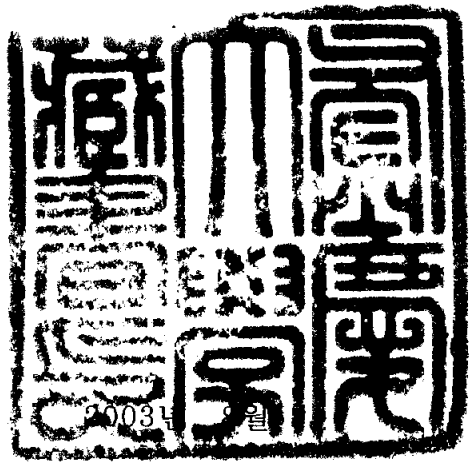
공학석사 학위논문

유해사이트 차단시스템의 설계 및 구현

지도교수 여 정 모

이 논문을 공학석사 학위논문으로 제출함



부경대학교 산업대학원

전 산 정 보 학 과

황 명 국

황명국의 공학석사 학위논문을 인준함

2003년 6월 21일

주 심 공학박사 박 만 곤 (인)

위 원 공학박사 김 영 봉 (인)

위 원 공학박사 여 정 모 (인)

< 목 차 >

< 표 차 례 >

< 그림차례 >

An implementation and design of the blocking system against harmful sites

Myung-Gook Hwang

Dept. of Computer and Information,
Graduate School of Industry,
PuKyung National Univesity

Abstract

Dysfunction of information that follow in popularization of internet is risen to social problem and problem by share of indiscreet information is serious level too. Space of imagination called internet can contact to objectionable material without limit if user wants and if the target is teenagers, gravity of damage becomes worse. Effort to check access of harmful site is consisting at each field each floor and various method to intercept access of harmful site by link of effort was presented. This treatise discusses about layout of harmful site blocking system and implementation way to propose blocking module that blocking harmful site in driver level of NIC (Network Interface Card) belonging to data link class of network, and construct web server that manage this module.

1. 서론

컴퓨터와 초고속 통신망 회선의 보급에 따라 급속도로 인터넷 환경이 성장하였지만 인터넷상의 음란, 폭력, 사이버도박, 불온사상 등의 유해한 정보가 청소년들에게 아무런 선별 없이 제공됨으로써 부모들의 우려와 사회적 문제를 야기하고 있다. 인터넷상의 유해정보로부터 청소년들을 효과적으로 보호하기 위한 기술적인 대책이 시급히 마련되어야 한다.

현재 개발되어 있는 유해사이트 차단 프로그램은 어플리케이션 계층에서 구현되어 구동되는 방식이 주류를 이루고 있으며 차단 방법으로 도메인을 이용하는 방식과 현재 시행중인 내용 등급제[3]를 이용하는 방식이 있다.

도메인을 이용하는 경우 사이트의 도메인을 DB로 구축 목록을 만들어 이용하며 차단 방법은 차단 목록만을 이용하거나 허용 목록만을 이용하는 방식을 이용하고 있다.

차단 목록만을 이용한 방식은 인터넷의 접근이 자유롭지만 유해 사이트에 쉽게 노출되며 차단 목록의 주기적인 갱신 문제가 해결되어야 한다.

허용목록만을 이용하는 경우 유해 사이트에 노출되지는 않지만 인터넷의 접근이 제한적이다.

내용등급제는 사이트의 특정부분에 컨텐츠의 등급과 관련된 표기양식을 기술하여 브라우저에서 정의된 등급의 분류 정책에 따라 필터링 하는 방식이다.

본 논문에서 제안하는 방식은 도메인을 이용하여 유해사이트를 차단하는 방식으로 차단 목록과 허용 목록을 동시에 이용하며 유해 사이트 차단 모듈을 관리하는 별도의 서버와 지속적인 통신을 통하여 목록의 갱신이 용이하며 정책적인 부분이 유해사이트 차단모듈을 관리하는 서버에 적용되어 내용등급제와 같은 방법도 지원할 수 있는 유연성을 가지고 있다.

그리고 유해 사이트 차단 모듈은 드라이버 수준에서 구현되어 사용자에게는 투명한 동작 환경을 제공하게 되며 목록을 별도로 관리하지 않아도 된다.

2. 관련연구

본 장에서는 유해사이트의 현황과 차단 시스템의 개발 동향 그리고 기존의 차단 시스템의 구현 방법에 관하여 설명한다.

유해사이트 차단 방법은 응용프로그램들이 인터넷 접속 프로그램과 운영체제 사이에서 동작하며 사용자의 인터넷 접속을 감시한다. 이런 응용프로그램들은 인터넷 접속 프로그램의 종류에 따라 영향을 받는다.

유해사이트에 대한 기술적인 차단방법을 차단유형에 따라 분류하면 차단 방법에 의한 구현방법과 차단 위치에 의한 구현방법 두 가지로 분류할 수 있다.

2.1 유해사이트 현황

유엔교육과학문화기구(UNESCO)가 파악한 1999년말 인터넷 상 정보 중 약 10%가 음란 및 폭력물이며, 음란·폭력 사이트는 10만개 이상인 것으로 추정하고있다.

표 1의 결과[1]는 인터넷을 통하여 한 번이라도 유해사이트에 접촉을 한 경험이 있는 사람의 비율을 나타내고 있다.

표 1. 유해사이트 접촉 여부

	2000년		2001년	
	사례수	비율(%)	사례수	비율(%)
있다	2,967	81.0	1,603	80.1
없다	696	19.0	397	19.9
{전 체}	3,663	100.0	2,000	100.0

유해사이트의 접촉경로를 조사한 그림 1의 결과[1]와 같이 유해사이트의 접촉경로로 인터넷의 이용 비율이 80~90%에 달한다.

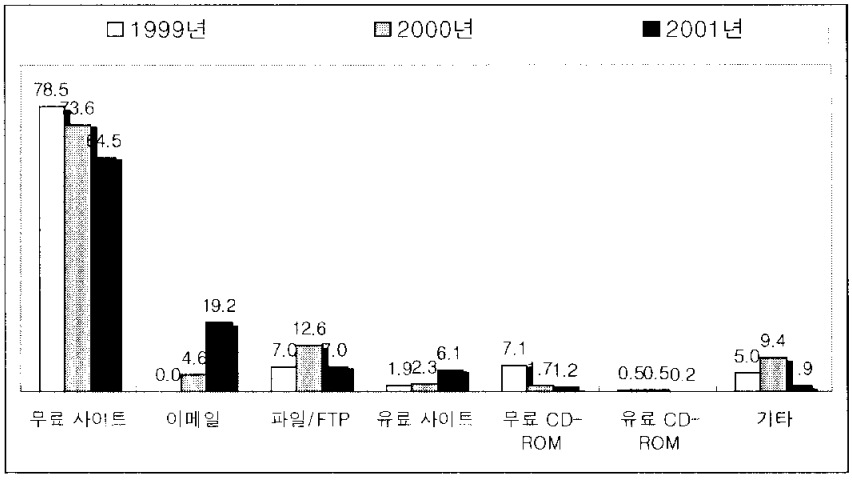


그림 1. 연도별 유해사이트 접촉 경로

그림 2는 컴퓨터의 활용 정도에 따라 유해사이트의 접근 경

로[1]를 나타내고 있으며 결과는 인터넷을 검색할 수 있는 정도의 수준을 가진 사용자도 쉽게 인터넷에서 유해사이트를 접근한다는 것을 알 수 있다.

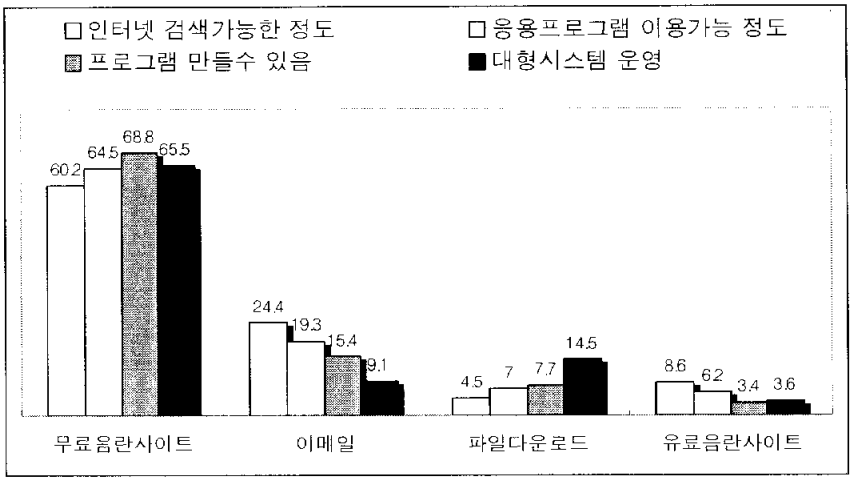


그림 2. 인터넷 활용정도에 따른 유해사이트 접촉 경로

2.2 기존의 유해사이트 차단 시스템 현황

단일 PC에서 수행되는 개인용 유해정보차단 제품은 1995년 7월에 사이버 패트롤을 시작으로 일반에게 판매되기 시작하였으며 현재는 미국 마이크로 시스템즈 소프트웨어사의 사이버 패트롤, Surfwatch, CYBERsitter, Net Nanny, 한국전산원이 개발한 NCApatrol 1.5가 있으며, PLUS TECH에서 개발한 수호천사3.0, 수호천사 배포판, 한국정보공학에서 개발한 안티엑

스, 아이탑에서 개발한 파로스 등이 있다.

이러한 소프트웨어들은 유해정보 주소목록에 포함된 유해한 사이트를 차단하고, 부모가 사용중에 추가로 찾은 유해사이트를 주소목록에 입력하여 차단시킬 수 있으며, 또한 차단하고 싶은 단어를 입력하여, 이러한 단어가 포함된 유해정보를 차단할 수도 있다.

그리고 사이버 패트롤은 요일별로 인터넷 사용 가능 시간대를 설정하거나 최대 인터넷 사용시간을 설정하여 어린이의 인터넷 사용시간을 제한할 수 있으며 자녀들의 인터넷 사용내역을 기록하여 부모가 볼 수 있도록 해 줌으로써 자녀들의 건전한 인터넷 활용을 유도할 수 있다.

네트워크용 유해정보차단 S/W는 인터넷 접속 관문에서 주소목록에 포함되어있는 유해사이트를 차단한다.

기존의 클라이언트용 소프트웨어를 서버용으로 확장한 제품과 인터넷 프락시 소프트웨어에 유해정보 차단 기능을 추가한 제품으로 분류된다. 미국 마이크로시스템즈 소프트웨어사의 Cyber Patrol Proxy와 Secure Computing 사의 Web Track, 한국전산원의 NCApatrol Proxy 1.0 등이 대표적인 네트워크용 유해정보차단 소프트웨어이다.

표 2는 기존의 차단 시스템을 정리 하였다.

표 2. 기존의 국내외 차단 시스템

국가	개발사	제품명	특징
미국	마이크로시스템즈	사이버 패트롤	· 개인용 · 차단목록 · 단어선별 차단기능
미국	마이크로시스템즈	Surfwarch	· 개인용 · 차단목록 · 단어선별 차단기능
미국	마이크로시스템즈	CYBERsitter	· 개인용 · 차단목록 · 단어선별 차단기능
미국	마이크로시스템즈	Net Nanny	· 개인용 · 차단목록 · 단어선별 차단기능
미국	마이크로시스템즈	사이버 패트롤 프록시	· 네트워크용
미국	Secure Computing	Web Track	· 네트워크용
한국	한국전산원	NCApatrol	· 개인용 · 차단목록 · 이용시간 통제
한국	한국전산원	NCApatrol proxy	· 네트워크용
한국	플러스기술	수호천사	· 개인용 · 차단목록 · 이용시간 통제
한국	인터정보	컴지기	· 개인용 · 차단목록
한국	이엠테크놀리지	웹그린	· 개인용 · 차단목록
한국	에이엠테크	아이키퍼	· 개인용 · 차단목록
한국	스마트시스템	SecureDesk	· 개인용 · 차단목록 · IC 카드를 이용하여 연령별 통제
한국	아이탑	파로스	· 개인용 · 차단목록
한국	인터피아월드	지키미	· 개인용 · 차단목록
한국	한국정보공학	안티엑스	· 개인용 · 차단목록
한국	넷피아닷컴	WebClean21	· 개인용 · 내용등급

2.3 기존의 유해사이트 차단 시스템의 구현방법

유해사이트의 차단 방법에따라 분류하면 차단목록에 의한 선별(Black List Filtering), 허용목록에 의한 선별(White List Filtering), 내용등급에 의한 선별(Neutral Label Filtering)이 있다.

유해사이트를 차단하는 위치에따라 분류하면 네트워크 상의 관문에서 차단하는 방법과 클라이언트인 개인용 PC에서 차단하는 방법이 있다.

2.3.1 차단 방법에 따른 분류

그림 3은 차단 소프트웨어가 차단 목록이나 허용 목록을 이용하여 클라이언트의 접근을 제어하는 방법의 개요이다.

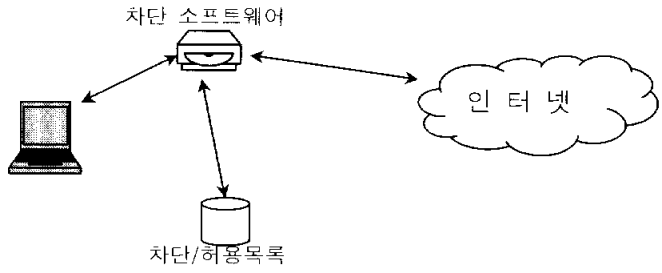


그림 3. 차단/허용 목록에 의한 차단 방법

1) 차단목록에 따른 차단

유해한 정보를 제공한다고 판단되는 사이트만을 차단한다. 이 방법은 인터넷에서 제공되는 대부분의 정보에 접근이 가능하여 유해 사이트에 노출될 수 있다. 이를 보완하기 위해 수신 정보의 키워드 및 구문 검색을 통해 유해정보를 포함하고있는 사이트를 차단하는 방법을 함께 사용한다.

이 방법의 단점은 차단 목록의 주기적인 갱신이 어렵고, 포괄적 기준 설정으로 인한 무차별 차단의 가능성이 있다. 또한 차단 목록 제공자에 의한 실질적인 검열권 행사의 우려가 있다. 현재 상용 차단 소프트웨어가 대부분 이 방법에 근간을 두고 있다.

2) 허용목록에 따른 차단

정보의 내용이 건전하다고 검증되어 허용목록에 등록된 사이트에 대해서만 접근을 허용하고 이외의 사이트는 모두 차단하는 방식이다.

이 방안은 유해정보에 노출될 위험은 거의 없으나 접근할 수 있는 사이트가 극히 제한적인 것이 단점이다. 일반적으로 허용목록을 관리하는 관리자는(부모 또는 선생님 등) 지정된 패스워드를 입력함으로써 등록되지 않은 새로운 사이트로의 접근이 가능하도록 하고 있다. 이 방법은 학교 등의 특수 환경

에서 사용하기에 적합하다.

3) 내용등급에 따른 차단

일정 기준에 의해 정의된 내용등급에 따라 차단하는 방식이다. 인터넷정보에 유해정도를 나타내는 등급 값을 삽입하여, 인터넷 사용자가 인터넷정보에 접근할 때 웹브라우저나 차단 소프트웨어에서 인터넷 정보에 삽입되어 있는 등급 값이 관리자가 미리 해당 사용자에게 설정해 놓은 허용 등급 값보다 크지 않을 때만 접근하도록 하는 방법이다.

웹브라우저나 차단 소프트웨어에서 등급에대한 정보를 가지고 있으며 허용목록에 의한 방법과 마찬가지로 관리자의 관리를 필요로 한다. 그러나 차단·허용 목록을 보유할 필요가 없으며 연령에 따라 차별적인 등급값을 적용할 수 있다. 그러나 웹서비스를 하는 해당 사이트에 의해 등급이 표기되므로 등급값을 표기 하지 않았을 경우 효력이 없다.

그림 4.[3]는 내용등급에 의한 차단 방법의 개요를 보여준다.

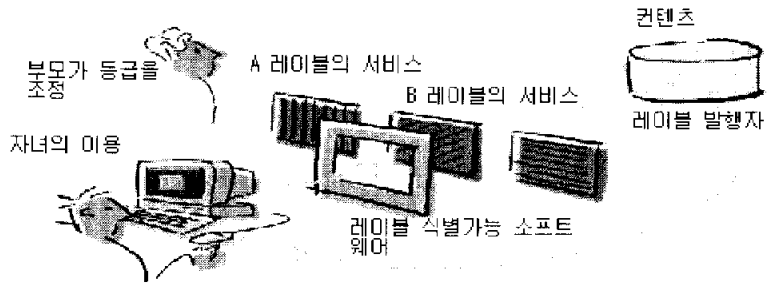


그림 4. 내용등급에 의한 차단 방법

2.3.2 차단위치에 따른 분류

1) 네트워크 상에서의 차단

라우터나 방화벽이 설치되어 있는 학교나 회사 등에서 사용될 수 있는 방법이다. 그림 3은 라우터 또는 프락시 서버를 통하여 운영되는 네트워크의 개요를 나타 내고 있다.

방화벽이나 별도의 서버에만 차단소프트웨어를 설치, 운영하므로 관리가 용이하며, 차단 소프트웨어가 클라이언트 시스템과는 별도로 존재하므로 소프트웨어 삭제 등의 위험으로부터 안전하다.

방화벽을 이용할 경우 로컬네트워크의 모든 호스트들이 방화벽이 설치된 서버를 통하여 인터넷에 접근하도록 하는 구조를 가지며 유해 사이트 차단 소프트웨어는 방화벽과 일체형이거나 방화벽과 다른 소프트웨어로 존재하게된다. 따라서 유지 및 관리는 쉬우나 서버시스템에 부하가 생기면 성능의 저하가

생길 가능성이 존재한다.

그리고 라우터를 이용하는 경우 라우터가 이해하는 것은 패킷의 IP(Inter- net Protocol)가 존재하는 네트워크 계층이다. 그러므로 라우터에서는 목적지 주소를 가지고 차단하고, 클라이언트용 웹브라우저에서 라우터 외부의 프락시(Proxy)를 설정해 놓고 사용하면, 라우터 측면에서는 목적지주소가 프락시 주소로 되어 있으므로 연결사이트가 유해사이트일지라도 차단할 수 없다.

다른 방법은 네트워크상에 유해정보 차단을 위한 전용서버를 연결하고 네트워크의 패킷을 모니터링하여 차단하거나 프락시 서버와 같이 특정서비스(예 : HTTP)에 대하여 서비스 중계를 하는 과정에서 유해정보를 차단할 수 있다. 이러한 방법도 단지 사이트 주소 목록에 의해서만 선별하기 때문에 내용등급 등을 이용한 탄력적인 선별은 불가능한 단점이 있다.

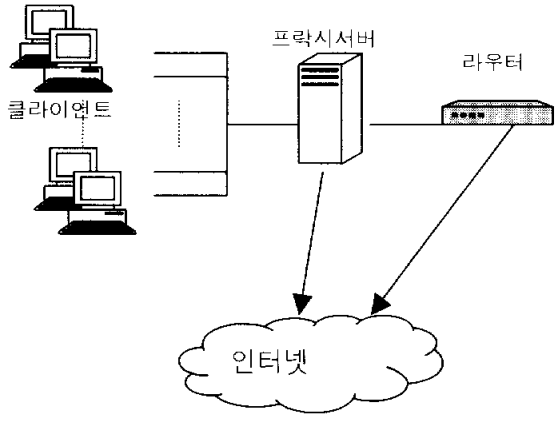


그림 5. 네트워크상에서의 차단 방법

2) 클라이언트상에서의 차단

성능 면에서 부하가 거의 없으나 모든 개인용 PC마다 차단 소프트웨어를 설치해야 하며, 컴퓨터에 초보자인 부모일지라도 자녀를 관리할 수 있어야 하고, 컴퓨터를 잘 다루는 어린이인 경우 부모 몰래 차단 소프트웨어에 설치된 안전장치를 무효화 시키는 방법[2]을 찾아낼 수 도 있으므로 안전성이 떨어진다.

3. 유해사이트 차단 시스템의 설계

본 장에서는 유해사이트 차단 시스템의 개요와 설계에 대하여 논의할 것이다. 유해사이트 차단 시스템의 설계는 2장에서 기술한 바와 같이 여러 가지 방법에서 어느 한가지 방법만으로 유해사이트를 차단하는 것은 어렵다는 점에 착안하여 허용목록과 차단목록을 같이 사용하도록 설계하였다.

유해사이트 차단 시스템의 클라이언트측 차단 시스템은 NIC(Network Interface Card) 드라이버의 모듈로 구현되어 드라이버의 일부처럼 동작을 한다. 따라서 사용자 브라우저의 형태 및 구성에 영향을 받지 않는다.

그리고 드라이버는 사용자 트래픽에 대하여 최소한의 간섭을 하도록 설계되어 있다. 즉 사용자가 웹에 접근하는 트래픽만을 실시간 감시의 대상으로 한다. 따라서 드라이버에서는 패킷을 분석할수 있는 능력이 요구되며 송수신시의 패킷을 분석하여 원하는 패킷의 ACCEPT 또는 DROP 여부를 결정한다.

3.1 개요

본 논문에서 구현한 모듈은 마이크로소프트사의 윈도우즈 운영체제를 기반으로 하며 ANSI. C로 구현하였다. 드라이버는 마이크로소프트사의 윈도우즈 운영체제상에서 동작되는 커널모드 드라이버인 미니포트(miniport)[4] 드라이버이다.

미니포트 드라이버의 특징은 하드웨어를 드라이버 제작자가 직접 제어할 수 있다는 것이다. 따라서 NIC와 직접적으로 연결되어 NIC를 제어할 수 있으며 표준 하드웨어 이외에도 개발자가 정의하는 하드웨어를 사용할 수 있다.

본 논문에서는 별도의 플래쉬 메모리를 NIC에 장착한 하드웨어를 사용한다고 가정 하였다. 플래쉬 메모리는 사이트의 목록을 저장하는 용도로 사용될 것이다. 그림 6은 전체 시스템의 개요를 설명하고 있다.

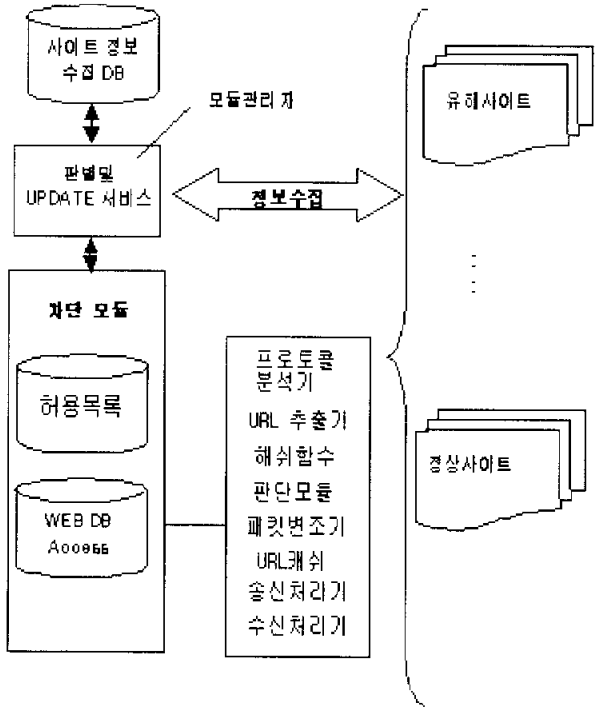


그림 6. 전체 시스템의 개요

유해사이트 차단 시스템은 차단 모듈에서는 허용목록을 가

지고 있으며 모듈 매니저에서는 차단 목록을 DB로 구축하여 사용한다.

3.2 차단 모듈의 구성 요소

본 논문에서 제안하고자 하는 유해사이트 차단시스템은 클라이언트에서 구동되는 드라이버와 드라이버의 모듈, 드라이버의 모듈에서 질의시 적절한 응답을 되돌려주는 판별서버로 구성되며 검색엔진과 그에 따르는 서버의 설계와 구현에 관한 내용은 포함하고 있지 않다. 그러나 완성된 차단 시스템에서는 검색엔진 및 DB서버가 반듯이 필요하다.

그림 7은 클라이언트에 장착되는 차단 모듈의 구성도 이다.

3.2.1 허용목록

NIC에 장착된 플래쉬 메모리에 저장되는 목록이다. 도메인이 저장되어야 하지만 길이가 도메인마다 다르므로 해쉬함수를 이용하여 16바이트의 고정된 길이로 만들어 저장하였다. 표 3.에서 플래쉬 메모리에 구성되는 허용 목록의 구성을 나타 내었다.

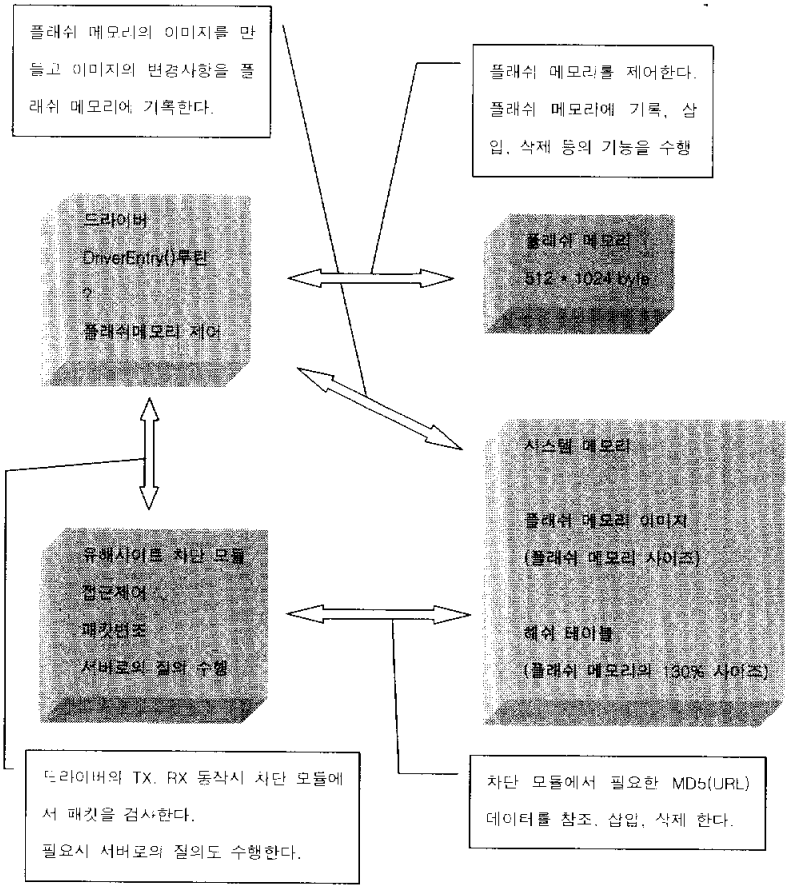


그림 7. 클라이언트에서의 차단 시스템 개요

표 3. 플래쉬 메모리내의 허용목록 구조

index	Hashed 도메인
0	AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
10	55 0A 4F 44 0C 3B 36 66 80 38 06 04 86 6A 1A 7B
20	05 2B 40 53 17 79 7A 59 3B 69 35 3F 0C 0A 05 28
:	:
:	:

플래쉬 메모리의 크기는 512kb의 크기를 가지며 그림 2에서 보이듯이 드라이버에서 플래쉬 메모리를 제어한다. 허용목록을 위해서 다음과 같은 사항들을 고려하여 설계하였다.

1) 허용목록의 갱신 방법

판별 서버에서는 허용목록과 차단목록을 동시에 유지하고 있다. 차단목록은 로봇 기반의 검색엔진을 통하여 목록을 업데이트 할 수 있으며 허용목록의 갱신은 사용자가 접근하려는 사이트를 분석하여 갱신이 이루어지도록한다.

사용자가 접근하려는 사이트의 분석은 사용자가 사이트에 접근시 플래쉬 메모리에 존재하지 않는 사이트일 경우 서버로 질의를 하게 되어 있음으로 질의 데이터에 사용자가 접근하고자 하는 사이트의 도메인을 데이터로 하여 서버에 보내 주면 되는 것이다. 이것은 클라이언트 시스템이 판별 서버로의 질의시 반듯이 필요한 데이터이기 때문에 가능하다. 플래쉬 메모리에 들어가는 내용은 허용목록이며 이러한 사이트는 차단목록과 마찬 가지로 수시로 사라지거나 생긴다. 게다가 사용자별로 방문하는 사이트도 제 각각일 것이다.

따라서 이러한 허용목록은 사용자가 요구하는 사이트를 플래쉬 메모리에 기록하는 것이 사용자를 위해서는 가장 바람직한 방법일 것이다. 그러나 모든 사용자의 플래쉬 메모리를 다르게 셋팅 할 수는 없다. 따라서 사용자가 자주 접속하는 사이

트를 플래쉬 메모리에 기록하도록 한다.

드라이버에서 판별 서버로 질의한 사이트 중에서 건전 사이트로 분류되어 있는 사이트에 대한 질의를 하였을 경우 응답 패킷에서 해당 사이트는 건전함으로 플래쉬 메모리에 기록하도록 응답한다.

만약 사이트가 분류되어 있지 않다면 서버에서는 질의 데이터중에 포함된 도메인스트링을 DB에 저장한 다음 전문 검색 요원의 판단아래 판단 결과를 DB에 저장 할 수 있다. 그리고 다음번 같은 사이트에 대한 질의가 오면 서버에서는 드라이버에 해당 사이트의 유해 여부에 따라 드라이버에 기록 여부를 알려준다.

따라서 플래쉬 메모리에 기록을 하게 되는 시점은 서버에서 유지하는 건전 사이트의 목록에 기록되어 있으며 사용자가 접근을 시도하는 사이트가 이러한 건전 목록에 존재하는 경우일 것이다.

2) 클라이언트에서의 허용목록 유지

클라이언트에서 허용목록은 NIC에 장착된 플래쉬 메모리에 기록되어 유지 된다. 따라서 허용목록의 유지는 플래쉬 메모리를 어떻게 관리하는가의 문제로 볼 수 있다.

플래쉬 메모리의 관리 기법은 단순하다. 실시간으로 플래쉬 메모리에 접근하여 쓰고 지우는 것은 상당한 딜레이 타임을

요구하며 보다 빠른 응답을 요구하는 사용자에 대하여 비효율적인 방법이다.

따라서 특정한 시간에 플래쉬 메모리를 쓰거나 지워야만 한다. 이 시점은 사용자가 컴퓨터를 사용하는데 불편을 느끼지 않는 시간이어야 하며 유해사이트 차단 시스템에서도 부담이 없는 시점을 이용 하여야 한다. 이러한 시점을 분류하면 다음의 표 4와 같다.

표 4. 플래쉬 메모리의 업데이트 시점

시 점	장 점	단 점
컴퓨터의 부팅시	쓰기 동작시 오동작 문제가 없음	플래쉬 메모리에 기록할 내용을 하드디스크에 기록하여 보관하여야 함
컴퓨터의 종료시	플래쉬 메모리에 기록할 내용을 하드디스크에 보관할 필요가 없음	쓰기 동작시 오동작이 일어남

이외에도 사용자가 컴퓨터를 사용중인 상황에서 플래쉬 메모리를 업데이트 할 수 있으나 실시간으로 플래쉬 메모리를 기록할 경우 딜레이 타임이 길어진다면 시스템 전체가 정지한 상태가 되어 사용자는 OS의 잘못된 동작으로 시스템이 비정

상인 것으로 오인할 수 있다. 따라서 컴퓨터의 종료시에 플래쉬 메모리의 기록과 삭제를 수행하는 것이 가장 효과적이다.

3) 사용자의 조작 가능성

허용목록은 사용자가 조작을 할 수 없도록 하여야 한다. 사용자가 허용목록을 조작할 수 있다면 차단 효과를 무력화 시킬수 있다.

3.2.2 WEB DB Access

질의를 할 수 있는 서버의 정보를 가지고 있다. 즉 사용자가 접근하려는 웹사이트가 클라이언트 차단 시스템의 허용목록에 존재하지 않을 경우 구축된 판별 서버로 질의하여 질의 결과에 따라 접근을 제어 하도록 하는 것이다. 이때 구축된 판별 서버의 DB는 허용목록과 차단목록을 함께 유지한다. 서버측의 시스템을 구축하기 위해 다음과 같은 사항들을 고려하여 설계하였다.

1) 차단목록의 갱신 방법

차단목록의 갱신은 주기적으로 웹을 검색하여 유해정보를 포함하는 사이트의 도메인을 DB에 갱신하는 방법을 사용한다.

따라서 효율적으로 웹의 정보를 검색할 수 있는 로봇 기반의 검색엔진을 제작하여 사용한다.

2) 클라이언트 시스템의 관리 및 제어

클라이언트 시스템은 사용자가 사용하고 있는 드라이버와 NIC상에 장착되어 있는 플래쉬 메모리를 지칭한다. 상술한 바와 같이 플래쉬 메모리의 제어는 드라이버에서 하지만 내용의 삽입, 삭제는 판별 서버에서 전적으로 책임을 진다. 드라이버는 판별 서버의 결정에 따라서 플래쉬의 내용을 삽입 또는 삭제한다.

3.2.3 도메인 추출기

HTTP 헤더[9]에서 도메인 부분을 추출하는 모듈이다.

이부분은 패킷 필터링이라 불리는 기법을 사용한다. 패킷 필터링은 어디서, 무엇을 필터링 하는지에 따라 수행하는 기능이 다르다고 할 수 있다. TCP/IP헤더의 내용을 분석하여 이미 정해진 룰에 따라 처리하는 방식이 있으며 또 다른 방식으로는 데이터 부분의 내용을 필터링하여 내용에 따라 정해진 룰을 적용시키는 방식이 있다.

내용에 기반한 필터링 방법은 정보의 흐름을 제한하기 위해 패킷의 데이터를 필터링 하는 방법으로 본 논문에서 제안하는

유해사이트 차단 시스템의 유해사이트 차단 모듈도 이 방법을 사용한다.

내용에 기반한 필터링 기술은 응용계층에서 생성되는 프로토콜을 이해하고 있어야 하며 낮은 계층의 프로토콜을 이해하여 필터링하는 방법보다는 상대적으로 구현이 어려우며 성능도 떨어진다.

유해사이트 차단 시스템의 필터링은 이런 내용에 기반한 필터링을 실시하되 모든 컨텐츠를 검사하는 방식이 아니라 HTTP request 페이지만을 검사하여 사용자의 목적지를 알아낸다.

3.2.4 해쉬함수

추출된 도메인을 16바이트의 해쉬 값으로 만든다. 사용 알고리즘은 MD5[11]이다.

드라이버에서 추출하는 도메인 데이터는 스트링으로 되어있으며 이는 플래쉬에 저장되는 허용목록의 리스트도 스트링으로 저장될 것을 요구한다. 그러나 도메인 데이터는 가변길이로 최대 255자까지의(255byte) 길이를 가질수 있다. 이러한 데이터를 그대로 사용한다는 것은 제한된 크기의 플래쉬 메모리를 사용하는데 있어 너무 비효율적인 방법이다. 게다가 스트링 데이터의 탐색과 유지 관리는 더욱 어렵다.

따라서 일정한 길이의 데이터로 만들 필요가 있으며 스트링

보다는 수치화 시키는 것이 탐색과 유지 관리등의 작업을 용이하게 만들 수 있다. 이러한 요구 조건에 따라 도메인을 MD5라는 해쉬 함수를 이용하여 128bit (16byte) 길이의 정량화된 수치로 대신한다.

그러므로 플래쉬 메모리에 저장되는 사이트의 도메인은 MD5로 해쉬화 되어 고정된 길이를 갖는 수치값으로 저장된다. 다음은 도메인을 MD5함수를 이용하여 수치화 시킨 예를 보여 주고 있다.

예) http://kr.yahoo.com -> MD5() ->
55 0A 4F 44 0C 3B 36 66 80 38 06 04 86 6A 1A 7B

위의 예와 같이 유해사이트 차단 시스템에서 사용되는 모든 도메인들을 MD5()를 이용 수치화 하여 플래쉬 메모리 또는 판별 서버의 DB에 저장을 하며 드라이버에서 HTTP request 패킷의 목적지 도메인을 추출하면 허용목록과 비교하기 위하여 해쉬화 한다.

3.2.5 프로토콜 분석기

송·수신되는 패킷들의 프로토콜[5]-[9]을 분석한다.

프로토콜의 분석은 클라이언트 차단 시스템의 기능 확장을 고려하여 설계된 부분이다.

프로토콜의 분석은 NIC가 이더넷에 사용되는 카드를 사용한

다라는 전제하에 데이터링크계층보다 상위 계층에서 사용되는 프로토콜들을 이해하고 있다는 것을 의미한다. 즉 데이터링크 계층에서 이더넷 헤더를 제외한 데이터를 분석하여 어떠한 프로토콜이 사용되었는지를 분석할 수 있다.

본 논문에서 제안하는 클라이언트 차단 시스템은 데이터링크 계층의 이더넷 프로토콜 외에 국내에서 널리 사용되고 있는 ADSL(Asymmetric Digital Subscriber Line)의 PPPover Ethernet 프로토콜, IP, TCP, UDP 그리고 HTTP를 지원하도록 설계되었다.

3.2.6 판단 모듈

사용자가 접근하려는 사이트의 접근 허용 여부를 판단한다. 판단은 추출된 도메인의 해쉬를 취한 값이 플래쉬 메모리에 기록되어 있는 허용목록에 있는지 검색하여 판단한다. 만약 사용자가 접근을 시도하는 사이트가 허용목록에 존재하지 않는다면 클라이언트 시스템은 서버로 질의하여 질의결과 데이터에 따라 접근을 제어할 것이다.

3.2.7 패킷 변조기

판단 모듈에서 모듈 매니저 서버로의 질의를 요구한다면 HTTP request 패킷을 모듈 매니저 서버로 전송되도록 변조하

여 질의에 필요한 데이터를 패킷에 삽입한다. 패킷의 변조는 IP spoofing 이라 불리는 기법을 참조하여 본 논문에서 제안한 시스템에 맞게 수정하여 사용 하였다.

3.2.8 도메인 캐쉬

차단 모듈에서의 판단 결과가 유해사이트 이면 허용 목록을 저장하는 플래쉬 메모리에 기록을 할 수 없으므로 임시메모리에 보관하여 사용자가 유해사이트에 재접속시 서버로의 질의 횟수를 줄인다. 그리고 모듈 매니저 서버측의 차단 목록에는 있더라도 사이트가 유해사이트인지의 여부가 정확히 판단되지 않았을 경우 역시 임시 메모리에 기록한다.

3.3 모듈 매니저 서버

서버측의 구성은 차단 모듈을 관리하고 차단 목록의 지속적인 업데이트가 가능하도록 설계한다.

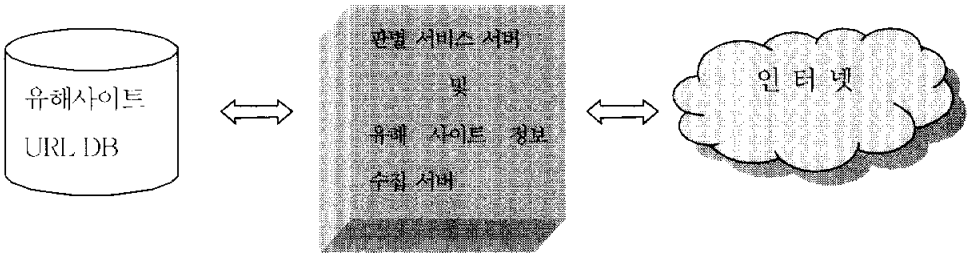


그림 8. 클라이언트 관리 및 유해사이트 정보 수집 서버의 개요

3.3.1 모듈 매니저

유해사이트 차단 시스템의 차단 모듈에서의 질의를 처리하거나 차단 모듈의 허용목록 업데이트와 서버측 차단 목록의 업데이트를 수행한다.

차단 목록의 수집은 검색 로봇을 이용하여 수집할 수 있다. 모듈 매니저의 역할은 차단 모듈을 관리하는 정책을 결정하는 것이다. 정책의 내용은 차단 모듈의 허용 목록 업데이트 또는 제거 방법, 차단 목록의 업데이트 또는 제거 방법, 차단 모듈에 적용할 카테고리(유해사이트의 분류, 예를 들어 폭력, 도박, 음란 등)의 범위 결정 등의 정책을 수립하여 모듈 매니저에 적용하여야 한다. 모듈 매니저의 차단 목록 수집에 의해 주기적으로 업데이트 된다.

4. 유해사이트 차단 시스템의 구현

본 논문에서 사용되는 차단 방법은 허용목록을 사용하여 차단하는 방법을 적용하였으며 모듈관리자에 의해 허용목록이 업데이트 되도록 설계되었다.

송·수신되는 패킷들은 차단모듈에서 감지해야 하는 HTTP 패킷이거나 모듈 매니저의 응답이거나 차단 모듈에서 처리가 필요하지 않은 패킷일수도 있다. 그러므로 패킷의 프로토콜을 분석하여 적절한 처리를 하여야 한다.

차단 모듈은 클라이언트에서 동작하도록 구현하되 어플리케이션이 아닌 드라이버 수준에서 구현하여 차단 시스템이 무력화되는 것을 보완하였다.

그리고 본 논문에서 구현한 도메인 추출기는 현재 널리 사용되고 있는 웹 프락시(Proxy) 서버를 통한 트래픽도 제어가 가능하도록 설계가 되었다.

웹 프락시를 통한 트래픽은 HTTP request부터 시작하는 것이 아니라 웹 프락시에 목적지 사이트의 도메인을 질의어나 데이터로 전송하는 방식을 취한다. 따라서 첫 번째 HTTP request 이후의 HTTP 패킷의 헤더를 지속적으로 분석하여야 한다는 단점이 있다.

4.1 차단 모듈의 송·수신 동작

그림 9에서 차단 모듈의 판단 모듈에서의 판별결과가 모듈 매니저 서버로의 질의를 요구하고 있으며 사용자의 HTTP request 패킷을 UDP(User Datagram Protocol)패킷으로 변조하여 판단에 필요한 데이터를 삽입하여 서버로 보내게 된다.

UDP을 사용하며 패킷이 유실 될 수 도 있으나 HTTP request 패킷을 보낸다는 의미는 사용자의 PC와 사용자가 접근하려는 웹 서버사이에 TCP의 three-way hand shake가 성공적으로 수행되었다는 의미이기도 하다.[8]

따라서 운영체제의 TCP(Transmission Control Protocol)/IP(Internet Protocol) 스택에서는 드라이버에 정상적인 HTTP request 패킷이 전달된 것으로 여기게 된다.

그러므로 모듈 매니저 서버에 보낸 UDP 패킷이 유실되어도 운영체제의 TCP/IP 스택의 재전송 메커니즘을 이용할 수 있게 된다. 그리고 UDP[10]의 속도는 TCP에 비해 빠르다. 그림 4의 (1)은 판단 모듈의 판별 결과가 허용사이트로 판단된 결과이며 HTTP request 패킷을 정상적으로 처리하도록 한다.

그림 10은 차단 모듈의 수신시 동작을 설명한다. 수신시 처리가 필요한 부분은 모듈 매니저로부터의 응답 패킷만을 처리한다.

본 논문에서 제안한 유해사이트 차단 시스템의 차단 모듈은 HTTP request 패킷을 대상으로 하여 차단하고 있어 유해 사이트일 경우 HTTP request 패킷은 유해 사이트에 도달하지

못한다.

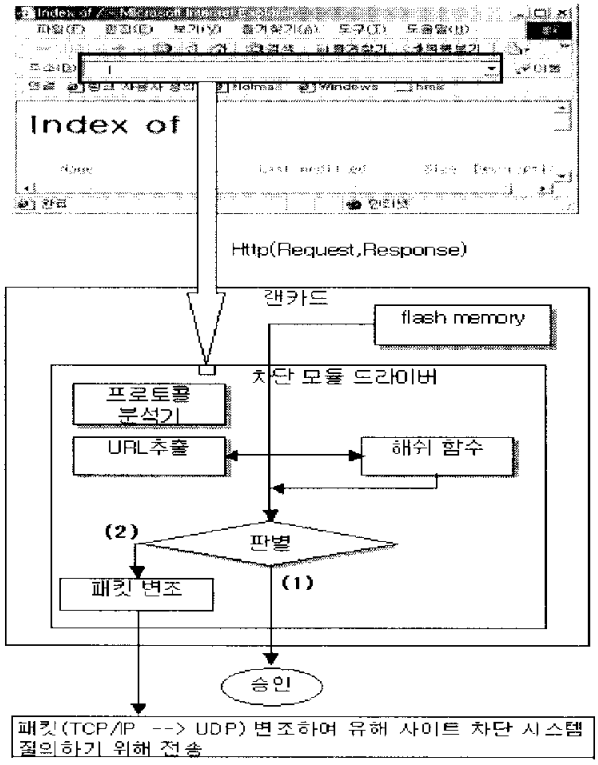


그림 9. 차단 모듈의 송신시 동작과정

따라서 HTTP request 패킷을 변조하여 모듈 매니저로의 질의를 수행하게 되며 질의에 대한 모듈 매니저의 응답 UDP 패킷을 수신 과정에서 처리한다.

즉 수신되는 HTTP 패킷이나 다른 용도의 패킷은 처리를 하지 않는 것이다.

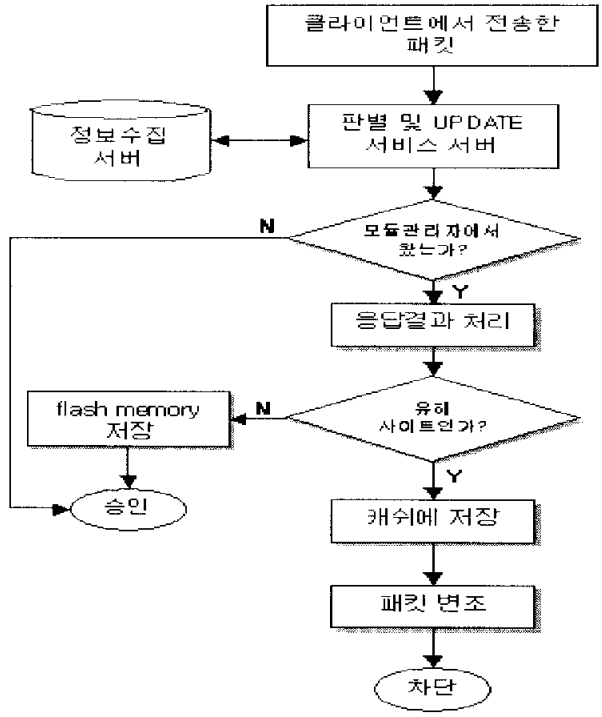


그림 10. 차단 모듈의 수신시 동작과정

모듈 매니저가 보낸 응답 패킷을 받았을 경우 그림 10과 같은 처리를 하게 되며 응답 패킷의 판단결과 데이터가 유해 사이트인 경우 응답 UDP패킷을 사용자가 접근하려던 웹사이트의 응답 HTTP redirection 패킷으로 변조하여 TCP/IP 스택에게 세션의 응답으로 보내주고 임시 메모리에 도메인의 해쉬값을 기록한다. 응답 패킷의 판단결과 데이터가 허용 사이트인 경우 해당 사이트의 도메인의 해쉬값을 플래쉬 메모리에 기록하게 된다.

4.2 패킷 필터링

유해 사이트 차단 모듈의 필터링 기능은 내용에 기반한 필터링 방식을 사용하고 있다. 다음 그림 6은 필터링 절차를 보여준다.

그림 9에서와 같이 핵심적인 내용은 HTTP 헤더의 내용을 검사하는 부분이다.

HTTP request 헤더의 내용중 필요한 부분은 도메인이 존재하는 필드 이다. 따라서 존재하는 도메인을 필터링하여 NIC의 플래쉬 메모리에 저장되어 있는 허용목록 사이트와 비교를 하여 허용 여부를 결정한다.

이러한 정보의 필터링을 위해서는 HTTP를 이해하고 있어야 한다. 그리고 헤더의 내용을 검색하기 위해서는 응용계층에서처럼 문자열 기반으로 검색을 하여야 하므로 특정 주소나 포트만을 필터링하는 방식 보다 상당히 효율이 떨어진다.

커널 기반의 소프트웨어인 드라이버에서는 응용계층에서 지원하는 함수들을 이용할 수 없으며 따라서 문자열을 검색하는 기능을 구현하여야 한다.

드라이버에서는 이런 일련의 패킷들을 다시 역캡슐화를 이용하여 사용자의 메시지를 알아내어 적절한 처리를 수행하게 되는 것이다.

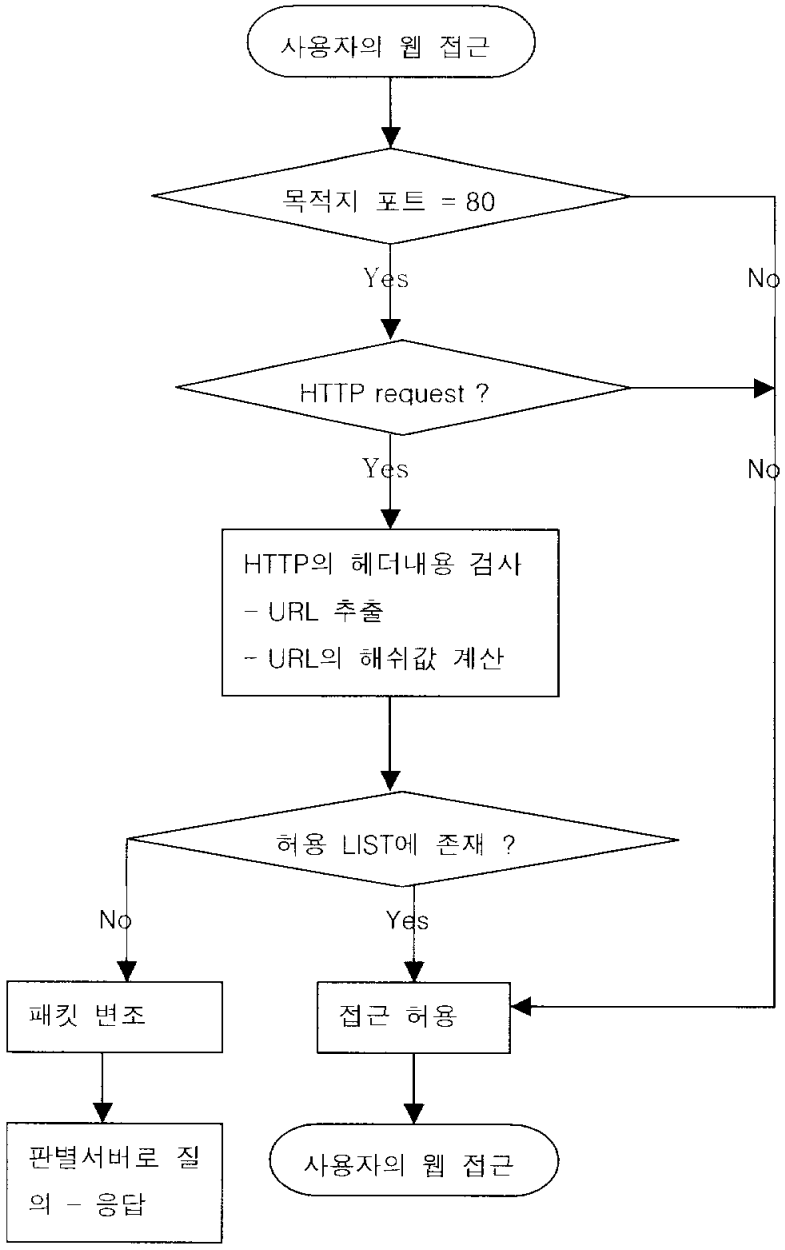


그림 11. 차단 모듈에서의 패킷 필터링

4.3 패킷 변조

패킷 변조는 의미 그대로 패킷의 내용을 수정하는 것이다. 패킷의 변조를 이해하기 위해서는 TCP/IP 프로토콜[5][8]의 동작에 대한 지식이 필요하다.

사용자가 접속하려는 사이트가 사용자 컴퓨터에 보관되어 있는 허용목록의 리스트(NIC의 플래쉬 메모리에 저장 되어 있다.)에 존재하지 않으면 세션의 방향을 바꾸어 본 논문에서 구현한 모듈 매니저 서버로 질의를 할 필요가 있다.

이런 사용자의 질의에 모듈 매니저 서버에서 사용자가 접근하려는 사이트의 유해 여부를 판별하여 응답을 보낸다. 응답의 결과가 유해 사이트라면 사용자의 연결을 차단 페이지로 리다이렉션을 하게 되고 건전 사이트라면 접근을 허용하는 것이다.

이렇게 사용자의 세션을 이용하여 패킷을 변조하여 내보냄으로 사용자와OS는 정상적인 접근이이루어지고 있다고 생각하게 된다. 따라서 TCP의 흐름제어를 방해하지 않으면서도 목적을 달성할 수 있는 것이다.

한가지 유의 할 점은 서버에서는 패킷의 리다이렉션에 관련되는 정보를 알 수가 없다. 리다이렉션에 관련된 정보를 사용자의 시스템 메모리에 보관할 수 도 있으나 세션의 관리와 성능에 따르는 문제가 생길 수 있다. 따라서 사용자의 패킷에서 추출된 정보를 질의 패킷에 같이 보내어 서버에서의 응답으로 리다이렉션에 관련된 정보를 취득하는 것이 효율적이다. 이런

방식으로 리다이렉션에 관한 정보를 드라이버에서 사용할 수 있다.

유해 사이트 차단 모듈에서의 패킷 변조는 송신시에 한번 그리고 수신시에 두번에 걸쳐 일어난다.

송신시의 패킷 변조는 모듈 매니저 서버로 질의 데이터를 보내기 위해 사용되고 수신시의 패킷 변조는 사용자의 접속을 리다이렉션 시키기 위해 사용된다.

패킷 변조시에는 TCP의 흐름을 방해하지 않기 위해 해당 세션의 흐름을 파악하고 있어야 한다. HTTP request 패킷이 나가는 상태는 3 way hand- shake가 이루어진 상태 이므로 이후의 상태를 예측하여 동작하도록 하면 된다.

송신시의 패킷 변조는 다음 표 5와 같은 조건과 과정으로 이루어진다.

표 5. 송신시의 패킷 변조

내용 / 위치	추출	삽입	변조 여부
IP 헤더	· Destination Address · Identification	· 모듈 매니저 서버의 주소 · 변조된 패킷길이 · 변조된 CheckSum	변조
TCP 헤더	· Sequence Number · Acknowledge Number · Source Port	· UDP로 대체 · 변조된 CheckSum	변조
HTTP 헤더 or 데이터	· 도메인	· 도메인의 해쉬값 · Destination Address · Identification · Sequence Number · Acknowledge Number · Source Port	변조

송신시의 패킷 변조는 사용자의 HTTP request 패킷을 서버로 질의하도록 하기 위해 UDP로 변조 한다.(UDP는 TCP보다는 신뢰성이 떨어지지만 속도가 빠르며 또한 세션을 맺을 필요가 없다는 장점이 있다.)

위의 질의문을 서버에서 처리를 하지만 리다이렉션에 관련되는 데이터는 서버에서 전혀 알 수 없는 상황 이므로 응답

시에 다시 클라이언트로 보내준다.

다음의 응답형식에는 질의시에 없는 데이터가 추가 되어 있다. 그것은 리다이렉션 페이지의 정보와 플래쉬 메모리의 관리에 사용되는 정보들이다.

수신시의 패킷 변조는 다음 표 6과 같은 조건과 과정을 통해 이루어진다.

표 6. 수신시의 패킷 변조

내용 / 위치	추출	삽입	변조 여부
IP 헤더		· Source Address · Identification · 변조된 패킷길이 · 변조된 CheckSum	변조
UDP 헤더		· TCP로 대체 · Sequence Number · Acknowledge Number · Destination Port · 변조된 CheckSum	변조
HTTP 헤더 or 데이터	· 판단결과 · 서비스 코드 · HTTP 리다이렉션 헤더	· HTTP 리다이렉션 헤더	변조

수신시의 패킷의 변조는 사용자에게 적절한 응답을 주기 위

해 사용되는 것이다. 사용자에게 적절한 응답을 주기 위해서는 모듈 매니저 서버로의 질의시에 이용한 사용자 세션의 복구가 필요하다.

즉 사용자의 HTTP request에 대한 응답을 해 주어야 하는 것이다. 수신시에 패킷의 변조가 필요하다는 것은 사용자가 접근을 시도 했던 사이트가 유해 사이트라는 판정을 받은 경우이다.

따라서 모듈 매니저 서버에서의 응답 패킷을 HTTP response 패킷으로 변조하여 사용자에게 되돌려 주어야 한다.

수신시의 패킷 변조는 송신시의 패킷 변조와 반대의 과정을 거쳐 패킷을 재조립한다. 그러나 데이터의 내용은 드라이버에서 응답 데이터의 해석을 끝내고 리다이렉션에 관련된 데이터를 채운다.

리다이렉션에 관련된 데이터는 HTTP 헤더로 이 헤더는 서버에서 결정하여 보내준다. 즉 서버측에서 리다이렉션 방향을 결정할 수 있는 것이다.

4.4 통신 방법 및 데이터 형식

서버로의 질의와 응답 절차는 사용자의 세션이 이루어진 이후에 일어나는 상황이다. 따라서 사용자에게 빠르고 적절한 응답을 해주어야 할 책임이 있다.

다음 그림 12는 사용자의 세션과 세션 도중의 질의와 응답

이 이루어지는 과정을 보여준다.

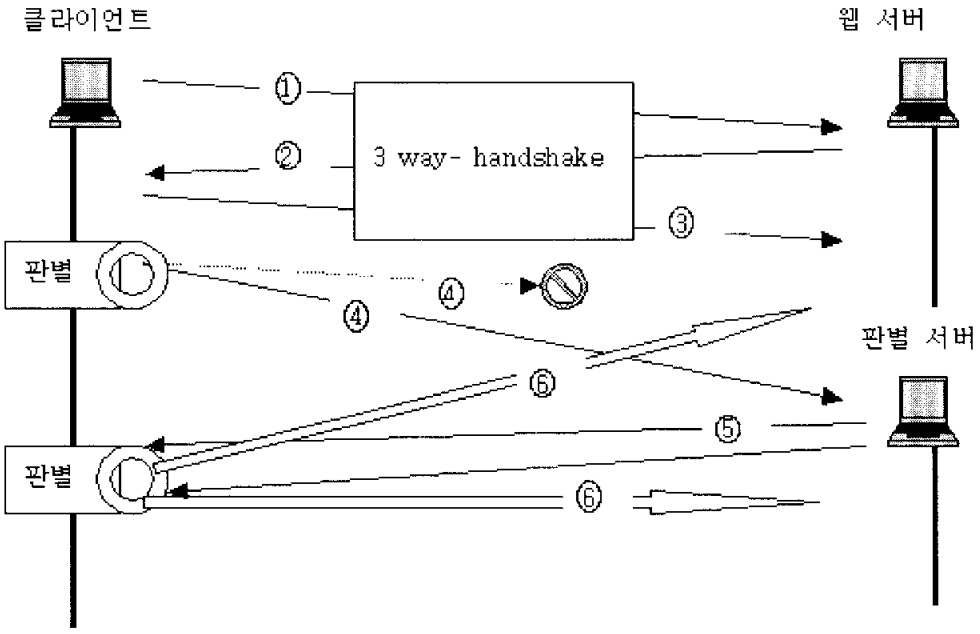


그림 12. 차단 모듈의 Session hijacking

그림 12에서 나타나는 통신과정은 차단 모듈의 유해 사이트 차단 과정 중 드라이버의 동작 부분을 제외하고 서버와의 통신에 관련 되는 부분을 설명하고 있다.

다음은 통신 과정에 대한 설명이다.

① 사용자의 호스트는 접속하려는 웹서버로 연결을 요구하는 TCP Syn 패킷을 보낸다.

② 웹서버가 존재하면 사용자의 호스트로 TCP Syn 패킷에 대한 응답과 자신의 TCP Syn 패킷을 보낸다.

③ 사용자의 호스트는 웹서버의 응답과 연결을 허락하는 패킷을 받음과 동시에 응답을 보낸다. 이로써 웹서버와 사

용자의 호스트는 세션을 시작할 수 있게 되었다.

④ 사용자의 호스트는 요청할 HTML문서의 위치와 요청형식이 담긴 HTTP 헤더를 웹서버로 보낸다.

이 순간 차단 모듈의 유해사이트 차단 기능이 동작을 하게 되어 드라이버에서 자체 판단을 할 수 없는 경우 판별 서버로 질의문을 보내게 된다. (여기서는 질의문을 보내는 것으로 가정을 하였으므로 이후의 상황에 대하여 설명을 계속한다.)

여기서 사용자 호스트의 HTTP request 헤더는 판별 서버로의 질의문으로 변경된다. 질의 패킷은 UDP 패킷이다.

⑤ 질의문에 대한 응답으로 판별 서버에서는 두개의 UDP 패킷을 보내게 된다. 이것은 사용자가 접근하려는 웹서버가 유해 사이트일 경우 리다이렉션을 염두에 둔 것이다.

⑥ 응답 패킷의 판별 여부에 따라 사용자의 접근을 허락 또는 거부를 하게 된다. 사용자의 접근을 거부하게 되면 응답이 왔던 것 처럼 보이기 위해 리다이렉션(판별 서버에 있는 접근 거부 페이지로)을 하게 된다.

현재 질의와 응답에 사용되는 통신 형태는 신뢰성이 없는 UDP를 사용하고 있다. UDP를 사용하는 이유는 다음과 같다.

(i) TCP와 같이 연결을 유지하기 위한 작업이 없으므로 속

도가 TCP보다는 빠르다.

(ii) UDP를 사용하더라도 신뢰성을 보장한다. 현재 사용자가 사용하는 브라우저에서는 TCP를 사용하고 있으며 본 논문의 유해차단 모듈은 TCP 세션을 가로채어 본 논문의 서버로 질의를 보낸다.

이 과정에서 유해차단 모듈의 질의 또는 응답 패킷이 소실된다면 사용자의 브라우저는 자신의 TCP세션에서 문제가 있는 것으로 생각을 하게 된다. 따라서 유해차단 모듈이 가로채었던 HTTP request 패킷이 분실 되었거나 웹서버의 응답이 분실 된 것으로 착각을 하게 되는 것이다.

그러므로 사용자의 브라우저는 다시 HTTP request 패킷을 내보내게 된다. 이로써 유해 차단 모듈은 UDP를 사용하지만 TCP의 신뢰성을 이용 할 수 있게 된다.

리다이렉션은 사용자가 접근을 시도한 사이트가 유해 정보를 포함하고 있을 때 일어나게 된다.

사용자가 유해 사이트에 접속시 리다이렉션을 시키지 않는다면 사용자는 TCP의 응답 대기 시간동안 아무런 응답이 없는 상태로 기다려야 하며 결과로는 서버를 찾을 수 없다는 브라우저의 메시지를 받게 된다.

이런 대기 상태와 네트워크의 상태에 대한 오해를 피하고 사용자에 즉각적인 응답을 주기 위해 리다이렉션을 사용한다.

다음 그림 13은 리다이렉션의 개요를 나타낸다.

그림 13에서 보이듯이 리다이렉션 과정은 두개의 UDP 즉 응답 패킷을 사용자가 접속을 시도한 웹서버의 응답 패킷으로 속여 목적을 달성한다.

데이터들은 드라이버에서 서버로의 질의시 사용자의 HTTP request 패킷을 변조하여 판별 서버로의 질의 패킷으로 재조립을 하는 과정에서 리다이렉션에 필요한 데이터를 따로 분류하여 질의 패킷에 데이터로 저장한다.

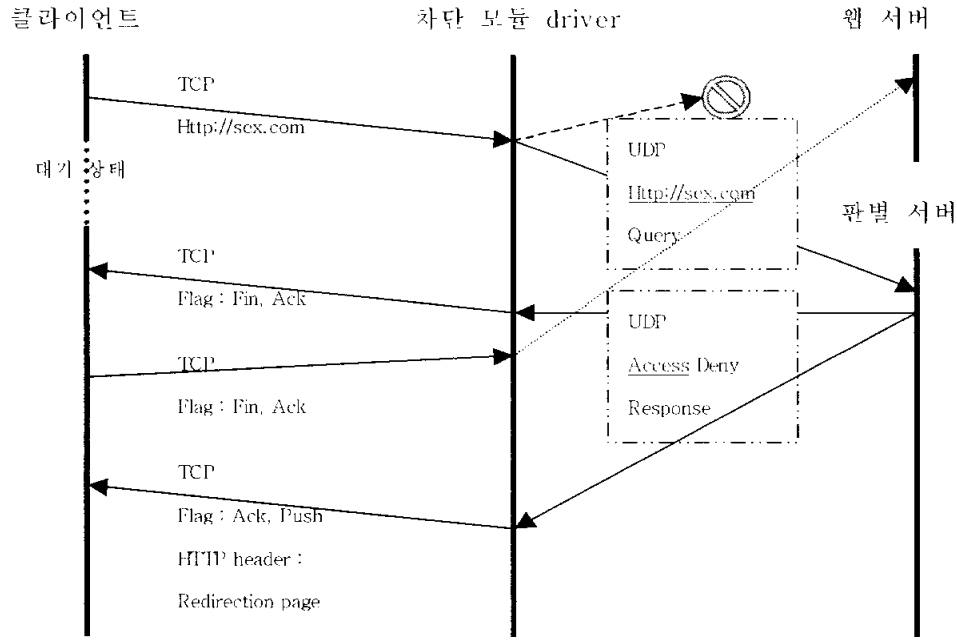


그림 13. 접근 거부에 의한 리다이렉션의 절차

그리고 서버에서 응답 패킷에 이 데이터들을 저장하여 드라이버에게 보내면 드라이버에서 수신 받은 패킷에 포함되어 있는 데이터들을 이용하는 구조이다. 이렇게 되면 드라이버에서

는 세션을 감시하거나 데이터를 저장하는 등의 번거로운 작업을 피할 수 있다.

이러한 방법이 가능한 것은 TCP의 흐름이 예측 가능하기 때문이며 언제나 같은 흐름을 유지하기 때문이다.

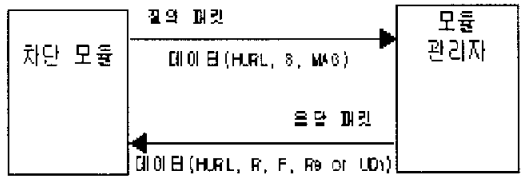


그림 14. 차단 모듈과 모듈관리자와의 통신

그림 14는 모듈 매니저와 드라이버의 질의 응답에 대하여 설명하고 있다. 차단 모듈의 질의 데이터 형식과 모듈 매니저의 응답 데이터 형식은 다음 표 7과 같다.

표 7. 차단 모듈과 모듈관리자와의 통신 데이터 형식

데이터형식	내 용
H	· 도메인 해쉬를 취한 도메인 데이터
S	· 서비스 코드
MAC	· 사용자 랜카드의 이더넷 어드레스
R	· 모듈 매니저의 판단결과
F	· Flag 데이터의 존재 유·무와 종류 길이를 표기
Re	· Redirection에 사용될 HTTP 헤더의 내용
UD	· 허용 목록에 업데이트될 데이터 · 데이터의 크기는 16바이트 이며 해쉬 함수를 이용하여 얻어 지는 도메인의 해쉬값

서비스 코드는 현재의 차단 모듈을 관리하기 위한 코드이다. 서비스 코드의 역할은 차단되는 유해정보의 카테고리를 식별하도록 도와주는 역할을 한다. 즉, 사용자의 차단 모듈은 음란정보를 가진 사이트만을 차단하고 싶을 경우 서비스 코드의 값은 음란 정보에 대한 값을 가지게 되는 것이다.

모듈 매니저는 서비스와 차단 모듈의 관리를 위한 체계적인 정책을 필요로 하며 허용 목록의 업데이트와 관련되는 정책사항을 정의할 필요가 있다.

허용 목록의 업데이트는 허용 목록의 버전정보를 모듈 매니저가 DB로 구축하여 관리하며 차단 모듈과 DB의 허용 목록 버전 정보와의 연관은 MAC으로 해결할 수 있다.

그리고 이러한 버전 정보를 통하여 차단 모듈의 질의시 마다 응답 패킷에 업데이트 데이터의 삽입여부를 결정할 수 있다.

4.5 플래쉬 메모리 관리방법

전용 NIC에는 플래쉬 메모리가 탑재되어 있으며 플래쉬 메모리의 용도는 유해 사이트의 차단 모듈에서 이용하는 허용목록이 유지 된다.

허용목록은 플래쉬 메모리의 용량이 한계가 있음으로 공장의 초기값은 일부분만을 채워넣는다. 이후의 목록은 사용자가 자주 접근하는 건전 사이트의 도메인을 서버에서 분석하여 건

전사이트로 판별된 사이트의 도메인 해쉬값이 플래쉬 메모리에 업데이트 된다.

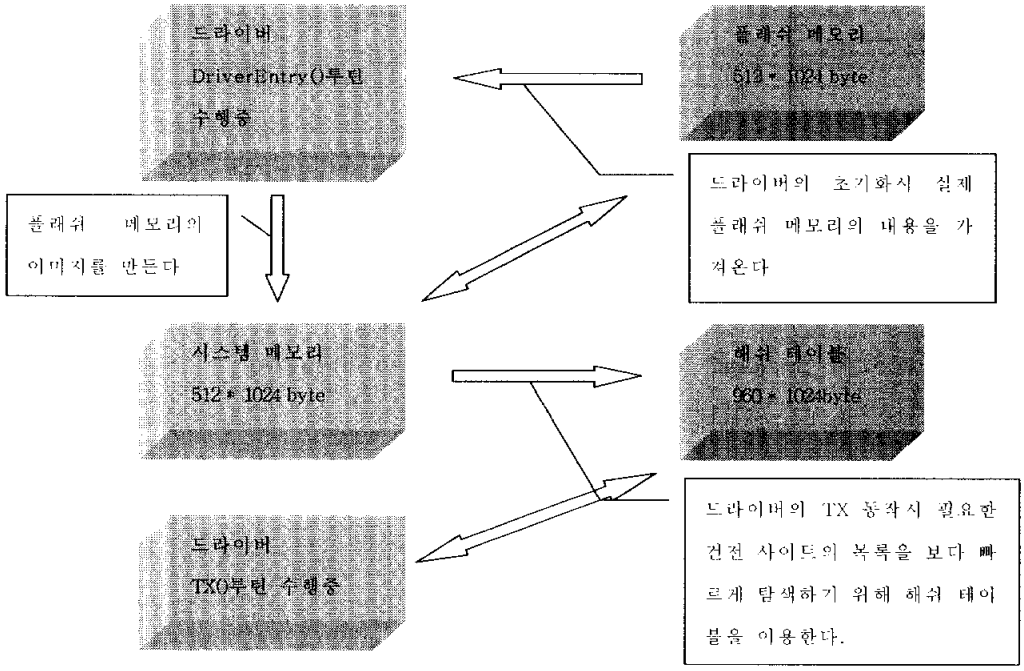


그림 15. 플래쉬 메모리의 관리와 탐색기법의 개요

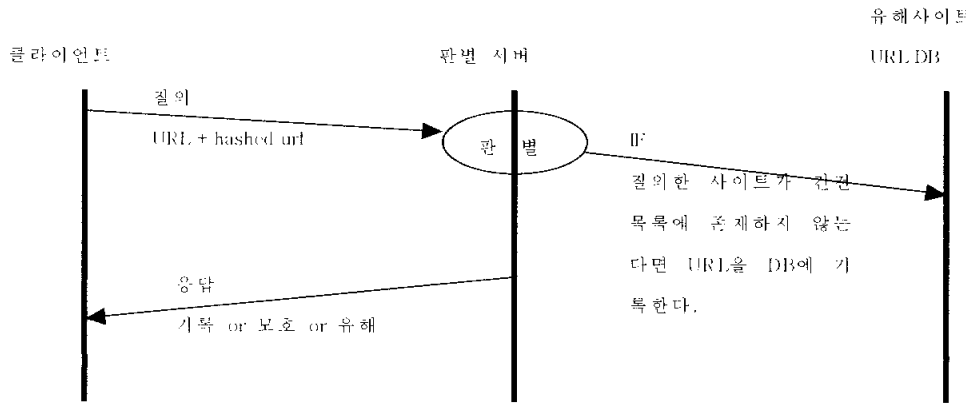


그림 16. 플래쉬 메모리의 업데이트

플래쉬 메모리에 실시간으로 기록되는 도메인들은 서버에서 건전 사이트라는 판별이 되어 있는 것들이다. 그림 16을 보면 응답 패킷에서 3가지의 상태 코드를 볼 수 있다.

이 상태 코드는 표 7에 정의된 데이터 형식의 F에 정의된다. 상태 코드의 의미는 다음의 표 8과 같다.

표 8. 모듈 매니저 서버의 응답 내용

상태코드 \ 의 미	
기록	• 플래쉬 메모리에 사이트를 기록 • 서버의 건전 사이트 목록에 존재
모호	• 서버의 DB에는 존재를 하지만 아직 판별이 되지 않은 사이트 • 유해 사이트의 목록에도 없으며 건전 사이트의 목록에도 존재 하지 않는 사이트.
유해	• 서버의 유해 사이트 목록에서 발견된 사이트 • 드라이버에서 이런 응답을 받으면 리다이렉션

이러한 상태 코드에 따라 드라이버에서는 플래쉬에 기록을 하게 된다. 그러면 실제로 플래쉬 메모리에 기록되는 것은 기록이라는 응답 코드외에는 플래쉬 메모리에 기록을 할 수 없다.

모호라는 코드는 현재 서버에서 판별이 되지 않았거나 DB

에 사이트가 존재 하지 않는 경우 이다. 이런 경우 사용자가 접근하려는 사이트가 유해한 사이트인지 건전한 사이트인지 알 수 없게된다. 그렇다고 무조건 사용자의 접근을 제한할 수도 없다.

따라서 이러한 경우 일시적으로만 접근을 허용한다. 즉 플래쉬 메모리에는 업데이트 시키지 않지만 해쉬 테이블에 삽입시켜 놓아 이후의 접근을 허용하는 것이다. 해쉬 테이블은 실제의 플래쉬 메모리 내용을 포함하고 있지만 플래쉬 메모리의 업데이트에는 영향을 미치지 않는다.

5. 시뮬레이션 및 분석

본 논문에서 구현한 유해 사이트 차단 시스템은 2장에 기술된 방법들의 단점들을 상당 부분 보완하였으며 모듈 매니저의 구현 시 체계적인 정책을 수립하여 적용한다면 유연한 구조의 시스템을 구축할 수 있다.

즉 내용등급의 적용은 모듈 매니저의 정보수집 기능에 내용등급에 관련된 사이트의 정보를 수집하도록 하는 검색 로봇을 컴포넌트로 추가하여 적용시킬 수 있으며 차단 모듈의 수정을 필요로 하지 않는다.

그림 17은 구현된 유해사이트 차단 시스템의 결과를 보여주고 있다. 사용자가 유해사이트로 접근을 시도 하면 차단 페이지가 브라우저에 나타난다.

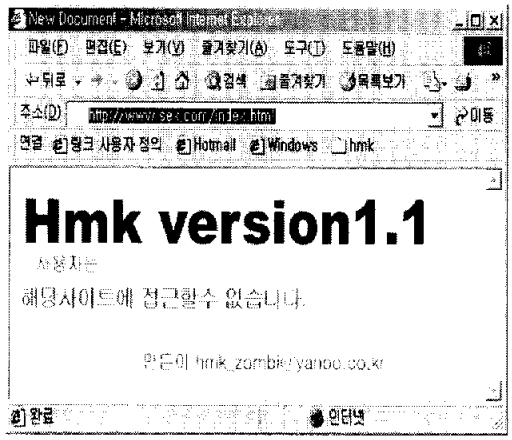


그림 17. 구현된 유해 차단 시스템을 적용한 시스템에서의 테스트 결과

표 9는 구현된 유해 차단 시스템과 기존의 제품들과의 차단율을 비교한 것이다.

표 9. 차단율과 접근 성공률 비교

	Black List 차단 실패율(%)	White List 허용 실패율(%)
제안 시스템	16	12
수호천사 배포판	31	38
파로스	31	25
안티엑스	46	63

차단 모듈의 동작환경이 운영체제의 커널 부분에서도 하위 레벨이므로 유해 사이트의 차단 이외에도 다양한 응용이 가능하다. 예를 들어 기초적인 침입차단 시스템을 구축한다면 드라이버는 모든 트래픽을 직접적으로 처리하고 있으므로 침입차단 모듈을 드라이버에 추가하여 침입차단 시스템을 구축할 수 있다.

표 10은 기존에 구현된 유해 사이트 차단 시스템과 본 논문에서 구현한 시스템의 장·단점을 나타내었다.

표 10. 유해사이트 차단 시스템의 장·단점

	장점	단점
허용 목록	유해사이트에 노출되지 않는다.	접근 가능한 사이트가 제한적이다.
차단 목록	인터넷에 제공되고 있는 대부분의 정보에 접근 가능	차단목록의 주기적 갱신이 어렵다. 유해사이트에 쉽게 노출된다.
내용 등급	다양한 환경에 적용 가능한 융통성과 보안성을 제공	차단 S/W와 웹브라우저에서 등급시스템을 지원하지 않거나 등급표시가 안된 사이트에는 내용등급에 의한 선별방법의 적용이 불가능
제안 방법	상술한 세가지 방법을 모두 적용 가능, 허용 목록의 갱신가능	차단 목록의 갱신이 어렵다.

6. 결론

기존의 유해 사이트 차단 소프트웨어와 본 논문에서 제안한 유해 사이트 차단 시스템은 다음과 같은 차이점이 있다. 첫째, 관리의 용이성이다. 본 논문에서 구현한 유해사이트 차단 시스템은 허용 목록이 지속적으로 업데이트 되도록 설계되어 있으며 목록의 작성도 실시간으로 사용자가 자주 방문하는 사이트를 위주로 작성하도록 구현 되어있다. 둘째, 사용자의 시스템에서 투명하게 동작한다. 셋째, 호환드라이버가 아닌 전용 NIC를 가지는 드라이버 방식으로 제작한다면 성능의 향상과 함께 클라이언트 방식의 유해 사이트 차단 소프트웨어의 단점인 차단 모듈의 무력화를 보완 할 수 있다.

현재 상용화되어 있는 유해 사이트 차단 소프트웨어들은 클라이언트용이며 일부 네트워크용 차단 소프트웨어가 존재하지만 성능의 개선이 시급한 상황이다. 네트워크와 PC의 성능 향상에 따라 본 논문에서 제안한 유해 사이트 차단 모듈의 관리를 전담하는 서버도 성능의 향상이 이루어 져야 한다. 현재 본 논문에서 구현된 모듈관리자는 DB의 관리와 갱신, 그리고 정보의 수집에 따르는 문제점들을 안고 있다. 또한 차단 모듈의 체계적인 관리 정책이 추가로 필요하여 차단 모듈의 관리 체계에 따라 사용자에게 보다 질 좋은 서비스가 가능하리라 예상된다.

참 고 문 헌

[1] 박영우, 이종화, "2001년 정보화 역기능 실태조사", 한국정보보호홍원 , pp. 47-48, 2001.

[2] 선우종성, 이병만, 김남욱, 홍성명, 송 의, "NCApatrol 1.5 개발보고서", 한국전산원, pp. 5-10, 1998.

[3] Paul Resnick, Jim Miller, "PICS: Internet Access Controls Without Censorship", Communications of the ACM, 1996, vol. 39(10), pp. 87-93..

[4] "Driver Developent Toolkit", Microsoft coporation, http://msdn.microsoft.com, June 2000.

[5] Postel,J., "Internet Protocol", RFC 791, USC/Information Sciences Instutute, September 1981.

[6] Reynolds, J., Postel, J., "Assigned Numbers", RFC 1340, USC/Information Sciences Instutute, July 1992.

[7] Postel, J., "User Datagram Protocol", RFC 768, USC/Information Sciences Instutute, August 1980.

[8] Postel, J., "Transmission Control Protocol", RFC 761, USC/Information Sciences Instutute, January 1980.

[9] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Berners-Lee, T., "Hypertext Transfer Protocol --HTTP/1.1", RFC 2068, January 1997.

[10] Richard Stevens, W., "TCP/IP Illustrated: the protocol", Addison-Wesely, 1994.

[11] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

감사의 글

처음 접하는 대학원 생활을 무사히 마칠 수 있을까 염려하며 시작하였는데 벌써 졸업이라 생각하니 뿌듯한 마음과 함께 아쉬움이 남습니다.
그 동안 많은 분들의 도움에 작으나마 정성을 다하여 감사의 마음을 전하고자 합니다.

배움의 길을 다시 시작하면서 언제나 힘과 용기를 심어주시며 격려해주신 이경현 교수님께 진심으로 감사의 말씀을 드립니다. 그리고 기꺼이 새로운 지도교수님이 되어주셔서 미숙한 저를 이끌어 주신 여정모 교수님께도 진심으로 감사의 말씀을 드립니다..
그리고 바쁘신 시간 속에서도 논문 심사를 위해 신경을 써 주신 박만곤 교수님, 김영봉 교수님께도 또한 진심으로 감사의 말씀을 드립니다. 2년 반 동안 배움의 길을 열어주셨던 윤성대, 박흥복, 박지환, 정순호, 김창수, 이경현, 김영봉 교수님들께 진심으로 감사드립니다.
낯설은 학교 생활에 언제나 본보기가 되어주신 연구실의 신 원 선배님, 이준석 선배님, 힘든 생활속에도 언제나 변치 않은 모습으로 연구실을 이끌어 가는 양수정 선배님, 신정화 선배님, 양종필 선배님, 박영호 선배님, 서 철 선배님, 조현호 선배님, 언제나 열심히 하시는 동기인 박남현 선생님, 김재권 선생님, 정상영 선생님, 박지철군, 김희연씨, 그리고 한해 늦게 인연이 되어 같이 공부한 위성균 선생님, 조선제 선생님, 심미선씨, 이영경씨, 이하 연구실 가족 모두에게 다시 한번 감사드립니다.
함께 공부한 임범춘, 김영하이하 모든 동기들, 옆에서 힘이 되어준 친구 성렬, 그리고 도움의 말을 아끼시지 않았던 동철 선배, 경섭 선배, 경훈 선배, 태훈 선배, 태윤 선배, 동렬 선배, 영웅 선배, 시환 선배, 민수 선배, 학기 중에 편안히(?) 공부할 수 있는 여건을 마련해 주신 구성진 사장님, 김경철 이사님, 안무경 이사님, 정영태 차장님, 조현호, 권봉재, 그리고 동서대학교 동기들, 친구 성현, 성원, 상훈, 태민 함께 했던 선, 후배님들에게도 감사의 마음을 전합니다.

끝으로 항상 믿어주신 아버지, 어머님, 동생 태원 고맙습니다. 그리고 사랑합니다. 저와 인연이 되었던 모든 분들에게 고개 숙여 감사의 마음을 전합니다.