인터넷 서비스 미 사용 계정 차단을 위한 서비스 로그 분석기

정규철* · 이진관* · 이대형* · 장혜숙* · 이종찬* · 박기홍*

요 약

인터넷이 확산과 더불어 보안의 문제도 증가하고 있다. 이로 인해 네트워크 보안과 서비스에 대한 관리자의 책임 또한 더욱더 중요시 되고 있다. 본 논문에서는 서비스 로그를 분석하여 멀티 도메인 환경에서 장시간 사용되지 않는 사용자 계정에 초점을 맞추어 관리자로 하여금 시스템 보안의 틈새를 찾고 이를 해결 할 수 있는 방안을 제시하였다. 이를 위해 SLA(Service Log Analyzer)를 구현하여 각 서비스들이 수행될 때 기록되는 각각의 로그를 분석한다. 그 결과 서비스를 사용한 계정 이름의 수를 포함한 UUL(Used User List)를 구축하고 일정기간 사용하지 않는 계정을 찾아내고 계정정보의 종료 시한을 수정한다.

The Service Log Analyser for Blocking Unused Account on Internet Services

Kyu Cheol Jung* · Jin-Kwan Lee* · Dae-Hyung Lee* · Hae-Suk Jang*
Jong-Chan Lee* · Kihong Park*

ABSTRACT

The fact that since Internet has been spreaded widely to people, Many security problems also have been grown too much. Due to sudden growth, administrator's responsibility for secure network and services has been growing more and more. This paper represents how to prevent account which didn't use for long period on multi domains environment using service log analysis. hence administrator can find security hole on systems and can dealing with it. The Service Log Analyzer is that loading log file which are written by each service and analyzing them. as a result it makes a list named UsedUserList contains a number of account names which uses specific services. When the time has come - means cron job schedule time, User Usage Shifter is the next runner. it's mission is finding the person who didn't used service for a specific period of time. Then modifying the expire day of the account information.

Key words : SLA(Service Log Analyser), xNIX, UUL(Used User List)

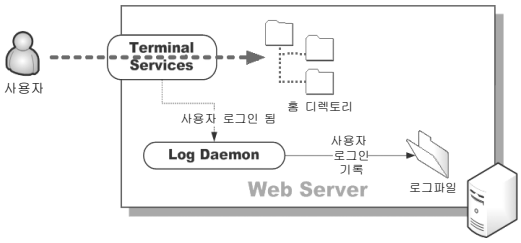
* 군산대학교 컴퓨터과학과

1. 서 론

인터넷을 이용하면서 사용자들은 자신의 계정이 있는 서버를 이용하여 웹 서비스, 파일전송 서비스, 터미널 서비스 등을 수행하기 원한다. 그러나 특정 업무에 종사하지 않을 경우 일시적인 사용에 그치고 만다. 이로 인해 각 서버에는 사용하지 않는 각 서비스계정이 산재 되어 시스템의 부하 및 위부 침입의 경로가 되기도 한다[1]. 특히 장시간 사용하지 않는 홈페이지 계정의 경우 펄(Perl) 스크립트나 PHP등 일반적인 언어로 작성된 공개 게시판을 보유하고 있다[2]. 이들은 수시로 보안 업데이트의 등의 관리가 필요하지만 사용자는 일시적인 구축만 해놓고 방치하는 경우가 종종 발생한다. 그 예로 필자가 근무하는 대학의 웹 서버의 사용자중 보안 업데이트를 하지 않고 장시간 방치된 공개게시판을 통해 해커가 침입하여 서버 기능을 정지시키고 데이터를 유출하고 파기하는 사건이 발생하였다. 본 논문에서는 이렇게 사용되지 않는 계정을 로그 분석기를 통해 사용여부를 추출하고 일정기간 사용되지 않은 것으로 나타날 경우 서비스를 사용하지 못하도록 조치를 취하고자 한다. 단 중요한 개인 정보가 계정에 포함되어 있을 수 있기 때문에 계정을 삭제하지 않고 단지 특정 서비스들만을 정지 시킨다. 본 논문의 구성은 다음과 같다. 제 2장에서는 사용자 사용빈도측정과 문제점을 논하고 제 3장에서는 서비스 로그 분석기에 대해 설명한다. 제 4장에서는 로그 분석기의 성능을 평가하고 마지막 제 5장에서는 결론 및 향후과제를 논하고자 한다.

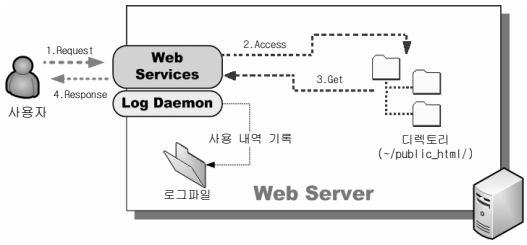
2. 사용자 사용빈도측정과 문제점

사용자가 터미널 서비스(SSH, Telnet)를 통하여 웹 서버에 접근을 하여 작업을 하였다고 가정을 하자. 우선 처음에 사용자가 TCP 프로토콜로 접속을 하였다면 터미널 서비스는 (그림 1)과 같이 사용자 인증을 통하여 사용자를 사용자 디렉토리에 접근하게 해준다. 이때 터미널 서비스는 그 접속 과정(어디서 어떤 IP에서 접속을 하였는지 등)을 로그 서비스에게 알려줌으로 로그가 기록되게 된다[3].



(그림 1) 터미널 서비스를 통한 웹 서버 접근

그러나 일정 규모의 웹서버 경우는 상당히 많은 사용자를 보유하고 있어 이 같은 경우 상당한 시간의 흘러 더 이상 사용하지 않는 사용자 경우 정지를 시키거나 삭제를 해야 하는 경우가 생기게 된다. 문제는 어느 기준을 두고 조치를 취할 것이냐가 관건이 되게 된다. 가장 간단한 방법은 디렉토리를 검색하여 최근의 데이터가 없을 경우, 그 사용자는 더 이상 사용을 하지 않는다고 가정을 하는 것 이다. 그런데 문제는 “사용자가 홈페이지나 사용자 디렉토리 내의 파일을 변경하지 않았다고 해서 사용하지 않고 있을까?”하는 것이다.



(그림 2) 웹 서비스를 통한 디렉토리(홈페이지) 접근

(그림 2)를 보게 되면 먼저 사용자가 웹 서버의 웹 서비스에게 특정 페이지를 요청하게 된다. 그러면 웹서비스는 요청한 디렉토리를 확인한 후 응답을 돌려주게 되는데 여기서 응답은 특정 서비스

페이지이거나 오류 코드 일 수 도 있다[4].

여기서 문제가 나오게 되는데 분명히 사용자 디렉토리는 사용되고 있는 상태이나 어느 파일 하나 변경된 것이 없으나 오랜 기간 사용하지 않는 사용자를 추려내는 작업에서 제외 대상이 될 수가 없게 된다. 다행히 (그림 2)에서 보게 되면 사용내역을 기록하는 것을 볼 수 있는데 이 기록은 특정 사용내역이 아니라 접속 상황이나 서비스의 상태 등을 모두 기록한다. 따라서 이 로그에는 사용자가 요청하고 응답한 내용 또한 들어가 있다

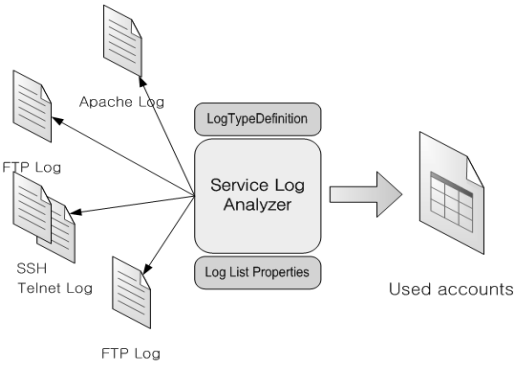
3. Service Log Analyzer(SLA)

3.1 로그를 통한 이용 사용자 검출

먼저 로그를 분석하기 위해서는 소스(Source)가 있어야 할 것이다. 정책에 따라 사용자에게 지원하는 서비스가 다루는데 본 논문에서는 다음 서비스가 사용 가능하다는 가정 하에 진행을 하도록 하였다.

- *웹 서비스(World Wide Web Service)* : 웹 서비스는 가장 흔히 사용되고 있는 서버의 정책이라고 할 수 있을 것이다. 본 논문에서는 아파치 2.x 버전을 사용하였다.
- *파일 전송 서비스(File Transport Service)* : 계정내의 파일이나 디렉토리를 관리하기 위한 서비스로 여기서는 홈페이지 내용을 갱신할 목적으로 사용한다고 가정하겠다.
- *터미널 서비스(Terminal Service)* : 터미널 서비스는 사용자가 서버에 접속하여 원격으로 여러 작업을 수행하고 파일 편집 작업이나 명령을 내리기 위한 텍스트 모드의 환경을 말하며 여기서는 SSH와 Telnet을 말하고 있다.

(그림 3)은 서비스 로그 분석기(SLA)의 위치를 보여 주는데 SLA가 어떠한 로그를 읽어 들여야 하는지는 Log List Properties라는 곳에 관리자가 직접 명시를 해주게 된다[5].



(그림 3) 서비스로그분석기(Service Log Analyzer)

3.1.1 Log List Properties

Log List Properties는 단순한 텍스트 파일로서 SLA가 시작시 읽어 들이는 로그 정보 파일이라고 할 수 있다. 각 프로퍼티들은 '='을 기준으로 키(key)와 값(value)으로 매칭 되게 되는데 항목들이 늘어날수록 SLA가 분석해야 하는 로그 양은 많아지게 됨으로 분석시간이 길어지게 된다.

(그림 4)은 Log List Properties의 내용이다. 여기서는 XXX_LOG형식을 사용하여 나타내었는데, XXX는 어떤 서비스인가를 나타내고 있으며 로그 형식 정의(Log Type Definition) 파일에 이 서비스가 어떤 형식을 사용하여 로그를 남기는지 반드시 정의해 놓아야 한다. 그렇지 않을 경우 이 항목은 무시하게 된다.

```
APACHE_LOG = /var/log/apache2/access_log
FTP_LOG = /var/log/auth.log
SSH_LOG = /var/log/auth.log
```

(그림 4) Log LIst properties

3.1.2 로그 형식 정의(Log Type Definition)

<표 1>은 log_type_def.properties 파일에 들어가는 항목을 나타낸 것으로 형식은 두 가지로 나누었다.

〈표 1〉 로그 형식 정의

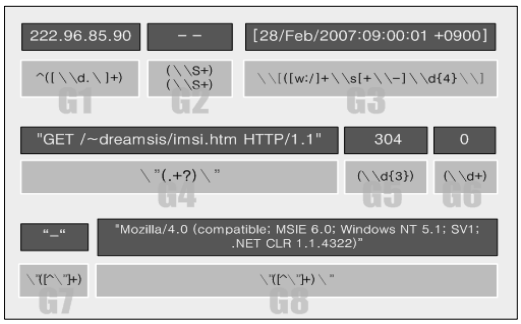
형식(TYPE)	서비스(Services)
Standalone	서비스에 종속된 로그 시스템을 이용하는 서비스 *(예) HTTP(Apache)*
Syslog (시스템 로그)	시스템에서 제공하는 로그 서비스를 이용하는 서비스 *(예) SSH, FTP, Telnet* 등

SLA는 이 형식을 프로퍼티 항목으로 넣어놓고 로그 리스트에 정의된 서비스들의 타입을 알아내도록 되어있다.

3.1.3 웹서버 로그 항목 분리 및 그룹화

범용으로 쓰이고 있는 아파치 서버의 경우 (그림 5)의 로그 형태를 가지고 있다.

먼저 각 서비스는 정의된 형식을 사용하여 로그를 저장하는데 이 형식을 정규식(Regular Expression) 패턴을 사용하여 항목들을 뽑아내도록 한다[6, 7].



(그림 5) 패턴을 이용한 필요 항목 추출 (아파치 LOG)

(그림 5)대로 정규 표현식을 이용하게 되면 그룹화가 가능하다는 것이다. 차후에 필요 항목만 따로 모아 여러 가지 분석을 할 수 있게 된다. 본 논문에서 필요한 사항은 사용 기록 이므로 G3, G4 그룹의 사용자 - '~'로 시작하는 부분이 사용자이다 - 만을 뽑아내어 객체를 생성하도록 하였다. 사용자가 요청하는 G4항목 경우 여러 가지 형식으로 저장되어 있을 수 있는데 본 논문에서는 다음 항목만을 범주에 포함시켜 작업을 하였다.

〈표 2〉 아파치 로그 분석 범주

클라이언트 사용 메서드	요청한 서버 자원 형식	프로토콜
GET	• 미포함 /, /index.htm, /home/x/index.htm, 그외 • 포함 /~account /~account/index.htm /~account/directory/file	구분안함
POST		
HEAD		
PROPFIND	범주 미포함	
OPTIONS		

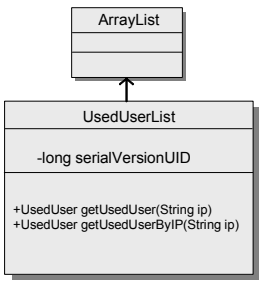
모든 로그가 분석대상이 될 수는 없다. 왜냐하면 모든 로그를 분석하게 되면 그만큼 비용이 들기 때문에 가급적 필요 없는 로그는 버리는 것이 작업 시간을 줄이는데 도움이 되기 때문이다. 또한 <표 2>에서 포함되는 항목을 제외한 요청된 서버 자원 형식은 해킹이나 사용자의 입력 오류 등에 기인하고 있으므로 차후에 다른 목적으로도 이용 가능 하다고 판단된다.

3.1.4 사용자객체 및 사용자리스트 객체의 생성

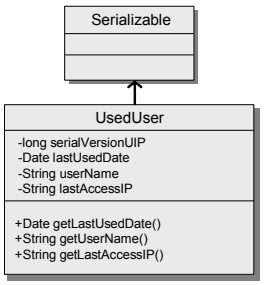
SLA가 처리하고 난 최종 결과물로 본 논문에서는 UUS(User Usage Shifter)로 보내기 위한 일종의 임시 장소라고 할 수 있다. 이는 MVC(Model, View, Controller)모델과 같이 View와 Control 사이에서 쓰이는 것과 비슷하다고 할 수 있겠다. 실제 본 논문의 구현에서는 단순히 사용되지만 차후 관리 프로그램이나 분석 프로그램 쪽으로 확장이 필요할 때 필요한 데이터가 될 가능성이 높으므로 따로 구현을 하도록 하였다.

(그림 6)의 UsedUserList 객체 경우는 Serializable을 구현하고 있는데 이것은 리스트(객체)를 파일 시스템에 저장을 해놨다가 필요시에 불러들일 수 있도록 함이다. (그림 8)은 SLA의 순차 다이어그램을 나타내는데 먼저 SLA가 시작이 되고나면

로그들을 분석하면서 사용된 사용자가 나타날 경우 (그림 7)형식을 가지는 사용된 사용자 객체를 생성, UsedUserList 객체에 저장을 한다.



(그림 6) 사용된 사용자 리스트



(그림 7) 사용된 사용자 객체

모든 로그 분석이 끝났을 경우 UUS에게 파일 형태의 UsedUserList를 넘기게 된다.

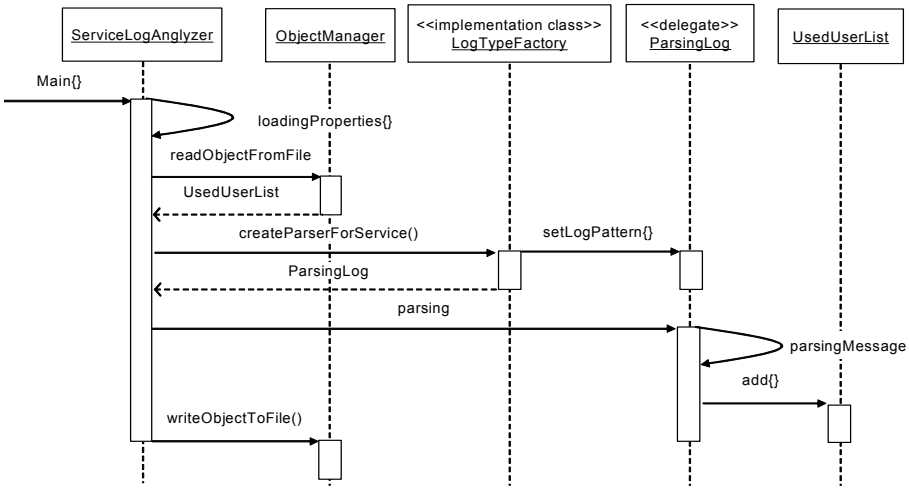
여기서 파일 형태로 저장하는 이유는 단순한 파라미터 형식으로 넘길 경우, 단지 UUS만이 사용하게 되지만 오브젝트 파일 형태로 저장할 경우 다른 관리프로그램에서도 불러 들여 사용 가능하기 때문이다.

3.2 Daemon과 Cron Job

많은 SLA 로그 분석은 시스템에 상당한 영향을 미치게 된다. 예로 학내 홈페이지 서버의 경우, 웹 서비스 하루 로그 크기는 약 40~60MB 정도 도달하는데 약 한 달분을 계산하면 약 1.5G 정도 한다고 볼 수 있다. 규정상 로그는 최대 약 6개월 최소 3개월 분량을 보관을 해야 하는데 이는 상당히 고가인 디스크 사용량의 상당부분이 아닐 수 없다.

현대의 로깅 시스템은 이렇게 늘어나는 로그들을 방치하는 것을 방지하기 위해 대부분 로그 로테이트(Log Rotate)라는 기술을 사용하여 로그가 대책없이 늘어나는 것을 방지하고 있다. 이것의 주요 기술은 날짜별로 로그를 끊어 내어내는 것이다. 더 나아가 압축을 하는 경우도 있다. 이점을 착안하여 SLA는 로그가 나눠지는 그 시간 이후에 클론(Cron) 작업(Job)에 등록, 특정 시간에 작동되도록 하고 있다. 일반적인 로그 로테이트가 끊나는 시점은 다음날로 넘어가는 시점으로 하고 또한 가장 시스템 사용률이 적은 새벽 3시를 기준으로 잡고 있다.

SLA는 독자적인(Standalone) 프로그램이면서도 메모리 상주 방식이 아닌 Cron에 의해서 작동되게 하였다. 만일 데몬 형식의 프로그램이 되게 된다면



(그림 8) SLA 순차 다이어그램

여러 서비스 로그를 계속 모니터링을 해야 하게 되는데, 이 경우 여러 문제점이 발생할 소지가 있기 때문이다. 또한 특정 클론 데몬은 스케쥴링을 하는데 있어 시스템 부하량, 시스템 가동유무 등을 판단하여 차후에 다시 작업을 할 수 있도록 해주는 것도 있는데 이런 특징들을 이용하면 관리자가 SLA를 현 시스템에 맞게 설정 할 수 있을 것이다.

4. 실험 평가

실험 평가를 위해 현재 운영 중인 시스템에 영향을 미치지 않기 위하여 Linux PC를 구성하고 다중 도메인 환경이라는 가정 하에 진행을 하였다.

4.1 가상 평가 시스템 구축

실질적으로 디렉토리 서버와 통신하는 부분은 UNIX와 Linux가 약간씩 차이가 있으나 구현 부분에 대해서는 언급하지 않았다. 왜냐하면 LDAP 프로토콜을 사용하여 통신하는 부분은 어느 플랫폼에서든 동일하기 때문이다[8].

〈표 3〉 평가 시스템 환경

분류	세부내용	
H/W	Pentinum4 2.8GB Memory 512MB HardDrive 160GB	
S/W	OS	Gentoo LinuxTM
	LDAP	OpenLDAPTM 2.3.33
	LDAP모듈	nss_ldap v254 pam_ldap v183
	WEB 서버	Apache 2.0.59-r2
	SSH 서버	OpenSSH 4.5_p1_r2
	FTP 서버	PureFTP 1.0.21-r1

<표 3>은 가상의 환경을 구현하기 위해 사용된 시스템이다. 실제 로그 데이터는 운영 중인 서버를 테스트하기는 어렵기 때문에 가상으로 구성을 하였고 또한 사용자 정보를 분석하기 위해 사용자 파일(file)에 관련된 사항들은 직접적으로 서버에 영향을 미치지 않으므로 운영 중인 서버 내에서 작업을 하였다. 다음으로 서비스 로그 분석 관련은 실제 운영 중인 교내 공용 웹 서버에서 로그파일을 평가 시스템에 복사하여 분석을 하고 평가하였으며 로그 파일은 1621개의 계정에 대한 2006년 10월에서 12월 3개월간의 자료를 가지고 분석을 하였다.

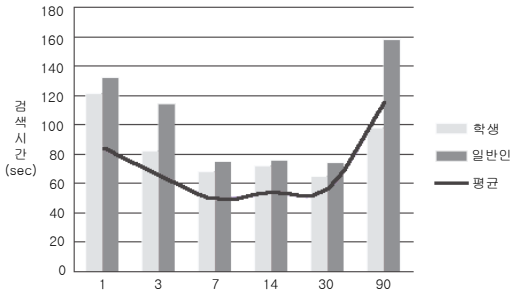
4.2 SLA와 UUS 구조의 비교 평가

4.2.1 xNIX[1] 검색 툴[2]을 이용한 사용된 사용자 계정검색

사용자 계정이 사용되었는지 알아내기 위해서는 파일 시스템이 변경되어있는가를 확인하면 된다.

검색도구로는 하지만 이는 파일 시스템을 검사하는 관계로 상당한 I/O 시간을 요구하게 되는데 (그림 9)는 각 3, 7, 14, 30, 90일 간 사용된 사용자 계정을 검색하는데 걸린 시간을 보여주고 있다.

디렉토리를 검색하는 시간을 보게 되면 일반인과 학생을 분리하여 검색을 할 경우 시간이 그다지 오래 걸리지 않지만 한꺼번에 검색할 경우 거의 두 배의 소요시간이 걸리는 것을 볼 수 있다.



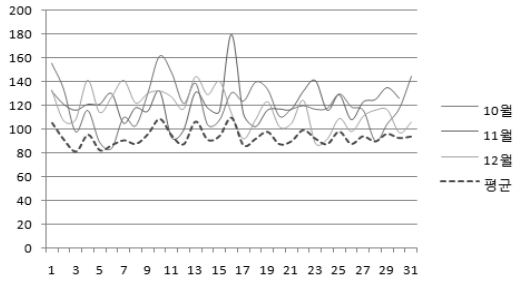
(그림 9) 기본 xNIX 시스템 툴을 이용한 사용된 사용자 검색

1) xNIX는 Unix계열과 Linux계열 OS의 총칭
2) time find/export -mtime 3 -exec ls -lrt {}\ ; Sed로 필터링

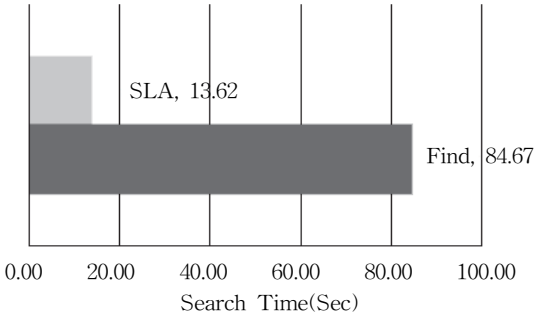
위와 같은 방법을 사용하여 특정 기간 동안 사용된 파일만을 검색, 어떤 사용자의 계정의 데이터가 새로 갱신되었는지 파악할 수 있다. 하지만 위의 명령어를 사용하게 되면 상당한 파일 리스트가 출력되므로 필터링을 거쳐 사용자 정보를 뽑아내야 한다. xNIX의 기본 명령어를 사용하여 사용된 계정을 찾는 부분에서 최대의 단점은 파일 시스템을 전부 검사를 해야 하기 때문에 사용되지 않는 파일까지 전부 검사를 한다는 것이다.

4.2.2 SLA를 사용한 사용자 사용도 분석

(그림 10)은 2006년 10월부터 12월까지의 서비스 사용자 수를 보여주는데 사용된 서버 전체 사용자 1621개의 계정 전체 중 18%도 못 미치는 사용률을 보여주고 있다. 이것은 서비스 정책의 미흡으로 생긴 문제이고 또한 관리의 문제이다.



(그림 10) SLA를 이용한 월별 사용자 서비스 사용도



(그림 11) 하루 동안 사용된 사용자 계정 검색 속도 비교

SLA와 xNIX 기본 툴을 이용한 검색 시간 비교(그림 11)은 SLA와 기본 xNIX find 명령어를 통한 하루 동안 사용된 사용자 계정을 알아내는 속도를 비교한 것이다. 파일 시스템을 검색한 것보다 로그를 통하여 사용된 사용자 계정을 검색하는 시간이 훨씬 더 빠른 것을 볼 수 있다. 이는 파일 시스템을 사용하는 서버의 부하에도 영향을 미치는 것으로 짧은 시간에 검색을 마치는 것이 서버 사용률에도 영향을 덜 미치게 된다.

5. 결 론

본 연구에서는 사용되지 않는 사용자계정으로 인한 보안의 위협을 해결하기 위해 사용된 사용자 계정을 서비스 로그 분석기를 통해 로그를 분석하여 일정기간 사용되지 않는 계정을 추출하고 해당 서비스를 사용하지 못하도록 하였다. 이를 위해 로그의 패턴에서 웹 서비스의 사용되는 영역을 추출한 결과 약 18%만 일정기간 사용되는 것으로 파악되어 나머지 82%의 계정에 대해 보안을 유지할 수 있었다. 또한 유닉스 관리툴을 이용할 경우 사용자의 사용정도를 파악하는데 많은 시간이 소요됐으나 제안한 분석기를 통해 분석할 경우 약 75%의 속도 증진을 볼 수 있었다. 향후 로그 패턴 분석기를 다양한 서비스에 접목시키고 보다 쉽게 활용 가능하도록 GUI환경의 시스템이 구축되어야 할 것이다.

참 고 문 헌

[1] “유닉스 시스템 보안 취약성 분석 및 진단에 관한 연구”, 한국전산원, 1995.

[2] 박영호, “인천지역 학내망 보안에 대한 실태 분석 및 개선방안”, 인천대학교 석사학위논문, 2002.

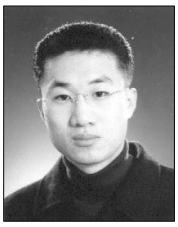
[3] HP, “Distributed Systems Administration Utilities”, pp. 70-73. 2005.
[4] “UNIX Desk Reference”, Peter Dyson, Sybex, 1997.
[5] 이대형, 이진관, 정규철, 장혜숙, 이종찬, 박기홍, “LDAP 기반의 사용자 계정관리 시스템 구현”, 인터넷정보학회 춘계학술대회, p. 2, 2006.
[6] R. BrandtSteven. “Regex Recipes”, Regular Expressions in Java http://www.javaregex.com, 2004.
[7] “Linux-Pam”, Kernel.org, http://www.kernel.org/pub/linux/libs/pam, 2005.
[8] Rob Weltman Dahbura Tony, “LDAP Programming With JAVA”, Addison-Wesley, pp. 4-31, 2000.

정 규 철

1996년 군산대학교 컴퓨터과학과 (이학사)
1999년 군산대학교 컴퓨터과학과 (이학석사)
2006년 군산대학교 컴퓨터과학과 (이학박사)
1999년～현재 군산대학교 컴퓨터과학과 강사

이 진 관

1996년 군산대학교 컴퓨터과학과(이학사)
2002년 군산대학교 컴퓨터과학과(이학석사)
2007년 군산대학교 컴퓨터과학과(이학박사)
2006년～현재 군산대학교 컴퓨터과학과 강사



이 대 형

1996년 군산대학교 컴퓨터과학과(이학사)
2007년 군산대학교 컴퓨터과학과(이학 석사)

장 혜 숙

2000년 군산대학교 컴퓨터과학과(이학석사)
2004년～현재 군산대학교 컴퓨터과학과(박사과정)

이 종 찬

1994년 군산대학교 컴퓨터과학과(이학사)
1996년 숭실대학교 컴퓨터학과 (이학석사)
2000년 숭실대학교 컴퓨터학과 (공학박사)
2000년～2005년 한국전자통신 연구원 선임연구원
2005년～현재 군산대학교 컴퓨터과학과 조교수

박 기 홍

1986년 숭실대학교 전자계산학과 (이학사)
1986년 숭실대학교 전자계산학과 (공학석사)
1995년 일본 토쿠시마대학교 지능정보과학과(공학박사)
1997년～1998년 영국 Middlesex Univ 객원 교수
2004년～2006년 NURi 사업 텔레메틱스 인력양성 사업단(군산대) 단장
2004년～현재 군산대학교 컴퓨터과학과 교수