는 파일은 규칙파일에 적용하여 4개의 False Alarm을 해결할 수 있다.

나머지 1개의 False Alarm은 ntvdm.exe 실행에 의한 것이다. ntvdm.exe는 16비트 프로세스가 32비트 플랫폼에서 실행할 수 있도록 해주는 윈도우즈 XP에 원래 존재하는 정상적인 실행파일이기 때문에 규칙파일에서 예외처리가 되어야 한다. 하지만 그림 11과 같이 로그를 자세히 살펴보면, ntvdm.exe이라는 실행 파일과 같은 이름의 프로세스가 생성되어 scs1.tmp라는 파일을 반복적으로 씀으로써 CPU의 사용률을 높여서 시스템을 마비시키기 때문에 악성으로 분류하였다. 즉, 정상적인 실행파일로 위장하여 공격을 할 수 있으므로 동적 분석의 결과인 로그파일을 정적분석하여 False Alarm을 낮출 수 있다.

4.4 악성페이지의 공격유형 분석

84개의 로그를 분석한 결과 표 3과 같은 결과를 얻을 수 있었다. 주로 파일생성 공격이 가장 많이 있었고, 레지스트리키 변경, 방화벽정책변경 순으로 공격비율이 높았다.

파일생성 공격의 예를 들어보면 그림 12의 로그와 같이, vv.com이라는 프로세스에 의해 특정 게임에 대한 사용자의 정보 등을 수집하여 특정 이메일 주소로 보내는 등의 행위를 하는 windf.exe, windf.hlp 파일을 생성 공격이 있었다.

또한, 그림 13과 같이 레지스트리 값을 변화시켜 특정 포트를 방화벽 정책에 적용시키는 행위도 있었다. 웹 서비스와 관련 없는 1114번 UDP포트(Mini SQL), 1115번 UDP포트(ARDUS Transfer)를 방화벽 정책에 적용시키는 것으로 보아 악성 행위라고 추정할 수 있다.

표 2 URL 분류 및 악성 URL의 실험 결과

분류	Seed URL	파생 URL	악성페이지 수			악성비율 (%)	소요시간 (초)
			1차 악성 페이지 수	False Alarm	2차 악성 페이지 수		
게임	100	3,448	14	0	14	0.406	16,423
성인	100	10,302	19	4	15	0.146	130,842
쇼핑	100	10,154	16	0	16	0.158	40,061
와레즈	100	8,718	1	0	1	0.012	15,444
정보커뮤니티	100	19,234	39	1	38	0.198	73,352
계	500	51,856	89	5	84	0.92	276,122

```
"process","7/8/2008 3:23:44.624","C:\Program Files\Internet Explorer\IEXPLORE.EXE",
"created","C:\WINDOWS\system32\ntvdm.exe"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
"file","7/8/2008 3:23:45.843","C:\WINDOWS\system32\ntvdm.exe","Write","C:\WINDOWS\Temp\scsl.tmp"
```

그림 11 ntvdm.exe 악성 로그

```
"file","7/8/2008 2:18:15.514","C:\Program Files\Internet Explorer\IEXPLORE.EXE","Write",
"C:\Documents and Settings\csss\Local Settings\Temporary Internet Files\Content.IE5\S5LVMIJY\H[1].exe"
"file","7/8/2008 2:18:15.514","C:\Program Files\Internet Explorer\IEXPLORE.EXE","Write",
"C:\Documents and Settings\csss\Local Settings\Temporary Internet Files\Content.IE5\S5LVMIJY\H[1].exe"
"process","7/8/2008 2:18:16.186","C:\Program Files\Internet Explorer\IEXPLORE.EXE","created","C:\WINDOWS\Temp\vv.com"
"file","7/8/2008 2:18:17.452","C:\WINDOWS\Temp\vv.com","Write","C:\WINDOWS\system32\drivers\windf.EXE"
"file","7/8/2008 2:18:17.452","C:\WINDOWS\Temp\vv.com","Write","C:\WINDOWS\system32\drivers\windf.EXE"
"file","7/8/2008 2:18:17.452","C:\WINDOWS\Temp\vv.com","Write","C:\WINDOWS\system32\drivers\windf.hlp"
"file","7/8/2008 2:18:18.468","System","Write","C:\WINDOWS\system32\drivers\windf.EXE"
"file","7/8/2008 2:18:19.468","System","Write","C:\WINDOWS\system32\drivers\windf.EXE"
"process","7/8/2008 2:18:19.905","C:\Program Files\Internet Explorer\IEXPLORE.EXE","terminated","C:\WINDOWS\Temp\vv.com"
```

그림 12 특정 게임의 사용자 정보 관련 악성 로그

```
"registry","7/8/2008 2:33:20.484","C:\Program Files\Internet Explorer\IEXPLORE.EXE","SetValueKey",
"HKLM\SYSTEM\ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\GloballyOpenPorts\List\1114:UDP"
"registry","7/8/2008 2:33:20.593","C:\Program Files\Internet Explorer\IEXPLORE.EXE","SetValueKey",
"HKLM\SYSTEM\ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\GloballyOpenPorts\List\1115:UDP"
"registry","7/8/2008 2:33:22.812","C:\Program Files\Internet Explorer\IEXPLORE.EXE","DeleteValueKey",
"HKLM\SOFTWARE\Microsoft\ESENT\Process\iexplore\DEBUG\Trace Level"
"registry","7/8/2008 2:33:22.812","C:\Program Files\Internet Explorer\IEXPLORE.EXE","SetValueKey",
"HKLM\SOFTWARE\Microsoft\ESENT\Process\iexplore\DEBUG\Trace Level"
"registry","7/8/2008 2:33:27.859","C:\Program Files\Internet Explorer\IEXPLORE.EXE","DeleteValueKey",
"HKLM\SYSTEM\ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\GloballyOpenPorts\List\1114:UDP"
"registry","7/8/2008 2:33:27.859","C:\Program Files\Internet Explorer\IEXPLORE.EXE","DeleteValueKey",
"HKLM\SYSTEM\ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\GloballyOpenPorts\List\1115:UDP"
```

그림 13 특정 포트를 방화벽 정책에 적용시키는 악성 행위 로그

표 3 악성 페이지의 공격유형 분석결과

공격(악성)의 종류	게임	성인	쇼핑	와레즈	정보 커뮤니티	계
파일생성	5	0	14	0	23	42
레지스트리키 변경	5	14	0	1	9	29
방화벽정책변경	4	1	0	0	5	10
인증서접근	0	0	2	0	0	2
프록시정책변경	0	0	0	0	1	1
계	14	15	16	1	38	84

5. 결 론

웹 해킹 공격은 악성코드 유포를 통한 개인정보 유출과 직결되어 있어 근본적인 대책을 서둘러야 한다. 웹 사이트 해킹 피해가 사이트 회원은 물론 사이트 방문자들의 개인정보 유출과 직결되는 만큼 기업들의 웹 사이트 보안을 의무화할 필요가 있다. 그러나 웹 해킹과 악성코드 유포 공격이 결합되면서 해결책을 찾기가 쉽지 않다. 본 논문에서는 악성 웹 사이트에 의한 보안 위협을 최소화하는 방안을 체계적으로 제안하였다. 즉, 가상머신 환경에서 능동적으로 웹 페이지를 방문하여 행위기반으로 사용자 시스템의 보안 침해여부를 판단하는 악성 URL 탐지 및 필터링 시스템을 구현하였다. 이때, 자가진단 기법을 사용하여 신뢰성을 높이고, 방문 URL의 메시지 박스를 자동으로 처리하여 메시지박스 URL이 20%일 때 약 4.3배, 메시지박스 URL이 30%일 때 약 6.7배 탐지속도를 높일 수 있었다. 또한 악성 URL들을 분석한 결과, 게임 사이트에서 악성 URL이 가장 많이 발견되었으며 주로 파일생성 및 레지스트리 변경 공격이 많음을 확인할 수 있었다.

참 고 문 헌

[1] N. Proves, D. McNamee, et. al., "The Ghost In The Browser Analysis of Web-based Malware," *Proc. of the first USENIX workshop on hot topics in Botnets*, Apr. 2007.

[2] Niels Provos, Google's Anti-Malware Team, "All Your iFrame Are Point to Us," *Google Technical Report provos-2008a*, February 11, 2008.

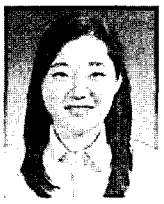
[3] Alexander Moshchuk, Tanya Bragin, et. al., "A Crawler-based Study of Spyware on the Web," *Proc. of the 2006 Networks and Distributed System Security Symposium*, pp.17-33, Feb. 2006.

[4] Christian Seifert, "Know Your Enemy: Malicious Web Servers," The Honeynet Project, *KYE paper*, Aug. 2007.

[5] Kathy Wang, "Using Honeyclients for Detection an Response Against New Attacks," MITRE, http://www.cerias.purdue.edu/assets/symposium/2008-panels/Wang-Honeyclients-CERIAS-Symposium-18Mar08-v2.pdf

[6] Yi-Min, et. al., "Strider HoneyMonkeys: Active, Client-Side Honeypots for Finding Malicious Websites," To be appear in *IEEE* Transactions on Computers, May 2007.

[7] Yi-Min Wang, Doug Beck, et. al., "Automated Web Patrol with Strider HoneyMonkeys," *Proc. of the Networks and Distributed System Security Symposium*, pp.35-49, Feb. 2006.



장 혜 영

2003년 단국대학교 이과대학 전산통계학과(이학사). 2005년 단국대학교 컴퓨터과학및통계학과(이학석사). 2007년 단국대학교 정보컴퓨터과학과 박사과정수료. 관심분야는 컴퓨터 보안, 임베디드 소프트웨어 등

김 민 재

2008년 단국대학교 컴퓨터과학과(이학사) 2010년 단국대학교 컴퓨터학과(공학석사) 관심분야는 리버스 엔지니어링, 클라이언트 허니팟, 시스템최적화

김 동 진

2009년 단국대학교 컴퓨터과학과(이학사) 2009년 단국대학교 컴퓨터학과 석사과정 중. 관심분야는 컴퓨터보안 등

이 진 영

2009년 단국대학교 컴퓨터과학과(이학사) 2009년~현재, 단국대학교 컴퓨터학과 컴퓨터과학(공학석사). 관심분야는 난독화, 패킹, 지적재산권 보호

김 홍 근

1985년 서울대학교 컴퓨터공학과(공학사) 1987년 서울대학교 컴퓨터공학과(공학석사). 1994년 서울대학교 컴퓨터공학과(공학박사). 1994년~1996년 한국전산원 1996년~2009년 한국정보보호진흥원. 2009년~현재 한국인터넷진흥원. 관심분야는 병렬처리, 컴퓨터 보안, 소프트웨어 보안 등

조 성 제

1989년 서울대학교 컴퓨터공학과(공학사) 1991년 서울대학교 컴퓨터공학과(공학석사). 1996년 서울대학교 컴퓨터공학과(공학박사). 2001년~2002년 미국 University of California, Irvine 객원연구원. 2009년~현재 미국 University of Cincinnati 객원연구원. 1997년 3월~현재 단국대학교 컴퓨터학부 교수 관심분야는 컴퓨터보안, 시스템소프트웨어, 실시간스케줄링, 임베디드 소프트웨어 등