고 상호작용 클라이언트 허니팟을 이용한 실행 기반의 악성 웹 페이지 탐지 시스템 및 성능 분석

(Execution-based System and Its Performance Analysis for Detecting Malicious Web Pages using High Interaction Client Honeypot)

김 민 재 † 장 혜 영 ††
(Min-Jae Kim) (Hye-Young Chang)

조 성 제 †††
(Seong-Je Cho)

요 약 Drive-by download와 같은 클라이언트 측 공격은, 악의적인 서버와 상호작용하거나 악의적인 데이터를 처리하는 클라이언트 애플리케이션의 취약점을 대상으로 이루어진다. 전형적인 공격은 특정 브라우저 취약점을 악용하는 악성 웹 페이지와 관련된 웹 기반 공격으로, 클라이언트 시스템에 멀웨어를 실행하거나 클라이언트의 제어를 악의적인 서버에게 완전히 넘겨주기도 한다. 이러한 공격을 방어하기 위해, 본 논문에서는 Capture-HPC를 이용하여 가상머신에서 실행기반으로 악성 웹 페이지를 탐지하는 고 상호작용(high interaction) 클라이언트 허니팟을 구축하였다. 이 실행기반 탐지 시스템을 이용하여 악성 웹 페이지를 탐지하고 분류하였다. 또한 가상머신의 이미지 개수 및 한 가상머신에서 동시 수행하는 브라우저 수에 따른 시스템 성능을 분석하였다. 실험 결과, 가상머신의 이미지 수는 하나이고 동시 수행하는 브라우저의 수가 50개일 때 시스템이 적은 리버팅 오버헤드를 유발하여 더 나은 성능을 보였다.

키워드 : Drive-by download, 고 상호작용 클라이언트 허니팟, 가상머신, 실행기반 탐지, 성능 분석

Abstract Client-side attacks including drive-by download target vulnerabilities in client applications that interact with a malicious server or process malicious data. A typical client-side attack is web-based one related to a malicious web page exploiting specific browser vulnerability that can execute malware on the client system (PC) or give complete control of it to the malicious server. To defend those attacks, this paper has constructed high interaction client honeypot system using Capture-HPC that adopts execution-based detection in virtual machine. We have detected and classified malicious web pages using the system. We have also analyzed the system's performance in terms of the number of virtual machine images and the number of browsers executed simultaneously in each virtual machine. Experimental results show that the system with one virtual machine image obtains better performance with less reverting overhead. The system also shows good performance when the number of browsers executed simultaneously in a virtual machine is 50.

Key words : Drive-by download, High interaction client honeypot, Virtual machine, Execution-based detection, Performance analysis

1. 서 론

최근 서버 시스템의 보안 관리, 방어체계 등이 강화되면서 공격 경로가 서버 측에서 비교적 방어체계가 취약한 클라이언트 측으로 전이되고 있다[1-4].

SecurityFocus의 2008년 자료에 의하면, 클라이언트 공격 중에서도 브라우저를 이용한 공격이 급증하고 있다. 또한, 일반 사용자가 악의적인 스크립트가 삽입된 페이지에 접속만 하더라도 공격 코드가 실행되는 피해가 늘고 있다. 이런 공격은 패치 되지 않은 웹 브라우저의 취약점이 가장 큰 원인이 되고 있다. 이처럼 웹 브라우저의 취약점을 이용하여 사용자가 인지하지 못하는 사이에 공격코드를 다운로드하여 실행하는 공격을 drive-by download 공격이라고 한다. 이렇게 다운로드 된 악성 코드는 스파이웨어, 키 로거, 원격제어 등 다양한 형태로 존재하기 때문에 더 많은 피해가 우려된다. 실제로 10억 개의 URL 중 3백만 개 이상의 악성 URL들이 drive-

· 본 연구는 2009년 단국대학교 교내연구비 지원으로 이루어졌음.
· 이 논문은 2009 한국컴퓨터종합학술대회에서 'High interaction 클라이언트 허니팟을 이용한 실행 기반의 악성 웹 페이지 탐지 및 성능 분석'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 단국대학교 컴퓨터학과 컴퓨터과학
6500cc@gmail.com
†† 학생회원 : 단국대학교 정보컴퓨터과학과 컴퓨터과학
hystella@gmail.com
††† 정 회 원 : 단국대학교 공과대학 컴퓨터학부 교수
sjcho@dku.edu
(Corresponding author임)

논문접수 : 2009년 8월 14일
심사완료 : 2009년 10월 5일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제12호(2009.12)

by download 공격과 관련 있다는 통계가 있다[1,2].

본 논문에서는 drive-by download 공격으로부터 클라이언트(웹 브라우저 이용자)를 보호하기 위해, '웹 브라우저를 통한 웹 페이지 방문 시 해당 웹 페이지의 악성 여부를 탐지하여 분석하는 클라이언트 허니팟 시스템'을 구축하고 실험하였다. 구축 시스템은 가상머신(Virtual Machine, 이하 VM) 상에서 능동적으로 웹 페이지를 방문하며, 웹 페이지 방문 후 VM의 상태변화를 조사하여 악성 여부를 판단한다. VM 기반의 클라이언트 허니팟 시스템에서 수많은 웹 페이지를 분석할 때 시스템 성능이 매우 중요하다. 이에 본 논문에서는 하나의 물리머신(host OS) 상에 구동하는 VM 이미지(guest OS)의 개수와 할당된 메모리의 크기에 따른 악성 웹 페이지 탐지 성능도 분석하였다.

논문의 구성은 다음과 같다. 2장에서는 클라이언트 허니팟을 이용한 악성 웹 페이지 탐지와 가상머신 관련 연구를 기술한다. 3장에서는 가상머신 기반 Capture-HPC를 적용한 고 상호작용 클라이언트 허니팟에 대해 설명한다. 4장에서는 대상 클라이언트 허니팟에서 가상머신 이미지의 개수와 할당 메모리 용량에 따른 시스템 성능을 분석하고, 5장에서 결론을 맺는다.

2. 관련 연구

2.1 고 상호작용 클라이언트 허니팟

최근 서버 측 공격보다 클라이언트 측 공격이 증가함에 따라, 허니팟의 클라이언트 형태인 클라이언트 허니팟(client honeypot)이 등장하였다. 클라이언트 허니팟은 허니팟 클라이언트라고도 불리며, 물리머신인 것처럼 가장한 가상머신 상에서 웹 브라우저를 이용하여 웹 페이지를 능동적으로 방문하여 해당 페이지의 악성 여부를 탐지한다. 의심스러운 웹 페이지를 능동적으로 방문한다는 점에서 수동적으로 공격을 기다렸던 기존 허니팟과 차이가 있다[5]. 클라이언트 허니팟은 웹 페이지 분석뿐 아니라 ssh, ftp, smtp 등의 분석에도 적용이 가능하다.

클라이언트 허니팟은 서버가 클라이언트 허니팟 상에서 활용할 수 있는 기능적인 상호작용(functional interaction) 수준에 따라 고 상호작용(실제 시스템에 필적할 정도로 충분한 기능을 가짐)과 저 상호작용(전체적으로 실제 시스템의 일부 기능만 가지며 경량으로 구축할 때 유용)으로 분류된다. 최근 고 상호작용 클라이언트 허니팟 연구가 활발히 진행 중에 있다. 대표적으로, 워싱턴 대학의 Spycrawler[3], 뉴질랜드 빅토리아 대학의 Capture-HPC, 허니넷 프로젝트(Honeynet Project)[4], MS 허니몽키(HoneyMonkey)[6], MITRE의 허니클라이언트(HoneyClient)[7], Vrije University Amsterdam의 SHELIA 등이 있다.

2.2 가상머신

고 상호작용 클라이언트 허니팟은 보통 VM 기술을 사용하여 악성 웹 페이지 탐지를 위해 사용한다. VM은 침입탐지 기법에 사용되며 그 사용 형태에 따라 그림 1과 같이 두 가지로 구분된다[8].

첫째는 샌드박스 VM으로 VM 내부를 고립하여 웹이나 P2P등을 통해 다운로드한 프로그램 등 신뢰할 수 없는 어플리케이션을 실행하게 된다. 특정 어플리케이션 내부의 악성 코드가 실행되더라도 그 효력이 VM 내부로만 제한되어, 실제 컴퓨터는 안전하게 유지되어 중요 정보를 보호할 수 있다. 둘째는 시큐어드 가상머신(Secured VM)으로 VM 내부를 안전한 공간으로 설정하여 중요 정보를 보호하는 형태이다. 이 구조를 적용하면 업무 컴퓨터에 해킹 툴 혹은 악성 코드가 실행되더라도 시큐어드 VM 내의 정보는 보호받게 된다.

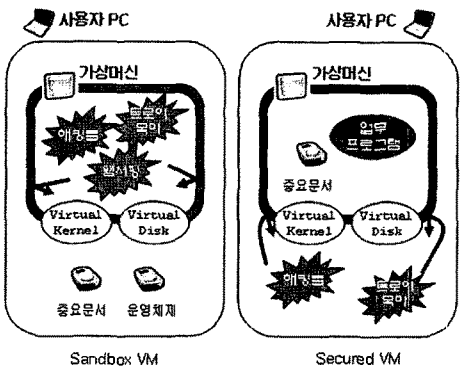

그림 1 가상머신의 사용 형태에 따른 분류

VM의 경우 물리시스템에서 구축되는 VM 이미지의 개수와 각 VM을 구성하는 자원(메모리와 하드디스크 크기 등)의 상황에 따라 성능이 크게 차이난다. 따라서 성능이 좋은 VM 환경을 구축하는 것이 필요하다.

본 연구진은 '악성 웹 페이지 탐지 및 필터링 시스템'을 통합한 프로토타입을 구축하여 발표한 적이 있다. '악성 웹 페이지 탐지 시스템'은 전용 머신에 설치되어 웹 페이지를 자동으로 수집하는 크롤링 시스템과 고 상호작용 클라이언트 허니팟으로 구성된다. 여기서 악성으로 탐지된 웹 페이지들을 블랙리스트로 관리한다. '필터링 시스템'은 사용자 PC에 설치되어 브라우저와 연동하며, 탐지 시스템이 관리하는 블랙리스트를 다운받아 블랙리스트 기반으로 악성 웹 페이지를 필터링한다.

본 논문에서는 악성 웹 페이지 탐지 시스템의 개선에 대한 연구로, 실행기반으로 의심스러운 웹 페이지들 분석할 때 어떤 VM 구성이 좋은 성능을 내는 지를 파악하고 또한 악성 웹 페이지들을 분류한다.

3. 고 상호작용 악성 웹 페이지 탐지 시스템

3.1 시스템 구성

악성 웹 페이지 탐지를 위해 본 논문에서는 뉴질랜드 빅토리아 대학의 Capture-HPC 엔진을 이용하였으며, 그 구조가 그림 2에 나타나 있다. Capture-HPC는 크게 Capture server와 Capture client로 나눌 수 있다.

Capture server는 방문하고자 하는 웹 페이지들의 리스트를 만든 후, VMware API를 호출하여 VMwear Server를 구동시킨다. VMware Server 상에 guest OS 이미지가 로드되고, 그 이미지 상에 Capture client들이 실행된다. Guest OS는 악성 URL 탐지를 위해 사용자 환경을 재연해 놓은 OS 이미지이다. Capture server로부터 전달받은 웹 페이지들을 실제 방문하기 위해 Capture client는 웹브라우저를 구동한다.

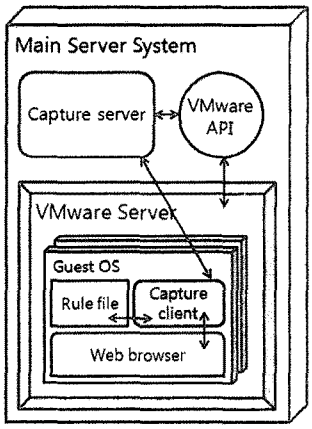

그림 2 Capture-HPC의 악성 URL탐지 시스템 구조

Capture client 및 GuestOS는 분석할 웹 페이지들을 방문하면서 또는 방문 후에 프로세스, 레지스트리, 파일 시스템 등의 상태변화를 모니터링하고, rule file에 정의되어 있는 규정에 따라 '악성' 혹은 '안전'상태를 판단하여, 진행상황이나 판단결과를 Capture server에게 전달한다. '악성' 판단이 나올 경우 Capture server는 VMware API를 이용하여 다른 웹 주소를 방문하기 위해 guest OS를 리버팅(reverting, 기존의 온전한 OS상태로 되돌림)한다. 왜냐하면 이전에 방문했던 웹 페이지에 의해 guest OS가 오염되어 정확하게 판단할 수 없는 경우가 존재하기 때문이다.

3.2 가상머신 기반의 악성 웹 페이지 탐지 시스템

고 상호작용 클라이언트 허니팟을 이용한 악성 웹 페이지 탐지를 위해 VM 기반으로 실행한다. 그 이유는 크게 두 가지이다. 첫 번째, 악성 웹 페이지가 실제 서버 시스템에서 실행된다면 서버를 감염시킬 것이며, 이로 인해 다음 웹 페이지를 분석하기 전에 실제 시스템은 재설치 되어야 한다. 즉, 실제 시스템으로 악성 URL 주소를 방문할 때마다 깨끗한 상태로 되돌리기 위해 재설치해야 하므로 많은 비용과 시간이 든다. 이에 반해 VM 기반으로 웹 페이지를 분석할 경우, 악성 코드가 유입되더라고 해당 VM과 guest OS만 다시 리버팅하면 된다.

두 번째, 운영체제/브라우저의 종류에 따라 실행이 되지 않는 악성 웹페이지 또한 존재한다. VM을 이용하면 하나의 시스템 안에서 여러 운영체제를 동시에 구동시킬 수 있으므로 비용과 시간을 단축시키며 여러 환경에서 다양하게 실험할 수 있다.

4. 실험 및 결과 분석

고 상호작용 클라이언트 허니팟은 새로운 악성 코드도 탐지할 수 있는데, 이를 위해 실시간으로 시스템 변화를 모니터링 해야 한다. 또한, 대부분의 웹 페이지는 계속 변경되거나 업데이트되므로 처음에 악성/안전 상태로 분석되었던 페이지가 나중에는 안전/악성 페이지로 바뀔 수 있기 때문에 주기적인 검사가 필요하다. 따라서 클라이언트 허니팟은 많은 컴퓨팅 시간과 자원을 필요로 하므로, 효율적인 탐지 방식이 필요하다.

시스템성능 분석을 위해 인텔 Core2Quard CPU, 8GB RAM, 500GB 7200rpm Serial-ATA HDD를 장착한 서버에 MS Server 2003을 구동시키고, Capture HPC 2.5.1-384 엔진을 설치하였다. 탐지용 클라이언트 시스템을 구동할 Guest OS로 VMware 서버1.0.6에 WindowsXP를 설치하여 사용자 PC를 재연하였다.

먼저, CaptureHPC 옵션을 이용하여 한 guestOS(이하, 이미지)마다 동시에 띄우는 웹 브라우저의 개수를 10개, 20개, 30개, 50개, 70개로 나누어 실험하였다. 이는 한 번에 악성 여부를 분석하는 URL 개수를 의미한다. 또한 이미지 개수(1개, 2개, 4개, 8개)와 각 이미지에 할당되는 메모리 크기(512MB, 1024MB, 3600MB)를 변화시키면서 리버팅 횟수, 시간 등을 측정하였다. 검사를 위한 url은 무작위로 수집하고, 카테고리별 분류를 위해 웹 검색방법을 이용하였으며, 이를 크롤링하였다.

4.1 웹 브라우저 개수에 따른 성능 분석

각 이미지마다 동시에 실행하는 웹 브라우저 수를 조절하면서 실험하였다(각 브라우저마다 하나의 페이지를 방문함). 20개로 설정된 디폴트 상태에서 40개의 페이지를 검사할 경우 악성 페이지가 없다면 2번의 실험으로 분석을 끝낼 수 있다. 이 경우 동시에 너무 많은 브라우저를 로드하면 자원이 고갈되어 오히려 더 나쁜 성능을 보일 수 있으므로 trade-off를 조절하는 것이 필요하다.

성능분석을 위해, 검색엔진, 쇼핑, 게임, 성인, 커뮤니티 등 다섯 검색어를 www.google.co.kr와 www.naver.

표 1 브라우저 수에 따른 시간 및 리버팅 횟수

(시간단위: sec)

웹 브라우저 / 성능측정	10개	20개	30개	50개	70개
초기 VM 생성시간	32.00	32.01	33.00	32.01	32.34
총 검사시간	11,559.95	9,685.51	7,285.51	6,224.53	6,344.52
페이지당 검사시간	21.99	18.27	13.84	11.78	11.86
리버팅 전체 시간	563.05	549.02	365.00	330.03	412.05
리버팅 1회 평균 시간	14.85	14,84	15,21	15.00	15.26
리버팅 횟수	38 회	37 회	24 회	22 회	27 회

com에 적용하여 페이지를 각각 100개씩 수집하였다. 수집된 500개의 url을 가지고, 동시에 실행되는 웹 브라우저의 수를 조절하면서 각각 2회씩 실험하여 그 평균을 구했으며, 그 결과가 표 1에 나타나 있다.

guestOS가 악성 웹 페이지 방문으로 오염되었을 경우에는 VMware 리버팅 기법을 이용하여, guestOS를 정상상태로 되돌린 후 검사를 계속해야 한다. 악성 웹 페이지가 없더라도 한번 검사 후에 다음 검사를 위해 리버팅을 해야 한다. 리버팅은 검사시간 외에 추가시간을 요구하게 되므로, VMware의 리버팅 횟수를 최소화하는 것이 필요하다 이를 위해, 검사에서 수행된 리버팅 횟수와 전체 리버팅 소요시간, 1회 리버팅의 평균 시간, 초기 VM 이미지 생성 시간을 측정하고 분석하였다(표 1 참조). 표를 보면, 동시에 실행되는 웹 브라우저 개수에 따라 리버팅 횟수가 변화함을 알 수 있다. 리버팅 1회 평균 시간은 대체로 약 15초 정도로 비슷했으며, 웹 브라우저 50개일 경우 리버팅 횟수가 22회로 가장 적은데, 이는 동시에 여러 페이지를 검사하기 때문에 그만큼 리버팅 횟수가 줄어들기 때문이다. 하지만 그 보다 더 많은 70개를 동시에 검사할 경우에는 서버 자원의 고갈로 웹 페이지 오류 등이 발생하고 리버팅이 증가하여 성능이 저하되는 역효과가 있다.

표 1의 초기 VM 생성시간이란, VMware가 실행되고 가장 처음 리버팅할 때의 시간이다. 이 시간은 리버팅 1회 평균시간 보다 긴데, 이유는 메모리 재사용으로 하드디스크에 저장되어 있는 스냅샷 된 이미지가 리버팅이 되면서 메모리에 적재되므로, 이때 많은 컴퓨팅 시간을 소모하기 때문이다. 하지만 2번 째 리버팅부터는 OS의 메모리 정책에 따라 하드디스크의 내용을 다시 메모리에 적재시키지 않고 재사용하기 때문에 적은 시간으로도 리버팅이 가능하다.

검사 시간의 경우, 한 이미지에 동시에 50개의 브라우저를 띄울 때 약 6,224초로 가장 좋은 성능을, 10개의 브라우저를 띄울 때 약 11,559초로 가장 나쁜 성능을 보였다. 리버팅 시간을 제외한, 한 페이지 당 평균 검사시간은 50개 브라우저 경우 약 11초였고, 가장 성능이 좋지 않은 10개 브라우저 경우 약 21초가 소요되었다.

이 실험을 통해, 탐지 성능이 리버팅 횟수와 동시에 띄우는 웹 브라우저 수와 밀접한 관계가 있음을 알 수 있었다. 주목해야 할 사항은 Capture HPC는 리버팅을 위해 divide and conquer 알고리즘을 이용한다는 점이다. 예로, 30개의 페이지가 동시에 분석될 때 악성 페이지가 하나 있다면, 그 페이지를 식별하기 위해 리버팅하면서 15개 → 8개 → 4개 → 2개 → 1개 순으로 분할하여 검사한다. 따라서 동시에 분석되는 페이지들 그룹이 크고 악성 페이지가 있다면 리버팅 횟수가 많아지고 분석 시간도 길어진다. 결과적으로 현재 시스템 자원에서 악성 웹 페이지 비율이 낮으면 각 이미지에서 동시에 수행하는 브라우저의 수를 50개 이상으로 하는 것이 좋고, 악성 웹 페이지 비율이 높으면 동시 수행하는 브라우저 수를 10개로 하는 것이 좋다. 실제로 검사에서 어떠한 웹 페이지가 악성일지 예상할 수 없으므로 주기적인 실험 분석에 따른 적정한 설정이 중요하다.

4.2 메모리 용량과 이미지 수에 따른 성능 분석

VM의 메모리 용량과 이미지 개수에 따른 성능을 분석하기 위해, 각 이미지에서 동시 수행하는 브라우저의 수를 디폴트인 20개로 하고, 무작위로 추출된 500개의 웹 페이지들을 대상으로 실험하였다.

500개의 웹 페이지를 분석한 실험 결과가 표 2에 나타나 있다. 512MB 메모리 용량의 한 guestOS 상에서 분석했을 때 약 2시간이 소요되었고, 각각 512MB 메모리 용량을 가진 8개의 guestOS에서 그 웹 페이지들을 나눠서 분석했을 때 약 3시간이 소요됐다. 멀티태스킹으로 성능이 좋아질 것으로 추측되었던 다중 이미지의 경우에 오히려 더 수행 속도가 느렸다. 그 이유는, 다중 이미지를 사용할 경우에 guestOS의 스냅샷이 존재하는 하드디스크에서의 병목현상으로 리버팅에서 많은 시간을 소모하기 때문이다.

하나의 guestOS 이미지 상에서 3600MB는 VM에 할당할 수 있는 최고의 RAM 크기 이므로, 이미지가 여럿일 경우에는, 전체 3600MB 내에서 각 이미지마다 RAM

표 2 메모리 용량과 이미지 수에 따른 소요시간

RAM / #Image	512MB	1024MB	3600MB
1	122분	107분	108분
2	153분	131분	-
4	168분	122분	-
8	190분	-	-

을 균등 할당하였다. 실험에서 각 이미지에 1024MB 이상을 할당하더라도 더 이상의 성능 향상이 없었다. 이는 검사대상 url 수를 50개로 줄여서 실험할 때도 마찬가지였으며, RAM의 크기는 1024MB가 적절하였다.

4.3 악성 웹 페이지 분석·분류

국내 웹 페이지를 성인 사이트, 와레즈 사이트, 커뮤니티 사이트, 쇼핑 사이트, 기업 사이트, 게임 사이트 카테고리로 나누어 분석하였다. 이 분류는 중국 Peking 대학의 사이트 분류[9]를 참조하고 국내에서 유행하는 쇼핑 사이트 카테고리를 추가하여 결정되었다. 웹 페이지 수집을 위해 검색엔진에 6개의 키워드로 각각 50개의 대표 URL(seed URL)을 적용하였다. 6개의 분류된 사이트에서 각각 4,800개의 파생 페이지를 수작업을 통해 얻었다. 총 28,800개(=6×4,800개)의 웹페이지의 악성 여부를 판단한 결과 표 3에서와 같은 통계를 얻을 수 있었고, 전체 약 27시간 50분이 소요됐다.

실험 결과, 전체 28,800개의 URL 중 119개, 즉 0.41% 비율로 악성으로 의심되는 웹 페이지가 있었다. 이 중에서 커뮤니티 사이트와 기업 사이트가 0.95%, 0.87% 순서로 높은 비율을 보였고, 와레즈 사이트의 경우 악성 페이지가 발견되지 않았다. 커뮤니티, 기업 사이트에서 공격 행위가 탐지되는 이유를 분석한 결과, 게시판이 주요 공격대상 임을 파악할 수 있었다. 따라서 사용자들은 가능하면 인증된 사이트의 게시판을 이용하고, 게시판 운영자들은 수시로 악성 코드를 유포하는 게시물이 있는지 확인하는 것이 필요하다. 반면 악성 페이지가 거의 발견되지 않은 성인 사이트, 와레즈 사이트, 게임 사이트는 주로 사용자가 다운로드 받는 컨텐츠 위주로 이루어져 있기 때문에 drive-by download 공격이 아닌 다른 형태의 악성 코드 유포가 있을 것으로 보인다.

표 3 분석 대상 URL들의 분류

사이트 분류	Seed URL수	파생 페이지 수	악성 페이지 수	비율
성인	50 개	4800 개	1 개	0.02%
와레즈	50 개	4800 개	0 개	0%
커뮤니티	50 개	4800 개	46 개	0.95%
쇼핑	50 개	4800 개	28 개	0.58%
기업	50 개	4800 개	42 개	0.87%
게임	50 개	4800 개	2 개	0.04%
계	300 개	28800 개	119 개	0.41%

5. 결론 및 향후 방향

본 논문에서는 drive-by download 공격과 같은 클라이언트 측 공격을 사전에 탐지하고, 피해를 줄이는 방법으로 고 상호작용 클라이언트 허니팟을 이용한 악성 웹 페이지 탐지 시스템을 구축하고 실험하였다. 그 결과, 검사 대상 중에 악성 페이지의 숫자가 적을수록 동시에 많은 웹 페이지를 검사하는 것이 효율적이었다. 28,800개의 웹 페이지에서 119개의 페이지가 악성으로 판단되었으며, 커뮤니티 사이트와 기업 사이트, 쇼핑 사이트의 게시판에서 drive-by download 공격이 빈번히 발생함을 확인할 수 있었다.

guestOS(가상머신) 이미지 수와 RAM 크기에 따른 전체 수행 시간을 측정하여 비교한 결과 1024MB 이상의 단일 가상머신 이미지에서 가장 좋은 성능을 보였다.

향후 탐지된 악성 페이지의 공격 유형과 분류를 좀 더 구체화 하고 리버팅 횟수를 줄이는 방법에 대한 연구를 진행할 계획이다.

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] Yi-Min, et. al., "Strider HoneyMonkeys: Active, Client-Side Honeypots for Finding Malicious Web-sites," *Appear in IEEE Transactions on Computers.*

[6] 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.

[7] Kathy Wang, "Using Honeyclients for Detection an Response Against New Attacks," *MITRE.*

[8] VMCraft web site 가상 머신의 구성 및 분류 http://www.vmcraft.com/technology/vm.vm

[9] J. Zhuge, T. Holz, J. Guo, X. Han, and W. Zou, "Studying Malicious Websites and the Underground Economy on the Chinese Web," *Proc. of the 2008 Workshop on the Economics of Information Security*, June 2008.