Journal of the Korea Industrial Information Systems Research Vol. 20 No.2, Apr. 2015 :65-72
http://dx.doi.org./10.9723/jksiis.2015.20.2.065 ISSN:1229-3741

시그니처 기반 안티 바이러스 성능 향상 기법에 대한 연구

(A Performance Enhancement Scheme for Signature-based Anti-Viruses)

조 민 재[1], 신 지 선[2]*
(Min Jae Jo and Ji Sun Shin)

요 약 안티바이러스는 단말에서 악성소프트웨어를 탐지하는데 있어 널리 사용되는 솔루션이다. 이 중 시그니처 기반 안티바이러스는 가장 기본적인 탐지방법으로 파일과 악성소프트웨어의 시그니처를 비교하여 탐지한다. 최근 악성소프트웨어의 수가 급격히 증가함에 따라 시그니처 기반 안티바이러스의 탐지 시간이 증가하고 시간당 처리량이 줄어들면서 성능 저하 문제가 발생되고 있다. 본 논문에서는 이를 극복하기 위해 제시된 주요 연구 결과를 살펴보고 이를 개선한 새로운 성능향상 솔루션을 제시한다. 특히, 본 논문의 솔루션은 성능향상 수준이 가장 높은 솔루션으로 알려진 SplitScreen과 비교하여, 클라이언트의 작업을 줄이고, 시그니처 서버와의 통신비용을 줄여 안티바이러스 솔루션의 성능향상에 기여하였다.

핵심주제어 : 시그니처 기반 안티바이러스, 블룸필터, 클라우드, 클라이언트 성능

Abstract An anti-virus is a widely used solution for detecting malicious software in client devices. In particular, signature-based anti-viruses detect malicious software by comparing a file with a signature of a malicious software. Recently, the number of malicious software dramatically increases and hence it results in a performance degradation issue: detection time of signature-based anti-virus increases and throughput decreases. In this paper, we summarize the research results of signature-based anti-viruses which are focusing on solutions overcoming of performance limitations, and propose a new solution. In particular, comparing our solution to SplitScreen which has been known with the best performance, our solution reduces client-side workload and decreases communication cost.

Key Words : Signature-based Anti viruses, Bloom filters, Cloud, Client-side Performance

1. 서 론

Ahnlab의 ASEC 보고서에 의하면 2014년 12월 한 달 간 탐지된 악성소프트웨어 수는 2,695만 5,828건이며 수집된 악성소프트웨어 샘플 수는 607만 9,293건이다[1]. 또한 McAfee 보고서에 의하면 2014년 3분기에 전체 악성소프트웨어 샘플이 3억 개를 넘겼으며 작년 대비 76%가 증가하였다[2].

이러한 악성소프트웨어를 탐지하기 위해 안티바이러스를 이용한다. 안티바이러스는 악성소프트웨어를

* Corresponding author
본 연구는 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2013R1A1A3009163)
Manuscript received April 3, 2015 / Revised April 16, 2015 / Accepted April 17, 2015
1) 세종대학교 컴퓨터공학과
2) 세종대학교 정보보호학과, 교신저자(jsshin@sejong.ac.k)

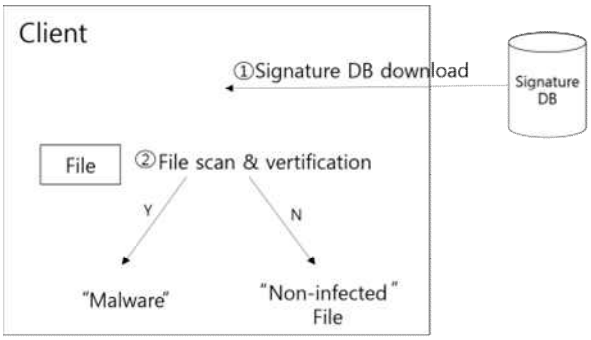


Fig. 1 Original Signature-based Anti-viruses

발견하고 삭제하는 소프트웨어이다. 안티바이러스에서 악성소프트웨어를 발견하기 위해서 사용되는 방법에는 시그니처 기반 탐지[3], 휴리스틱(heuristic) 기반 탐지[4], 행위 기반 탐지[5,6] 등이 있다. 그 중 시그니처 기반 탐지는 안티바이러스의 가장 기본적인 솔루션으로 파일이 악성코드의 패턴을 가지고 있는지 검사하는 방식이다. 여기서 악성코드의 패턴을 시그니처(signature)라고 하며, 새로 악성코드가 발견되면 그 시그니처를 데이터베이스에 추가하여 악성코드들의 시그니처 데이터베이스를 갖춘다. 기초적인 시그니처 기반 안티바이러스는 클라이언트에 전체 시그니처 데이터를 다운받아서 직접적으로 파일 검사를 진행한다(<Fig .1> 참고). 데이터베이스의 크기가 증가하면 탐색 시간이 증가하게 된다. 또한 메모리 소비가 증가하고 시간당 처리량이 줄어들면서 시그니처 매칭 작업의 전체적인 성능이 저하하는 문제가 발생한다.

최근에는 악성소프트웨어가 증가함에 따라 악성소프트웨어의 시그니처 수도 수억 개가 존재하게 되어 안티바이러스의 엔진 크기도 증가하게 된다. 시그니처 기반 안티바이러스는 수억 개의 시그니처 데이터베이스와 파일을 비교하여 탐색을 진행하여야 하므로 탐색 속도 개선이 시그니처 기반 안티바이러스의 중요한 요소이다.

본 논문에서는 기초적인 시그니처 기반 탐지의 한계를 극복하는 기법들을 살펴본다. 특히, 클라이언트의 작업에 대한 성능 향상 방법들을 이해한다. 또한, 이를 개선한 솔루션을 제안하여 시그니처 기반 안티바이러스 성능 향상에 기여하고자 한다.

1.1 시그니처 기반 안티바이러스 솔루션

앞서 이야기했듯이 안티바이러스의 대표적인 탐지 방법으로 시그니처 기반 탐지, 행위 기반 탐지, 휴리스틱 기반 탐지가 있다.

시그니처 기반 탐지는 파일에 악성코드의 패턴을 가지고 있는 지를 단순 비교하는 방식이기 때문에 유사한 패턴을 가진 악성코드는 탐지하지 못한다. 이를 보완하기 위해 휴리스틱 기반 탐지는 특정 패턴을 비교하여 유사한 패턴을 가진 악성코드도 탐지 가능한 방식이다. 행위 기반 탐지는 악성코드의 패턴 특징을 탐지하는 것이 아니라 악성소프트웨어가 실행될 때 행동을 기반으로 하여 탐지하는 방법이다. 시그니처 패턴을 통한 탐지는 안티바이러스 외에도 침입탐지[7-11]에도 이용된다.

1.2 ClamAV

ClamAV[12]는 가장 널리 사용되는 오픈 소스 안티 바이러스 솔루션이다. ClamAV의 시그니처는 2가지 타입의 시그니처를 포함한다. 첫 번째 타입의 시그니처는 전체 파일이나 세그먼트를 MD5(message-digest algorithm 5)로 해싱한 시그니처이다. 다른 시그니처는 바이트 패턴 시그니처로써 정규 표현식과 같은 문법을 가진 커스텀 언어로 작성된다. 바이러스를 탐색할 때 ClamAV는 처음에 몇 가지 사전검사 단계를 수행하며 각 입력 파일을 시그니처 데이터베이스와 검사한다. MD5 검사는 시그니처 데이터베이스에 있는 MD5와 파일의 MD5를 검사하며, 정규표현식 검사는 파일에 해당하는 정규표현식이 포함되어 있는지를 검사한다.

2. 관련 연구(Related Works)

시그니처 탐색의 성능을 향상시키기 위해서 3가지 방법이 사용되고 있다. 먼저 사전 검사(pre-matching) 방식[13,14]으로 사전 검사는 먼저 의심되는 파일을 미리 검사하여 추출한 다음에 실제로 정밀 탐색을 진행하는 방법이다. 다음으로, 클라우드를 사용하는 방식[15]이 있다. 클라우드 방식은 파일을 클라우드 서버에 보내어 클라우드에서 파일에 대한 감염 여부

를 검사하여 결과를 보내주는 방식이다. 마지막으로 사전검사와 클라우드 방식을 조합하여 사용하는 방식이다[16]. 각각의 솔루션 현황을 정리하면 <Table 1>과 같다.

Table 1 Signature-based Anti-Virus Solutions

	Pre-matching	Cloud
HashAV[13]	∨	
SigMatch[14]	∨	
CloudAV[15]		∨
SplitScreen [16]	∨	∨

2.1 사전 검사(pre-matching)

<Fig. 2>에서 보듯이 사전 검사는 감염되지 않은 파일을 먼저 필터링하여 성능을 향상하는 방법이다. 전체 시그니처 데이터를 블룸필터로 해싱하여 검사하고자 하는 파일과 사전 검사를 진행하여 의심되는 파일을 걸러낸 뒤 의심 파일을 전체 시그니처에서 탐색하여 감염 여부를 판단한다. 사전 검사를 통해서 감염되지 않은 파일이 걸러지므로 파일에 대해서 전체 시그니처 데이터베이스에서 확인하는 과정을 피한다. 따라서, 메모리 접근을 줄임으로써 캐시 미스 발생이 줄어들게 되며 이 때문에 시그니처 데이터가 증가하여도 메모리 소비가 급격하게 증가되지 않으므로 전체적인 성능이 향상된다. 이러한 사전 검사를 통해서 성능 향상하는 방법은 HashAV, SigMatch, SplitScreen들이 있다.

2.1.1 HashAV

HashAV[13]는 전체 시그니처 데이터를 해싱하여 캐시에 존재하는 블룸 필터에 저장한다. HashAV는 저장된 블룸 필터의 해싱 값과 검사하고자 하는 파일을 해싱한 값을 비교하여 값이 같을 경우에는 의심되는 파일로서 실제 검사 단계로 넘어가 검사를 진행한다. 값이 다를 경우에는 감염되지 않은 파일이므로 검사를 더 진행하지 않는다. 블룸 필터의 성능

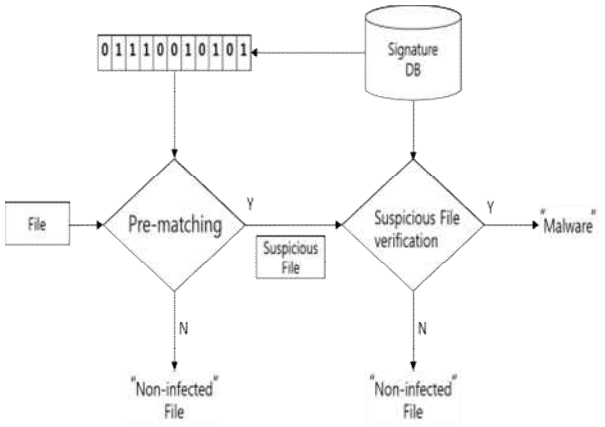


Fig. 2 Signature-based Anti-viruses with a pre-matching procedure

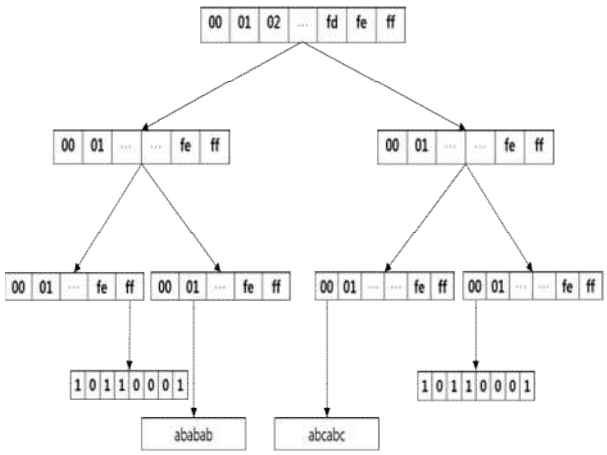


Fig. 3 An example of SigMatch Algorithm with b=3 and β=5

을 향상시키기 위해서 해시 함수의 선택, 블룸 필터의 크기 등의 요소를 설정한다. 시스템마다 성능을 최적화시키기 위해서 시스템 성능에 맞는 요소들을 프로그램을 수행하여 선택한다.

2.1.2 SigMatch

SigMatch[14]는 HashAV와 검사 방법은 비슷하다. 다만, SigMatch는 단순한 블룸 필터를 사용하는 것이 아니라 SigTree를 사용한다. 트리 구조가 블룸 필터보다 속도가 빠르며 블룸 필터는 트리 구조에 비해 메모리 소비가 적다는 장점이 있어 SigTree는 트리 구조와 블룸 필터를 조합하였다. SigTree의 트리나 블룸 필터에 시그니처가 인덱스된다. 각 SigTree

노드는 256크기의 배열로 구성되어 있으며 최상위 노드는 자식 노드를 가지고 있으며 리프 노드(leaf node)는 블룸 필터에 연결되며 짧은 시그니처 경우는 링크드 리스트에 연결된다.

SigMatch는 시그니처를 SigTree에 할당하며 파일의 감염 여부를 검사하고자 할 때 파일의 시그니처 값을 SigTree에서 탐색하여 존재하는지 찾으며 해당 파일의 시그니처가 SigTree에 존재할 경우 실제 검사 단계로 넘어가 검사를 진행하여 실제로 파일이 감염이 되었는지 검사한다.

SigTree에서 시그니처 탐색은 <Fig. 3>을 통해서 자세히 설명한다. <Fig. 3>의 SigTree에는 아래 4개의 인덱싱 된 시그니처가 있다. b는 SigTree의 깊이이며, β는 블룸 필터에서 인덱스 되는 시그니처 길이이다.

Signature 1 : abcd0001ffabababababab
Signature 2 : 00fe01abababab
Signature 3 : fdff00aaabadcdef
Signature 4 : efacabcdfd0100abcabc

트리의 깊이는 3, 블룸 필터에서 인덱스 되는 시그니처는 5바이트로 설정하였다. Sig1의 인덱스 값 '0001FFABABABABAB'을 SigTree에서 탐색을 할 때 상위 노드의 시그니처 인덱스 값에서부터 시작한다. 최상위 노드에서 '00' 인덱스을 찾은 후 다음 자식 노드를 탐색한다. 자식 노드에서 '01' 인덱스 값을 찾은 후 다음 인덱스가 가리키는 곳을 탐색한다. 마지막 인덱스가 가리키는 곳의 값은 링크드 리스트나 블룸 필터이다. Sig1의 경우에는 짧은 시그니처가 아니기 때문에 블룸필터에서 'ABABABABAB' 값이 존재하는 지 확인한다. 해당 값이 존재하면 Sig1을 실제 검사 단계로 넘어가 검사를 진행한다.

2.2 클라우드(Cloud)

사전 검사 이외에도 시그니처 기반 안티 바이러스의 성능을 향상시키기 위한 방법으로 클라우드 방식을 사용한다. <Fig. 4>와 같이 클라우드 방식은 클라이언트에서 검사하고자 하는 파일을 서버에 보내 해당 파일에 대해 감염 여부를 검사한 뒤 결과를 클라이언트에 보낸다. 클라우드 방식은 전체 시그니처

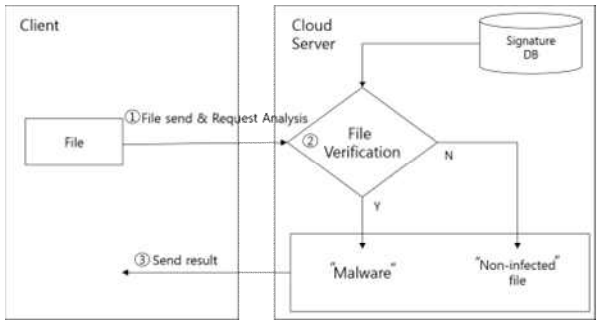


Fig. 4 Cloud-based Anti-viruses

데이터와 검사를 서버에서 하므로 클라이언트의 리소스 소모가 줄어든다. 그렇기 때문에 리소스가 중요한 요소인 디바이스에서 클라우드 방식을 적용하는 게 효과적이다. 클라우드 방식을 사용하여 성능을 향상시키기 위한 방법은 CloudAV에서 제안하였다.

CloudAV[15]는 클라우드에서 악성소프트웨어를 탐색하는 시그니처 기반 안티 바이러스이다. CloudAV는 탐색의 정확성을 'N-version protection'을 적용하여 악성소프트웨어 탐색을 진행한다. CloudAV는 클라이언트에서 악의적인 파일이나 원하지 않은 파일이 시스템에 들어오면 해당 파일을 서버에 보내 분석을 한다. 서버에서 분석 시 N-version protection을 사용한다. N-version protection은 10개의 안티 바이러스 엔진과 2개의 행위 기반 탐색 엔진을 사용하여 분석한다. 이러한 N-version protection을 사용하는 이유는 악성소프트웨어가 발견되어 이를 안티 바이러스에 적용되는 기간을 줄이기 위해서 이다. 하지만 CloudAV는 파일을 서버에 보내 분석하기 때문에 민감한 데이터를 다루는 유저일 경우 프라이버시를 침해할 수 있다[2,3].

2.3 조합 방식

시그니처 탐색의 성능을 향상시키기 위해서 사전 검사와 클라우드 방법 두 가지를 조합하여 사용하기도 한다. 두 방법의 조합은 사전 검사를 통해서 의심되는 파일만 걸러내어 클라우드에서 의심되는 파일에 대한 부분 시그니처 파일을 받아서 검사하기 때문에 경량화된 클라이언트를 제공하여 성능이 증가하게 된다.

SplitScreen[16]은 사전 검사와 클라우드 방법을 두

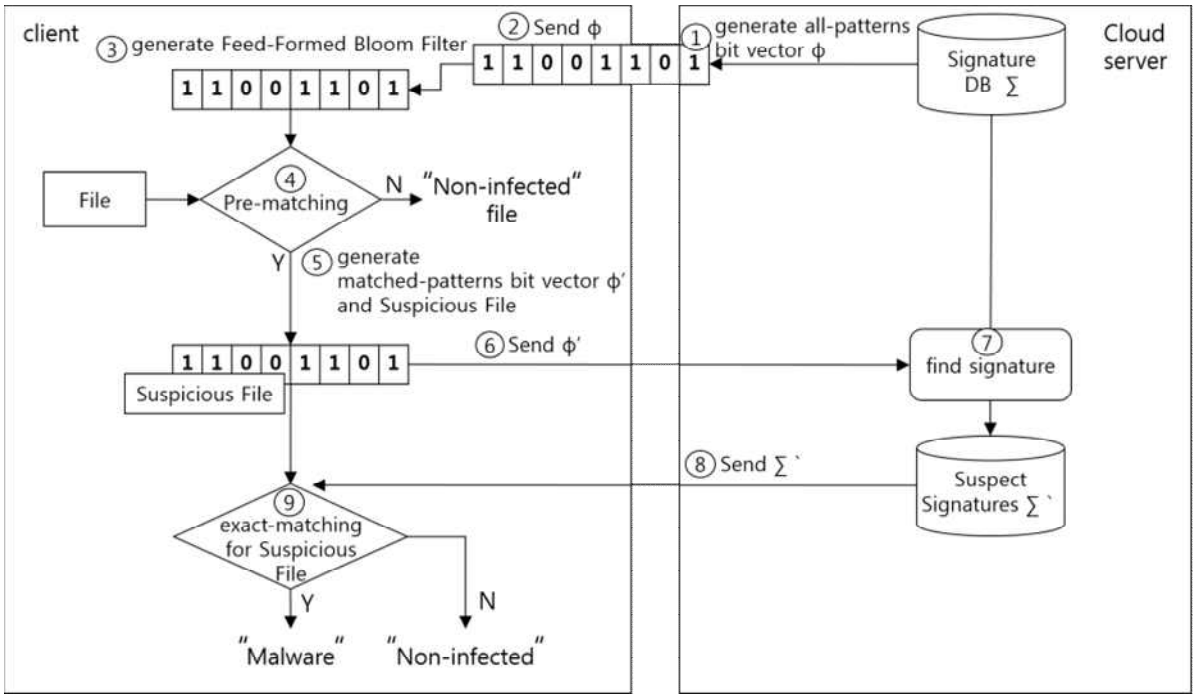


Fig. 5 The Architecture of SplitScreen

가지를 조합한 방법이다. SplitScreen은 서버에서 시그니처의 일부분이 해싱된 블룸 필터를 받아와 사전 검사를 통해서 의심되는 패턴을 만들어 서버에 보내고, 서버에서 전체 시그니처와 의심되는 패턴을 탐색하여 의심되는 패턴의 실제 시그니처 데이터를 클라이언트에 보내어서 실제 패턴 매칭을 통해 파일의 감염 여부를 확인한다. <Fig. 5>에서는 SplitScreen의 구조를 설명하고 있다. 먼저 SplitScreen는 서버에서 최근 악성소프트웨어 시그니처를 통해서 전체 패턴을 가진 All-patterns bit vector ϕ를 생성하고 이를 클라이언트에게 보낸다. 클라이언트는 서버로부터 받은 all-patterns bit vector ϕ을 통해 Feed Forward Bloom Filters(FFBF)을 생성하며 클라이언트는 FFBF을 통해서 파일을 사전 검사한다. 파일 중 all-patterns bit vector ϕ에 대응되는 패턴은 matched-patterns bit vector ϕ'에 기록한다. 사전 검사를 통해 matched-patterns bit vector ϕ'과 의심 파일을 생성한다. 클라이언트는 bit vector ϕ' 서버에 보내며 서버는 bit vector ϕ'에 해당하는 시그니처를 전체 시그니처에서 검색하여 의심 시그니처 파일 $\sum'$을 클라이언트에 전송한다. 클라이언트는 받은 시그니처 $\sum'$을 통해서 의심 파일에 대해서 실제 매칭을 진행한다. SplitScreen은 MD5 시그니처와 정규표현식 시그니처에 대한 탐색 모두 다루고 있다.

SplitScreen은 전체 시그니처 파일을 클라이언트에 저장하는 것이 아니라 사전 검사를 통해 의심되는 파일에 대한 시그니처만 서버에서 받아오기 때문에 경량화된 클라이언트를 제공하게 되며 사전 검사를 통해서 감염되지 않는 파일을 빠르게 필터링 할 수 있어 성능이 향상된다. 뿐만 아니라 SplitScreen은 전체 파일을 서버에 보내는 것이 아니라 부분 파일만 보내기 때문에 CloudAV와 달리 유저의 프라이버시 침해를 줄일 수 있다.

3. 제안하는 성능향상 기법

본 논문에서 제안하는 방식은 시그니처 기반 안티바이러스의 성능 향상을 위한 방법으로 사전 검사와 클라우드를 조합한 방식의 새로운 솔루션을 제안한다. SplitScreen의 솔루션은 정규표현식 탐색에 비중을 크게 두고 있다.

Table 2 Analysis on ClamAV Database (June 2012) [14]

	Basic	Regex	MD5
Signature Ratio	6.06%	0.61%	92.32%
Detection time Ratio	63.86%	27.94%	8.2%

한편, ClamAV의 시그니처에서 MD5는 92%를 차지하며 ClamAV의 전체 탐색 시간중 8.2%을 차지한다[17]<Table 2>. 본 논문에서는 MD5 시그니처에 초점을 맞추어 SplitScreen의 솔루션을 개선하여 MD5 시그니처 탐색 성능 향상의 방법을 제시한다.

<Fig. 6>은 제안하는 방식의 시스템 구조도이다. 제안하는 방식은 먼저 클라이언트에서 파일의 블룸필터(δ)을 생성한다. 생성한 블룸필터 δ을 서버에게 전송 한다. 서버는 클라이언트로부터 받은 블룸필터 δ을 시그니처 데이터베이스($\sum$)의 블룸필터와 사전검사를 진행한다. 사전 검사를 통해 시그니처와 일치하는 블룸필터(δ')을 생성하고 블룸필터(δ')에 해당하는 시그니처($\sum'$)를 생성한다. 서버는 해당하는 시그니처($\sum'$)을 클라이언트에 보내 클라이언트는 서버로 받은 시그니처를 통해 실제 매칭을 진행한다. 클라이언트는 새로운 파일이 생성되거나 다운받으면 파일의 블룸필터를 생성하여 서버에게 전송한다.

시그니처 기반 솔루션으로 본 논문에서 제안하는 방식과 SplitScreen의 정규표현식에 대한 탐색 솔루션을 조합하여 사용하는 것을 제안한다. 이로써 시그니처 기반 안티바이러스의 성능 향상에 기여할 수 있다.

<Table. 3>과 <Table. 4>에 기존 솔루션과 본 논문에서 제안하는 솔루션을 비교하였다. 구체적으로 본 논문에서 제안하는 솔루션의 특징을 살펴보면 다음과 같다:

- 사전 검사를 서버에서 수행하므로 클라이언트의 작업량이 줄어든다.
- 새로운 시그니처가 추가될 때 빠르게 대응할 수 있다.
- 파일의 해쉬값만을 서버에 보내므로 CloudAV와 같은 사용자의 프라이버시를 침해하지 않는다.
- SplitScreen의 경우 3번의 통신이 이루어지지만 본 논문에서 제안하는 방식은 2번의 통신을 통해 파일을 검사하므로 좀 더 효율적이다.

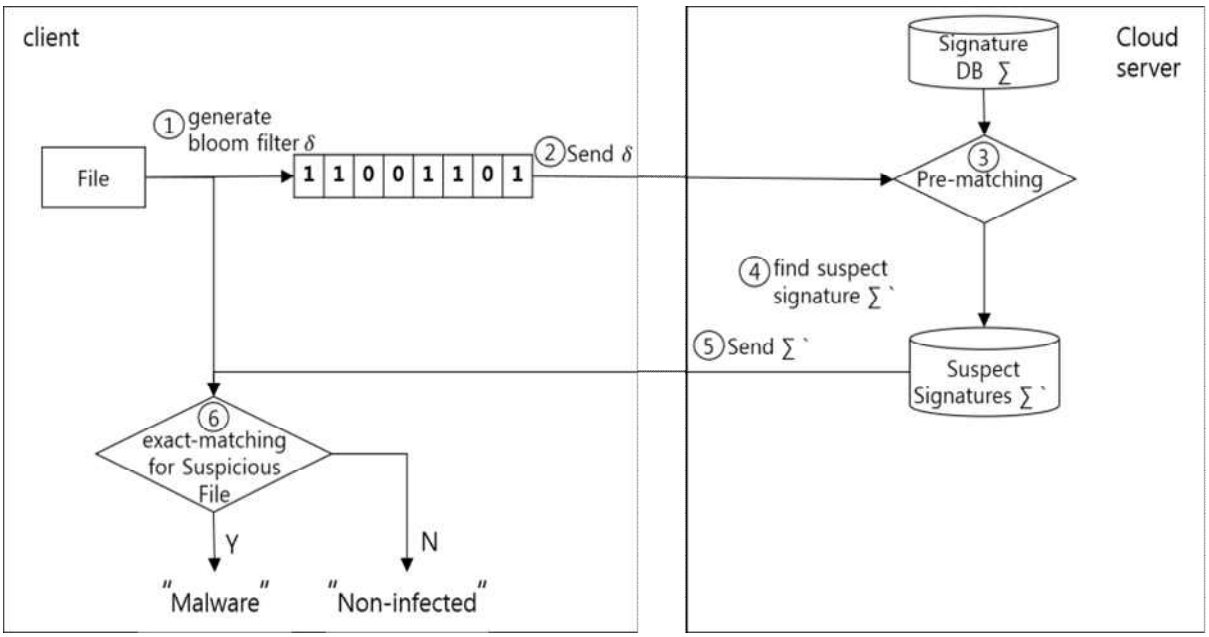


Fig. 6 The Architecture of Our Solution

Table 3 Performance Evaluation and Comparison

	CloudAV	Split Screen	Our Solution
Communication Cost	2	3	2
Invasion of Privacy	O	X	X
Real-time Response	O	O	O

4. 결 론

본 논문에서는 시그니처 기반 안티바이러스 성능 향상을 위한 방법인 사전 검사 방식과 클라우드 방식 2가지를 살펴보았다. 사전 검사 방식은 시그니처 데이터 사전 검사를 통하여 의심되지 않은 파일을 필터링함으로써 성능 향상시키는 방식이다. 이때 사전검사를 위해서 다양한 해싱 방법이 이용된다. 클라우드 방식은 클라이언트에서 파일을 탐지하지 않고 컴퓨팅 파워가 높은 서버, 즉 클라우드에 파일을 보내 탐지하는 방식이다. 시그니처 기반 안티바이러스 성능 향상 방법은 <Table. 4>에 정리하였다.

본 논문은 시그니처 기반 안티바이러스 성능 향상을 위한 연구 방법 동향을 정리함으로써 향후 시그니처 기반 안티바이러스의 탐지 솔루션 개발에 도움을 주고자 한다. 또한 본 논문의 제안 방식은 서버에서 사전 검사를 수행함으로써 클라이언트의 작업량과 통신비용을 감소시켰으며 새로운 시그니처에 대해서 빠르게 대응할 수 있다는 장점이 있다. 또한 파일의 해싱 값만 다루기 때문에 사용자의 프라이버시를 침해하지 않아 시그니처 기반 안티바이러스의 성능 향상에 기여한다.

Table 4 Summary of Solutions

Solution	Properties
HashAV	Pre-matching • Bloom Filters for Hashing
CloudAV	Cloud-based No Privacy Protection
SplitScreen	Pre-matching • Feed Forward Bloom Filters for Hashing Cloud-based Privacy Protection
SigMatch	Pre-matching • SigTree(Bloom Filters + Tree) for Hashing
Our Solution	Pre-matching • Bloom Filters for Hashing Cloud-based Privacy Protection

References

[1] ASEC Report, http://download.ahnlab.com/asecReport/ASEC_Report_Vol.60_Kor.pdf

[2] McAfee Report, http://www.mcafee.com/kr/resources/reports/rp-quarterly-threat-q3-2014.pdf

[3] J.O. Kephart and W.C. Arnold. 1994. "Automatic Extraction of Computer Virus Signatures." In Proc.of the 4th Virus Bulletin Int'l Conf. Virus Bulletin Ltd., Abingdon, pp. 178-184.

[4] Arnold, W. and G. Tesauro, "Automatically generated Win32 heuristic virus detection", in 10th Virus Bulletin International Conference (VB2000), pp. 51-60, 2000.

[5] Cohen, F.: Computer viruses. Ph.D. thesis, University of South California, 1986

[6] Cohen, F.B.: Computer viruses: Theory and experiments. Comput. Secur. Vol 6, No.1, pp. 22-35, 1987

[7] Eun Jun Yoon, Hyun Sung Kim and Ki Dong Bu, "An Intrusion Detection System Using Pattern Classification", Proceedings of the Korea Society for Industrial Systems Conference, 2002.

[8] Eun Jun Yoon, Hyun Sung Kim and Ki Dong Bu, "Intrusion Detection System using Pattern Classification with Hashing Technique", Journal of the Korea Industrial Information System Society, Vol. 8, No. 1, pp. 75-82, 2003.

[9] Hyun Chul Cha, "A Solution for Timing Gap

Problems on Network Intrusion Detection Systems", Journal of the Korea Industrial Information System Society, Vol. 7, No.1, pp. 1-6, 2001.

[10] Seon Cheol Choi and Hyun Chul Cha, "A Detection Method for Network Intrusion using the NFR", Proceedings of the Korea Society for Industrial Systems Conference, 2001.

[11] Jae Min Son, Hyun Sung Kim and Ki Dong Bu, "A Scheme for Protecting Security Rules in Intrusion Detection System", Journal of the Korea Industrial Information System Society, Vol. 8, No.4, pp. 8-16, 2003.

[12] ClamAV, http://www.clamav.net/index.html

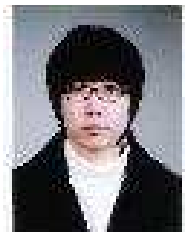
[13] Erdogan, Ozgun, and Pei Cao. "Hash-AV: fast virus signature scanning by cache-resident filters," International Journal of Security and Networks 2.1, pp. 50-59, 2007

[14] Kandhan, Ramakrishnan, Nikhil Teletia, and Jignesh M. Patel. "SigMatch: fast and scalable multi-pattern matching." Proceedings of the VLDB Endowment3.1-2 ,pp. 1173-1184, 2010

[15] Oberheide, Jon, Evan Cooke, and Farnam Jahanian. "CloudAV: N-Version Antivirus in the Network Cloud." USENIX Security Symposium, pp. 91-106, 2008

[16] Cha, Sang Kil, et al. "SplitScreen: Enabling efficient, distributed malware detection." Communications and Networks, Vol 13, No. 2, pp. 187-200, 2011

[17] Tran Ngoc, T, Hieu T T, Ishii H, and Tomiyama S," Memory-efficient signature matching for ClamAV on FPGA.", In Communications and Electronics (ICCE), 2014 IEEE Fifth International Conference on, pp. 358-363, 2014



조 민 재 (Min Jae Jo)

- 학생회원
- 세종대학교 컴퓨터공학과 학사
- 세종대학교 컴퓨터공학과 석사과정
- 관심분야 : 모바일 보안, 네트워크 보안

신 지 선 (Ji Sun Shin)

- 정회원
- 서울대학교 컴퓨터공학과 학사
- 메릴랜드 주립대학(University of Maryland at College Park) 컴퓨터과학과 박사
- 삼성SDS 책임연구원
- 세종대학교 정보보호학과 조교수
- 관심분야 : 정보보호, 암호학, 컴퓨터 보안