

접근통제서버를 활용한 보안시스템 구현

팀 명 I. O. S
(Innovation of security)

지도교수 유 승 재 교수님

팀 원 심 재 완
이 동 재
이 주 곤
김 광 혁
서 유 현

2014. 06

요 약 문

1. 연구 배경

IT기술이 급속도로 발전함에 따라 정보의 가치가 증대되고 있습니다. 많은 데이터 즉 빅 데이터를 이용하여 사업의 지표로 삼는다거나 빅 데이터를 이용하여 가까운 미래의 관심사를 예측할 수도 있습니다. 즉, 정보를 이용하여 많은 사람들의 관심사인 돈을 벌수가 있는 것입니다.

또한 정보의 역효과로 인하여 많은 개인 유출 사고가 빈번히 발생되고 있습니다. 많은 사람들이 다른 사람의 정보를 불법적으로 얻어 그 정보를 이용하여 돈을 버는 식의 정보의 역효과의 발생도 더욱 빈번히 발생되고 있는 실정입니다.

이러한 정보의 역효과를 방지하기 위하여 접근이 허용되어 있는 정당한 사용자만이 정보에 접근할 수 있도록 하는 필요성이 대두되고 있습니다. 이러한 현상에 맞춰 정보의 기밀성을 보장하기 위한 방법을 생각하다 접근통제 서버의 필요성을 느끼게 되었습니다.

2. 연구 목적 및 필요성

요즘 들어 정보의 가치가 증대됨에 따라 정보에 대한 많은 보안 공격들이 실행되고 있는 시점입니다. 따라서 특정 서버를 보호하기 위한 방법으로 허용된 사용자가 아닌 사람의 접근을 통제할 필요성이 증가되고 있습니다. 그러한 무분별한 접근을 제어하고 통제하기 위하여 접근통제 서버를 구축하여야 할 필요성이 대두되고 있습니다.

또한 관리자들의 원활한 관리를 돕기 위하여 관리자 페이지를 구축하여 관리자들이 사용자들의 로그기록, 특정 명령어 사용 통제, 접속시간 등을 관리자 페이지에 입력하여 관리자가 특정 서버를 원활하게 관리할 필요성이 있다고 생각합니다.

3. 연구 내용

접근통제 서버란 기록을 담고 있는 정보를 보호하기 위하여 기록에 대한 접근을 제한하거나 허용하는 것을 말합니다. 정보를 담고 있는 서버를 만일 무분별하게 접근할 수 있게 된다면 그 정보의 가치는 하락하게 될 것입니다. 따라서 그러한 정보를 보호할 방법이 필요하다는 생각에 접근통제 서버를 구축하게 되어 무분별한 접근을 막고 허용된 사용자들에게만 접근을 허용하여 정보의 가치를 증대시키고자 하였습니다.

뿐만 아니라 관리자 페이지를 구축함으로써 인하여 특정 서버에 접근한 사용자들의 시간과 사용한 명령어의 출력 또한 위험한 명령어의 통제를 통해 서버의 보안을 강화하였으며 만일 보안에 문제가 발생될 시 사용자들의 사용여부를 확인하여 빠른 대응이 가능하도록 노력하였습니다.

4. 연구 결과

요즘 들어 많은 정보의 가치가 증대됨에 따라 많은 정보 유출 사건이 발생하고 사회적으로 정보 유출에 대한 관심이 증대되고 있는 시점입니다. 그러한 현상에 대응하기 위해 정보 유출에 대한 대응으로 접근통제 서버를 구축하여 무분별한 정보에 대한 접근을 막게 되어 특정 정보의 무분별한 접근을 막고 허용된 사용자만이 접근할 수 있게 하여 정보의 가치를 높이고 정보의 기밀성을 보장할 수 있었습니다.

관리자의 입장에서는 관리자 페이지를 통하여 보다 쉽게 사용자들을 통제할 수 있었으며 만일 정보의 유출 사건이 발생할 경우에는 빠르게 대응할 수 있었습니다.

목 차

요 약 문	1
I. 서론	3
1. 접근통제의 개요	4
2. 연구 목적	5
3. 연구 개요	5
4. 개발 구성	6
II. 접근통제 시스템	7
1. 접근통제서버 설치 및 구현	7
2. 소스코드	9
III. 연구 결과	19
1. 모니터링 상황	19
IV. 결론	21
1. 최종 연구 결과 보고	21
V. 참고문헌	21
VI. 부록	22
1. 발표ppt	22

I. 서론

I. 접근통제(Access Control)의 개요

가. 접근통제의 정의

- 자원에 대한 비인가된 접근을 감시하고, 접근을 요구하는 이용자를 식별하고, 사용자의 접근 요구가 **정당한 것인지를 확인, 기록하고**, 보안정책(Security Policy)에 근거하여 접근을 승인하거나 거부함으로써 **비인가자에 의한 불법적인 자원접근 및 파괴를 예방하는** 하드웨어, 소프트웨어 및 행정적인 관리(Administration)을 총칭함
- 접근통제는 각 자원에 대한 기밀성, 무결성, 가용성 및 합법적인 이용과 같은 정보보호 서비스에 직접적으로 기여하게 되며 이러한 서비스들의 권한부여를 위한 수단

나. 접근통제의 3요소

구분	내 용
접근통제 정책	- 시스템 자원에 접근하는 사용자의 접근 모드 및 모든 접근제한 조건 등을 정의
접근통제 메커니즘	- 시도된 접근 요청을 정의된 규칙에 대응시켜 검사함으로써 불법적 접근을 방어
접근통제 보안모델	- 시스템의 보안요구를 나타내는 요구명세로부터 출발하여 정확하고 간결한 기능적 모델을 표현

다. 접근 통제 조건

- VDC(Value-Dependent Control)
 - 객체의 기밀성이 현재 저장된 값에 따라서 다양
- MUC(Multi-User Control)
 - 다수 사용자(주체)가 연합하여 요청할 경우의 접근통제정책을 지원수단
- CBC(Context-Based Control)
 - 외부적인 요소에 의존하여 객체에 접근을 제어하는 정책 (하루의 특정시간, 사용자의 위치 등)

라. 접근통제 메커니즘]

- **ACL(Access Control List)**
 - 어떤 주체가 객체에 어떤 행위를 할 수 있는지 표현
- CL(Capability List)
 - 주체에 대하여 저장된 접근 허가 목록
- SL(Security Label)
 - 객체에 부여된 보안 속성 정보의 집합

2. 연구 목적

최근 인력의 부주의로 인한 보안 사고가 잇따라 발생하면서 인력의 권한 관리 및 시스템 접근 통제 정책의 중요성이 부각되고 있다.

특히 서버 환경의 확산과 더불어 서버의 자료와 보안이 크게 중요시 되고 있다.

접근통제서버는 이러한 문제를 해결하고자 비인가 사용자의 서버 접속을 불허하고 중요 자원에 대한 사용자의 무분별한 접근을 통제하며 작업내역을 관리자가 확인하는 접근통제서버를 구축하여 서버 보안체제를 구현하고자 한다.

관리자는 웹페이지를 통해 사용자가 서버에 접속한 상황을 실시간으로 확인할 수 있고 로그아웃한 사용자에 대한 로그도 확인할 수 있다.

또한 사용자가 현재 터미널에서 작업하는 내용을 확인하고 사용자가 사용하는 명령어를 확인함으로써 의심가는 사용자에 대한 제제를 가할 수 있다.

3. 연구 개요

접근통제의 목적은 컴퓨팅 자원, 통신 자원 및 정보자원 등에 대하여 허가되지 않은 접근을 방어하는 것이다. 허가되지 않은 접근이란 불법적인 자원의 사용, 노출, 수정, 파괴와 불법적인 커맨드의 발행을 포함하고 있다. 즉, 접근통제는 각 자원에 대한 기밀성, 무결성, 가용성 및 합법적인 이용과 같은 정보보호 서비스에 직접적으로 기여하게 되며 이러한 서비스들의 권한부여를 위한 수단이 된다.

대부분 컴퓨터 시스템의 사용자는 시스템을 사용하기 위하여 식별과 인증이라고 하는 검사과정을 통하여 시작된다. 식별과 인증은 각 시스템 자원을 보호하기 위한 외부의 1차적인 보호계층이다. 접근통제 결정은 요청자의 신분이 완전히 인증되기 전까지는 수행될 수 없다. 인증의 강도는 접근통제의 개별적인 정책에 의존적인 부분일 수 있다. 즉, 인증의 강도에 따라 자원 접근대상 및 접근모드를 제한하는 정책 시행이 가능하므로 고유의 접근통제 정책을 위배하지 않는 조화된 보안정책 추진전략이 필요하다.

인증이 성공하면 각 시스템 지원에 대한 사용자의 요청을 보안정책이 적용된 접근통제 절차에 따라서 허용여부를 인가 받는다. 접근통제 시스템의 분석은 기능적으로 3가지 요소적 측면으로 구분하여 관찰할 수 있다.

시스템 자원에 접근하는 사용자의 접근 모드 및 모든 접근제한 조건 등을 정의하는 접근통제 정책

시도된 접근 요청을 정의된 규칙에 대응시켜 검사함으로써 불법적 접근을 방어하는 접근통제 메커니즘

시스템의 보안요구를 나타내는 요구명세로부터 출발하여 정확하고 간결한 기능적 모델을 표현하는 접근통제 관련 보안모델

개방형 정보통신망에서 접근통제는 실제의 어떤 개방 시스템에 있는 물리적 실체, OSI계층의 특정한 실체, 파일과 같은 논리적 실체, 그리고 일반적 사용자와 같은 다양한 형태의 실체들과 연관된다. 접근통제를 위한 일반적 모델에서 능동적인 실체의 집합을 개시자(initiator) 또는 주체(subject)라고 하며, 수동적 자원의 집합을 타겟(target) 또는 객체(object)라고 부른다. 그러나, 본 연구에서는 주체/객체 용어가 현대의 컴퓨터와 통신 분야에서 널리 사용되고 있는 개념과 혼동될 수 있으므로 개시자/타겟으로 통일하여 사용한다. 그러나, 특정 논문 등의 참고 문헌에서 주체 및 객체로 사용한 내용을 인용할 경우는 주체/객체로 일부 사용한다.

접근통제의 결정은 어떤 개시자가 어떤 타겟에 대하여 어떤 목적을 갖고, 어떤 조건하에서 접근할 수 있는지를 다루는 문제이다.

즉, 이러한 결정은 접근통제정책에 반영이 되고, 접근 요청은 접근 정책을 시행하는 접근통제 메커니즘을 통하여 시행된다.

접근통제정책은 다음과 같이 다양한 형태로 서술될 수 있다.

권한부여의 과정에서 어떤 정책은 기관의 부서별로 모든 결정이 제어되거나, 또는 특정 타겟에 대하여 개인별 권한부여가 서술될 수 있다.

사용자 및 타겟들이 공통의 처리를 위하여 함께 그룹을 형성하여 서술될 수 있다.

어떤 정책이 시스템 요소에 의하여 강제적으로 시행될 수 있는 일반적 규칙들로 서술될 수 있다.

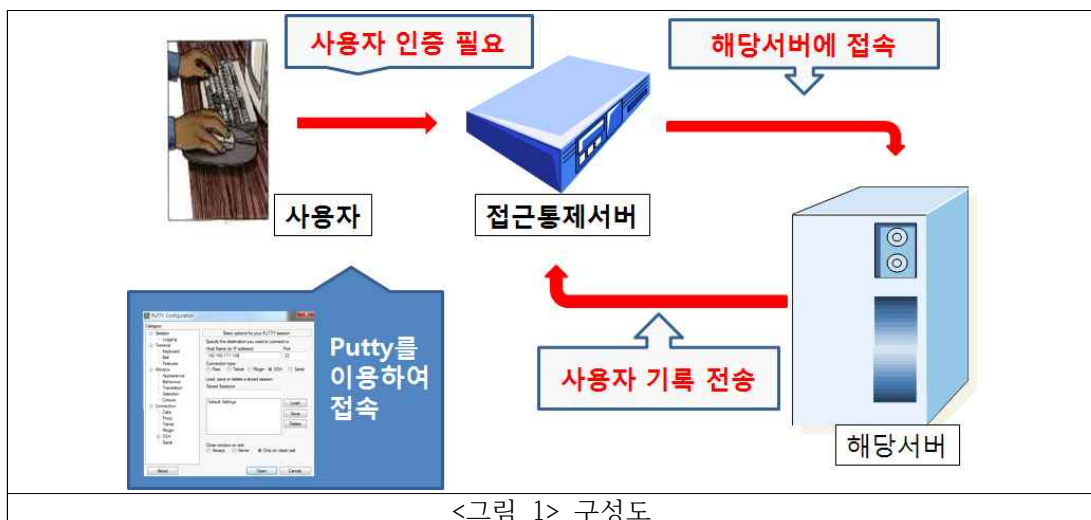
미 국방성에서 기밀 분류된 방법으로부터 유래하는 접근통제 정책은MAC(Mandatory Access Control)과 DAC(Discretionary Access Control)으로 널리 알려져 있다. MAC정책은 자동적으로 시행되는 어떤 규칙에 기반하고 있다. 그러한 규칙을 실제로 시행하기 위하여 사용자와 타겟에 대해서 광범위한 그룹 형성이 요구된다. DAC 정책은 특별한 사용자별로 정보에 대한 접근을 제공하고 추가적 접근통제를 그 사용자에게 일임한다.

OSI 보안 구조에서는 MAC/DAC 용어를 사용하지 않고 신분-기반(identity-based)과 규칙-기반(rule-based) 정책으로 구분하고 있다. 실제적인 목적에 있어서 신분-기반과 규칙-기반 정책은 각각 DAC 및 MAC 정책과 동일하다.

신분-기반 정책은 개인-기반(Individual-Based Policy: IBP)과 그룹-기반(Group-Based Policy: GBP) 정책을 포함한다. 한편, 규칙-기반 정책은 다중-단계(Multi-Level Policy: MLP)와 부서-기반(Compartment-Based Policy: CBP)정책을 포함한다. 이외에 직무-기반(role-based) 정책은 신분-기반과 규칙-기반 정책의 양쪽 특성을 갖고 있다. 또한, 이러한 정책들은 서로 연합될 수 있으며, 임계값 의존 제어(Value-Dependent Control: VDC), 다중 사용자 제어(Multi-User Control: MUC) 및 배경-기반 제어(Context-Based Control: CBC) 등의 추가적 수단을 사용하여 제한 될 수 있다.

접근통제 메커니즘은 접근 행렬의 열을 표현하는 ACL(Access Control List), 접근 행렬의 행을 표현하는 CL(Capability List), 제어 대상에 레이블을 붙이는 SL(Security Label)을 기본적으로 생각할 수 있다. 그리고 이러한 3 가지 정보를 종합적으로 생각하는 통합정보 메커니즘, 각 파일에 접근통제를 위한 비트들을 부가하여 제어하는 Protection Bit(PB), 파일의 접근 권한을 검증하기위한 패스워드 등의 기법이 있다.

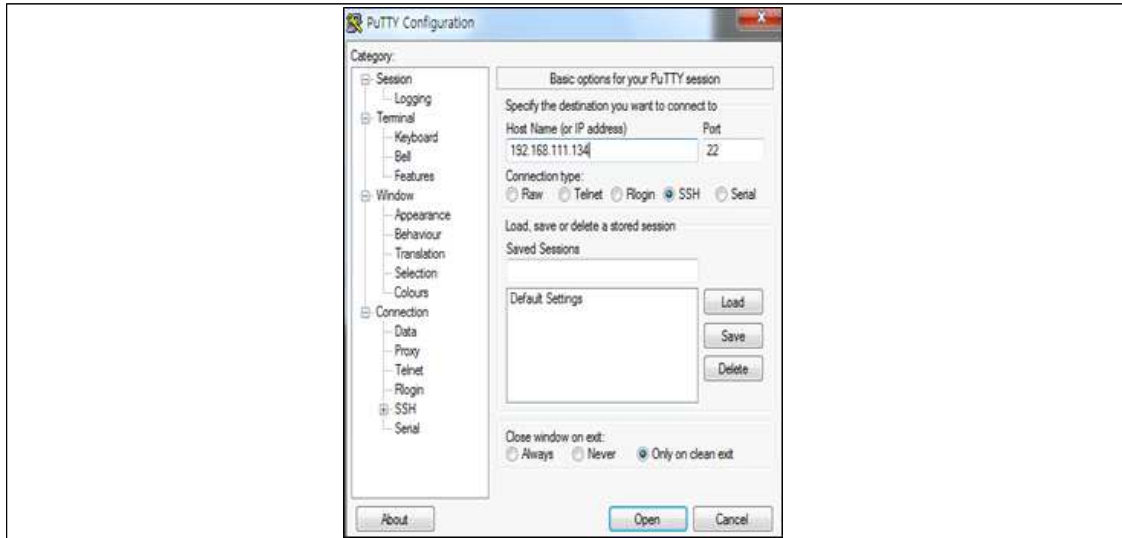
4. 개발 구성



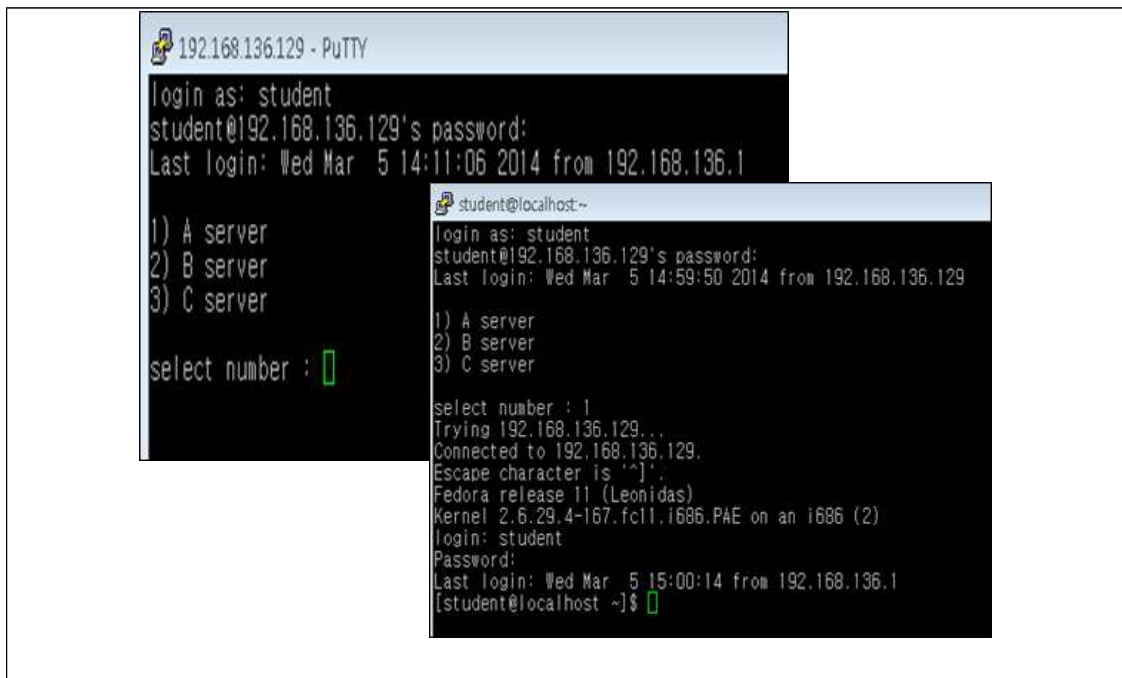
II. 접근통제 시스템

1. 접근통제서버 설치 및 구현

1. 윈도우에서 putty를 사용하여 접근통제서버로 접속



2. 통제서버로 접속하게 되면 sh프로그램이 자동으로 실행되면서 서버목록을 보여줌



3. 접근통제서버는 lsyncd를 통해 서버에서 데이터를 실시간으로 백업

```
[root@localhost ~]# lsyncd --no-daemon /home/backup2/ 192.168.131.131:/gon
syncing /home/backup2/ -> 192.168.131.131:/gon
Fri Mar 14 08:17:14 2014: Starting up
Fri Mar 14 08:17:14 2014: watching /home/backup2/
Fri Mar 14 08:17:15 2014: ---entering normal operation---
Fri Mar 14 08:17:44 2014: CREATE of test in /home/backup2// --> 192.168.131.131:/gon/
Fri Mar 14 08:17:48 2014: CLOSE_WRITE of test in /home/backup2// --> 192.168.131.131:/gon/
Fri Mar 14 08:28:44 2014: CREATE of test4 in /home/backup2// --> 192.168.131.131:/gon/
Fri Mar 14 08:28:48 2014: CLOSE_WRITE of test4 in /home/backup2// --> 192.168.131.131:/gon/
```

```
[root@localhost backup2]# cat > test5
[root@localhost backup2]# ls
test test2 test3 test4 test5
[root@localhost backup2]#
```

```
[root@localhost gon]# ls
test test2 test3 test4 test5
[root@localhost gon]#
```

4. 관리자는 웹을 통해 현황관리, 서버관리, 사용자관리, 시스템관리를 확인

2. 관리자 페이지

현황 관리 서버 관리 사용자 관리 시스템 관리

현황판

총 서버수	총 사용자 수
3	3

2. 관리자 페이지

현황 관리 서버 관리 사용자 관리 시스템 관리

서버 상태 서버 현황

서버 상태

서버명	상태
A서버	정상
B서버	정상
C서버	정상

2. 관리자 페이지

현황 관리 서버 현황 사용자 관리 시스템 관리

서버 상태 서버 현황

서버 현황

NO	서버명	분류	OS	IP	비고
1	파일서버	공유	LINUX	10.26.0.216	
2	개발서버	개발	LINUX	10.26.0.100	
3	웹서버	개발	LINUX	10.26.0.45	

2. 관리자 페이지

현황 관리 서버 관리 사용자 관리 시스템 관리

사용자 접속로그 사용자 현황 사용자 지역 사용자 접속로그

사용자 접속로그

사용자명	접속시간	접속명	IP	로그아웃 시간
fedora	10:21	Fri Apr 18	10.0	10:21
fedora	21:45	Wed Apr 16	10.0	10:24
test1	20:59	Wed Apr 16	10.26.0.45	20:59
test1	20:57	Wed Apr 16	10.26.0.45	20:58
test4	20:57	Wed Apr 16	10.26.0.45	20:57
test4	20:56	Wed Apr 16	10.26.0.45	20:57
test4	20:54	Wed Apr 16	10.26.0.45	20:54
test4	20:52	Wed Apr 16	10.26.0.45	20:54
test4	20:51	Wed Apr 16	10.26.0.45	20:52
fedora	20:51	Wed Apr 16	10.26.0.45	20:51
fedora	20:50	Wed Apr 16	10.26.0.45	20:51
test4	20:49	Wed Apr 16	10.26.0.45	20:50
test4	20:47	Wed Apr 16	10.26.0.45	20:49
test4	20:46	Wed Apr 16	10.26.0.45	20:47
fedora	20:45	Wed Apr 16	10.26.0.45	20:46
fedora	20:44	Wed Apr 16	10.26.0.45	20:45
fedora	20:43	Wed Apr 16	10.0	10:24
fedora	20:43	Wed Apr 16	10	10:24
root	16	boots Wed Apr	boots	(5=21139)
root	10:18	Wed Apr 16	10.26.0.190	0x0x0x
vtmp	10:10:04	Wed Apr 16	Wed	

2. 소스코드

2-1. 메인 페이지

```
<!doctype html>
<html lang="ko">
<head>
    <title>test</title>

<!--html5 기본구조 css--> <link rel="stylesheet" href="html/master.css" type="text/css"
media="screen" />
<!--상단 메뉴 css--> <link rel="stylesheet"
href="html/style.css" media="screen">

</head>
<body>

<!-------
= 로고와 홈페이지 이름
----->
    <header>
        <h1> <IMG SRC="html/img/logo.jpg" WIDTH="40"BORDER="0" ALT="">
        &nbsp;관리자 페이지 </h1>
    </header>

<body>
<!-------
= 메뉴
----->
<nav>
    <ul class="menu">

    <li><a href="top_main.html">현황 관리</a></li>
    <li><a href="#">서버 현황</a>
    <ul>
        <li><a href="top_main.html">관리</a></li>
        <li><a href="#" >관리 2</a></li>
        <li><a href="#">관리 3</a></li>
    </ul></li>
    <li><a href="#">사용자 관리</a>
    <ul>
        <li><a href="top_list.php" >사용자 현황</a></li>
        <li><a href="top_board.php" >사용자 이력</a></li>
        <li><a href="top_list2.php" >사용자 접속로그</a></li>
    </ul>

    </li>
    <li><a href="#">시스템 관리</a>
    <ul>
```

```
<li><a href="top_chmod.html">명령어 통제</a></li>
<li><a href="#" >시스템 2</a></li>
<li><a href="#">시스템 3</a></li>
</ul></li>

</ul>
</nav>
<br /><br />

<span class="spantitle">현황</span>
<div align="center"><br>
<table width="450" height="109" border="1">
<tr>
<td width="217"><div align="center"><strong>총 서버수</strong></div></td>
<td width="217"><div align="center"><strong>총 사용자 수</strong></div></td>

</tr>

<tr>
<td width="217"><div align="center">3</div></td>
<td width="217">&nbsp;</td>

</tr>
</table>
</div>

</body>
</html>
```

2-2. 서버리스트

```
<!doctype html>
<html lang="ko">
<head>
    <title>test</title>

<!--html5 기본구조 css-->    <link rel="stylesheet" href="html/master.css"
type="text/css" media="screen" />
<!--상단 메뉴 css-->                <link rel="stylesheet"
href="html/style.css" media="screen">

</head>
<body>

<!-------
= 로고와 홈페이지 이름
----->

    <header>
        <h1> <IMG SRC="html/img/logo.jpg" WIDTH="40" BORDER="0" ALT="">
        &nbsp;   관리자 페이지 </h1>
    </header>

<body>
<!-------
= 메뉴
----->

<nav>
    <ul class="menu">

        <li><a href="index.html">현황 관리</a></li>
        <li><a href="#">서버 현황</a>
        <ul>
            <li><a href="#" target="main">관리</a></li>
            <li><a href="#">관리 2</a></li>
            <li><a href="#">관리 3</a></li>
        </ul></li>
        <li><a href="#">사용자 관리</a>
        <ul>
            <li><a href="list.php" target="main">사용자 현황
        </a></li>
            <li><a href="history.php" target="main">사용자 이력
        </a></li>
            <li><a href="test.html" target="main">접근 권한</a></li>
```

```

        </ul>

        </li>
<li><a href="#">시스템 관리</a>
        <ul>
                <li><a href="chmod/chmod.html" target="main">명령어 통제
</a></li>
                <li><a href="#" >시스템 2</a></li>
                <li><a href="#">시스템 3</a></li>
        </ul></li>

</ul>
</nav>
<div id="mainContent">
<br />
<span class="spantitle">사용자 접속이력</span>
<br /><br />

<table width="900" border="0" cellspacing="0" cellpadding="0"
style="padding-left:0px;">
<tbody>
        <tr>
                <td>
                         </td>
        </tr>
        <tr>
                <td style="padding:0px; background-color:#565656;
vertical-align:absmiddle; text-align:middle;
background-image:url('./imgs/board_R02.gif');" align="middle">
                        <table cellpadding="0" cellspacing="0" width="900"
border="0">
                                <tbody>
                                        <tr style="background-color:#52c9b5;">
                                                <td width="7"></td>
                                                <td width="171"
class="boardtitlebar">사용자 id</td>
                                                <td width="306"
class="boardtitlebar">접속시간</td>
                                                <td width="208"
class="boardtitlebar">접속일</td>
                                                <td width="199"
class="boardtitlebar">IP</td>
                                                <td width="199"

```

```

class="boardtitlebar">로그아웃 시간</td>
                                <td
                                width="9"></td>
                                </tr>
<tr class="listline">
<td width="7"></td>
<td width="171" class="mtitle">
<? $res=shell_exec("last -R | awk '{print $1}'"); echo '<pre>'.$res.'</pre>'; ?>
</td>
<td width="306" class="muser">
<? $res=shell_exec("last -R | awk '{print $6}'"); echo '<pre>'.$res.'</pre>'; ?>
</td>
<td width="208" class="muser">
<? $res=shell_exec('LC_TIME="en_US.UTF-8" last -R | awk W'{print $3,$4,$5}W');
echo '<pre>'.$res.'</pre>'; ?> </td>
<td width="199" class="muser">
<? $res=shell_exec("last | awk '{print $3}'"); echo '<pre>'.$res.'</pre>'; ?>
</td>
<td width="199" class="muser">
<? $res=shell_exec("last -R | awk '{print $8}'"); echo '<pre>'.$res.'</pre>'; ?>
</td>
<td width="9"></td>
                                </tr>
                                <tr align="center">
                                <td colspan="6">
                                <div id="button"></td>
                                </tr>
                                </tbody>
                                </table>
                                </td>
                                </tr>
                                <tr >
                                <td align="center">
                                
                                </td>
                                </tr>
</tbody>
</table>
</div>
</body>
</html>

```

2-3. 사용자 로그인 페이지

```
<!doctype html>
<html lang="ko">
<head>
    <title>test</title>

<!--html5 기본구조 css-->    <link    rel="stylesheet"    href="html/master.css"
type="text/css" media="screen" />
<!--상단 메뉴 css-->                <link                rel="stylesheet"
href="html/style.css" media="screen">

</head>
<body>

<!-------
= 로고와 홈페이지 이름
----->
    <header>
        <h1>    <IMG    SRC="html/img/logo.jpg"    WIDTH="40"BORDER="0"
ALT=""> &nbsp;   관리자 페이지 </h1>
    </header>

<body>
<!-------
= 메뉴
----->
<nav>
    <ul class="menu">

<li><a href="index.html">현황 관리</a></li>
<li><a href="#">서버 현황</a>
<ul>
        <li><a href="#" target="main">관리</a></li>
        <li><a href="#" >관리 2</a></li>
        <li><a href="#">관리 3</a></li>
    </ul></li>
<li><a href="#">사용자 관리</a>

    <ul>
        <li><a href="list.php" target="main" >사용자 현황</a></li>
        <li><a href="history.php" target="main">사용자 이력
</a></li>

        <li><a href="test.html" target="main" >접근 권한</a></li>
    </ul>
</nav>
```

```
        </li>
<li><a href="#">시스템 관리</a>
    <ul>
        <li><a href="chmod/chmod.html" target="main">명령어 통
제</a></li>
        <li><a href="#" >시스템 2</a></li>
        <li><a href="#">시스템 3</a></li>
    </ul></li>

</ul>
</nav>
<style type="text/css">
.title
{
padding-top: 3px;
font-family: 돋움,verdana,arial;
font-size: 14px;
color: #000000;
background: lightcoral;
font-weight:900;
}

.holsu
{
padding-top: 3px;
font-family: 돋움,verdana,arial;
font-size: 13px;
color: #000000;
background: gainsboro;
}

.jjagsu
{
padding-top: 3px;
font-family: 돋움,verdana,arial;
font-size: 13px;
color: #000000;
background: white;
}

.input_80
{
width:80%;
color:#333333;
```

```

font-size:11px;
font-family:돋움,Dotum,AppleGothic,sans-serif;
height:20px;
border-top:1px solid #B0B0B0;
border-right:1px solid #E1E1E1;
border-bottom:1px solid #E1E1E1;
border-left:1px solid #B0B0B0;
}
</style>

<!-- 스타일 시트 작성 끝 -->

<!-- @ 오픈/저장할 파일 선택 ----- -->

<table cellpadding='10' cellspacing='1' border='0' align='center'>
<!-- @ 오픈된 데이터 리스트 ----- -->
<tr><br>
</tr>
<tr>

<? // php 시작!!!!
echo("
<td><b><h3><font color='black'> 명령어 사용내역 </font></h3></b></td>
</tr>");
?>

<tr>
<table class='input_80'align='center'>
<?
$filedir_full = 'history'; // 출력할 파일 경로명

if(!file_exists($filedir_full)) return: //파일이 없을경우 처리하지 않음

$fp = fopen($filedir_full, "rt");

```



```

$linecnt = 0;    //파일 줄 카운트
$datacnt = 0;    //데이터 카운트

while(!feof($fp)) {          //파일길이만큼 반복

    $buffArr = array();

    $buff = fgets($fp, 2048);
                //fgets->문자열을 읽음
    $buffArr = explode(",", $buff);    //.단위로 데이터 잘라내기 ex)
hostname,cpu,output,...

    $cnt=sizeof($buffArr);

    /*카운트 시작*/
    if($datacnt==0 || $datacnt==1) { //첫번째줄과 두번째줄은 count 패스
    }else{
        if($cnt<$i){ //마지막줄 출력방지를 위해 사용. /

            break;
        }
        $linecnt++; //카운트 시작
    } //if end

    /*리스트 출력 시작_첫번째줄 - 첫번째줄과 두번째줄은 카운트 안함.*/
    if($datacnt==1){
        echo ("
        <tr>

        ");
    }else if($datacnt==1){ //두번째줄은 무의미한 =====이런표시라 출력안하고 건너뛰기

    }else if(($linecnt%2)==1){ //홀수일경우 gainsboro색으로 교차 출력
        echo ("
        <tr>
        <td class='holsu'>$linecnt</td>
        ");
    }else{ //짝수일경우 white색으로 교차 출력
        echo ("
        <tr>

```

```

        <td class='jjagsu'>$linecnt</td>
    ");
} //if end

/* for문 시작 */
for($i=0; $i<sizeof($buffArr); $i++){

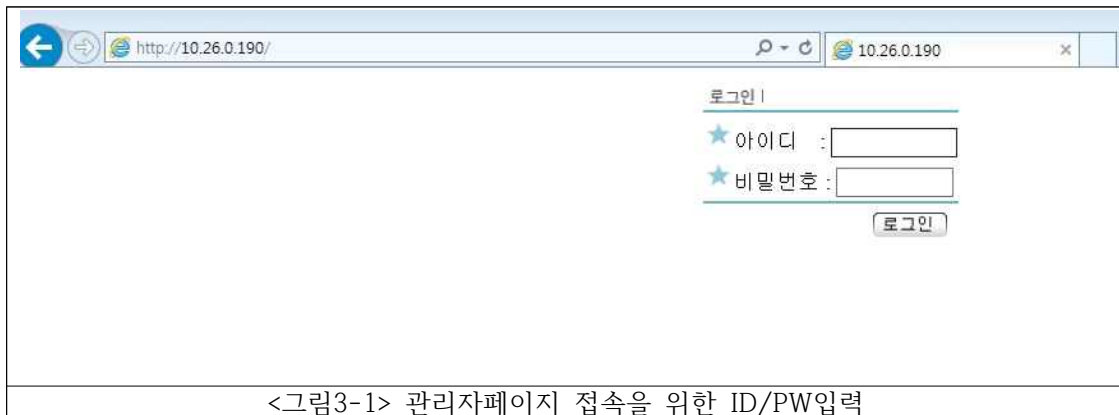
    if($i==0) $datacnt++;
    $data_arr[$datacnt][$i]= $buffArr[$i]; //데이터 배열에 저장
    $data_arr[$datacnt][$i]= str_replace("\'", "'", $data_arr[$datacnt][$i]); //쌍따옴
표처리
    $data_arr[$datacnt][$i]= trim($data_arr[$datacnt][$i]); //사이트 공백 처
리

/* 본문 리스트 출력 시작
첫번째 줄일경우 - 항목명 출력*/
if($datacnt==0){
    echo ("
        <td class='title'>$buffArr[$i]</td>
    ");
}
else if($datacnt==2){
    continue; //두번째줄 건너뛰기
}
else if(($linecnt%2)==1){ //홀수일경우 gainsboro색으로 교차 출력
    echo ("
        <td class='holsu'>$buffArr[$i]</td>
    ");
}
else{ //짝수일경우 white색으로 교차 출력
    echo ("
        <td class='jjagsu'>$buffArr[$i]</td>
    ");
} //if end
} //for end
echo ("</tr>");
} //while end
?>
</table>
</tr>
</table>
</body>
</html>

```

III. 연구 결과

1. 모니터링 상황



ios 관리자 페이지

현황 관리 서버 관리 사용자 관리 시스템 관리

사용자 ID

사용자 계정

fedora	🔍
test1	🔍
gon	🔍
wan	🔍

<그림3-4> 현재 등록되어 있는 사용자들의 명령어 내역을 확인

ios 관리자 페이지

현황 관리 서버 관리 사용자 관리 시스템 관리

사용자 접속로그

사용자 id	접속시간	접속일	IP	로그아웃 시간
fedora	22:21	Wed May 21	:0.0	logged
fedora	22:20	Wed May 21	:0	logged
reboot	21	boot Wed May	boot	(17:10)
fedora	22:19	Wed May 21	:0.0	down
root	22:17	Wed May 21	10.26.0.45	22:17

<그림3-5> 사용자의 모든 접속내역을 확인

ios 관리자 페이지

현황 관리 서버 관리 사용자 관리 시스템 관리

개발서버

su사용	사용가능 처리	rm사용	사용가능 처리	ps사용	사용가능 처리	ping사용	사용가능 처리
su제한	사용불가 처리	rm제한	사용불가 처리	ps제한	사용불가 처리	ping제한	사용불가 처리

파일서버

su사용	사용가능 처리	rm사용	사용가능 처리	ps사용	사용가능 처리	ping사용	사용가능 처리
su제한	사용불가 처리	rm제한	사용불가 처리	ps제한	사용불가 처리	ping제한	사용불가 처리

<그림3-6> 서버별로 사용자에게 특정 명령어를 통제할 수 있는 페이지

IV. 결론

1. 최종 연구 결과 보고

최근 들어 IT산업의 발전으로 인해 정보의 가치가 엄청나게 증가하고 있습니다. 빅 데이터를 많은 사업의 지표로 삼는 경우도 많아지고 있습니다. 즉, 쉽게 말해 정보를 이용하여 돈을 버는 상황이 많이 발생하고 있습니다. 따라서 많은 사람들이 정보를 얻기 위해 노력을 하고 때로는 불법적인 방법으로 정보를 얻으려고 하고 있는 실정입니다. 요즘 들어 뉴스나 신문에서 쉽게 정보 유출에 관한 문제를 많이 볼 수가 있는데 그것이 요즘 실정을 잘 반영하고 있다고 생각합니다.

저희는 이러한 정보의 유출을 막기 위하여 많은 고민을 하다 정당히 허가된 사용자에게만 정보를 제공할 수 있도록 접근통제 서버를 생각하게 되었습니다. 정당히 허가된 사용자만이 서버에 접근하여 자신이 원하는 작업을 수행할 수 있고 자신이 원하는 정보를 얻어갈 수 있도록 접근통제 서버를 만들고 운영하겠다는 생각을 하게 되었고 그러한 기능을 하는 접근통제 서버를 구축하였습니다.

뿐만 아니라 접근통제 서버를 구축하였다 하더라도 다른 문제가 발생할 수 있는 상황이 제기될 수 있기 때문에 이러한 상황에 대응하기 위하여 관리자만이 사용할 수 있도록 관리자 페이지를 구축하게 되었습니다. 관리자가 관리자 페이지를 이용하여 사용자들을 쉽게 통제할 수 있고 만일의 상황에 쉽게 대응할 수 있게 하기 위해 웹 프로그래밍을 이용하여 관리자 페이지를 구축하였습니다.

저희에 연구로 인하여 조금 더 쉽게 정보를 보호할 수 있고 관리자가 좀 더 쉽게 사용자들을 관리하고 정보의 유출에 더욱 빠르고 쉽게 대응할 수 있는데 조금이나마 도움이 되었으면 하는 바람입니다.

V. 참고문헌

[1] 박성수 저

“리눅스 서버관리 실무 바이블 3.0”

[2] <http://www.phpschool.com/> “PHP스쿨”

[3] <http://blueriver35.blog.me/60112940504> “txt파일 웹페이지로 보여주기”

[4] <http://blog.naver.com/necall?Redirect=Log&logNo=80188858458> “버튼제어”

VI. 발표ppt

**접근통제서버를 활용한
보안시스템 구현**




2014.4.29

지도교수 : 유승재 교수님

I O S
Innovation of security

목 차

- 1 조원 소개 및 역할
- 2 주제 선정
- 3 개발 내용
- 4 접근통제서버 운영
- 5 진도 점검
- 6 결론 및 건의



조원 소개 및 역할

조원	역할
조장 심재완	졸업작품 총괄
조원 이동재	통제서버 구축
이주곤	통제서버 구축
강광혁	서버-통제서버 인증구축
서유현	관리자 페이지 구축



주제 선정

➤ 주제

- Telnet이나 SSH로 접속하는 비인가 사용자의 직접적인 시스템 접근을 통제하고 사용자 권한을 제한하는 접근통제서버 구축

➤ 주제 선정 사유

- 최근 인력의 부주의로 인한 **보안 고가 잇따라 발생**하면서 인력의 권한 관리 및 시스템 접근통제 정책의 중요성이 부각
- 특히 서버 환경의 확산과 더불어 **서버의 자료와 보안**이 크게 중요시
- ※ **비인가 사용자의 서버 접속을 봉쇄**하고, 중요 자원에 대한 **사용자의 무분별한 접근을 통제**하며, **작업내역을 관리자가 확인**하는 접근 통제서버를 구축 ⇒ **서버 보안체제 구현**



개발 내용

개발 환경

- 운영체제 : Fedora 11
- 개발언어 : Shell Script.

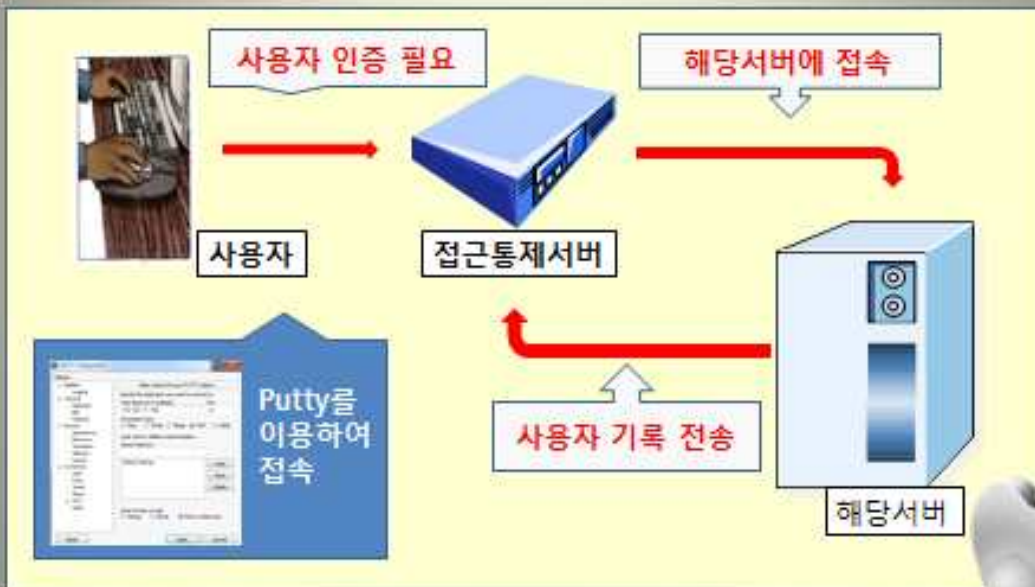
PHP

HTML



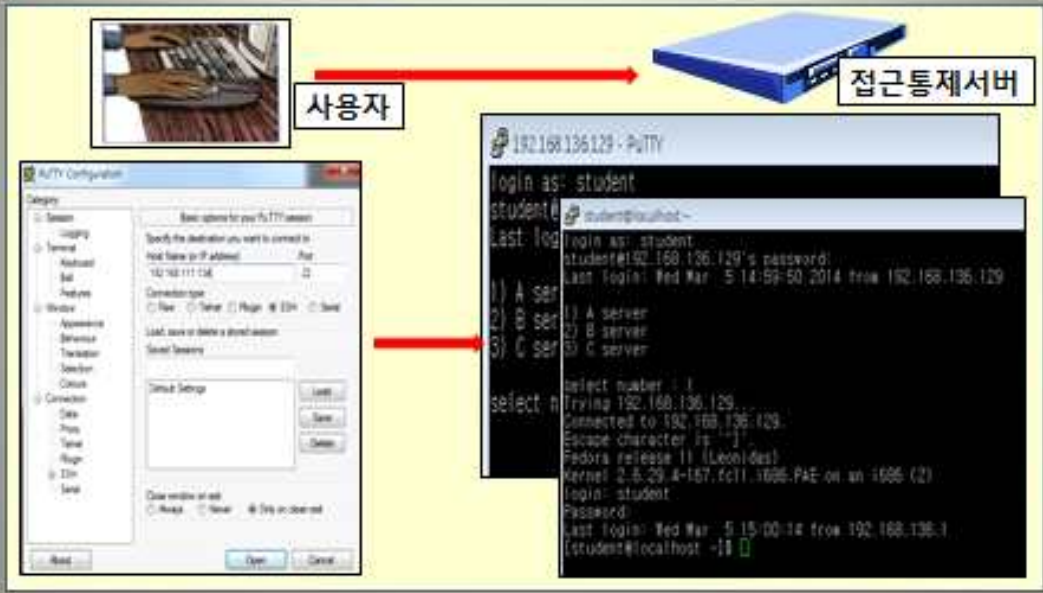
개발 내용

시스템 구성



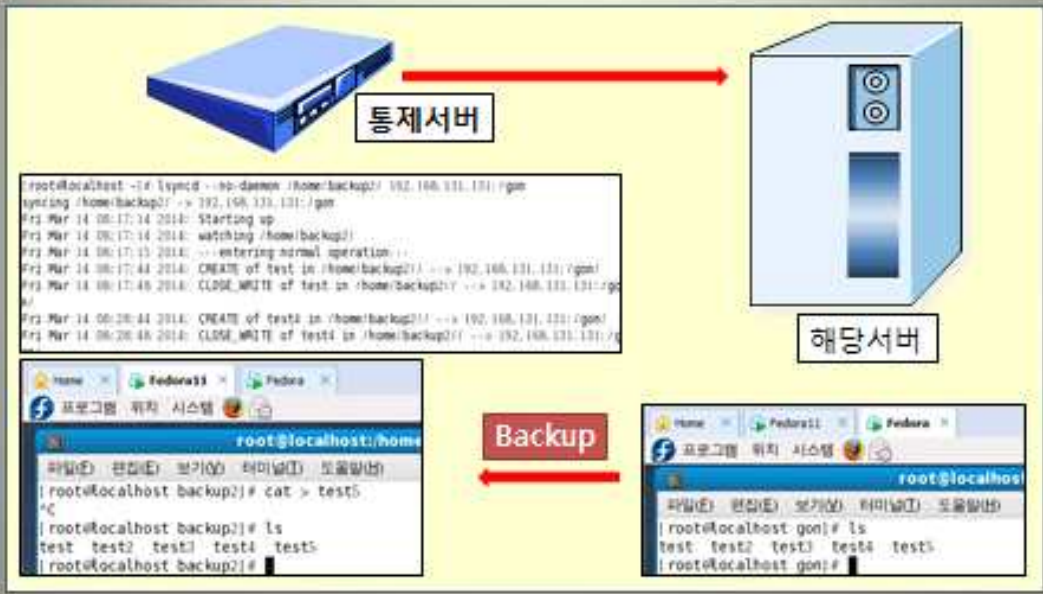
개발 내용

사용자-통제서버



개발 내용

서버간 연동



접근통제서버 운영

> 접근통제서버 구동 → 관리자 페이지(현황판)

105 관리자 페이지

관리자 페이지

현황 관리 서버 관리 사용자 관리 시스템 관리

INNOVATION OF SECURITY

총 서버 수 총 사용자 수

현황판 ⇒ 총 서버 수, 사용자 수를 클릭

접근통제서버 운영

> 서버 상황 및 서버 현황 정보

105 관리자 페이지

현황 관리 서버 현황 사용자 관리 시스템 관리

105 관리자 페이지

현황 관리 서버 현황 사용자 관리 시스템 관리

서버 현황

No.	서버명	분류
1	주요서버	중요
2	보조서버	보통

서버 상태

서버명	상태
주요서버	[ON]
보조서버	[ON]

서버 현황 ⇒ 서버의 상세정보 클릭

서버 상태 ⇒ 서버들의 가동상태 클릭

접근통제서버 운영

> 사용자 관리 및 사용자 이력

2. 관리자 페이지

권한 관리 | 서버 관리 | 사용자 관리 | 시스템 관리

사용자 현황

사용자 관리 ⇒ 사용자 이력

2. 관리자 페이지

권한 관리 | 서버 관리 | 사용자 관리 | 시스템 관리

사용자 이력

사용자 검색

Admin | test1 | test2

사용자 이력 ⇒ 사용자별 사용 이력 출력

접근통제서버 운영

> 사용자 접속로그

2. 관리자 페이지

권한 관리 | 서버 관리 | 사용자 관리 | 시스템 관리

사용자 접속로그

접속로그 ⇒ 사용자의 모든 접속상황 출력

이름	접속시간	접속명	IP	로그아웃 시간
Admin	20:21	Web_Appl_1.0	192.168.0.100	20:21
Admin	20:40	Web_Appl_1.0	192.168.0.100	20:29
test1	20:59	Web_Appl_1.0	10.10.10.99	20:59
test1	20:57	Web_Appl_1.0	10.10.10.95	20:59
test2	20:58	Web_Appl_1.0	10.10.10.95	20:57
test2	20:58	Web_Appl_1.0	10.10.10.99	20:58
test2	20:52	Web_Appl_1.0	10.10.10.95	20:58
test2	20:51	Web_Appl_1.0	10.10.10.95	20:52
Admin	20:53	Web_Appl_1.0	10.10.10.95	20:51
Admin	20:55	Web_Appl_1.0	10.10.10.95	20:51
test5	20:48	Web_Appl_1.0	10.10.10.93	20:50
test5	20:47	Web_Appl_1.0	10.10.10.95	20:48
test5	20:48	Web_Appl_1.0	10.10.10.93	20:47
Admin	20:45	Web_Appl_1.0	10.10.10.95	20:48
Admin	20:49	Web_Appl_1.0	10.10.10.93	20:45
Admin	20:49	Web_Appl_1.0	192.168.0.100	20:49
Admin	20:49	Web_Appl_1.0	192.168.0.100	20:49
test3	20:49	Web_Appl_1.0	192.168.0.100	logged
test3	18	Web_Appl_1.0	192.168.0.100	20:49
test3	18:18	Web_Appl_1.0	192.168.0.100	20:49
test3	18:18	Web_Appl_1.0	192.168.0.100	20:49

접근통제서버 운영

> 명령어 통제

2 관리자 페이지

원형 관리
서버 관리
사용자 관리
시스템 관리

명령어 통제
시스템 2
시스템 3

A서버

ms사용	사용가능 처리	ms사용	사용가능 처리	ps사용	사용가능 처리	ms사용	사용가능 처리
ms제한	사용불가 처리	ms제한	사용불가 처리	ps제한	사용불가 처리	ms제한	사용불가 처리

B서버

ms사용	사용가능 처리	ms사용	사용가능 처리	ps사용	사용가능 처리	ms사용	사용가능 처리
ms제한	사용불가 처리	ms제한	사용불가 처리	ps제한	사용불가 처리	ms제한	사용불가 처리

C서버

ms사용	사용가능 처리	ms사용	사용가능 처리	ps사용	사용가능 처리	ms사용	사용가능 처리
ms제한	사용불가 처리	ms제한	사용불가 처리	ps제한	사용불가 처리	ms제한	사용불가 처리

명령어 통제 ⇒ 사용자의 특정 명령어 사용 통제

진도 점검

	9월	10월	11월	12월	1월	2월	3월	4월	5월
주제 선정									
자료수집 및 개인공부									
관리자 모니터링 구축									
수정 및 테스트									

> 주제 조정

- 인증서버 ⇒ 공개서버로 접속하는 과정에서 인증을 받는 부분이 해결 되지 않아 현재 방식으로 비공.
- 사용자관리 ⇒ 접속로그에서 날짜별 및 검색기능을 추가하고자 했으나 해결하지 못함

결론 및 의견

결론

- ▶ 해당 서버에 LSYNC를 사용하여 서버에 대한 정보를 실시간 백업하여 통제 서버로 보내줌으로써 서버에 대한 정보를 실시간으로 받아옴.
- ▶ 관리자 페이지를 구축하여 현황관리, 서버관리, 사용자 관리, 시스템 관리로 분류하여 간편하게 서버를 관리할 수 있음.

의견

- ▶ 향후 관리자 페이지 스타일링 예정.
- ▶ 의견 수렴 후 관리자 페이지의 시스템 관리 페이지에 추가 기능 삽입 예정.
→ 계속적으로 기능 보안 및 추가 예정

*Do you have any
Questions?*

Thank You

