HTTP Header 정보의 변조를 통한 악성코드 분석과 대응방안

임원규* · 허건일** · 박원형** · 국광호**

요 약

최근 새로운 유형의 악성코드 발생이 꾸준히 증가하고 있으며 점점 지능화, 고도화되면서 그 형태 또한 다양한 형태로 변화하고 있다. 정보화산업의 발달로 정보의 경제적, 금전적 가치가 높아지면서 정보유출 악성코드로 인한 그 피해 또한 점점 더 증가하고 있다. 본 논문은 HTTP Header 정보 중 User-Agent의 일반적인 사용기법에 대해 알아본다. 또한, User-Agent 정보의 변조를 통한 다양한 악성코드 제작기법을 연구하고 이에 대한 기술적·정책적 대응방안을 제안한다.

Malware Analysis and Policy Counterplan Against a Transformation of HTTP Header Information

Won Gyu Lim* · Geon Il Heo** · Won Hyung Park** · Kwang-Ho Kook**

ABSTRACT

Nowadays, the occurrence of Malware is steadily increasing. The Malware is also becoming more intelligent, advanced and changing into various types. With the development of the information industry, the economic and monetary value of the information is going up and the damage due to the leaked information by the Malware is also increasing. This paper investigates the general usage of the User-Agent in the HTTP Header, studies the Malware production techniques by transformation of the User-Agent information and suggests the technical and political counterplan against them.

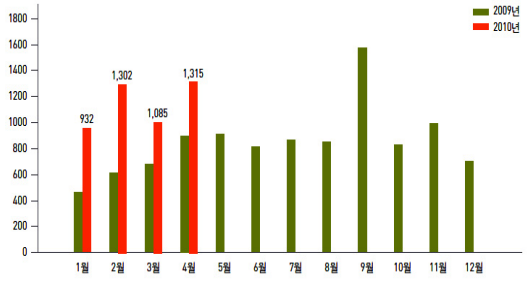
Key words : User-Agent, Malware, Snort

접수일 : 2010년 5월 21일; 채택일 : 2010년 6월 22일

* 서울산업대학교 IT정책대학원 산업정보시스템전공
** 서울산업대학교 산업정보시스템공학과

1. 서 론

최근 악성코드의 발생은 꾸준히 증가하고 있으며 점점 지능화, 고도화되면서 그 형태 또한 다양한 형태로 변화하고 있다. 인터넷과 정보기술의 발달로 전자상거래, 전자정부, 원격교육, 원격진료 등 주요 사회경제 활동이 정보통신 기술을 기반으로 급격히 변화하고 있어 정보의 경제적, 금전적 가치가 높아지고 있다. 이에 따라 새로운 악성코드로 인한 그 피해 또한 점점 더 증가하고 있다. 아래 (그림 1)은 지속 증가하고 있는 우리나라의 웜·바이러스 현황을 보여주고 있다.



(그림 1) 월별 웜·바이러스 신고 건수[1]

본 논문에서는 HTTP Header의 User-Agent 변조를 통한 새로운 악성코드 기법을 알아보고 이에 따른 기술적·정책적 대응방안에 대해 연구하고자 한다.

2. 관련 연구

2.1 HTTP Header

HTTP(HyperText Transfer Protocol)는 클라이언트와 서버 사이에 이루어지는 요청/응답(request/response) 프로토콜이다. HTTP는 Request Message와 Response Message 두 개의 메시지로 구분할 수 있다. HTTP의 기본동작은 웹브라우저가 웹서버로 Request Message를 보내면, 웹 서버는 웹 브라우저에게 받은 Request Message에 대한 Response Message를 보낸다. 메시지는 스타트라인, 헤더영역, 메시지바디로 구성되어 있다. 메시지의 Header는 크게 General Headers와 Request Headers로 구분된다. 다음 <표 1>는 General Headers에 관한 표이다.

〈표 1〉 General Header[1]

header	function/example
Connection	접속옵션을 지정
	Connection : Keep Alive
Date	메시지가 작성된 시간
	Date : Tue, 15 Nov 1994 08 : 12 : 31 GMT

아래 <표 2>는 Request Headers에 관한 표이다.

〈표 2〉 Request Headers[2]

header	function/example
Accept	응답으로 허용하는 데이터타입
	Accept : */*
Accept-Language	응답으로 허용하는 언어
	Accept-Language : ko
Accept-Encoding	응답으로 허용하는 encoding type
	Accept-Enconding : gzip, deflate
Host	요구하는 자료를 가지고 있는 host와 port
	Host : www.snut.ac.kr
User-Agent	웹 브라우저 정보
	User-Agent : Mozila/4.0(compatible : MSIE 6.0; Windows NT 5.1

HTTP Header 중 User-Agent 옵션은 사용한 웹 브라우저가 어떤 것인지를 웹 서버에게 알려주는 역할을 한다.

위의 옵션값을 통해 웹 서버는 현재 요청을 하고

〈표 3〉 User-Agent 사례

User-Agent : Mozilla/4.0(compatible; MSIE 8.0; Windows NT 5.1; .NET CLR 2.0.50727; .NET CLR 1.1.4322)

있는 클라이언트가 운영체제는 XP를 사용하고 있으며 .NET 1.x와 .NET 2.x가 설치되어 있는 PC에서 MS 익스플로러 8.0을 사용하여 요청을 하고 있다는 정보를 알 수 있다. 이를 통해 웹 서버측에서는 User-Agent별 특성에 따른 서로 다른 서비스를 제공할 수 있다.

2.2 검색로봇 방지

로봇배제표준(Robots Exclusion)은 검색로봇이 정보 수집을 위해 사이트를 방문할 경우를 대비하여, 검색로봇에 대한 웹 사이트의 디렉토리 및 파일들에 대한 검색조건을 명시해 놓은 국제규약이다[3].

최근 들어 웹 검색엔진의 성능이 과거에 비해 월등히 향상되면서 구글 등 일부 검색엔진에서는 HTML로 작성된 웹 페이지의 내용은 물론 웹 사이트에 올려놓은 PDF나 DOC같은 문서파일 내용까지도 검색이 가능해지고 있어 검색엔진에서는 적절한 검색어만으로 기업 및 개인의 정보를 유출할 수 있다. 특히 동창회・동아리・학교 홈페이지 등은 주소록이나 회원근황 파일을 자료실 등에 올려놓은 경우가 많아 검색엔진을 통한 개인정보 유출 위험도가 증가하고 있다. 이에 따라 홈페이지 운영자들의 기술적인 대비책 마련이 필요하다. 일차적인 조치로는 '인터넷 검색엔진 배제표준(Robots Exclusion Protocol)' 즉, 로봇배제표준을 적용한다.

정상적인 검색로봇은 사이트 방문시 가장 먼저 웹사이트의 최상위 주소에 있는 로봇배제표준파일(robots.txt)을 확인하고 파일에 명시된 디렉터리와 파일에 대한 검색 조건 및 로봇의 배제사항을 준수하면서 웹 사이트의 컨텐츠를 수집한다.

robots.txt 파일의 내용은 크게 2가지 부분으로 나뉜다. 로봇의 이름을 적는 부분(User-Agent), 방문을 허용하지 않을 디렉터리를 적는 부분(Disallow)으로 구분된다.

〈표 4〉 robots.txt 문법[4]

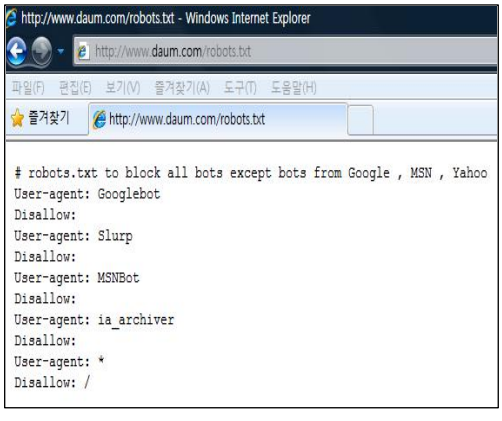
User-Agent	* 은 모든 User-Agent 이고, 특정 User-Agent를 지정할 경우, 봇 이름을 지정
Allow or Disallow	Disallow에서는 모든 하위 문서에 대해 배제를 의미하고, 아무것도 지정하지 않으면 허가를 의미

각 검색엔진 마다 다른 검색봇의 이름을 가지고 있어 각 검색봇의 이름을 User-Agent에 명시함으로써 검색봇의 접근을 제한할 수 있다.

〈표 5〉 검색봇의 User-Agent

검색엔진	검색봇의 이름
네이버	cowbot
구글	Googlebot
MSN	MSNBot
야후	Slurp
엠파스	empas

아래(그림 2)를 통하여 다음(DAUM) 사이트는 Slurp, MSNBot, ia-archiver봇의 접근만을 허용하고 있는 것을 확인 할 수 있다.



(그림 2) 다음(Daum) 사이트의 robots.txt

3. HTTP Header 정보를 이용한 악성코드

3.1 User-Agent 변조를 통한 악성코드 유포

웹 서버측에서 User-Agent별로 다른 서비스를 제공할 수 있다는 점을 이용한 새로운 악성코드 발생이 가능하다.

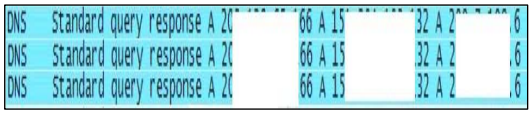
새로운 악성코드는 PC의 User-Agent부분을 특정한 User-Agent로 변조한다. 웹 서버는 User-Agent가 변조된 클라이언트를 구분하여 클라이언트의 정보를 가져온다.

해킹 메일이나 악성코드 유포 사이트에서 악성코드에 감염된 후 다음과 같은 증상이 발생한다.

〈표 6〉 User-Agent 변조 패킷

```
GET/Util/editplus2/aa.txt HTTP/1.1
If-Modified-Since : Sun, 04 Apr 2010 04 : 27 : 33 GMT
If-None-Mat ch : W/"90868ca1 : 1bed"
User-Agent : RookIE/1.0
Host : www.sansarang.net/Util/editplus2/aa.txt
```

User-Agent가 RookIE/1.0으로 변경된 것을 미상의 해커가 악성코드에 감염된 PC만을 구별하기 위해 변조 한다. 그리고 특정 브라우저 정보를 변경하여 감염된 PC이외 접근을 제안하는 효과가 있다. 이후 다운받은 악성코드가 실행되면 임의의 서비스와 프로세스가 등록되고 등록된 서비스와 프로세스 및 파일은 사용자 프로그램으로부터 은닉된다. 또한 미상의 해커가 운용 중인 관리프로그램의 위치를 참조하기 위해 특정도메인에 주기적으로 접근한다.



(그림 3) 악성도메인 주기적 접속

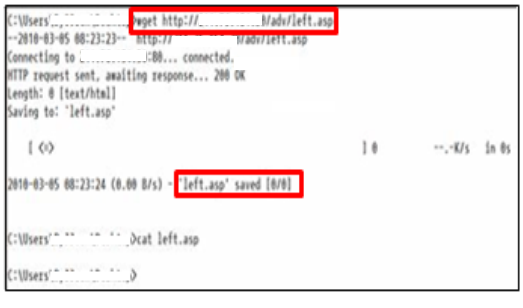
그 외 시스템의 모든 자원을 조작하기 위한 사용자 제어, 레지스트리 및 파일 조작, 키보드 입력 기록 및 개인 방화벽을 우회하기 위한 코드 삽입 기법을 포함한다. 또한, 봇넷으로 악용되기도 하며 이를 통해 DDoS(분산 서비스거부) 공격이 발생하기도 한다.

3.2 User-Agent 변조를 통한 악성코드 난독화

앞서 User-Agent 변조를 통해 악성코드에 감염된 PC만이 악성코드를 다운로드 받을 수 있는 기법을 설명하였다.

다음 기술은 User-Agent를 이용해 특정 브라우저가 접근 했을 때 악성코드를 보여주는 기법이다. 웹 서버 측에서는 특정 웹 브라우저의 취약점을 이용한 공격을 위해 특정 웹 브라우저의 User-Agent 값으로 변조하여 취약점을 가진 웹 브라우저에서만 악성코드를 확인할 수 있도록 하는 기술이다.

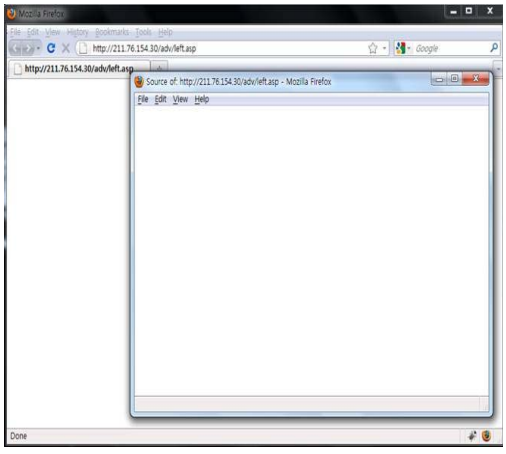
MS06-014 취약점[5]을 이용하는 악성코드를 전파하는 left.asp 파일의 악성코드를 확인하려 하였다. 우선 wget을 이용하여 파일을 다운로드하면 다음(그림 4)와 같다.



(그림 4) wget을 이용한 다운로드

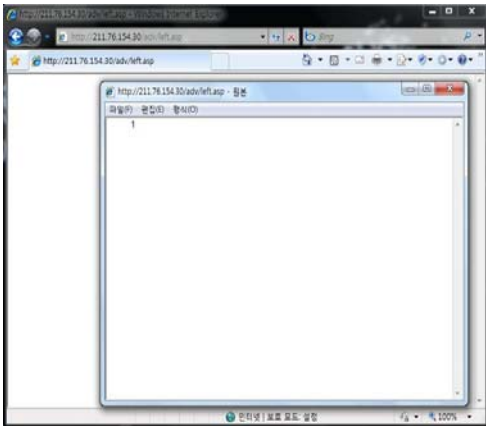
위의 그림으로 left.asp 파일을 다운 받지 못한 것을 확인할 수 있다.

다음 방법으로 파이어폭스를 이용해 접속해 보면 다음(그림 5)와 같이 아무런 내용을 볼 수 없는 것을 확인할 수 있다.



(그림 5) 파이어폭스를 이용한 접속화면

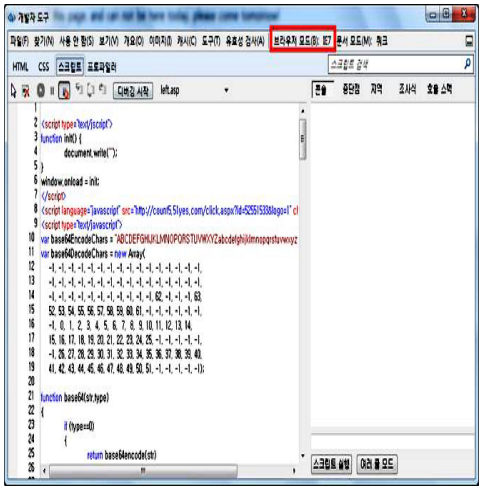
마지막으로 MS 익스플로러 8을 이용해서 확인하여도 동일한 결과를 얻었다.



(그림 6) MS 익스플로러 8을 이용한 접속시

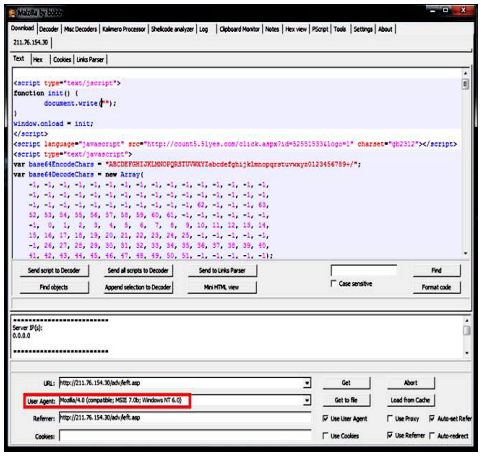
이것으로 특정 브라우저에서만 내용을 확인할 수 있을 것으로 추측할 수 있고, MS 익스플로러 8의 개발자 도구를 이용하여 MS 익스플로러 7환경으로 확인하였다. 개발자 도구는 MS 익스플로러 8 버전에만 있는 도구로 단축키 F12를 누르면 확인할 수 있다.

MS 익스플로러 7로 변경한 후 (그림 7)과 같은 스크립트를 확인할 수 있었다.



(그림 7) MS 익스플로러 변경 후

위의 내용으로 보아 User-Agent를 확인하여 스크립트의 노출 여부를 결정하는 것으로 확인 되었다. left.asp 파일로 유포되는 악성코드는 MS 익스플로러 6 또는 7버전의 취약점을 이용한 것으로 추측할 수 있다.



(그림 8) 파이어폭스에서 User-Agent 변경

이와 같은 방법으로 해커는 특정 웹브라우저에서만 스크립트를 나타나게하는 방법을 이용하여 분석의 어려움을 더하기도 한다.

이런 방법으로 유포되는 악성코드를 방지하기 위해서는 파이어폭스를 이용하여 User-Agent 값을 변경하는 방법이 있을 것이다. 변경 방법은 아래의(그림 8)과 같다.

4. 대응방안

4.1 탐지패턴 제작을 통한 예방

미상의 해커에 의해 감염된 시스템을 탐지하기 위해서는 여러 가지 방안을 고려하여 탐지규칙을 제작하여야 하며, 탐지규칙 작성시 가장 효과적인 방법으로 현재 가장 보편적으로 쓰고 있는 스노트(Snort)[6]를 기반으로 탐지규칙 생성 및 탐지방법을 기술한다[7, 8]

〈표 7〉 User-Agent 변조 탐지 패턴

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS content : "User-Agent\: RookIE?";
```

새로운 악성코드를 탐지하기 위해서는 감염된 PC가 특정 도메인으로 User-Agent가 RookIE으로 아웃바운드로 접속시도할 때 탐지할 수 있는 탐지패턴을 제작하였다.

(그림 9) 실제 악성코드 감염 탐지 패킷

보안관제 시스템(IDS)에서 탐지패턴을 주입하여 악성코드 감염 패킷을 정확히 탐지하였다. User-Agent가 변조된 악성코드 감염 PC는 100% 탐지가 가능하였다.

4.2 HTTP Header 정보 인증

앞서 알아본 HTTP Header 옵션에서 User-Agent는 서버측에 브라우저 정보를 제공해 주는 역할만 하고 있다. 이는 User-Agent 옵션이 현재 일련의 정보만을 제공하는 역할을 할 뿐 특별한 기능을 제공하고 있지 못하다. 그리고 미상의 해커가 User-Agent의해 변조를 할 때 아무런 인증절차 없이 변조된다는 것을 확인하였다. 이는 HTTP Header 정보는 아무 런 인증권한 없이 변조가 가능하다는 의미이다. 그래서 User-Agent 옵션 변경시 최소 관리자권한에 의해서 변경이 가능하도록 보안절차를 강화해야 새로운 악성코드에 대응할 수 있다.

5. 결 론

본 논문은 User-Agent를 변조를 통한 새로운 악성코드 기법에 대한 정책적 대응방안에 관한 연구이다. 이를 통해 정보보호담당자들은 감염된 피해시스템에 대한 정확한 분석 및 대응방안을 모색해야 한다. 이는 최근 미상의 해커들이 우리나라의 중요정보를 절취하는 것은 물론이고 사이버테러를 위한 신규 악성코드를 제작하고 있으며 앞서 알아본 기법 이외에도 수많은 변종 악성코드가 국내로 유입되고 있는 실정이다. 앞으로 신규 악성코드를 예방하기 위해서는 개인의 보안의식은 물론 정보보호 담당자들의 침해사고 대응 능력을 강화해야 하며, 국내 바이러스 백신 업체를 통해 신종 악성코드에 대한 보안 업데이트를 지속 갱신해야 한다.

향후 User-Agent 변조를 통한 악의적인 시나리오 등 여러가지 대응방안에 대해 좀 더 연구해야 할 것이다.

참 고 문 헌

[1] 인터넷 침해사고 동향 및 분석 월보, www.krcert.or.kr, 2010. 4.
[2] 웹브라우저를 통한 장비 제어, www.sollae.co.kr, 2003.
[3] 로봇배제표준, 구글로봇배제 적용 사용 설명서, 한국인터넷진흥원, 2008. 10.
[4] 홈페이지 개인정보 노출원인과 대응방법, 한국인터넷진흥원, 2007. 6.
[5] http://microsoft.com/technet/security/bulletin/ms06-014.mspx, Microsoft Security Bulletin MS06-014.
[6] 박원형, 정보유출 악성코드 분석을 통한 개선된 탐지 규칙 제작 연구, 정보보안논문지, 2008. 12.
[7] 조명휘, 정보보호 시스템 평가를 위한 NP 기반 네트워크 공격패킷 생성모델, 한남대학교, 2005.
[8] 박남열, 행위 기반의 패턴생성을 통한 알려지지 않은 악성코드 탐지기법, 전남대학교, 2006.

임 원 규

2008년 서울산업대학교 산업정보시스템공학과 (학사)
현재 서울산업대학교 IT정책대학원 석사과정 (산업정보시스템전공)

허 건 일

2004년 서울산업대학교 산업정보시스템공학과 입학
현재 서울산업대학교 산업정보시스템공학과 네트워크보안 Lab 연구원

박 원 형

2002년 서울산업대학교 산업정보시스템공학과 (공학사)
2005년 서울산업대학교 정보산업공학과(공학석사)
2009년 경기대학교 정보보호학과 이학박사(정보보호전공)
현재 서울산업대학교 산업정보 시스템공학과 겸임교수

국 광 호

1979년 서울대학교 공과대학 (공학사)
1981년 서울대학교 대학원 (공학석사)
1984년 청주대학교 산업공학과 전임강사
1989년 Georgia Institute of Technology, U.S.A (공학박사)
1993년 한국전자통신연구원 선임연구원
현재 서울산업대학교 산업정보시스템공학과 교수